Imperial College London
Department of Mechanical Engineering

A Joint Eulerian-Lagrangian Method for the Solution of Multi-Scale Flow Problems

Michael W. A. Pettit

August 2013

Supervised by Prof. Dr.-Ing. A. M. Kempf, Dr. A. J. Marquis

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering of Imperial College London
and the Diploma of Imperial College London

Abstract

Numerical convection schemes in a time-dependent Computational Fluid Dynamics simulation suffer from numerical diffusion, where a transported scalar quantity experiences a total diffusivity greater than any physical diffusivity due to the viscosity of the surrounding fluid. This work aims to develop and test a novel convection method, combining the Eulerian and Lagrangian frameworks, to eliminate numerical diffusion.

A conserved scalar quantity is decomposed into low- and high-frequency components. The low-frequency field is transported in the Eulerian framework using a high-order Central Differencing Scheme, which has a negligible numerical diffusivity and good cost efficiency, but exhibits oscillatory behaviour around sharp changes in gradient. The high-frequency information is described with (computationally expensive) massless Lagrangian particles, where the prescribed particle 'density' provides a balance between accuracy and computational cost. Particles are convected by interpolating the underlying velocity field on to the particle position, while particle diffusion is described using a stochastic Wiener process. After transport an Eulerian representation of the Lagrangian particle field is constructed and added to the low-frequency scalar component, to recover the overall transported field.

Formulations of the *joint Eulerian-Lagrangian method* for Direct Numerical Simulation and Large-Eddy Simulation are proposed. Re-initialisation (particle addition) and particle removal are implemented to maintain accuracy and to reduce the computational cost of the method. The accuracy of the Lagrangian reconstruction is improved through the development and application of localised filtering and deconvolution algorithms. The method is applied in two and three dimensions, where it is effective in removing numerical diffusion, but introduces noise into the scalar field due to the point-like nature of the particles. While the method is at least twice as expensive as traditional Eulerian simulations at the same grid resolution, it is capable of delivering better accuracy and considerably greater cost efficiency than Eulerian simulations at higher resolutions.

To E, S, F, R and N

'Since they had no way of knowing that it would later be regarded as
one of the truly difficult peaks of North America, they simply went ahead and climbed it.

That, in my mind, is the way to climb a mountain.'

William Henry Jackson

Declaration of Originality

The work presented in this thesis is entirely that of the author,
except where the previous efforts of others have been cited as appropriate.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under
a Creative Commons Attribution Non-Commercial No Derivatives licence.
Researchers are free to copy, distribute or transmit the thesis on the condition
that they attribute it, that they do not use it for commercial purposes and
that they do not alter, transform or build upon it. For any reuse or redistribution,
researchers must make clear to others the licence terms of this work.

Acknowledgements

I must start off by thanking Prof. Dr.-Ing. Andreas Kempf of the University of Duisberg-Essen, and Dr. Fred Marquis of Imperial College London, for the fantastic support and guidance they have given me over four (or more) years. Without Andreas' intervention I would never have had the opportunity to undertake a PhD; without Fred's enthusiastic teaching while I was an Undergraduate, I probably would not have been interested in Fluid Mechanics in the first place. You have both been most excellent supervisors, from whom I could not have wished for more, and great acquaintances.

The generous financial support of the Engineering and Physical Sciences Research Council through the Doctoral Training Award scheme at Imperial College London has obviously been essential and has been very gratefully received, while the High-End Computing Studentship that I was very fortunate to be awarded has opened up many exciting new opportunities. Many thanks to David Henty, Lorna Smith, and all of the other staff and students at the Edinburgh Parallel Computing Centre, University of Edinburgh, for making my time on the HPC course an enjoyable and educational one.

For the work on Turbulent Opposed Jets presented in Chapter 11, Prof. Kempf and I would like to thank Prof. Alessandro Gomez and Dr. Bruno Coriton at the Yale Center for Combustion Studies for their efforts in our collaborative contribution to the 33$^{\text{rd}}$ Combustion Symposium. We also thank Dr. Rob Barlow, of Sandia National Laboratories, as the main organiser of the 'Turbulent Non-Premixed Flames Workshop' at which the collaboration was established.

To my colleagues at Imperial College, past and present: Stefan Wysocki, Philipp Geipel and Nakul Prasad, who have given the study of philosophy a whole new meaning; Roger 'Otto' Robinson, for the introduction to mountain biking, and a new way to injure myself; Olli Stein and Jeremy Floyd, for helping me through the early stages of my PhD; Abdel Rahbiou, for the education in politics and useful French phrases; Yagos Pesmazoglou, Ben Franchetti, Fabrizio Cavallo-Marincola, Terence Ma and Miriam Rabaçal; and to my new friends in Duisburg: Fabian Proch, Andreas Rittler, Thuong Nguyen, Peter Janas, Martin Reith, Claudia Weise, Nejra Sikalo, and Dr. Irenäus Wlokas, I wish you all the very best.

I am indebted to the Royal School of Mines, Imperial College 2nd XI and East London Hockey Clubs, who have given me something to look forward to each weekend, particularly during the long weeks of writing up.

I must also thank the teaching staff at Truro School, especially to Dr. Peter Allen, whose enthusiasm for science was highly influential; to Tim Tall, who taught me (perhaps unsuccessfully) to 'think on paper, not in wood'; and to Andy Crawford, for many years of patient advice.

To my family and friends in Cornwall, London, and elsewhere; to my sisters and brothers-in-law; to my nieces and nephews, thank you all for your continuing love and support, I could not have done it without you. I owe a special thanks (and probably a drink or two) to Katie Vowles and Sammy Jones, whose efforts to proof-read the final version of this thesis have gone far beyond my hopes and expectations.

And finally, to Dehydys, for the never-ending motivation and encouragement, and for making it all so very enjoyable.

Contents

1 Introduction **21**

1.1 Motivation . . . 23
1.2 Objectives . . . 23
1.3 Thesis Structure . . . 24

2 Theory and Simulation of Fluid Flow **25**

2.1 Fundamental Theory . . . 25
2.1.1 Assumptions . . . 25
2.1.2 Equations of Fluid Motion . . . 26
2.1.3 Turbulence . . . 28
2.1.4 Turbulent Scales . . . 29
2.2 Time-Averaged Simulations . . . 30
2.3 Time-Dependent Simulations . . . 32
2.3.1 Direct Numerical Simulation . . . 32
2.3.2 Large-Eddy Simulation . . . 33
2.4 Summary . . . 36

3 Simulation Numerics **37**

3.1 Spatial Discretisation: Finite Volume Method . . . 37
3.2 Temporal Discretisation . . . 38
3.3 Time-Step Width . . . 40
3.4 Pressure Correction . . . 40
3.4.1 The SIMPLE Algorithm . . . 40
3.4.2 The Fully Conservative Approach . . . 41
3.5 Sub-Grid Scale Turbulence Models . . . 41
3.5.1 Smagorinsky Model . . . 42
3.5.2 Dynamic Smagorinsky (Germano) Model . . . 42
3.5.3 Other SGS Turbulence Models . . . 44
3.6 Sub-Grid Scale Scalar Flux Models . . . 45
3.6.1 Gradient Diffusion Model . . . 46
3.6.2 Similarity Model . . . 46
3.6.3 Mixed Model . . . 46
3.7 Boundary Conditions . . . 47

3.7.1 Velocity Inflow Conditions . . . 47
3.7.2 Other Boundary Conditions . . . 50
3.8 Summary . . . 50

4 Discretisation of Fluxes **51**
4.1 Convection in the Finite Volume Method . . . 51
4.2 Desirable Properties of Differencing Schemes . . . 52
4.3 Low Order Schemes . . . 53
4.3.1 First-Order UDS . . . 53
4.3.2 Second-Order CDS . . . 55
4.3.3 Second-Order LUDS . . . 57
4.4 Higher-Order Schemes . . . 58
4.4.1 Third-Order QUICK . . . 58
4.4.2 Higher-Order CDS . . . 59
4.4.3 ENO Schemes . . . 61
4.4.4 WENO Schemes . . . 62
4.5 TVD Schemes . . . 64
4.6 Summary of Convective Schemes . . . 70
4.7 Diffusion in the Finite Volume Method . . . 70

5 Formulation of the Joint Eulerian-Lagrangian Method **72**
5.1 Review of the Literature . . . 72
5.2 Theoretical Formulation . . . 75
5.2.1 The Eulerian-Lagrangian Decomposition . . . 76
5.2.2 Lagrangian Representation . . . 79
5.2.3 Treatment of Transport Terms . . . 79
5.2.4 Lagrangian Reconstruction . . . 84
5.3 Implementation – Initialisation . . . 84
5.3.1 Decomposition . . . 84
5.3.2 Particle Distribution . . . 93
5.3.3 Particle Assignment . . . 96
5.4 Implementation – Time-Stepping . . . 98
5.4.1 Eulerian Convection and Diffusion . . . 98
5.4.2 Lagrangian Convection and Diffusion . . . 99
5.4.3 Lagrangian Reconstruction . . . 103
5.4.4 Re-initialisation . . . 104
5.5 Summary . . . 108

6 Test Case 1: The Zalesak Disc **110**
6.1 Case Objectives . . . 110
6.2 Case Description . . . 110
6.2.1 Velocity Field . . . 112

6.2.2 Solid Body . . . 112
6.3 Simulation Numerics . . . 112
6.4 Results and Discussion . . . 114
6.4.1 Eulerian Simulations . . . 116
6.4.2 Eulerian-Lagrangian Simulations . . . 119
6.4.3 Tabulated Results . . . 122
6.5 Conclusions . . . 126

7 Test Case 2: Zalesak Modified **127**
7.1 Case Modification . . . 127
7.2 Results and Discussion . . . 129
7.2.1 One Revolution, High Schmidt Number . . . 129
7.2.2 One Revolution, Low Schmidt Number . . . 133
7.2.3 100 Revolutions, High Schmidt Number . . . 136
7.3 Conclusions . . . 142

8 Further Development **144**
8.1 Particle Removal . . . 144
8.1.1 Formulation . . . 145
8.1.2 Testing . . . 146
8.1.3 Application in Higher Dimensions . . . 150
8.2 Low-Pass Filtering . . . 151
8.2.1 Metrics . . . 152
8.2.2 Filtered Field . . . 154
8.2.3 Filter Localisation . . . 155
8.2.4 Testing: Two Dimensions . . . 159
8.2.5 Testing: Three Dimensions . . . 163
8.3 Inverse Filtering . . . 164
8.3.1 Deconvolution . . . 165
8.3.2 Deconvolution Localisation . . . 168
8.3.3 Testing . . . 170
8.3.4 Other Methods . . . 172
8.4 Application to the Zalesak Cases . . . 173
8.4.1 Particle Removal . . . 173
8.4.2 Low-Pass and Inverse Filtering . . . 177
8.5 Conclusions . . . 178

9 Test Case 3: The Single Vortex **181**
9.1 Case Objectives . . . 181
9.2 Case Description . . . 182
9.2.1 Velocity Field . . . 182
9.2.2 Solid Body . . . 183

9.3 Simulation Numerics . . . 184
9.4 Results and Discussion . . . 186
9.4.1 Eulerian Simulations . . . 187
9.4.2 Eulerian-Lagrangian Simulations . . . 196
9.4.3 Tabulated Results . . . 202
9.4.4 Localised Deconvolution and Filtering . . . 206
9.4.5 Standardised Performance . . . 208
9.5 Conclusions . . . 210

10 Test Case 4: The Mixing Layer **212**
10.1 Case Objectives . . . 212
10.2 Case Description . . . 213
10.3 Simulation Numerics . . . 216
10.4 Flow-Field Development . . . 220
10.5 Results and Discussion . . . 224
10.5.1 Scalar Fields . . . 224
10.5.2 Scalar Topology . . . 237
10.5.3 Computational Cost . . . 248
10.6 Conclusions . . . 255

11 Test Case 5: The Turbulent Opposed Jet **258**
11.1 Case Objectives . . . 258
11.2 Case Description . . . 259
11.3 Simulation Numerics . . . 260
11.3.1 Precursor Simulation . . . 261
11.3.2 Mixing Layer Simulations . . . 263
11.4 Results and Discussion . . . 265
11.4.1 Precursor Simulation . . . 266
11.4.2 Mixing Layer Simulations . . . 271
11.5 Conclusions . . . 279

12 Conclusions and Further Work **281**
12.1 Conclusions . . . 281
12.2 Further Applications . . . 284
12.2.1 Downstream Mixing in a Bluff Body Jet . . . 284
12.2.2 Mixing in a Stirred Tank . . . 285
12.3 Further Development . . . 285
12.3.1 Re-Initialisation . . . 286
12.3.2 Localised Low-Pass Filtering and Deconvolution . . . 286
12.3.3 Resolving the Batchelor Scale . . . 287
12.3.4 The Eulerian-Lagrangian Method for Density Transport . . . 288

List of Tables

4.1 Coefficients for the approximation of ϕ_e up to eighth order-of-accuracy [114]. . 60
4.2 Flux limiters from the literature. Non-TVD limiters are shown in the lower part. Limiters marked (*) are 'symmetric' (Eq. 4.33). 68
4.3 Coefficients for the approximation of $\left.\frac{d\phi}{dx}\right|_e$ up to eighth order-of-accuracy [114]. 71

6.1 Numerical schemes to be tested with the Zalesak disc. 113
6.2 Measurements of accuracy and cost for Cases Z1 to Z9. 123

7.1 Numerical schemes tested with the modified Zalesak disc. 128
7.2 Measurements of accuracy and cost for Cases ZM1 to ZM4. 133
7.3 Measurements of accuracy and cost for Cases ZM5 to ZM8. 135
7.4 Measurements of accuracy and cost for Cases ZM9 to ZM12. 143

8.1 Particle removal test results. 149
8.2 Simulation parameters for filter localisation tests. 152
8.3 Maximum change in total error, uniform vs. localised filtering. 162
8.4 Maximum change in total error in 3D, uniform vs. localised filtering. 163
8.5 Total error before and after localised deconvolution. 170
8.6 Total error before and after localised deconvolution, in three dimensions. 172
8.7 Impact of particle removal, Cases Z8 and Z9. 173
8.8 Impact of particle removal, Cases ZM3 and ZM4. 174
8.9 Impact of particle removal, Cases ZM7 and ZM8; CDS8 for comparison. 176
8.10 Impact of particle removal, Cases ZM11(3) and ZM12(2). 176
8.11 Impact of filtering (LPF) and deconvolution (HPF) for the original Zalesak cases. 177
8.12 Impact of filtering for the modified Zalesak cases. 178
8.13 A comparison of total error and time to solution for a selection of the Eulerian (italicised) and E-L simulations presented so far, including particle removal (PR), low-pass filtering (LPF), and deconvolution (HPF). 179

9.1 Numerical schemes to be tested with the single vortex case. 185
9.2 Measurements of accuracy and cost for Cases SV1 to SV6, after $t = 10$ s. 202
9.3 Case SV4: With and without deconvolution (HPF); with modified filtering (LPF). 207
9.4 Case SV6: With and without deconvolution (HPF); with modified filtering (LPF). 208

10.1 Simulation configurations for the mixing layer study. 219

10.2 Computational cost of the mixing layer simulations. 248

11.1 Mixing layer simulation configurations for the TOJ study. 264

List of Figures

2.1 The energy cascade. 29

3.1 Different grid structures: (a) uniform, orthogonal Cartesian; (b) non-uniform, orthogonal Cartesian; non-uniform, oblique Cartesian; (d) unstructured. 38
3.2 Filtering of random noise using a combination of two 1D filters: (a) original field, (b) filtered in j-direction, and (c) subsequently filtered in i-direction. . . . 49

4.1 A 1D domain containing a function ϕ discretised at points i, and a velocity u. . 51
4.2 The first-order Upwind approximation. 54
4.3 Two 1D functions transported by UDS: (a) sharp, (b) smooth. 54
4.4 The second-order Central approximation. 56
4.5 Two 1D functions transported by CDS2: (a) sharp, (b) smooth. 56
4.6 Oscillation reduction with increasing smoothness for CDS2. 57
4.7 The second-order Upwind approximation. 57
4.8 Two 1D functions transported by LUDS: (a) sharp, (b) smooth. 58
4.9 The third-order Quadratic Upwind approximation. 59
4.10 Two 1D functions transported by QUICK: (a) sharp, (b) smooth. 59
4.11 The fourth-order Central approximation. 60
4.12 Two 1D functions transported by CDS8: (a) sharp, (b) smooth. 60
4.13 Oscillation reduction with increasing smoothness for CDS2 and CDS8. 61
4.14 Two second-order polynomials used to construct an ENO approximation. . . . 62
4.15 Two 1D functions transported by second-order ENO: (a) sharp, (b) smooth. . . 62
4.16 Three third-order polynomials used to construct a WENO approximation. . . . 63
4.17 Two 1D functions transported by fifth-order WENO: (a) sharp, (b) smooth. . . 64
4.18 Two 1D functions transported by MinMod TVD: (a) sharp, (b) smooth. 66
4.19 Two 1D functions transported by SuperBee TVD: (a) sharp, (b) smooth. 67
4.20 Two 1D functions transported by CHARM TVD: (a) sharp, (b) smooth. 67
4.21 Sweby diagram of 18 schemes, including the 14 flux limiters in Table 4.2. The shaded area represents the bounds for second-order TVD schemes. 69

5.1 The Eulerian-Lagrangian decomposition. 76
5.2 Random walks of 20 particles. 82
5.3 PDFs generated from 10^6 random walks. 83
5.4 Sinc function filter kernel (a); typical filter response (b). 86
5.5 Sinc function filter kernel with reduced kernel size (a); typical filter response (b). 87

5.6 Sinc filter windowed with an exponential function (a); typical filter response (b). 88
5.7 Top-hat filter kernel of width 3Δ (a); typical filter response (b). 89
5.8 Gaussian filter kernel (a); typical filter response (b). 90
5.9 For various Gaussian filter widths: (a) kernel, (b) response, (c) smoothness. . . 92
5.10 Reducing standard deviation and maximum error with increasing L^{ϱ}. 94
5.11 Increasing computational cost with increasing L^{ϱ}. 94
5.12 Moiré pattern resulting from cell-centred particle distributions. 94
5.13 Generation of ghost particles on neighbour processes, and inflow particles. . . . 97
5.14 Transported and derived scalars described by Eulerian and Lagrangian methods, at constant or variable density and viscosity. 102
5.15 Scalar contributions of four Lagrangian particles in a 2D Eulerian frame. 103
5.16 Global re-initialisation, showing accumulating Lagrangian errors. 105
5.17 Step function: initial and filtered, with derivatives, compared to the filter kernel. 106

6.1 Zalesak disc: Schematic representation. The shaded area represents $\phi = 1$. . . . 113
6.2 Zalesak disc: Initial conditions after discretisation. 114
6.3 Zalesak disc: E-L reconstruction. 114
6.4 Case Z1: UDS. 117
6.5 Case Z2: QUICK scheme. 117
6.6 Case Z3: Osher TVD ($\beta = 1.0$). 118
6.7 Case Z4: Osher TVD ($\beta = 2.0$). 118
6.8 Case Z5: CHARM TVD. 119
6.9 Case Z6: SuperBee TVD. 119
6.10 Case Z7: E-L solution applying CDS2 for the Eulerian phase. 120
6.11 Case Z7: Eulerian component. 121
6.12 Case Z7: Lagrangian component. 121
6.13 Case Z8: E-L method with CDS8, $L^{\varrho} = 1000$. 122
6.14 Case Z9: E-L method with CDS8, $L^{\varrho} = 100$. 122
6.15 A visual representation of boundedness (a) and total error (b) for Cases Z1 to Z9. 124
6.16 The additional computational cost of Lagrangian particles. 125

7.1 Modified Zalesak disc: Initial conditions 128
7.2 Modified Zalesak disc: The Eulerian solution applying diffusion only, $Sc = 10$. . 130
7.3 Case ZM1: CHARM TVD; $Sc = 10$. 131
7.4 Case ZM2: SuperBee TVD; $Sc = 10$. 131
7.5 Case ZM3: E-L method with CDS8, $L^{\varrho} = 1000$; $Sc = 10$. 132
7.6 Case ZM4: E-L method with CDS8, $L^{\varrho} = 100$; $Sc = 10$. 132
7.7 Modified Zalesak disc: The Eulerian solution applying diffusion only, $Sc = 1$. . 134
7.8 Case ZM5: CHARM TVD; $Sc = 1$. 135
7.9 Case ZM6: SuperBee TVD; $Sc = 1$. 135
7.10 Case ZM7: E-L method with CDS8, $L^{\varrho} = 1000$; $Sc = 1$. 136
7.11 Case ZM8: E-L method with CDS8, $L^{\varrho} = 100$; $Sc = 1$. 136

7.12 Case ZM9: CHARM TVD; 100 revolutions, $Sc = 100$. 138
7.13 Case ZM10: SuperBee TVD; 100 revolutions, $Sc = 100$. 138
7.14 Case ZM11: E-L method with CDS8, $L^{\varrho} = 1000$; 100 revolutions, $Sc = 100$. . 139
7.15 Case ZM11: Eulerian phase. 139
7.16 Case ZM11: Lagrangian phase. 139
7.17 Case ZM11(2): With CDS12 for the Eulerian phase. 141
7.18 Case ZM11(3): With $\sigma_f = 15\Delta$ for the initial decomposition. 141
7.19 Case ZM12: E-L method with CDS8, $L^{\varrho} = 100$; 100 revolutions, $Sc = 100$. . . 142
7.20 Case ZM12(2): With $\sigma_f = 15\Delta$ for the initial decomposition. 142

8.1 Initial decomposition. 146
8.2 Particle removal: Without transport. 147
8.3 Diffused field, without particle removal applied. 148
8.4 Particle removal: After (a) 1000, (b) 2000, (c) 4000 and (d) 6000 steps. 149
8.5 Effect of varying filter width on: lower boundedness (a) ΔB_L and (b) $|\Delta B_L|$ (%); upper boundedness (c) ΔB_U and (d) $|\Delta B_U|$ (%); total error (e) E and (f) E (%); change in total variation (g) ΔTV and (h) ΔTV (%); symmetry error (i) E_{sym} and (j) E_{sym} (%). 153
8.6 Effect of filtering the Lagrangian field (red) or the E-L summation (black). . . . 154
8.7 Decreasing σ_f with an increasing rate of change in scalar gradient. 156
8.8 Variation in N_f for different top-hat filter widths. 157
8.9 Effective Gaussian filter width for different top-hat filter widths. 158
8.10 Initial field: Second derivative of ϕ_f^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding σ_f; (e) pre- and (f) post-filtered error field. 160
8.11 Case F05LS: Second derivative of ϕ_f^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding σ_f; (e) pre- and (f) post-filtered error field. 161
8.12 Post-filtering total error: uniform (black) and localised (red). 162
8.13 Post-filtering total error, in 3D: uniform (black) and localised (red). 164
8.14 Lagrangian reconstruction: centred particle distribution. 165
8.15 Lagrangian reconstruction: uniform random particle distribution. 166
8.16 The high-pass filter kernel j required for the deconvolution of ϕ_f. 167
8.17 Reducing deconvolution error with increasing deconvolution operations. 168
8.18 Variation of N_d required to attain 0.01% deconvolution error with $\left|\nabla^2\phi_f\right|_{\max}$. . 169
8.19 Initial field: Second derivative of ϕ^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding N_d; (e) pre- and (f) post-deconvolution error field. . 171
8.20 Case Z8: Transported Eulerian phase with and without particle removal. 174
8.21 Case ZM3: Transported Eulerian phase with and without particle removal; initial Eulerian component for comparison. 175
8.22 Case ZM12(2): Transported Eulerian phase with and without particle removal. 177

9.1 Single vortex: Contour plot of the stream function described in Eq. 9.1. 183
9.2 Single vortex: Schematic representation. The shaded area represents $\phi = 1$. . . 184

9.3 Single vortex: Approximate analytical solution at $t = 5$ s. 185
9.4 Single vortex: PDF of the initial scalar field; assessed region (red box). 187
9.5 Case SV1: CDS8, $\Delta = 1.00$ cm; scalar field at (a) $t = 5$ s; (b) $t = 10$ s. 188
9.6 Case SV2: CHARM TVD, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 190
9.7 Case SV3(a): SuperBee TVD, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 193
9.8 Case SV3(d): SuperBee TVD, $\Delta = 0.40$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 194
9.9 Case SV3(f): SuperBee TVD, $\Delta = 0.20$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 195
9.10 Case SV4: Detail of the reconstruction near to the 'tail' of the scalar field. . . . 196
9.11 Case SV4: (a) Particle positions; (b) high-resolution Lagrangian reconstruction. 197
9.12 Case SV4: Slice through the scalar field at $x = 1$ m, without deconvolution. . . 198
9.13 Lagrangian reconstructions after particle translation by 0.5 units. 198
9.14 Case SV4: E-L method, $L^{\varrho} = 500$, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 200
9.15 Case SV6: E-L method, $L^{\varrho} = 20$, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF. 201
9.16 Case SV3: Dependence of (a) upper boundedness, (b) total error on Δ. 203
9.17 Variation of (a) upper boundedness, (b) total error with time, applying a constant diffusivity to the initial SV conditions. 204
9.18 Case SV3: Dependence of computational cost on Δ. 204
9.19 E-L simulations: Dependence of computational cost on particle number. 205
9.20 E-L simulations: Total and added particle numbers for (a) SV4; (b) SV5; (c) SV6. 206
9.21 Standardised performance. (*With no deconvolution, and modified filtering). . 209
9.22 Standardised performance, Zalesak and modified Zalesak simulations. 210

10.1 Initial mixing layer bulk flow profile. 214
10.2 Initial mixing layer flow-field; (a) horizontal and (b) vertical velocities. 221
10.3 Mixing layer flow-field at $t = 2.5$ ms; (a) horizontal and (b) vertical velocities. . 222
10.4 Mixing layer flow-field at $t = 12$ ms; (a) horizontal and (b) vertical velocities. . 223
10.5 Effective diffusivity with varying Sc, for a selection of constant D_{num}. 227
10.6 Case ML5-ELb: Lagrangian field at $t = 12$ ms, with flow-field and scalar contours. 228
10.7 Case ML1: Scalar field at $t = 20$ ms, $Sc = 0.7$; (a) Eulerian, (b) E-L method. . 229
10.8 Case ML2: Scalar field at $t = 20$ ms, $Sc = 5$; (a) Eulerian, (b) E-L method. . . 230
10.9 Case ML3: Scalar field at $t = 18$ ms, $Sc = 20$; (a) Eulerian, (b) E-L method. . . 231
10.10 Case ML4: Scalar field at $t = 13$ ms, $Sc = 100$; (a) Eulerian, (b) E-L method. . 232
10.11 Case ML5: Scalar field at $t = 12$ ms, $Sc = 1000$; (a) Eulerian, (b) E-L method. 233
10.12 Case ML4: (a) Eulerian; (b) E-L method; (c) E-L without particle removal. . . 234
10.13 Case ML5: (a) Eulerian; (b) E-L method; (c) E-L without particle removal. . . 234

10.14 Case ML5: Scalar field at $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm. 236
10.15 $|\nabla\phi|$ conditional on ϕ, for an analytical 1D profile with varying Sc, after $t = 20$ ms. 239
10.16 Case ML2: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method. 240
10.17 Case ML3: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method. 241
10.18 Case ML4: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method. 241
10.19 Case ML5: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm. 242
10.20 Mixing layer thickness: area between $\phi = 0.01$ and $\phi = 0.99$ (shaded) divided by length of $\phi = 0.5$ contour (dotted line). 244
10.21 Mixing layer thickness with varying Sc at $t = 12$ ms. 245
10.22 Case ML3: Scalar dissipation rate at $t = 20$ ms; (a) Eulerian, (b) E-L method. 246
10.23 Case ML5: Scalar dissipation rate at $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm. 247
10.24 E-L simulations: Dependence of computational cost on particle number. 252
10.25 Total active; added; removed particle numbers, for Cases ML1-EL to ML5-EL, (a) to (e) respectively. *Without particle removal. 254

11.1 Schematic of a single nozzle. Dimensions in mm, not to scale. 260
11.2 Precursor simulation: Horizontal velocities, (a) instantaneous, (b) fluctuation; vertical velocities, (c) instantaneous, (d) mean; lateral velocities, (e) instantaneous, (f) fluctuation; viscosity ratio r_ν, (g) instantaneous, (h) mean. 269
11.3 Precursor simulation: Velocities from PIV and LES; along the axis: vertical (a) mean, (b) fluctuation; (c) horizontal fluctuation; across the nozzle exit: (d) vertical mean; horizontal (e) mean, (f) turbulence intensity. 270
11.4 Precursor simulation: Normalised auto-correlations for u, v, w velocities. 270
11.5 Mixing layer simulations: Velocities from PIV and LES; along the axis: vertical (a) mean and (b) fluctuation, (c) horizontal fluctuation. 271
11.6 Mixing layer simulations: Viscosity ratio r_ν; (a) instantaneous, (b) mean. . . . 272
11.7 Case TOJ2-EL: Volume-rendered instantaneous fields; (a) mixture fraction; (b) horizontal, (c) vertical, and (d) lateral velocities. 272
11.8 Case TOJ2: Instantaneous mixture fraction from (a) Eulerian, (b) E-L simulations. 273
11.9 Eulerian vs E-L scalar statistics at $t = 0.5$ s: (a) $Sc = 0.7$, (b) $Sc = 1000$. . . . 274
11.10 Analytical, Eulerian and E-L scalar profiles for a 1D oscillating interface. 275
11.11 E-L components, Case TOJ2-EL: Eulerian (a) mean, (b) fluctuation; Lagrangian (c) mean, (d) fluctuation. 277
11.12 Particle addition, Case TOJ2-EL: Lagrangian source (a) mean, (b) fluctuation. 277
11.13 Particle removal, Case TOJ2-EL: Probability (a) mean, (b) fluctuation; Eulerian source (c) mean, (d) fluctuation. 278
11.14 Particle numbers, Case TOJ2-EL: Total active, added, and removed. 279

Nomenclature

Lower-case Latin

Symbol	Meaning	Unit
a	Low-pass filtering parameter (sharpness limit)	–
a_j	Runge-Kutta weighting (previous update)	–
a_1, b_1	Top-hat filter kernel weights (one dimension)	–
a_2, b_2, c_2	Top-hat filter kernel weights (two dimensions)	–
a_3, b_3, c_3, d_3	Top-hat filter kernel weights (three dimensions)	–
b	Low-pass filtering parameter (filter width limit)	m
b	Low-pass filter kernel	–
b_j	Runge-Kutta weighting (current update)	–
c	Intercept, linear fit	–
c_1, c_2	Arbitrary constants for standardised performance	–
d	Number of dimensions	–
dt	Time-step width	s
dW	Incremental Wiener step	–
f	Particles per unit volume	–
f	External force	$\mathrm{kgm/s^2}$
f_s	Sampling frequency	1/s
g	Acceleration vector due to gravity	$\mathrm{m/s^2}$
g_1, g_2	Independent Gaussian-distributed random numbers	–
h	(Implicit) Large-Eddy Simulation (LES) filter kernel	–
h_j	Runge-Kutta weighting (time-step width)	–
i, j, k	Cartesian (logical) co-ordinates	–
i', j', k'	Local Cartesian co-ordinates (relative to i, j, k)	–
j	High-pass filter kernel	–
k	Turbulent kinetic energy per unit mass	$\mathrm{m^2/s^2}$
k_ϕ	Scalar spectral density	m
l	Characteristic length	m
m	Gradient, linear fit	–
n	Normalised integral length-scale	–
n_f	Number of filter support points	–

Lower-case Latin (cont.)

Symbol	Meaning	Unit
p	Pressure	$\mathrm{kg/ms^2}$
q	Runge-Kutta update	–
r	Radius	m
r	Gradient ratio	–
r^α	Uniformly-distributed random noise field	–
r_ν	Viscosity ratio	–
s	Distance	m
s^p	Particle position	m
t	Time (Eulerian framework)	s
u, v, w	Velocity components	m/s
u_1, u_2	Independent uniformly-distributed random numbers	–
w	WENO scheme weighting	–
$\hat{w}$	WENO scheme weighting (intermediate value)	–
w_ϕ^p	Lagrangian particle scalar contribution	–
w_{TH}	Top-hat filter width	m
x, y, z	Cartesian (physical) co-ordinates	m
x', y', z'	Local Cartesian co-ordinates (relative to x, y, z)	m

Upper-case Latin

Symbol	Meaning	Unit
A	Area	$\mathrm{m^2}$
B	Flux limiter value	–
B	Low-pass filter kernel (three-dimensional)	–
$\mathcal{B}$	Fourier transform of low-pass filter b	–
B_L	Lower boundedness measure	–
B_U	Upper boundedness measure	–
C	Germano model value	–
C^*	Germano model value (estimate)	–
CFL	Courant-Friedrichs-Lewy (CFL) number	–
C_K	Kolmogorov constant	–
C_s	Smagorinksy model constant	–
C_{sim}	Similarity model constant	–
C_w	WALE model constant	–
C_ϕ	Obukhov-Corrsin constant	–
C_1	Sutherland constant	$\mathrm{kg/ms\sqrt{K}}$
D	Diffusivity	$\mathrm{m^2/s}$
D_{num}	Numerical diffusivity	$\mathrm{m^2/s}$
D_{sgs}	Diffusivity (modelled)	$\mathrm{m^2/s}$

Upper-case Latin (cont.)

Symbol	Meaning	Unit
E	Total error	–
E_m	Parallel efficiency	–
E_{sym}	Symmetry error	–
F^c	Convective flux	m^3/s
F^d	Diffusive flux	m^3/s
H	Filter frequency response	–
$\mathcal{I}$	Deconvolution update	–
$\mathcal{J}$	Fourier transform of high-pass filter j	–
Kn	Knudsen number	–
L	Integral length-scale	m
$\mathcal{L}_{ij}$	Resolved shear stress tensor	m^2/s^2
$\mathcal{L}^*_{ij}$	Resolved shear stress tensor (deviatoric part)	m^2/s^2
L^{ϱ}	Lagrangian particle density	–
$\mathrm{LES_IQ}_{\nu}$	Viscosity-based quality index	–
M_c	Convective Mach number	–
N	Number of cells	–
N_d	Number of deconvolution operations	–
N_{DNS}	Number of DNS grid points	–
N_f	Number of filter operations	–
N_f^{lim}	Number of filter operations (limited)	–
N_i, N_j, N_k	Computational domain sizes (cells)	–
N_{MC}	Monte Carlo samples	–
N_p	Number of Lagrangian particles	–
P	Probability	–
P_{MC}	Particle removal probability	–
Pe	Péclet number	–
Pr	Prandtl number	–
R	Radius (alternate)	m
$\bar{R}$	Specific gas constant	J/kgK
$R_{\min}$	Minimum effective radius (for a dimensionless scalar)	–
R_{uu}	Longitudinal auto-correlation function	–
R_{vv}, R_{ww}	Transverse auto-correlation functions	–
R^{α}	Filtered uniformly-distributed random noise field	–
Re	Reynolds number	–
Re_t	Turbulent Reynolds number	–
Re_{Δ}	Grid Reynolds number	–
Re_{λ}	Taylor Reynolds number	–
S	Source or sink terms (for a dimensionless scalar)	1/s
Sc	Schmidt number	–

Upper-case Latin (cont.)

Symbol	Meaning	Unit
Sc_t	Turbulent Schmidt number	–
S_{ij}	Rate-of-strain tensor	1/s
T	Temperature	K
T	Total time	s
$\mathcal{T}$	Integral time-scale	s
T_{ij}	Residual shear stress tensor (sub-test filter scale)	m^2/s^2
T_s	Sutherland temperature	K
TV	Total variation	–
V	Volume	m^3
W	Wiener step	–
Z	Mixture fraction	–

Lower-case Greek

Symbol	Meaning	Unit
α_{ij}	Dynamic Smagorinsky model test-filtered tensor	m^2/s^2
α^p	Lagrangian particle weighting	–
α_{TH}	Top-hat filter scale factor	–
β	Flux limiter parameter (Osher and Sweby schemes)	–
β	Particle removal parameter (second derivative)	–
β	Sinc filter width factor	–
β	WENO scheme smoothness indicator	–
β_{ij}	Dynamic Smagorinsky model grid-filtered tensor	m^2/s^2
γ	Linear WENO scheme weighting	–
γ	Particle addition parameter	–
δ	Discretisation error	–
δ_{ij}	Kronecker delta	–
δ_ϕ	Mixing layer thickness	–
ϵ	Arbitrary small positive quantity	–
ε	Dissipation rate	m^2/s^3
η_B	Batchelor length-scale	m
η_K	Kolmogorov length-scale	m
η_ϕ	Obukhov-Corrsin length-scale	m
θ	Angle	rad
κ	Curvature (for a dimensionless scalar)	–
λ	Convergence factor for deconvolution	–
λ	Mean free path	m
λ	Taylor microscale	m
λ	Particle removal parameter (gradient)	–

Lower-case Greek (cont.)

Symbol	Meaning	Unit
μ	Dynamic viscosity	kg/ms
μ	Mean (Gaussian distribution)	–
ν	Kinematic viscosity	m^2/s
ν_t	Turbulent (modelled) viscosity	m^2/s
ξ	Position (Lagrangian framework)	m
ρ	Density	kg/m^3
ρ^E	Density (Eulerian phase)	kg/m^3
ρ^L	Density (Lagrangian phase)	kg/m^3
σ	Standard deviation (Gaussian distribution)	–
σ_f	Gaussian filter width	m
τ	Time (Lagrangian framework)	s
τ_{ij}	Shear stress tensor	kg/ms^2
τ_{ij}^{sgs}	Residual shear stress tensor (sub-grid scale)	m^2/s^2
τ_i^{ϕ}	Residual scalar flux vector (sub-grid scale)	m/s
ϕ	Generic scalar quantity	–
ϕ^E	Eulerian phase of a generic scalar quantity	–
ϕ_f	Filtered scalar quantity	–
ϕ_F	Generic scalar quantity (face-centred)	–
ϕ^L	Lagrangian phase of a generic scalar quantity	–
ϕ''_{max}	Second derivative, upper limit (w. r. t. cell co-ordinates)	–
ϕ''_{min}	Second derivative, lower limit (w. r. t. cell co-ordinates)	–
ϕ'_{min}	Gradient, lower limit (w. r. t. cell co-ordinates)	–
χ	Instantaneous scalar dissipation rate	1/s
ω	Angular velocity	rad/s
ω	Wavenumber	1/m

Upper-case Greek

Symbol	Meaning	Unit
Δ	Cell size, LES filter width	m
$\digamma$	Reconstruction filter kernel	–
Φ	Fourier transform of ϕ	–
Ψ	Stream function	m^2/s

1 Introduction

The study of the Mechanics of Fluids has formed a key part of the topic of Engineering since the earliest days of science, when Archimedes first outlined the fundamental principles of hydrostatics. Today, the disciplines of Fluid Mechanics and Fluid Dynamics are more important and relevant than ever: in the current global climate, there is an unquestionable need for improvements in all forms of energy acquisition, generation and expenditure. Understanding the flow within a combustion chamber may reveal ways in which such a device can be modified to make more effective use of the fuel it consumes, or to reduce those emissions which may be harmful to the environment. This applies as much to the internal combustion engine of an automobile as it does to the gas turbine of an aircraft or the boiler of a coal-fired power station. Similarly, a considerable appreciation of the governing physics of flows is essential as the demand for alternative, 'renewable' sources of energy increases, with wind turbines and deep water tidal generation lighting the way for the future. Reducing losses due to flow phenomena such as turbulence and drag finds significance from the surface coating of the hull of a cargo ship, to the lining of a trans-Siberian natural gas pipeline or the rear wing of a high-performance Formula 1 sports car. Fortunately for the ardent Fluid Dynamicist striving to comprehend and solve all of these highly challenging problems, these flows – that at once appear to be entirely chaotic and unpredictable – are governed by a relatively simple set of equations.

The solution of these equations has come a long way since the pioneering work of the 19^{th} Century physicists Claude-Louis Navier and George Gabriel Stokes, after whom the governing laws of Fluid Mechanics are known. As partial differential equations an analytical solution is rarely forth-coming, and would be infeasible for a flow of any practical use. However, even during the first half of the 20^{th} Century – well before the advent of computers, at least in mainstream use – scientists such as Richardson [1], Thom [2], and Kawaguti [3], to name a few, were attempting numerical solutions to problems ranging from the flow around a cylinder to the prediction of atmospheric weather conditions. Their work relied on the principle of spatial discretisation, where a region of interest is divided into a number of smaller domains, and a 'Reynolds-Averaged' form of the Navier-Stokes equations (known as the RANS approach) would then be solved 'step-by-step' to provide a numerical approximation to their integration. It should be noted that such early work constituted a jaw-dropping effort: to quote from the 1953 paper of Kawaguti [3], 'numerical integration ... took about one year and a half with twenty working hours every week, with a considerable amount of labour and endurance'. Thankfully, we now have computers to do the majority of the work for us, meaning that Kawaguti's simulation of a cylinder in a viscous fluid could be performed in a matter of seconds, as opposed to months or years. *Computational Fluid Dynamics*, or CFD, now represents the state-of-the-art for the solution of fluid flows.

The field of CFD has recently seen advances in the feasibility of very large and complex numerical simulations, due to the continuing increase in performance of modern computing resources. Presently we find ourselves in a time of High-Performance Computing that could only be dreamed of by Richardson, Thom and Kawaguti; the ability to complete very large and complex simulations is more likely to be limited by the cost of access to High-Performance systems, rather than the performance of those systems themselves. The November 2012 publication of the TOP500® list of the world's most powerful supercomputers [4] includes 23 architectures capable of performing in excess of one PetaFLOP (10^{15} Floating Point Operations per second, as determined through the assessment of the LINPACK benchmark [5]) with the top spot going to the 560640-core, 17.6 PetaFLOP Cray XK7 'Titan' machine of the US Department of Energy (DoE) Oak Ridge National Laboratory. Assuming that the current trends in performance are sustained, it is likely that the first ExaFLOP (10^{18} FLOPs) machines will appear within the next ten years: the Cray 2 supercomputer achieved a performance of one GigaFLOP in 1986; the TeraFLOP milestone was subsequently reached in 1997, by the Intel ASCI Red machine at Sandia National Laboratories, USA, while the PetaFLOP barrier was broken in 2008 by the IBM Roadrunner of the US DoE's Los Alamos facility.

Over the last decade or so, the increasing accessibility of High-Performance systems has encouraged a move away from studying flows solely by numerical solution of the classical RANS equations, to generating time-dependent information using Direct Numerical Simulation (DNS) or Large-Eddy Simulation (LES). These approaches, although considerably more computationally expensive than RANS, are often preferred as they allow small-scale and transient structures to be observed and designed for. The increased detail provided by DNS and LES may be essential for accurate predictions of the complex flows that are found in typical industrial applications, particularly for reactive studies where the myriad chemical interactions between a fuel and an oxidiser must also be considered (although RANS continues to be developed, and remains a powerful tool, for such cases).

The RANS approach provides the quickest means of predicting a wide range of problems as it solves a set of equations averaged in time, and is therefore the standard method used in the majority of contemporary commercial codes. Direct Numerical Simulation, on the other hand, provides time-dependent information by solving the governing equations repeatedly over very small time-steps, while the grid on to which the flow is discretised is fine enough to capture the smallest scales of turbulence. Using such a fine spatial resolution means that the number of grid points required to simulate a useful volume becomes extremely large (a typical DNS may used in excess of 10^9 grid points to simulate a domain of around 10 cm^3), while many time-steps must be completed before a meaningful physical duration has been simulated. The LES approach is similar to DNS in many respects, but alleviates the problem of excessive computational expense by applying a spatial filter to the flow-field, to remove the smallest turbulent structures. This means that the grid resolution and time-step width may be increased, but also necessitates the introduction of a numerical model to account for the effect of those small features that have been filtered out.

Computational Fluid Dynamics simulations are frequently more intricate than merely calculating the flow-field around or through a prescribed geometry. Many of the examples included above, such as the aircraft gas turbine, require the transport of a scalar quantity by that flow-field to be determined as well. In the example of the turbine the transported quantity might be the concentra-

tion of fuel throughout the combustor stage; without knowing this, it would be impossible to know where ignition occurs, what temperatures the turbine blades experience, and how much potentially harmful nitrous oxide is released into the atmosphere. The spatial discretisation schemes that are used to approximate this transport thus form an integral part of any transient numerical simulation, and the fidelity of these schemes is paramount for an accurate prediction.

1.1 Motivation

The motivation for this work stems from the inaccuracies inherent in traditional spatial discretisation schemes, used to approximate the convective flux between two points in a simulated domain, in typical DNS and LES. Depending on the scheme used, and the topology of the transported scalar quantity, the transport of that quantity may lead to either oscillatory behaviour, resulting in instabilities or unbounded results, or numerical diffusion, where the scheme is incapable of accurately describing a sharp change in scalar gradient. Such numerical artefacts are clearly undesirable: the more favourable outcome, for the CFD Engineer at least, is that numerical oscillations in a simulation will simply cause it to fail or deliver obviously implausible results; far more ominous, however, are the simulations that appear to deliver reasonable results, but that are, in fact, quite wrong, due to the over-estimation of diffusive fluxes within the simulated domain. In these cases, what the Engineer observes on the computer screen may differ significantly from what they will measure once their design has been put into production.

1.2 Objectives

The purpose of this work is to develop and test a novel method for the approximation of convective fluxes within the Eulerian framework of a DNS or LES, by combining the Eulerian description with a Lagrangian (particle-based) technique. In the present context, such a method could potentially help to improve the prediction of combustion processes and products, lead to improved combustor design, and facilitate the reduction of hazardous emissions at the design stage for future applications. The method would also be applicable to problems relating to the mixing of fluids, particularly at increased Schmidt numbers, or for the simulation of two-phase flows. In addition, the wider Fluid Mechanics topics of Aerodynamics and Meteorology may benefit; the method could also find use in any other fields of science that depend upon the solution of transient partial differential equations containing a convective term. For example, within the other disciplines of Engineering, the wave equation and Lorenz equation are both widely used in a number of contexts; further afield, in the world of finance modelling, the Black-Scholes equation [6] describes the price of a stock option over time, and has become a cornerstone of the subject.

Before outlining the concept of the present work, it is instructive to briefly reiterate what the two terms mean in the context of Fluid Mechanics. In the Lagrangian framework, a single point or particle within the fluid flow is tracked; the Eulerian specification, on the other hand, considers a fixed point in space, where the fluid passing through that point is observed. A useful analogy is to imagine the movement of aircraft through a busy German airport: Leonhard Euler sits in the

Berlin Air Traffic Control tower, and observes the number of planes on the runway; Joseph-Louis Lagrange[1], meanwhile, is in the cabin of a flight from Turin, and checks his own position and velocity as he brings his aircraft in to land. If we consider the airspace of Western Europe to be a control volume, we can see that Euler and Lagrange form a part of the same system within that volume; both would be able to provide information regarding the mechanics of that system, although from very different viewpoints.

The proposed *joint Eulerian-Lagrangian method* is formulated around the decomposition of a scalar quantity into its high- and low-frequency components, by the convolution of that quantity with a low-pass spatial filter; the high-frequency component is subsequently determined as the difference between the initial field and the (low-frequency) filtered field. The smooth, filtered field is transported using a traditional Eulerian scheme, while the high-frequency part is described and transported using Lagrangian particles. After transport, an Eulerian description of the Lagrangian particle field is reconstructed by a weighted averaging from the particle positions on to the underlying Eulerian mesh, and the final scalar field is calculated as the summation of this reconstruction and the transported low-frequency contribution. It is hoped that such an approach will be capable of providing a more accurate description of the convective transport of a scalar quantity within a simulated flow, than would otherwise be possible through an Eulerian description alone.

1.3 Thesis Structure

The structure of this thesis is as follows. Chapter 2 outlines the theory behind the topic of Computational Fluid Dynamics, describes the equations that govern the simulation of a fluid flow, and introduces the main approaches for the numerical solution of that flow. Chapter 3 explains in greater detail how the governing equations are implemented, particularly within the frameworks of DNS and LES, and introduces the `PsiPhi` flow-solving code on which the present work is based. Chapter 4 concentrates on the spatial discretisation of the convective and diffusive flux terms appearing in the governing equations, and provides a more detailed explanation of the motivation for the joint Eulerian-Lagrangian method. Chapter 5 presents an in-depth description of the method, in both its theoretical formulation and its application within a computational code. In Chapter 6 the method is applied to a simple, two-dimensional, forced flow test case for the first time, and this case is subsequently modified to widen the scope of the test in Chapter 7. The results from these studies shed light on potential developments and improvements, which are explored in Chapter 8. An updated version of the method is then applied to a more complex two-dimensional forced flow case, in Chapter 9. Having been satisfied by the performance of the method in two dimensions, it is subsequently applied to a realistic, three-dimensional shear layer case, presented in Chapter 10 (with grid resolutions of a DNS, such that no sub-grid scale modelling is applied), and the LES of a Turbulent Opposed Jet (requiring the inclusion of a turbulence model), in Chapter 11. Finally, Chapter 12 draws conclusions regarding the performance and applicability of the joint Eulerian-Lagrangian method, and suggests possible avenues for further work.

[1] Born Giuseppe Luigi Lagrancia

2 Theory and Simulation of Fluid Flow

This chapter begins by outlining the theory of Fluid Mechanics, and describing the equations that govern the simulation of a fluid flow. It then explains how these equations are applied to the solution of that flow in a Computational Fluid Dynamics (CFD) simulation.

2.1 Fundamental Theory

Although the dynamics of 'real-world' fluids – such as breaking ocean waves, or tropical weather systems – may seem to be entirely chaotic, a limited set of assumptions and equations may be used to describe almost any flow of practical engineering use.

2.1.1 Assumptions

In the present work, three main assumptions are made. Firstly, the continuum assumption states that, although matter is made up of numerous discrete particles, at the macro-scale it is sufficient to treat that matter as being continuous. This means that individual inter-particle interactions may be neglected; without such an assumption, almost all materials and fluids science would become prohibitively complex. Continuum is assumed for the Fluid Mechanics approaches that will be covered in this chapter, and only becomes invalid when the characteristic length l of the flow (for example, the channel width) is comparable to the mean free path λ of molecules within that flow (typically in the order of tens of nanometres, at standard temperature and pressure). The dimensionless Knudsen number Kn is defined as the quotient of λ and l [7], and may be used to classify a flow in terms of the detail with which inter-particle interactions must be treated [8, 9, 10].

Secondly, fluids are assumed to be Newtonian, which is to say that when a shear stress τ is applied, the rate of strain (deformation) will be directly proportional:

$$\tau = \mu \frac{du}{dy} \tag{2.1}$$

Where the constant of proportionality is equal to the fluid dynamic viscosity μ. The majority of fluids behave in this manner, although a few notable exceptions exist: it would be possible to walk across a swimming pool filled with a cornflour suspension, as the impacts of one's feet will cause a local thickening (increased viscosity) and solidification. Conversely, liquefaction (decreased viscosity) of underlying sediments may exacerbate damage during an earthquake (as observed in Alaska, 1964 [11] and Kobe, 1995 [12]).

Finally, fluids are assumed to be incompressible at low speeds. Below a Mach number of 0.3 (around 100 m/s in air, at standard temperature and pressure) it is assumed that the density of a

fluid does not depend on pressure, i.e.:

$$\frac{d\rho}{dp} = 0 \tag{2.2}$$

Where ρ is the fluid density and p is the pressure. Many CFD codes rely heavily upon this assumption, and although a different approach is required for high-speed flows, many cases of interest may be treated as incompressible. It should be noted, however, that incompressibility does not imply constant density: while the assumption represents the independence of density on pressure, density may vary with temperature or chemical state (for example, in reacting flows).

2.1.2 Equations of Fluid Motion

A simple Fluid Mechanics problem may be solved by the application of two equations: through the conservation of mass, otherwise known as continuity, and the conservation of momentum. These are typically presented in a form known as the Navier-Stokes equations.

Continuity states that mass may be neither created nor destroyed:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{2.3}$$

Where t is time, x is position, u is velocity, and the subscript i represents the Cartesian directions. The first term of Eq. 2.3 represents the rate of change of mass within a given (control) volume, while the second term is the net convection of mass into or out of that control volume.

The conservation of momentum accounts for the motion of the fluid, deformation due to applied stresses, and external forces. This may be written as:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = \frac{\partial}{\partial x_j}\tau_{ij} - \frac{\partial p}{\partial x_i} + f_i \tag{2.4}$$

Where the stress tensor τ_{ij} represents deformation due to shear, and f_i represents the sum of any external forces. Applying the assumption that the fluid in question is Newtonian (Eq. 2.1), and Stokes' hypothesis that the trace of the stress tensor (i.e. the sum of the normal stresses) is zero, the term τ_{ij} in Eq. 2.4 may be expressed as:

$$\tau_{ij} = \mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) \tag{2.5}$$

Where δ_{ij} is the Kronecker symbol ($\delta_{ij} = 1$ for $i = j$; $\delta_{ij} = 0$ otherwise). Assuming that the only external force acting upon the fluid is acceleration due to gravity g, and substituting Eq. 2.5 into Eq. 2.4, we have:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] - \frac{\partial p}{\partial x_i} + \rho g_i \tag{2.6}$$

Finally, it is convenient to express the first term on the RHS of Eq. 2.6 in terms of the rate-of-strain tensor S_{ij}:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \tag{2.7}$$

Which results in:

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = \frac{\partial}{\partial x_j}\left(2\mu S_{ij}\right) - \frac{\partial p}{\partial x_i} + \rho g_i \tag{2.8}$$

The terms of Eq. 2.8 represent, from left to right: the rate of change of momentum within the control volume; the net convection of momentum into or out of the control volume; the dissipation of momentum due to viscous forces; net forces due to pressure acting on the control volume, and the action of gravity (buoyancy). Equations 2.3 and 2.8 are some of the most significant and useful in the field of Fluid Mechanics; analytical solution of the (time-averaged) Navier-Stokes equations would provide a (statistical) description for any macroscopic, incompressible, Newtonian flow. However, such analytical solutions are intractable for all but the simplest of problems, which motivates the use of a numerical approximation to their description.

In many cases the purpose of a CFD code lies beyond solely the solution of a flow-field. For example, a great deal of research continues to focus on reactive flows (i.e. combustion), and for this it is typically necessary to transport fuel, oxidiser, and other species mass fractions within the flow. For these quantities a general transport equation may be applied:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_j}(\phi u_j) = \frac{\partial}{\partial x_j}\left(D_\phi \frac{\partial \phi}{\partial x_j}\right) + S_\phi \tag{2.9}$$

Where ϕ is a conserved scalar quantity, D_ϕ is the diffusivity of that quantity, and S_ϕ represents any sources (additive) or sinks (subtractive). From left to right, the terms of Eq. 2.9 then represent: the net rate of change of the quantity ϕ within the control volume; the convection of ϕ across the boundaries of the control volume; the diffusion of ϕ across those boundaries, and the creation or consumption of ϕ within the control volume. A typical transported scalar might be the mixture fraction Z of a fuel, for which Eq. 2.9 would be written as:

$$\frac{\partial Z}{\partial t} + \frac{\partial}{\partial x_j}(Z u_j) = \frac{\partial}{\partial x_j}\left(D_Z \frac{\partial Z}{\partial x_j}\right) + S_Z \tag{2.10}$$

Note that applying Eq. 2.9 for density ρ, which has no source term ($S_\rho = 0$) and is not diffused ($D_\rho = 0$), recovers the continuity equation; a similar transformation may be achieved for the momentum conservation equation, where viscosity is analogous to the momentum diffusivity. For other scalars, it is useful to define the ratio of momentum and mass diffusivity:

$$Sc = \frac{\nu}{D} \tag{2.11}$$

Where ν is the kinematic viscosity, defined as μ/ρ, and Sc is the dimensionless Schmidt number, which is typically of the order of unity for gas mixtures, but may be significantly higher for liquids.

The Schmidt number is closely related to the Prandtl number Pr, which is defined as the ratio of momentum and thermal diffusivity, and the Péclet number Pe, which is the ratio of convective transport to diffusive transport of a scalar quantity within a flow. (The Péclet number may be determined as the product of Sc (for mass transport) or Pr (for heat transport) with the Reynolds number Re, which will be introduced shortly).

2.1.3 Turbulence

Fluid flow may be classified as being either laminar or turbulent. A laminar flow is characterised by relatively high momentum diffusion and low momentum convection, such that viscous forces are dominant; fluid particles will tend to move parallel to one another, with little or no disruption to the particle trajectory. Accordingly, a laminar flow is often (but not always) steady, which is to say that it does not vary with time, and is one-dimensional. Conversely, a turbulent flow: is dominated by inertial forces; is characterised by its chaotic appearance; contains a large range of structure sizes; and is always three-dimensional.

In 1883, Reynolds attempted to formalise the classification of flow behaviour, by performing a series of experiments using streams of liquid dye in water running through transparent tubes [13]. Reynolds ascertained that the nature of the flow depended upon four factors: the velocity, dynamic viscosity and density of the fluid, and the characteristic dimension l of the flow (in this case, the pipe diameter). He therefore classified his flows using the dimensionless group now referred to as the Reynolds number [14]:

$$Re = \frac{\rho u l}{\mu} = \frac{u l}{\nu} \tag{2.12}$$

In practice the Reynolds number is only able to provide an indication of the nature of the flow, while the actual behaviour depends on considerably more factors. Typically, however, a flow will be laminar for $Re < 2000$. At higher Reynolds numbers a laminar flow will undergo a transition to turbulence, although under the correct conditions laminar flow can be maintained at relatively high speeds or for relatively low-viscosity fluids (i.e. high Re).

Reynolds also formulated a decomposition of the 'steady' and 'unsteady' components of the velocity in a turbulent flow, where the steady component is defined as:

$$\langle u_i(x_j) \rangle = \lim_{T \to \infty} \left(\frac{1}{T} \int_0^T u_i(x_j, t)\, dt \right) \tag{2.13}$$

Where the $\langle \cdots \rangle$ operator denotes the time average and T is the total time. The unsteady part then represents the instantaneous fluctuation about that mean, such that:

$$u_i(x_j, t) = \langle u_i(x_j) \rangle + u_i'(x_j, t) \tag{2.14}$$

Where u_i' is the instantaneous velocity fluctuation. The Reynolds decomposition expressed in Eq. 2.14 finds application in the formulation of the Reynolds-Averaged Navier-Stokes (RANS) equations, which will be covered in §2.2.

2.1.4 Turbulent Scales

The rotating features (eddies) within a turbulent flow have a wide range of sizes, from the order of metres (in large-scale flow domains, for example around an aircraft in flight) down to tenths of a millimetre. However, three measures are frequently used to provide an idea of the maximum, typical, and minimum structure sizes that are observed in a flow, and these are the integral length-scale, the Taylor microscale, and the Kolmogorov length-scale, respectively, each of which has a corresponding time- and velocity-scale associated with it.

To appreciate the significance of these scales, an understanding of Richardson's concept of the energy cascade [1] is beneficial. Within a turbulent flow the specific turbulent kinetic energy k is greatest in the large eddies, as they are the source of its production. As the large eddies progress through the flow, they break up; in doing so, they pass energy from the large scales to the smaller ones, while also losing energy as heat. Eventually, the eddies become so small that they start to dissipate under viscous forces, until they disappear from the flow completely. At this point all of the turbulent kinetic energy has been converted into thermal energy, and the rate at which this conversion occurs is referred to as the dissipation rate, ε. Figure 2.1 shows the concept of the energy cascade, with the three turbulent 'ranges' through which energy is transferred, where k is shown as a function of the wavenumber ω.

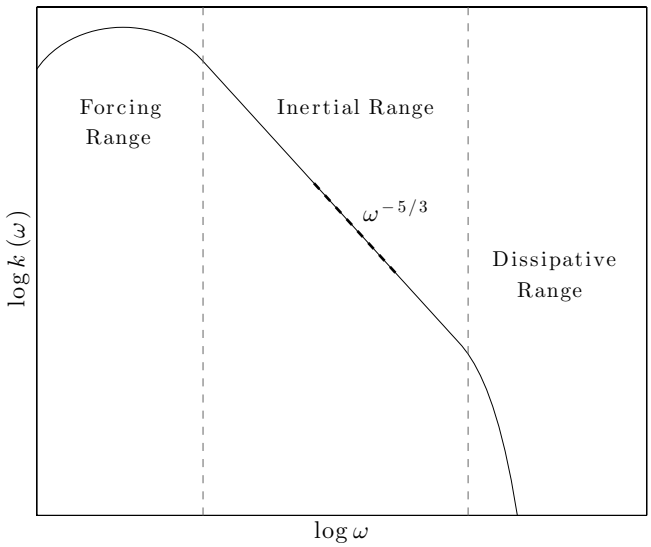


Figure 2.1: The energy cascade.

The integral scales, Taylor microscales, and Kolmogorov scales are thus the characteristic scales of the forcing, inertial, and dissipative turbulence ranges, respectively. The relationship between turbulent kinetic energy and wavenumber is described by the expression proposed by Kolmogorov [15]:

$$k(\omega) = C_K\, \varepsilon^{2/3}\, \omega^{-5/3} \tag{2.15}$$

Where C_K is the Kolmogorov constant, which is found to have a universal value of approximately

0.5 from experiments performed under a range of conditions [16]. The integral time-scale $\mathcal{T}$ may be calculated from integration of the normalised auto-correlation function $R_{uu}(\tau)$ of the axial velocity $u(t)$:

$$R_{uu}(\tau) = \frac{\langle u(t)\, u(t+\tau)\rangle}{\langle u^2\rangle} \tag{2.16}$$

$$\mathcal{T} = \int_0^\infty R_{uu}(\tau)\, d\tau \tag{2.17}$$

Note that the auto-correlation of the velocity across two points in time (at t and $t+\tau$) – rather than two points in space – relies on the application of Taylor's hypothesis, which assumes that the turbulence in u is statistically stationary. Essentially, the velocity auto-correlation is a measure of how much the velocity at a given point is related to velocities near to that point, and thus allows the extent of the rotating structures in the flow to be estimated. Calculation of the integral length-scale L, as the product of time-scale $\mathcal{T}$ and the bulk flow velocity, allows the turbulent Reynolds number Re_t to be found:

$$Re_t = \frac{u' L}{\nu} \tag{2.18}$$

Where, in this case, u' is taken to be the standard deviation of the axial flow velocity, which will be referred to as the velocity fluctuation. Richardson's initial concept of the energy cascade was expanded upon by Kolmogorov [15, 17], who proposed a length-scale to describe the size of the smallest eddies in a flow:

$$\eta_K = \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}} \tag{2.19}$$

The Kolmogorov scale η_K thus represents the lower limit of turbulent scales that must be resolved (or modelled) in an isothermal simulation. However, this assertion may not apply to reactive simulations, where chemical length- and time-scales may be significantly smaller. (In their textbook, Warnatz, Maas and Dibble indicate that chemical time-scales may be as short as 10^{-10} s, compared to turbulent time-scales in the order of 10^{-5} s; see [18], Fig. 7.10).

2.2 Time-Averaged Simulations

RANS simulation is, commercially, the most widely-used approach for the simulation of Fluid Mechanics problems, not least because it was one of the earliest methods to be widely accepted, and proves to be relatively inexpensive in terms of computational cost. The RANS equations are formulated by inserting the Reynolds decomposition (Eq. 2.14) for each of the terms appearing in the incompressible Navier-Stokes equations (Eqs. 2.3 and 2.8):

$$\frac{\partial}{\partial x_i}(\langle u_i\rangle + u_i') = 0 \tag{2.20}$$

$$\frac{\partial(\langle u_i\rangle + u_i')}{\partial t} + \frac{\partial}{\partial x_j}\left[(\langle u_i\rangle + u_i')(\langle u_j\rangle + u_j')\right] = \frac{\partial}{\partial x_j}\left[2\nu\left(\langle S_{ij}\rangle + S_{ij}'\right)\right] - \frac{1}{\rho}\frac{\partial(\langle p\rangle + p')}{\partial x_i} + g_i \tag{2.21}$$

Where density has been assumed to be constant. Averaging of Eqs. 2.20 and 2.21 over time, while considering that the mean of the fluctuating part of a quantity is zero (e.g. $\langle u_i' \rangle = 0$) and that the time derivative of a time-averaged quantity is zero, leads to:

$$\frac{\partial \langle u_i \rangle}{\partial x_i} = 0 \tag{2.22}$$

$$\frac{\partial}{\partial x_j}(\langle u_i \rangle \langle u_j \rangle) + \frac{\partial}{\partial x_j}(\langle u_i' u_j' \rangle) = \frac{\partial}{\partial x_j}(2\nu \langle S_{ij} \rangle) - \frac{1}{\rho}\frac{\partial \langle p \rangle}{\partial x_i} + g_i \tag{2.23}$$

Where the second and third terms on the LHS of Eq. 2.23 result from time-averaging of the convection term (second term of the LHS of Eq. 2.21) after it has been expanded. Finally, the chain rule is applied to the second term on the LHS of Eq. 2.23:

$$\frac{\partial}{\partial x_j}\left(\langle u_i \rangle \langle u_j \rangle\right) = \langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} + \langle u_i \rangle \frac{\partial \langle u_j \rangle}{\partial x_j} \tag{2.24}$$

Where the second term on the RHS of Eq. 2.24 is zero from continuity (Eq. 2.22). The terms of Eq. 2.23 may thus be re-arranged into:

$$\langle u_j \rangle \frac{\partial \langle u_i \rangle}{\partial x_j} = \frac{\partial}{\partial x_j}\left(2\nu \langle S_{ij} \rangle - \langle u_i' u_j' \rangle\right) - \frac{1}{\rho}\frac{\partial \langle p \rangle}{\partial x_i} + g_i \tag{2.25}$$

Notably, the time-averaging has created a new tensor term $\langle u_i' u_j' \rangle$, referred to as the Reynolds stresses, which is unknown. While new transport equations may be formulated to close this term, each new equation creates a further term that requires closure. The Reynolds stresses are therefore modelled, for example with the classical k–ε approach of Jones and Launder [19].

A similar decomposition and time-averaging may be performed for the general transport equation (Eq. 2.9). Substitution of the Reynolds decomposition of a scalar ϕ into its mean $\langle \phi \rangle$ and fluctuating ϕ' parts yields:

$$\frac{\partial(\langle \phi \rangle + \phi')}{\partial t} + \frac{\partial}{\partial x_j}\left[(\langle \phi \rangle + \phi')(\langle u_j \rangle + u_j')\right] = \frac{\partial}{\partial x_j}\left(D_\phi \frac{\partial(\langle \phi \rangle + \phi')}{\partial x_j}\right) + S_\phi \tag{2.26}$$

While subsequently averaging in time, considering that $\langle \phi' \rangle = 0$, and performing some re-arrangement as before, gives:

$$\langle u_j \rangle \frac{\partial \langle \phi \rangle}{\partial x_j} = \frac{\partial}{\partial x_j}\left(D_\phi \frac{\partial \langle \phi \rangle}{\partial x_j} - \langle u_j' \phi' \rangle\right) + S_\phi \tag{2.27}$$

Again there appears an unclosed term $\langle u_j' \phi' \rangle$, representing turbulent (or Reynolds) scalar fluxes, which requires modelling.

RANS simulations typically use relatively coarse grids, as all of the turbulent scales are modelled and do not need to be resolved. However, this means that it can be difficult to create a computational mesh that accurately represents the flow, and a degree of experience is required for the correct interpretation and analysis of simulation results. Another major drawback of RANS is that

it provides only time-averaged solutions, such that instantaneous representations of the flow are impossible (although coarsely time-dependent simulations are possible through application of the 'unsteady' RANS, or URANS, approach).

Research into RANS as a tool for industry and academia continues to deliver significant extensions to the approach, particularly with regards to the treatment of reaction modelling, even as the more detailed Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES) approaches become more practical. Extensions to RANS are often jointly applicable to LES, where one of the most notable developments that has been adopted for LES from RANS is the use of the steady flamelet approach for the modelling of non-premixed combustion, introduced by Williams [20]. This method considers the dependence of quantities such as density, temperature, or species mass fraction upon mixture fraction, where the simulation of a one-dimensional flame (at a much higher degree of chemical complexity than would feasible within the LES or RANS simulation itself) is used to create a 'look-up' reference table for these quantities.

The joint Probability Distribution Function (PDF) method was first introduced for combustion problems by Dopazo and O'Brien [21] as a way to improve the modelling of mixing and reaction in RANS simulations, while subsequent development was carried out by Pope [22] and Kollmann [23] amongst others. One of the more significant extensions is the Lagrangian PDF approach of Pope [24], although the computational expense of the method (which is perhaps comparable to that of a typical LES) has proved to be limiting to its application. The Conditional Moment Closure (CMC) approach was developed by Klimenko and Bilger [25], where reaction rate models are based on conditional mean quantities, and has demonstrated good performance when compared to data from DNS [26, 27]. Original implementations of CMC considered only first-order terms, where fluctuations around conditional means were assumed to be negligible, although the technique has been extended to account for second moments by Kronenberg et al. [28], Mastorakos and Bilger [29] and others.

Multiple Mapping Conditioning (MMC) is one of the most recent models, and has been developed by Klimenko and Pope [30], Klimenko [31] and Wandel and Klimenko [32]. MMC provides a compromise between the simpler CMC approach and the computationally-expensive joint PDF method by considering the 'importance' of each chemical species, where in the simplest case it is equivalent to CMC and in the most complex it replicates joint PDF. This new method has been successfully applied by Cleary and Kronenburg [33], Cleary and Klimenko [34], Vaishnavi and Kronenburg [35] and Vogiatzaki et al. [36], amongst others.

2.3 Time-Dependent Simulations

Of more relevance to the present work are the DNS and LES approaches, which provide a time-dependent numerical integration of the Navier-Stokes equations.

2.3.1 Direct Numerical Simulation

In a DNS the simulated domain is discretised on to a very fine computational grid, requiring a very short time-step width (q.v. §3.3), such that the Kolmogorov length- and time-scales are fully resolved. (As stated previously, however, in reactive simulations it is sometimes necessary to resolve

the relevant chemical scales, which may be even smaller). The smallest eddies within the flow are accurately represented by the grid, and no turbulence modelling is required. The immediate problem with DNS is then the significant computational expense involved: the groups of Chen at Sandia National Laboratories, USA [37], and Vervisch, at CORIA, INSA de Rouen, France [38], perform computations on grids of $O(10^9)$ points, with which it is only possible to simulate a very short duration of physical time[1]. The DNS approach is also restricted to relatively low Reynolds number flows: by considering the integral and Kolmogorov scales that must be resolved, an estimate of the required number of DNS grid points (N_{DNS}) may be determined:

$$N_{\mathrm{DNS}} > Re_t^{9/4} \tag{2.28}$$

Doubling the flow velocity fluctuation in a simulation therefore demands a five-fold increase in N_{DNS}, for exactly the same fluid, geometry, and domain size. For these reasons DNS has only become popular within the last few years, and becomes more applicable as High-Performance Computing resources become cheaper, more powerful, and more readily available. However, the continued feasibility of DNS requires that codes are ported to, and make effective use of, the latest systems. The most recently-developed machines consist of millions of simple processing cores (General Purpose Graphical Processing Units, or GPGPUs), and although they are capable of very rapid computations, they currently pose additional challenges for software development due to the way in which data transfer to the GPGPU must be carefully managed.

In order to ensure that the maximum possible accuracy is maintained, DNS codes usually employ very high-order spatial and temporal discretisation schemes: a typical example might apply a tenth-order scheme in space (such that the discretisation error δ is proportional to Δ^{-10}, where Δ is the grid spacing), and fourth-order in time (i.e. $\delta \propto dt^{-4}$, where dt is the time-step width).

Although DNS is currently infeasible for large-scale simulations of long duration, it finds use in the validation of new models such as the PDF, CMC and MMC methods described in §2.2. Large databases of DNS data have been created within many academic institutions for this purpose, and prove invaluable for cases where comparable experimental data does not exist.

2.3.2 Large-Eddy Simulation

LES has been developed to provide an intermediate solution between the RANS (§2.2) and DNS (§2.3.1) approaches. LES is performed in much the same way as a DNS, but the requirement of a fine grid resolution is relaxed by filtering out the smallest turbulent (or more generally, scalar) scales and replacing them with a modelled contribution. This spatial filtering may be described as the convolution of each simulated scalar quantity ϕ with a filter function $h(x_i - x_i')$:

$$\bar{\phi}(x_i, t) = \iiint_{-\infty}^{\infty} \phi(x_i', t)\, h(x_i - x_i')\, dx_i' \tag{2.29}$$

[1] Much of Chen's work has been performed on the Jaguar Cray XT5-HE system at the Oak Ridge National Laboratory, USA, which was, at one point, the fastest computer in the world [39]

Where $\bar{\phi}$ is a spatially-filtered quantity, and x_i' is a co-ordinate system local to x_i (i.e. $x_i' = 0$ at x_i). This results in the decomposition of a quantity into a resolved mean and an unresolved 'sub-grid' fluctuation, similar to the Reynolds decomposition of Eq. 2.14, but in space rather than in time. In the majority of LES codes the filter width Δ is chosen to be equal to the grid resolution (i.e. the finite volume cell size), as per the suggestions of Deardorff [40] and Schumann [41], such that the filtering is implicit. Applying this filter to the incompressible, constant density Navier-Stokes equations, with manipulation similar to that for the derivation of the RANS equations (Eqs. 2.22 and 2.25), yields:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2.30}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u_i u_j}) = \frac{\partial}{\partial x_j}\left(2\nu \bar{S}_{ij}\right) - \frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + g_i \tag{2.31}$$

The filtered product $\overline{u_i u_j}$ appearing in the second term on the LHS of Eq. 2.31 includes the sub-grid scale (SGS) turbulent stresses, and cannot be solved directly. This term is therefore dealt with by substituting the (known) quantity $\bar{u}_i \bar{u}_j$, and defining the residual stress tensor τ_{ij}^{sgs}, suggested by Leonard [42], as:

$$\tau_{ij}^{\text{sgs}} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j \tag{2.32}$$

This term must be treated with a closure model. The LES-filtered momentum conservation equation can then be written as:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{\partial}{\partial x_j}\left(2\nu \bar{S}_{ij} - \tau_{ij}^{\text{sgs}}\right) - \frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_i} + g_i \tag{2.33}$$

Where the chain rule has been applied to the derivative of the last term of Eq. 2.32 (cf. Eq. 2.24), and continuity (Eq. 2.30) has been substituted for one of the resulting terms.

It is also useful to derive an LES-filtered form of the general transport equation (Eq. 2.9), which reads:

$$\frac{\partial \bar{\phi}}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u_j \phi}\right) = \frac{\partial}{\partial x_j}\left(\bar{D}_\phi \frac{\partial \bar{\phi}}{\partial x_j}\right) + \bar{S}_\phi \tag{2.34}$$

Where it has been assumed that the LES-filtered diffusive term may be approximated as the product of the filtered terms:

$$\overline{D_\phi \frac{\partial \phi}{\partial x_j}} \approx \bar{D}_\phi \frac{\partial \bar{\phi}}{\partial x_j} \tag{2.35}$$

Once again there appears an unclosed term $\overline{u_j \phi}$ in Eq. 2.34, which in this case includes the sub-grid scale scalar fluxes. This is dealt with by defining a residual scalar flux τ_i^ϕ:

$$\tau_i^\phi = \overline{u_i \phi} - \bar{u}_i \bar{\phi} \tag{2.36}$$

The modelling of the SGS scalar fluxes may be achieved via a similar approach to that of the

SGS turbulent stresses, and will be covered in Chapter 3. Substituting Eq. 2.36 into Eq. 2.34 allows the general transport equation for LES to be written as:

$$\frac{\partial \bar{\phi}}{\partial t} + \bar{u}_j \frac{\partial \bar{\phi}}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\bar{D}_\phi \frac{\partial \bar{\phi}}{\partial x_j} - \tau_j^\phi\right) + \bar{S}_\phi \tag{2.37}$$

While these LES-filtered forms of the Navier-Stokes and transport equations are useful for simple, constant density cases, in the present context it is also necessary to consider forms that are applicable to the simulation of variable-density mixing and reacting fluids. In this case the density cannot be taken outside of the temporal and spatial derivatives appearing in the conservation equations, such that the application of a spatial filter (of the form in Eq. 2.29) generates filtered products (for example, $\overline{\rho u_i}$) that are unclosed. It is therefore common practice to apply a density-weighted, or Favre, average:

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \quad \rightarrow \quad \overline{\rho\phi} = \bar{\rho}\tilde{\phi}, \quad \text{with} \quad \phi = \tilde{\phi} + \phi'' \tag{2.38}$$

Where $\tilde{\phi}$ is a Favre-filtered quantity, and ϕ'' is the corresponding fluctuation. Following the approach of Vreman et al. [43], the application of a density-weighted filter to the Navier-Stokes and general transport equations subsequently yields:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0 \tag{2.39}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{u}_i\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_i\tilde{u}_j\right) = \frac{\partial}{\partial x_j}\left(2\mu\tilde{S}_{ij} + \bar{\rho}\tilde{\tau}_{ij}^{\text{sgs}}\right) - \frac{\partial \bar{p}}{\partial x_i} + \bar{\rho}\tilde{g}_i \tag{2.40}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{\phi}\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\phi}\right) = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\tilde{D}_\phi \frac{\partial \tilde{\phi}}{\partial x_j} + \tilde{\tau}_j^\phi\right)\right] + \bar{S}_\phi \tag{2.41}$$

Where, in this case, the rate-of-strain tensor $\tilde{S}_{ij}$, residual stress tensor $\tilde{\tau}_{ij}^{\text{sgs}}$ and residual scalar flux $\tilde{\tau}_j^\phi$ are based on (gradients of) Favre-filtered quantities, and the transported quantity ϕ has been multiplied by the local density. Following [44], it has also been assumed that the Favre-filtered diffusive term may be approximated as the product of the filtered terms:

$$\widetilde{D_\phi \frac{\partial \phi}{\partial x_j}} \approx \tilde{D}_\phi \frac{\partial \tilde{\phi}}{\partial x_j} \tag{2.42}$$

Recent extensions to LES include the development of Flame Surface Density (FSD) models for the simulation of premixed combustion, as suggested by Colin et al. [45], Fureby [46], Charlette et al. [47], Gülder [48], Knikker et al. [49], Weller et al. [50] and others, and Linear Eddy Modelling (LEM) developed principally by Kerstein [51, 52, 53, 54, 55, 56, 57]. Both of these techniques have been investigated by, for example, Stein [58]. Efforts to assess the accuracy of LES have been made by Celik et al. [59], Klein [60], Kempf et al. [61], Geurts and Frölich [62] and Meyers et al. [63], to name a few.

2.4 Summary

In this chapter the underlying theories of Fluid Mechanics have been introduced. The central assumptions of continuity, linear (Newtonian) deformation and incompressibility have been described; the mass and momentum conservation equations have been presented in their usual form as the Navier-Stokes equations, along with the general transport equation; and the key concepts of turbulence have been outlined.

The governing equations of Fluid Mechanics have also been manipulated into forms that allow them to be solved computationally, either temporally-averaged in the context of a Reynolds-Averaged Navier-Stokes (RANS) simulation, spatially-averaged for a Large-Eddy Simulation (LES), or directly in a Direct Numerical Simulation (DNS). In Chapter 3 we will consider in greater detail the practicalities of how these equations may be solved by DNS and LES. Other extensions to LES, such as the continuing development of sub-grid scale turbulence and scalar flux models, will also be covered.

3 Simulation Numerics

In this chapter the means by which we may recover approximate solutions to the equations introduced in Chapter 2 will be described, with an emphasis on the time-dependent approaches of Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES). The discussion will include the manner in which a problem is discretised in space and time; the use of pressure correction as a tool to solve the Navier-Stokes equations; the modelling of filtered (sub-grid scale) transport terms in an LES; and the necessary prescription of boundary conditions to the computational domain. The `PsiPhi` code that forms the basis for the development of the present work is also introduced.

The discretisation of convective and diffusive fluxes will be treated in depth in Chapter 4, as it constitutes the focus of the present work.

3.1 Spatial Discretisation: Finite Volume Method

In general, Computational Fluid Dynamics codes – like many other codes designed to simulate physical phenomena – apply a spatial discretisation to the governing equations, where continuous functions describing quantities of interest, such as momentum, density, or pressure, are stored at discrete points within a simulated domain. Although different approaches exist, the majority of DNS and LES codes use the Finite Volume Method (FVM). This is based upon the division of a domain (in one, two or three dimensions) into a number of small regions of finite size (length, area or volume, depending on the dimensionality of the problem). Typically the quantities of interest are stored at the centre of each computational 'cell' (control volume), and are assumed to follow some distribution across that cell. The chosen distribution may have some influence on the simulation of each quantity [64]; in the present work, however, a 'top-hat' distribution is assumed, i.e. the value of a scalar is constant throughout the cell.

A further consideration for the spatial discretisation of the governing equations is the shape and size of the cells into which the simulated domain is divided. Although many methods exist, they may be loosely divided into 'structured' and 'unstructured' grids, some examples of which are shown in Fig. 3.1.

Structured grids may apply either a uniform spacing of cells, as shown in Fig. 3.1(a), or a non-uniform spacing (Fig. 3.1(b)). However, the key feature of a structured grid is that the values within each cell may be stored using a Cartesian co-ordinate system. In other words, although the points of the grid may not necessarily be orthogonal (as in Fig. 3.1(c)), the data within the domain may be stored in a regular matrix or array. The local neighbours to a given point may be inferred from the data structure, rather than being explicitly stored. This is very useful in a CFD code, as the change in a scalar quantity at a point in space is typically defined in terms of the values at points

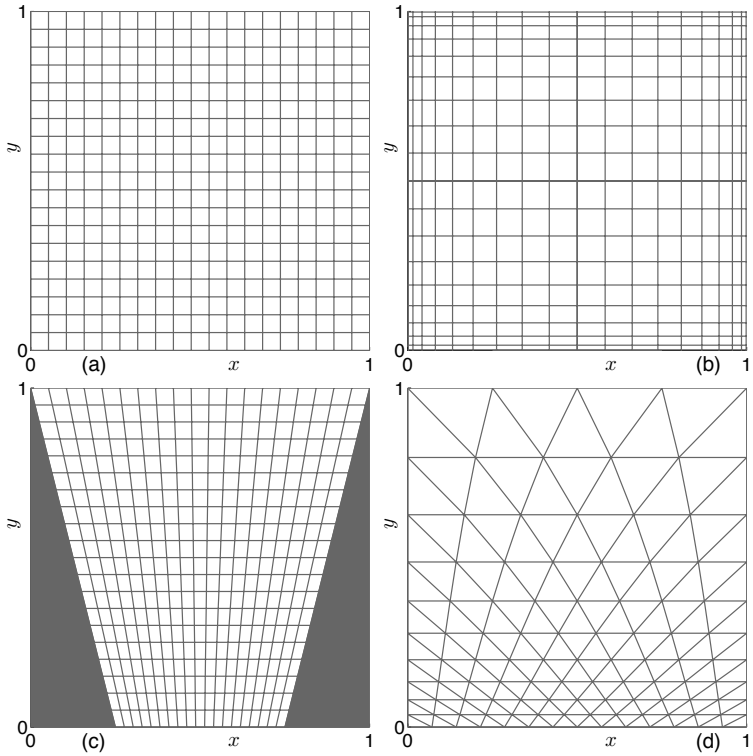


Figure 3.1: Different grid structures: (a) uniform, orthogonal Cartesian; (b) non-uniform, orthogonal Cartesian; non-uniform, oblique Cartesian; (d) unstructured.

adjacent to it. Structured grids may be designed to provide improved spatial resolution in particular regions of the simulated domain, as in Fig. 3.1(b), or to fit around the physical geometry of that domain, as in Fig. 3.1(c), where the shaded regions represent solid walls; however, such refinement or adaptivity will always be restricted by the necessity of the grid to conform to the underlying Cartesian structure.

The unstructured grid shown in Fig. 3.1(d) does not permit such a simple method of data storage, and therefore requires a more complex system of determining the local neighbouring cells; however, it has the distinct benefit of allowing a computational grid to exactly fit around a complex geometry, while also providing the option of locally increased or decreased spatial resolution where necessary.

Throughout its development the main focus of the `PsiPhi` code has been on a simple and computationally-efficient implementation of the governing equations. `PsiPhi` therefore applies a structured, equidistant (cubic) Cartesian grid for the discretisation of those equations.

3.2 Temporal Discretisation

Time-dependent solutions to the Navier-Stokes equations must be numerically integrated in time, where the solution is advanced from a set of initial conditions over a number of finite time-steps. Consider a time-dependent quantity ϕ that varies in time as some function $\mathcal{F}(\phi)$, so that:

$$\frac{\partial \phi}{\partial t} = \mathcal{F}(\phi) \tag{3.1}$$

The discretisation of this equation in time may be written as:

$$\phi(t+dt) \approx \phi(t) + dt\mathcal{F}(\phi(t)) \tag{3.2}$$

The solution to ϕ at time $t+dt$ is approximated from the previous solution at time t, plus some update proportional to the time-step width dt. This method is known as the Euler-explicit scheme, which is first-order accurate in time: the error in the solution at time $t+dt$ is proportional to dt^{-1}.

The means of time integration plays an important role in the temporal accuracy and stability of a simulation. While the Euler-explicit scheme is simple to understand and implement, it is prone to numerical instabilities that may result in divergent solutions. In the present `PsiPhi` code a third-order accurate Runge-Kutta method is applied (named after the German mathematicians C. Runge and M. W. Kutta, who originally developed the family of schemes to which this method belongs). In this case the update from the solution at t to that at $t+dt$ is performed across three sub-time-steps. The chosen scheme is formulated so that each intermediate step is a function only of the preceding step and a time-independent update, meaning that only two 'copies' of the solution must be stored at any one time. (Other third-order Runge-Kutta time integration schemes require all three of the initial and intermediate solutions for the computation of the solution at $t+dt$; this could pose a significant problem for an LES code, where computer memory is often at a premium). This 'low-storage' implementation of the Runge-Kutta method was proposed by Williamson [65], and may be written as:

$$q_j = a_j q_{j-1} + \mathcal{F}(\phi_{j-1})\, dt \tag{3.3}$$

$$\phi_j = \phi_{j-1} + b_j q_j \tag{3.4}$$

$$t_j = t_{j-1} + h_j\, dt \tag{3.5}$$

Where q_j is the update to the solution ϕ_j for the j^{th} sub-time-step, and the weightings a_j, b_j and h_j are determined for the chosen scheme. In the present work, the applied constants are:

$$a_j = \left\{0; \quad -\frac{5}{9}; \quad -\frac{153}{128}\right\} \tag{3.6}$$

$$b_j = \left\{\frac{1}{3}; \quad \frac{15}{16}; \quad \frac{8}{15}\right\} \tag{3.7}$$

$$h_j = \left\{\frac{1}{6}; \quad \frac{3}{10}; \quad \frac{8}{15}\right\} \tag{3.8}$$

An apparent drawback of the Runge-Kutta scheme is that the same computations (in $\mathcal{F}(\phi)$) that are performed once in an Euler-explicit scheme must be completed three times for a full third-order Runge-Kutta time-step. However, for a given accuracy in the solution the time-step width in a Runge-Kutta scheme may be significantly increased compared to that of an Euler-explicit calculation, such that over the course of a full simulation the computational expense is reduced.

It should be noted that many other time integration schemes exist, including semi-implicit, implicit, adaptive and hybrid versions. For a comprehensive summary of these schemes and their myriad variants, the reader is referred to the book by Butcher [66].

3.3 Time-Step Width

Where an explicit time integration scheme is used, a stability limit must be applied to the width of each time-step. In effect, 'information' must not be convected downstream by a distance greater than one cell width; in other words, information may not 'skip' cells. This criterion is formally described by Courant et al. [67], who proposed that a time-dependent simulation must adhere to the 'CFL' condition:

$$\mathrm{CFL} = \frac{|u|\, dt_{\mathrm{CFL}}}{\Delta} \leq 1 \tag{3.9}$$

Where CFL is a user-defined parameter between 0 and 1. The criterion is necessary, although not mathematically sufficient, to ensure the convergence of the solution. It should be noted, however, that the actual time-step width dt may require modification depending on the time integration scheme used, as described by Williamson [65], Shu and Osher [68], and Carpenter and Kennedy [69], where $dt \leq dt_{\mathrm{CFL}}$.

Typically a value approaching CFL = 1 is permissible for a non-reactive case. Note however that Eq. 3.9 applies to convective fluxes only (i.e. depends on the velocity field), and in some simulations the limit of stability may depend on the magnitude of the diffusive flux. For the specific case of a 'central' approximation for that flux (see §4.7) on a one-dimensional uniform grid, combined with an Euler-explicit advancement in time, it may be demonstrated by Fourier stability analysis that the necessary condition for a convergent solution is [70]:

$$\frac{D\, dt}{\Delta^2} \leq \frac{1}{2} \tag{3.10}$$

In the present work diffusive fluxes will typically be smaller than convective fluxes (i.e. the Péclet number $Pe \gg 1$), so that the expression in Eq. 3.9 is usually more restrictive.

3.4 Pressure Correction

In an incompressible Computational Fluid Dynamics code, the solution of the momentum transport terms (whether in the RANS context (Eq. 2.25), in LES (Eq. 2.33), or in a DNS (Eq. 2.8)) will typically result in a failure to satisfy the continuity equation. In order to maintain mass conservation it is therefore necessary to create a pressure field that corrects the mass within each computational cell. In this sense, according to Ferziger and Perić, the pressure term in the incompressible momentum conservation equation becomes more of a mathematical term than a physical one [71]; although its values are still physically representative of local pressure changes, the pressure field itself is not solved for directly.

3.4.1 The SIMPLE Algorithm

The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm was originally developed by Patankar and Spalding [72]. Introduced in the RANS framework, the SIMPLE algorithm (and its subsequent revisions by Patankar [73]) now forms the basis of the majority of commercial

RANS CFD packages. An excellent description of the method is provided in the textbook of Ferziger and Perić [71]; only a brief description is included here.

The algorithm starts from an initial estimate for the velocity components and pressure in each cell, which will typically consist of some known initial and boundary conditions at the start of a simulation; subsequently, the algorithm will take the solution from the previous iterative step. This estimate is used to solve the momentum equation, from which a set of velocities (not satisfying continuity) are found. The corrections required to these velocities (in order to satisfy continuity) may then be found from some correction to the pressure field, where the source term for the pressure correction is the mass error in each cell (i.e. the sum of the mass fluxes, as calculated from the predicted velocities). The pressure correction term is used to calculate velocity corrections, from which the pressure and velocity fields are updated. These new values are then used to solve the momentum equation again, from which a new mass error will result. The process of: i) solving the momentum equations; ii) calculating a pressure correction from the resulting mass error; iii) updating the pressure and velocity fields, is repeated until the corrections to the velocity and pressure terms have dropped below some target value. To advance in time the procedure described above is repeated with the boundary conditions to the domain suitably modified at each time-step, where the solution at one time-step forms the initial conditions for the next. For both time-averaged and time-dependent simulations the updates to the pressure and velocity fields within the iterative process may be under-relaxed, to promote the convergence of the algorithm.

3.4.2 The Fully Conservative Approach

As a semi-implicit scheme the SIMPLE algorithm is not applicable to the majority of time-dependent CFD simulations, which typically use an explicit advancement in time (§3.2). In the `PsiPhi` code the Fully Conservative Approach of Kempf is implemented. This method is based on a procedure of prediction and correction, and a detailed description is provided in [74].

The mass in each computational cell is transported using the current velocity field, in a 'predictor' step, and is then compared to the mass expected from the chemical state. For non-reactive, incompressible flows of a single fluid, the mass in each cell according to the chemical state will be constant; for reactive flows, the mass will depend on the local temperature and composition. The difference between the transported mass in each computational cell and the mass from the chemical state is then the error that would be induced by transporting the mass field with the current velocity field. This mass error forms the source term for a Poisson equation, which is solved in the `PsiPhi` code using an iterative, under-relaxed Gauss-Seidel algorithm. The resulting pressure correction is used to correct the pressure, velocity and momentum fields, so that when the transport routines are called a second time – in the corrected step – the mass transport induced by the velocity field continues to satisfy the continuity equation.

3.5 Sub-Grid Scale Turbulence Models

Filtering of the Navier-Stokes equations generates a closed and an unresolved stress component, the latter being, in the context of LES, the sub-grid scale (SGS) residual stress tensor τ_{ij}^{sgs} appearing

in Eq. 2.33. The residual stress term requires closure, and in many cases must be modelled to recover accurate results. A number of approaches have been suggested and tested in the literature; however, the majority are built upon the Boussinesq hypothesis [75], which states that the residual stress tensor is proportional to the traceless rate-of-strain tensor S_{ij}. In effect this implies that the impact of turbulent eddies within a flow may be modelled with an effective 'eddy' or 'turbulent' viscosity (where in a RANS simulation all of the eddies are accounted for by a modelled term). The Boussinesq hypothesis may be written in the context of LES as:

$$\tau_{ij}^{\text{sgs}} - \frac{1}{3}\tau_{kk}^{\text{sgs}}\delta_{ij} = 2\nu_t \bar{S}_{ij} \tag{3.11}$$

Where ν_t is the turbulent kinematic viscosity. An expression of the form in Eq. 3.11 is applicable for both the constant density (Eq. 2.33) and Favre-filtered (Eq. 2.40) forms of the momentum conservation equation. A number of approaches for the calculation of this turbulent viscosity are presented below. The Smagorinsky and Germano models, described here in comparative detail, are implemented in the `PsiPhi` code.

3.5.1 Smagorinsky Model

Smagorinsky [76] formulated an approximation for ν_t based on the Boussinesq hypothesis, which was extended by Lilly [77] and was subsequently applied to LES by Deardorff [40]. Smagorinsky's model may be written as:

$$\nu_t = (C_s\Delta)^2 |S| \tag{3.12}$$

Where $|S|$ is the magnitude of the resolved rate-of-strain tensor $\bar{S}_{ij}$:

$$|S| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{3.13}$$

The quantity C_s in Eq. 3.12 is the Smagorinsky constant, which represents a scaling factor for the modelled viscosity and typically takes a value $0.05 < C_s < 0.20$. The selection of a value for C_s may have a significant effect on the nature of the flow-field, where large values cause turbulent structures to be smoothed out. As the resolution of the flow field approaches that required of a DNS (where the Kolmogorov scale is resolved and no sub-grid fluctuations exist) it follows that the modelled contribution should tend towards zero; however, this should be accounted for by the dependence of ν_t on Δ, rather than its dependence on C_s, so that in theory the chosen value of C_s should not be grid-dependent.

In terms of computational cost and memory requirements, the Smagorinsky model is the cheapest of the SGS turbulence models described here.

3.5.2 Dynamic Smagorinsky (Germano) Model

Selecting a suitable value of C_s for the Smagorinsky model constitutes a challenging aspect of LES. A common question posed to the computational scientist is then why a particular value of C_s has been chosen for a given simulation, although there are guiding approximations: Deardorff [78] devotes

a paper to the topic. A solution to this problem is to formulate a relationship to approximate the model constant dynamically. A dynamic counterpart to the Smagorinsky model has been proposed by Germano et al. [79], and has been extended by, amongst others, Piomelli and Liu [80], whose implementation is described here.

The Germano model is built on the concept of applying two different filters to the velocity field of an LES. For convenience, the smaller 'grid' filter is chosen to be equal to the LES filter width (i.e. cell size, where implicit filtering is assumed), while a larger 'test' filter is applied to the grid-filtered velocities. This treatment yields a different residual stress tensor at each filter width, and these can be related to the resolved turbulent stresses using the identity proposed by Germano [81]:

$$\mathcal{L}_{ij} = T_{ij} - \widehat{\tau}_{ij} \tag{3.14}$$

Where $\widehat{\tau}_{ij}$ has been filtered at the test filter width. The sub-test-filtered residual stresses are determined from:

$$T_{ij} = \widehat{\overline{u_i}\overline{u}_j} - \widehat{\bar{u}}_i\widehat{\bar{u}}_j \tag{3.15}$$

While the test-filtered SGS residual stresses can be written as:

$$\widehat{\tau}_{ij} = \widehat{\overline{u}_i\overline{u}_j} - \widehat{\overline{u_i}\overline{u}_j} \tag{3.16}$$

The eddy-viscosity concept (Eq. 3.11) can then be applied for each of T_{ij} and τ_{ij}:

$$T_{ij} - \frac{1}{3}T_{kk}\delta_{ij} = -2\,C\widehat{\Delta}^2|\widehat{\bar{S}}|\widehat{\bar{S}}_{ij} = -2\,C\alpha_{ij} \tag{3.17}$$

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\,C\Delta^2|\bar{S}|\bar{S}_{ij} = -2\,C\beta_{ij} \tag{3.18}$$

Where $\widehat{\Delta}$ is the width of the test filter, which is commonly set to $\widehat{\Delta} = 2\Delta$. Note that in this case C is not squared; the values of C appearing in these and following equations are equivalent to C_s^2 in the Smagorinsky model (Eq. 3.12). Equations 3.17 and 3.18 are substituted into Eq. 3.14, yielding:

$$\mathcal{L}^*_{ij} = \mathcal{L}_{ij} - \frac{1}{3}\mathcal{L}_{kk}\delta_{ij} = 2\left(\widehat{C\beta_{ij}} - C\alpha_{ij}\right) \tag{3.19}$$

At this point there is a difficulty in solving for C, because it appears inside a test-filtering operation. However, this problem may be mitigated by assuming a known estimate C^*, which leads to:

$$-2\,C\alpha_{ij} = \mathcal{L}^*_{ij} - 2\,\widehat{C^*\beta_{ij}} \tag{3.20}$$

Equation 3.20 may now be re-arranged to calculate C directly. In addition, the sum of the squares of the residual may also be minimised, according to Piomelli and Liu, by contraction with α_{ij}:

$$C = -\frac{1}{2}\frac{\left(\mathcal{L}^*_{ij} - 2\,\widehat{C^*\beta_{ij}}\right)\alpha_{ij}}{\alpha_{mn}\alpha_{mn}} \tag{3.21}$$

It is common practice to convolute C with the test filter, as its values should vary smoothly in space over the scale of that filter. This also helps to reduce spurious high values of C that may potentially result as the denominator of the LHS of Eq. 3.21 tends to zero (for example, in a laminar flow). The final requirement is to provide a reasonable estimate for C^*, where Piomelli and Liu suggest using the value from the previous time-step ($C^* = C_{n-1}$, where n is the number of the time-step), or from a backward extrapolation of the form:

$$C^* = C_{n-1} + dt \left.\frac{\partial C}{\partial t}\right|_{n-1} + \dots \tag{3.22}$$

In the present work the latter approach is adopted. A linear (first-order) approximation is used, which is effectively an Euler-explicit scheme between the sub-time-steps of the third-order Runge-Kutta scheme described in §3.2. The rate of change of C is approximated as:

$$\left.\frac{\partial C}{\partial t}\right|_{n-1} = \frac{C_{n-1} - C_{n-2}}{h_{j-1}\, dt_{n-1}} \tag{3.23}$$

Where the time-step weights h_j are given in Eq. 3.8. The approximation of C^* may then be written as:

$$C^* = C_{n-1} + \frac{h_j\, dt_n}{h_{j-1}\, dt_{n-1}} \left(C_{n-1} - C_{n-2}\right) \tag{3.24}$$

The implementation of the Germano model proves to be expensive in terms of both calculation and memory requirements. A large number of intermediate quantities must be stored, many of which are symmetric second-order tensors (comprising six unique components per computational cell). In addition no fewer than four (explicit) filtering operations are required to determine the velocities $\widehat{\overline{u}}_i$, the product of velocities $\widehat{\overline{u}_i \overline{u}_j}$, the term $\widehat{C^* \beta_{ij}}$, and for filtering of the final calculated value of C. Values of C are determined plane-by-plane, such that, in a computational domain measuring $N_i \times N_j \times N_k$ cells, the memory requirement is reduced by a factor of approximately N_i. Although this increases the cost of the calculation to some degree, as some quantities that are calculated for slice i are re-calculated for slice $(i+1)$, the significance of the memory saving for use in practical LES far outweighs the associated reduction in efficiency.

This formulation of the Germano model permits negative values of C, representing the physical phenomenon of 'backscatter': although the average transferral of energy within the turbulence cascade (Fig. 2.1) is from the large scales to the small ones, Piomelli et al. [82] have demonstrated that a small amount of energy transfer in the opposite direction may also be expected. Negative values of C may pose a risk to the stability of a simulation, however, and it is therefore common to enforce positive values of C in the calculation of ν_t.

3.5.3 Other SGS Turbulence Models

A number of other modelling approaches exist in addition to the Smagorinsky and Germano procedures, and some examples are provided here.

A disadvantage of the Smagorinsky model is that it performs poorly near to walls. In reality all

turbulence is damped in the vicinity of a wall, so that the turbulent viscosity should approach zero; however, the Smagorinsky model continues to provide a non-zero value in near-wall regions, as a velocity gradient exists. Nicoud and Ducros [83] argued that any eddy-viscosity model built on the rate-of-strain tensor $\bar{S}_{ij}$ will not behave correctly near to walls, and they therefore proposed a Wall-Adapting Local Eddy-viscosity (WALE) model that is built around an alternative quantity. The WALE model has the significant advantage that it performs well in near-wall regions for LES [84, 85, 86]. However, as for the Smagorinsky model, it contains a (static) model constant C_w. The chosen value of C_w can be difficult to justify considering that it may depend on the geometry of the flow or the resolution of the computational grid. The computational cost and memory expense of the WALE model is estimated to be around a factor of four greater than that of the Smagorinsky approach, based on the number of floating point operations required to evaluate the turbulent viscosity at each point within the domain, as the operator for the model must be built in intermediate steps from the velocity components.

A dynamic formulation of the WALE model has been suggested and implemented by Baya Toda et al. [87]. A dynamic value for C_w is determined in a similar fashion to that of the Germano model, where the quantity $|\bar{S}|$ in Eqs. 3.17 and 3.18 is replaced by the WALE quantity. However, Baya Toda et al. found that such a formulation tended to over-predict the turbulent viscosity near walls, and subsequently implemented a 'Shear and Vortex Sensor' to limit the value of C_w accordingly. The computational cost of the dynamic WALE model is greater than that of the Germano implementation, as the cost of computing the WALE quantity is greater than that of computing $|\bar{S}|$. The excessive memory requirement of the model may be mitigated by calculating C_w plane-by-plane as before, with the associated reduction in computational efficiency.

Alternative implementations of the Germano model have been proposed by, amongst others: Lilly, who uses a least squares approach to formulate an alternative contraction to minimise the error in C [88]; and Zhou et al., who apply the scale similarity hypothesis of Bardina et al. [89] to evaluate the SGS residual stresses τ_{ij} [90]. The values of C determined from the model presented in §3.5.2, as well as from these alternate formulations, may be either locally averaged, averaged across planes (where it makes sense to do so), or globally averaged over the whole domain.

A promising new model has been developed by Nicoud et al. [91], where the operator $|\bar{S}|$ in the Smagorinsky model is replaced with an operator constructed from the singular values (i.e. the square roots of the eigenvalues) of the velocity gradient tensor for the resolved scales. The so-called σ-model has been shown to deliver results equivalent to or marginally better than those obtained from the Germano model, while being considerably less expensive; it also performs well near walls.

3.6 Sub-Grid Scale Scalar Flux Models

Filtering of the general transport equation (Eq. 2.9) generates a closed term and an unresolved residual (SGS) scalar flux, similar to the terms of the filtered Navier-Stokes equations. In LES, the latter appears in Eq. 2.37 as τ_i^{ϕ} (or, in a Favre-filtered form in Eq. 2.41, as $\widehat{\tau}_i^{\phi}$). A number of approaches for the modelling of the residual flux term have been suggested, and some examples are provided here.

3.6.1 Gradient Diffusion Model

The gradient diffusion model proposed by Eidson [92] applies a method similar to the eddy-viscosity formulation of Smagorinsky (§3.5.1). In this model, the SGS scalar flux is approximated as:

$$\tau_i^{\phi} = -D_{\text{sgs}} \frac{\partial \bar{\phi}}{\partial x_i} \tag{3.25}$$

Where the diffusivity accounting for turbulence at the sub-grid scale is defined as:

$$D_{\text{sgs}} = \frac{\nu_t}{Sc_t} = \frac{1}{Sc_t}(C_s\Delta)^2 \left|S\right| \tag{3.26}$$

Where Sc_t is the turbulent Schmidt number (often taken to be 0.7) and C_s is the Smagorinsky constant from the modelling of the SGS residual stresses. Where C is determined dynamically, the co-efficient $(C_s\Delta)^2$ in Eq. 3.26 should be replaced with $C\Delta^2$. A clear advantage of this model is, therefore, its simplicity in implementation, as the turbulent diffusivity D_{sgs} is proportional to the turbulent viscosity ν_t that has already been calculated to model the SGS residual stresses. This simplicity motivates the use of the gradient diffusion model in the `PsiPhi` code. However, a disadvantage is that (in the static case, at least) the model over-estimates the SGS scalar fluxes near to walls, as demonstrated by Mason [93] amongst others.

3.6.2 Similarity Model

The similarity model suggested by Bardina et al. [89] employs a different approach: unresolved scalar fluxes at the LES grid filter width are assumed to be proportional to unresolved scalar fluxes determined for some other test filter width. The difference between the sub-test-filter scalar fluxes and the test-filtered resolved fluxes is then used to approximate the SGS scalar fluxes as:

$$\tau_i^{\phi} = C_{\text{sim}} \left(\widehat{\bar{u}_i \bar{\phi}} - \widehat{\bar{u}}_i \widehat{\bar{\phi}} \right) \tag{3.27}$$

Where $\widehat{\bar{\phi}}$ is an LES-filtered quantity that has been filtered again at the test filter width, and C_{sim} is a model constant. While this model demonstrates advantages over the gradient diffusion model in near-wall regions [94, 95], it has also shown a tendency to cause instabilities in simulations [89].

3.6.3 Mixed Model

Bardina suggested adding the (dissipative) gradient diffusion model as a 'dampener' to the similarity model, to improve its stability. The resulting mixed model is then written as:

$$\tau_i^{\phi} = C_{\text{sim}} \left(\widehat{\bar{u}_i \bar{\phi}} - \widehat{\bar{u}}_i \widehat{\bar{\phi}} \right) - \frac{1}{Sc_t}(C_s\Delta)^2 \left|S\right| \frac{\partial \bar{\phi}}{\partial x_i} \tag{3.28}$$

Liu et al. [96] have demonstrated that the addition of the gradient diffusion model does not overly detract from the benefits gained through the use of the similarity model, as the former is

typically small in comparison to the latter. The model constants C_{sim} and C_s may be determined dynamically, as demonstrated by Porté-Agel et al. [97].

3.7 Boundary Conditions

A important element of any computational flow solver is the accuracy and suitability of the applied boundary conditions. The use of incorrect boundary conditions may lead to numerical instability, but in the worst case, the flow-field will develop incorrectly and any subsequent predictions will be inaccurate. Within a simulated domain each boundary is usually either an inlet or an outlet, and will need a condition to be prescribed for every scalar quantity that crosses it. In a typical LES boundary conditions must therefore be applied for density, components of momentum, and any other transported scalar; they may also be required for pressure, and for other derived quantities such as temperature. Arguably the most important of these are the velocity conditions from which the momentum components are determined.

3.7.1 Velocity Inflow Conditions

Ideally, for a fully-developed turbulent flow, the simulated domain would extend far upstream of the region of interest, allowing the flow to develop over a sufficient length to be insensitive to the inflow conditions. However, this approach is unlikely to produce the desired turbulence characteristics in the region of interest (such as the magnitude of velocity fluctuations, or the integral length-scale). In addition a very long simulation domain is unlikely to have the same acoustic characteristics as the actual geometry, and finally, the treatment of immersed boundaries (walls) in an LES is often overly dissipative. Even if the upstream region were to be configured to deliver the correct turbulence characteristics, by the time the flow reaches the region of interest it may be dominated by undesirable near-wall modelling effects.

The generation of artificial inflow data is therefore preferable, as it removes the need for such an extension to the simulated domain. In the simplest case a time-independent (mean) velocity profile may be applied to satisfy the target mass flow rate only. However, such a profile will take a long time to undergo a transition to turbulence within the domain (if such a transition occurs at all), requiring an increased simulation duration. The next logical step might be to add some random fluctuations, or noise, to the mean profile, but this results in all of the turbulent kinetic energy being contained in the smallest scales. The resulting energy spectrum bears little resemblance to Kolmogorov's energy cascade (Fig. 2.1), and all of the turbulent energy is rapidly dissipated.

A solution to this problem has been proposed by Klein et al. [98] and has been successfully applied in numerous cases, in the context of both DNS and LES [99, 100, 101, 102, 103, 104]. Subsequent modifications to reduce the computational cost of the method have been proposed by Kempf et al. [105, 106]. The approach used in the `PsiPhi` code is summarised below.

Klein's original method applies a discrete low-pass filter $B(i', j', k')$ to each point (i, j, k) within a field of random noise r^{α}, where α represents the components of the flow velocity in three Cartesian directions. The filtering increases the correlation within that noise until a desired integral length-scale L is achieved. Following the expression of Eq. 2.29, a discretised form of the convolution of

r^{α} with the filter B may be written as:

$$R^{\alpha}(i,\, j,\, k) = \sum_{i'=-n_f}^{n_f} \sum_{j'=-n_f}^{n_f} \sum_{k'=-n_f}^{n_f} B(i',\, j',\, k')\, r^{\alpha}(i+i',\, j+j',\, k+k') \tag{3.29}$$

Where n_f is the number of filter support points (i.e. the extent of the discrete filter kernel) in each direction. The filtered fields R^{α} are then shifted and scaled so that they have zero mean and a unit standard deviation, before being scaled by the prescribed turbulent kinetic energy. The scaled, filtered fields (with some mean velocity field super-imposed) may then be used as initial velocity conditions throughout the computational domain, or may be 'fed in' to the domain at the inflow plane.

The coefficients in the filter B required to recover the correct length-scale L may be determined from the modelled auto-correlation function R_{uu}, proposed by Batchelor [107], for fully-developed, homogenous turbulence:

$$R_{uu}(r) = \exp\left(-\frac{\pi r^2}{4L^2}\right) \tag{3.30}$$

In Kempf's proposed modification the radius r is expressed in the local co-ordinates $(i',\, j',\, k')$:

$$r^2 = \Delta^2 \left(i'^2 + j'^2 + k'^2\right) \tag{3.31}$$

While the target integral length-scale is normalised by the DNS or LES cell size:

$$n = L/\Delta \tag{3.32}$$

Substituting Eqs. 3.31 and 3.32 into Eq. 3.30:

$$R_{uu}(i',\, j',\, k') = \exp\left(-\frac{\pi(i'^2 + j'^2 + k'^2)}{4n^2}\right) \tag{3.33}$$

Resulting from the use of an exponential filter kernel, the RHS of Eq. 3.33 may be expressed as the product of three one-dimensional filters:

$$B(i',\, j',\, k') = b(i')\, b(j')\, b(k') \tag{3.34}$$

Comparing Eqs. 3.33 and 3.34, these one-dimensional coefficients can therefore be written generally as:

$$b(s') = \exp\left(-\frac{\pi s'^2}{2n^2}\right) \tag{3.35}$$

The fact that the filter B may be expressed in the manner of Eq. 3.34 allows for a significant reduction in computational effort: for a given point $(i,\, j,\, k)$ in the field r^{α}, the one-dimensional filter b can be applied separately three times, as opposed to applying the three-dimensional filter B

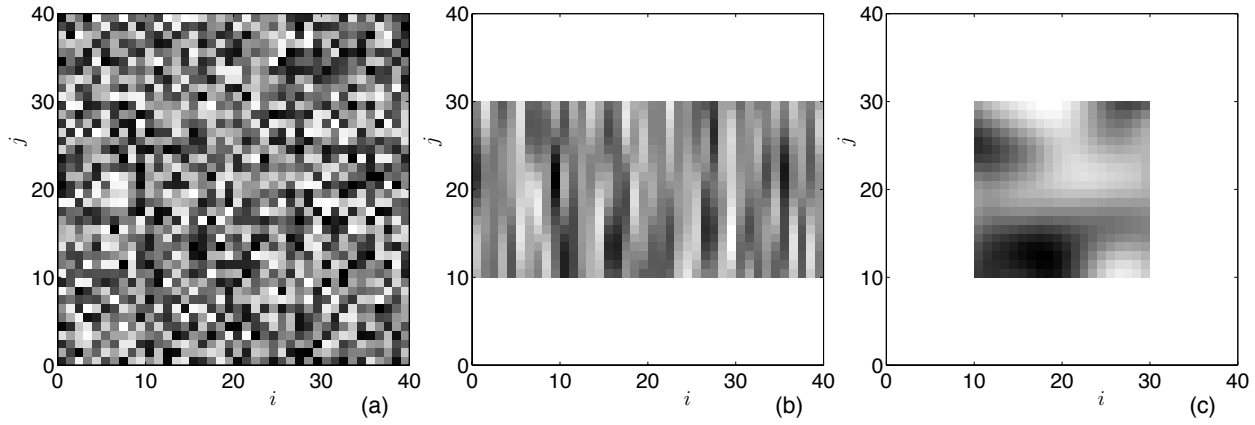


Figure 3.2: Filtering of random noise using a combination of two 1D filters: (a) original field, (b) filtered in j-direction, and (c) subsequently filtered in i-direction.

once. This operation may be shown by the substitution of Eq. 3.34 into Eq. 3.29:

$$R^{\alpha}(i,\,j,\,k) = \sum_{i'=-n_f}^{n_f} \sum_{j'=-n_f}^{n_f} \sum_{k'=-n_f}^{n_f} b(i')\,b(j')\,b(k')\,r^{\alpha}(i+i',\,j+j',\,k+k')$$

$$= \underbrace{\sum_{i'=-n_f}^{n_f} b(i') \underbrace{\sum_{j'=-n_f}^{n_f} b(j') \underbrace{\sum_{k'=-n_f}^{n_f} b(k')\,r^{\alpha}(i+i',\,j+j',\,k+k')}_{\text{filter in } k\text{-direction} \rightarrow r^{\alpha}(i,j,k)\,|_{k}}}_{\text{filter in } j\text{-direction} \rightarrow r^{\alpha}(i,j,k)\,|_{k,\,j}}}_{\text{filter in } i\text{-direction} \rightarrow r^{\alpha}(i,j,k)\,|_{k,\,j,\,i}} \tag{3.36}$$

A visual representation of the filtering process is shown in Fig. 3.2. The white areas indicate the boundary regions that cannot be filtered, necessitating the use of a domain larger than the inflow plane by n_f cells in each direction (where $n_f = 10$ in the example shown). The computational advantage that is gained by performing the filtering operation in this manner comes from the repeated re-use of previously-filtered values. Evaluation of Eq. 3.36 requires the multiplication and summation of $3(2\,n_f+1)$ values for each point in the domain, while Eq. 3.29 would require $(2\,n_f+1)^3$ multiplications and additions. For a typical LES, with a filter width $n_f \approx 15$ (based on $L = 5$ mm, $\Delta = 1$ mm, from which $n = 5$; then $n_f \approx 3\,n$ to ensure that the 'tails' of the exponential filter kernel are accurately described), this results in a computational speed-up of three orders of magnitude.

Although the underlying mean velocity field prescribed at the inflow should satisfy continuity, the turbulent flow field generated by Klein's (modified) method will not. This will therefore result in some additional work for the pressure correction scheme as 'slices' of the turbulent field are added at the inflow plane; however, this may be mitigated by linearly interpolating between these slices at the inflow. Such a method has proved to be satisfactory in the experience of the author and co-workers [99, 100, 108, 109].

3.7.2 Other Boundary Conditions

Boundary conditions must be applied at inflow planes for any other quantities for which a transport equation (Eq. 2.9) is solved. These are, however, typically simpler to prescribe than inflow velocities. Boundary conditions for quantities such as density and mixture fraction are often prescribed with a Dirichlet (constant value) condition, but as the boundary values for these quantities will not be modified over the course of a computational time-step, the condition is implied rather than explicit. Conditions for other transported quantities, such as enthalpy, may depend on the nature of the boundary: for example, for temperature, a Dirichlet condition should be used for inflows and isothermal walls, while a von Neumann (zero gradient) condition should be applied for outflows or adiabatic walls. Boundary conditions for the pressure field (determined from the predictor step described in §3.4) may be set using a von Neumann condition; however, such a condition here may cause pressure waves within the domain to be reflected, which can lead to unphysical flow behaviour. As an alternative a constant gradient condition, where the pressure at the boundary is linearly extrapolated from points local to (and normal to) the boundary, can be used.

Outflow conditions are also required for all scalar quantities, for which the von Neumann condition is usually sufficient. In the present work a positive outflow condition is additionally implemented, where momentum and velocity components are constrained to be positive in the direction normal to an outflow plane. This prevents fluid from being entrained (drawn) into the simulated domain, as, conceptually, no Dirichlet boundary conditions are available for such fluid. With the von Neumann condition already applied any entrained fluid would inherit its properties from the fluid already inside the domain, which could lead to, for example, the undesirable (and unrealistic) re-introduction of chemical species that had previously been transported out of the domain.

3.8 Summary

This chapter has focused on how the governing equations introduced in Chapter 2 may be solved numerically in the context of Large-Eddy Simulation (LES) and Direct Numerical Simulation (DNS). The Finite Volume Method (FVM) for the spatial discretisation of a domain has been discussed, along with the Euler-explicit and Runge-Kutta methods for the advancement of a solution in time. Limitations on the temporal resolution required for a convergent solution, due to the convective and diffusive transport of information within the domain, have also been considered.

With regards to the Navier-Stokes equations themselves the SIMPLE and Fully Conservative pressure correction methods have been examined, where these are used to ensure the satisfaction of continuity in the solution of the momentum conservation equation. In addition, a number of modelling techniques for the treatment of the (unclosed) sub-grid scale momentum and scalar flux terms, resulting from the spatial filtering applied in an LES, have been described. Finally, the prescription of boundary conditions – and in particular, the use of digital filtering for the generation of realistic turbulence at the inflow plane of the simulated domain – has been discussed.

Numerical approximations for the convective and diffusive transport terms of the Navier-Stokes and general transport equations have not been considered here, but will be treated in depth in Chapter 4.

4 Discretisation of Fluxes

Previous chapters have introduced the governing equations of Fluid Mechanics (§2.1), have described how an approximate numerical solution to those equations may be obtained (§2.2 and §2.3), and have explained how those equations are implemented in a computational code (Chapter 3). In this chapter a particular focus is given to the spatial discretisation of the convective and diffusive flux terms of the general transport equation (Eq. 2.9), as modified for application within the context of Large-Eddy Simulation (LES, Eqs. 2.37 and 2.41) or directly in a Direct Numerical Simulation (DNS).

In a traditional DNS or LES fluxes between cells are estimated in the framework of the Finite Volume Method described in §3.1. However, the estimation methods, or 'differencing schemes', for convective flux terms all suffer from various disadvantages, motivating the development of an alternative approach for their numerical approximation.

4.1 Convection in the Finite Volume Method

To understand how convective fluxes are spatially discretised, a one-dimensional domain containing a continuous conserved scalar quantity ϕ in a velocity field u is considered. The discrete values $\phi(i)$ are stored in uniformly-spaced computational cells, of width Δ, centred at $x(i)$ (where i refers to the index of the cell), as shown in Fig. 4.1. For convenience, each velocity $u(i)$ is stored at the right-hand ('east') boundary of the corresponding cell i.

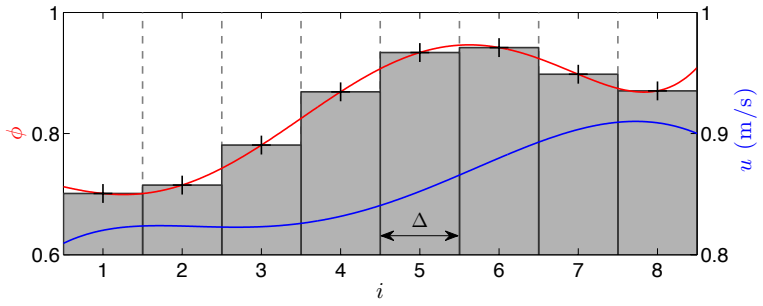


Figure 4.1: A 1D domain containing a function ϕ discretised at points i, and a velocity u.

The value of $\phi(i)$ is assumed to represent the mean (piece-wise constant) value of ϕ throughout the cell, as shown in Fig. 4.1. The diffusivity of the scalar is zero, and there are no source or sink terms. From the general transport equation (Eq. 2.9), the evolution in time of ϕ may then be

written as:

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u\phi)}{\partial x} = 0 \tag{4.1}$$

Assuming that the domain is in a steady state:

$$\frac{d(u\phi)}{dx} = 0 \tag{4.2}$$

Integration of Eq. 4.2 across a single cell (control volume), combined with the application of the divergence theorem, yields:

$$\int_V \nabla \cdot (u\phi)\, dV = \int_A (u \cdot n)\, \phi\, dA = 0 \tag{4.3}$$

Where n is the unit direction normal to the face of the cell, A is the area of each cell face, and V is the volume of the cell. Taking n to be positive in the positive x direction results in:

$$[(u \cdot 1)\, \phi A]_e + [(u \cdot (-1))\, \phi A]_w = 0 \tag{4.4}$$

Where the subscripts w and e denote the west and east faces of the cell, respectively. From Eq. 4.4 it is possible to define a general expression for the convective flux F^c across a cell i:

$$F^c(i) \approx \sum_F [(u \cdot n)\, \phi A]_F \qquad \text{with} \qquad F = \{e, w, \ldots\} \tag{4.5}$$

Where $F^c(i)$ is equal to zero for a steady state problem, or is equal to the rate of change of the scalar ϕ within cell i for a transient problem. Although Eq. 4.5 has been determined for a one-dimensional domain, where each cell has only two (point-like) faces, it may be shown that it is equally applicable to two- or three-dimensional cases.

In Eq. 4.5 the variables relating to the computational grid (the unit normal n and the cell face area A) are known, as is the velocity u at the cell face. All that is required, therefore, is an estimate of the scalar value ϕ_F, from the surrounding (known) values of $\phi(i)$.

4.2 Desirable Properties of Differencing Schemes

The manner in which ϕ_F is approximated forms one of the key aspects of both DNS and LES, as it strongly influences the solution of a time-dependent simulation. In terms of the properties that a differencing scheme should possess, accuracy is perhaps the most obvious; however, an increase in accuracy usually comes at the cost of requiring a larger number of neighbouring values for the approximation of ϕ_F. The extent of a scheme in this sense is referred to as its computational 'stencil'. Increasing the size of the stencil has implications both for the computational expense of a scheme, and for the application of that scheme around the domain boundaries (where there may not be enough 'space' for the stencil to be applied). The other important properties for a convective scheme are that it is bounded, conservative, and transportive.

Conservativeness A scheme must conserve the quantity that it is transporting, which is to say that it should be consistent in the way that cell-faced values are calculated: the estimated value ϕ_e for cell i must be equal to that of ϕ_w for cell $(i+1)$. The two cells share the same face, so it makes sense that the scalar value at that face should be approximated in the same way for each cell. Any discrepancy between the approximations would result in a net increase or decrease in the global sum of ϕ, as (assuming a positive velocity at that face) the efflux from cell i would not be equal to the influx into cell $(i+1)$. In practice conservativeness is ensured for all of the schemes that will described in this chapter, by assuming that the scalar approximation at the west face of a cell is equal to the approximation at the east face of the adjacent cell. In this way it is only necessary to calculate the east-face values for each cell, which also reduces computational expense.

Boundedness A convective transport scheme is 'bounded' if the upper and lower limits of a transported function are guaranteed to lie within the limits of the corresponding initial function (provided that the transported quantity is strictly conserved). This criterion has particular significance in, for example, the transport of a chemical species mass fraction within a flow: there cannot be less than 0% or more than 100% of a mass fraction, by definition. Unlike conservativeness, this is not a guaranteed property.

Transportiveness Finally, a convective transport scheme should take into account the direction of the flow. Information travels downstream: referring to Fig. 4.1, the action of the velocity u at the position of a cell face F will tend to cause the actual value of ϕ_F to become closer to the upstream value of ϕ. It therefore makes sense to base the approximation of ϕ_F on positions upstream of F, as the information at these positions will move closer to F over the course of the computational time-step. (Conversely it would not make sense to choose downstream values, as this information has already passed through, and will be moving away from, the position of F).

4.3 Low Order Schemes

Taking the desirable properties of §4.2 into consideration, the performance of some low order-of-accuracy differencing schemes will now be discussed. These are the first-order accurate Upwind Differencing Scheme (UDS), the second-order Central Differencing Scheme (CDS) and the second-order Linear Upwind Differencing Scheme (LUDS).

4.3.1 First-Order UDS

The most basic approximation that may be made is to assume that the value of ϕ at the cell face F is equal to the value of $\phi(i)$ in the adjacent cell. The transportiveness property is satisfied by choosing the point upwind of F when deciding whether to use the 'left' or 'right' adjacent value of $\phi(i)$. The approximation for ϕ_e (i.e. at the east face of cell i) may thus be written as:

$$\phi_e \approx \begin{cases} \phi(i) & \text{for} \quad u > 0 \\ \phi(i+1) & \text{for} \quad u < 0 \end{cases} \tag{4.6}$$

A corresponding expression for ϕ_w would be identical apart from a shifted index of i (by one cell to the left); however, conservativeness is satisfied by assuming that ϕ_w is equal to ϕ_e in the adjacent cell, so no explicit calculation of ϕ_w is necessary. A graphical visualisation of the approximation for ϕ_e in a positive velocity field is provided in Fig. 4.2, where the red dashed line indicates the extrapolation from the point i, and the red cross is the approximated value.

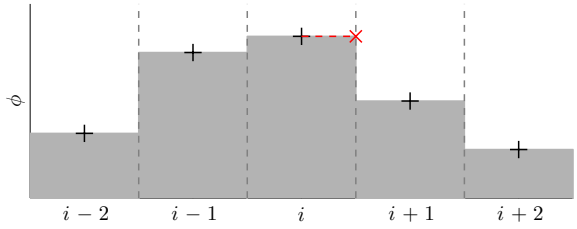


Figure 4.2: The first-order Upwind approximation.

It is possible to demonstrate that the error resulting from the use of UDS is proportional to the cell size Δ [110], as a Taylor expansion of this approximation is truncated after the first term. The Upwind Scheme is therefore said to be first-order accurate with respect to the cell size.

In order to gain some insight into how different schemes for the approximation of ϕ_F behave, two simple one-dimensional cases are considered. In the first case, a domain of unit length is divided into 100 computational cells, and a function ϕ is initialised with $\phi = 1$ in the range $x = [0.0 - 0.1]$ m, and $\phi = 0$ everywhere else. A constant unit velocity is applied from left to right throughout the domain, and the solution is advanced in time until 0.4 s have passed (i.e. when the interface between $\phi = 0$ and $\phi = 1$ should have reached 0.5 m). The second case is identical to the first, except that the initial 'sharp' step in ϕ is smoothed by applying a low-pass filter. In both cases advancement in time is performed using a (temporally) first-order accurate Euler-explicit method (Eq. 3.2).

For each case, the solution for ϕ after it has been transported numerically is compared to an analytical solution (created by moving the co-ordinate system of the initial function by 0.4 m to the left). The resulting fields, where first-order UDS has been applied, are shown in Fig. 4.3.

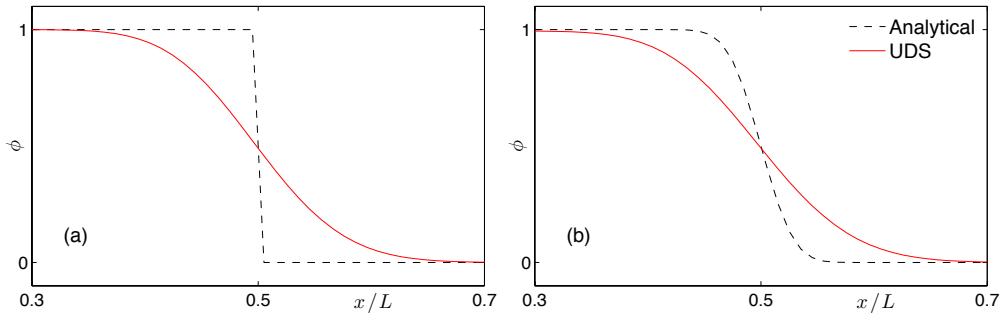


Figure 4.3: Two 1D functions transported by UDS: (a) sharp, (b) smooth.

The detrimental effect of applying UDS for convective transport is striking. In both cases the maximum scalar gradient at the interface has been dramatically reduced, while the 'corners' of the

transported function have become (in the second case, more) rounded. Both transported solutions are almost identical, even though the initial conditions are discernibly different. The implications for any meaningful simulation are obvious: using UDS will potentially result in a very wrong answer.

This consequence of using UDS is known as numerical (or 'false') diffusion, and it is an artefact that affects every numerical scheme to a lesser or greater extent. For an Upwind Scheme it is possible to determine a numerical diffusivity, similar to the physical diffusivity associated with the fluid viscosity, which is related to the width of the time-step. By considering the sum of the absolute differences between a field ϕ_c, transported by UDS over t time units with constant streamwise velocity u and a certain dt, and a corresponding field ϕ_d initialised at the final position of ϕ_c but instead diffused with a constant diffusivity D, it is possible to demonstrate that the difference between the fields is minimised by the function:

$$D = \frac{1}{2}\left(1 - u\,dt\right) \tag{4.7}$$

Where, in this case, the computational cells are assumed to have unit size, so that $u\,dt \equiv \mathrm{CFL}$. An interesting point to note is that the numerical diffusivity reaches zero for the limit of stability where CFL = 1. This may be more clearly understood by considering Eq. 3.2: given that the velocity field is positive, and by substituting Eqs. 4.5 and 4.6 into Eq. 3.2, it may be written that:

$$\phi(i,\, t + dt) = \phi(i,\, t) - dt\left(u\,\phi(i,\, t) - u\,\phi(i-1,\, t)\right) \tag{4.8}$$

For the case of $u\,dt = 1$, Eq. 4.8 reduces to:

$$\phi(i,\, t + dt) = \phi(i-1,\, t) \tag{4.9}$$

Equation 4.9 shows that the values of ϕ are exactly transported from left to right under the restrictions of unit cell size and a unit CFL number.

4.3.2 Second-Order CDS

The Central Scheme in its simplest form may be expressed as the linear interpolation of ϕ_F between its neighbours, which, for the east face of cell i, is written as:

$$\phi_e \approx \frac{1}{2}\left(\phi(i) + \phi(i+1)\right) \tag{4.10}$$

The approximation in Eq. 4.10 is shown graphically in Fig. 4.4. A Taylor expansion of this approximation shows that the error in ϕ_e is proportional to Δ^2 on a uniform grid, and it is therefore said to be second-order accurate in space. The two cases introduced in §4.3.1 are also considered with convection by second-order CDS (CDS2). The resulting fields are shown in Fig. 4.5, where the numerical diffusion attributed to UDS has been replaced by sizeable numerical oscillations.

Sengupta et al. [111] provide a detailed explanation for the emergence of these oscillations, and also point out that their magnitude is not reduced by increasing the grid resolution (even though

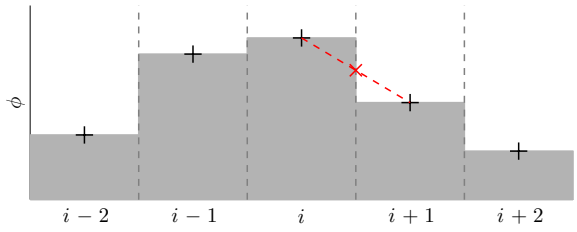


Figure 4.4: The second-order Central approximation.

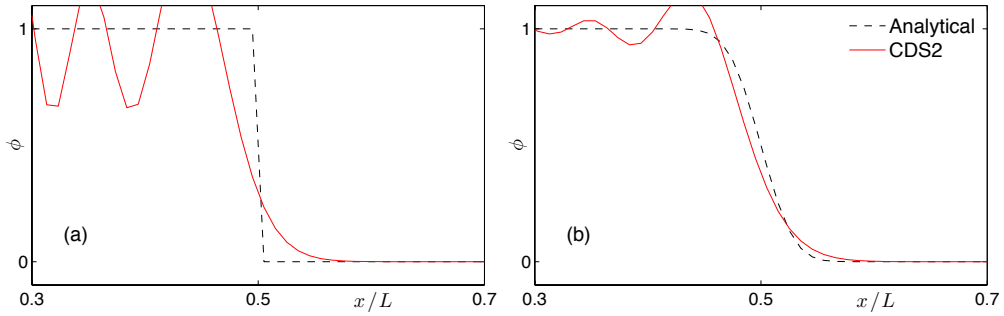


Figure 4.5: Two 1D functions transported by CDS2: (a) sharp, (b) smooth.

the scheme exhibits second-order spatial accuracy). The development of these oscillations is due to the influence of the point downstream of the cell face for which an approximation is being made, where the propagation of the oscillations in the upstream direction illustrates this 'downwind' effect. Considering Eq. 4.10 it is apparent that the approximation of ϕ_e is independent of the flow direction (i.e. the computational stencil is symmetric about the cell face), and CDS therefore lacks the property of transportiveness.

Figure 4.5(b) shows that the oscillations are lessened where the sharpness of the field is reduced, while the decay of those oscillations upstream of the sharp gradient is also increased. It is therefore suggested that the magnitude of the oscillations in the transported function is related to the second derivative, or the curvature, of the initial transported field. The (unsigned) curvature κ of a one-dimensional function ϕ is defined as:

$$\kappa = \left|\frac{d^2\phi}{dx^2}\right| \left[1 + \left(\frac{d\phi}{dx}\right)^2\right]^{-3/2} \tag{4.11}$$

The curvature will therefore tend to the absolute second derivative as the scalar gradient becomes small. A 'minimum effective radius' $R_{\min}$ can then be defined as a measure of the smoothness of the function ϕ:

$$R_{\min} = \frac{1}{\max(\kappa)} \tag{4.12}$$

The dependence of the oscillation amplitude (determined as the maximum difference between the initial and transported fields) on the minimum effective radius is shown in Fig. 4.6. The

trend shown is only applicable to this particular case (oscillations will also tend to increase with an increasing number of simulated time-steps, for example) so that this analysis provides only a qualitative indication. Figure 4.6 demonstrates that although CDS2 will generate oscillatory behaviour for rapid changes in gradient, those oscillations tend to zero as the smoothness of the scalar field increases. This suggests that there is a minimum acceptable effective radius beyond which the scalar field may be transported with a negligible oscillation magnitude; for this case, for example, $R_{\min} > 140$ corresponds to oscillations of less than 0.01%.

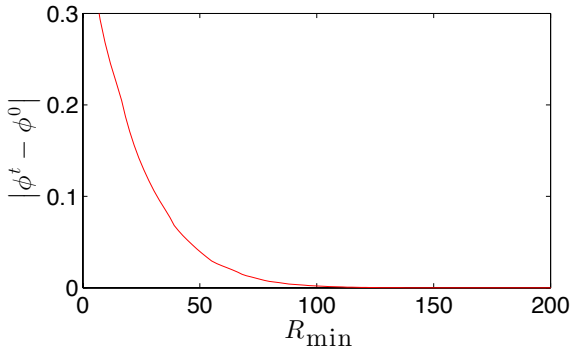


Figure 4.6: Oscillation reduction with increasing smoothness for CDS2.

4.3.3 Second-Order LUDS

The Linear Upwind Differencing Scheme (LUDS) estimates ϕ_F from an extrapolation of the gradient between the two points upwind of the cell face:

$$\phi_e \approx \begin{cases} \frac{1}{2}\left(3\phi(i) - \phi(i-1)\right) & \text{for} \quad u > 0 \\ \frac{1}{2}\left(3\phi(i+1) - \phi(i+2)\right) & \text{for} \quad u < 0 \end{cases} \tag{4.13}$$

Equation 4.13 is shown graphically in Fig. 4.7. As in the case of CDS2, the Taylor expansion – this time based on the two points upstream of ϕ_F – is truncated after the second term, indicating that LUDS is second-order accurate in space with respect to the cell size.

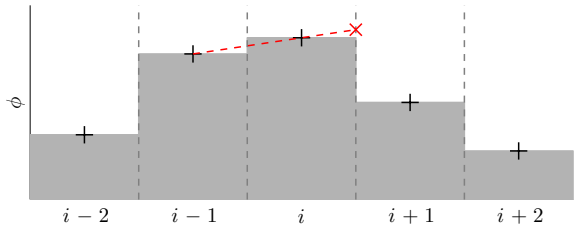


Figure 4.7: The second-order Upwind approximation.

Figure 4.8 shows the result of applying LUDS for the two one-dimensional cases introduced in §4.3.1. For this scheme the artefacts are reminiscent of CDS, but with a more damped oscillatory

behaviour downstream (as opposed to upstream) of the sharp scalar gradient.

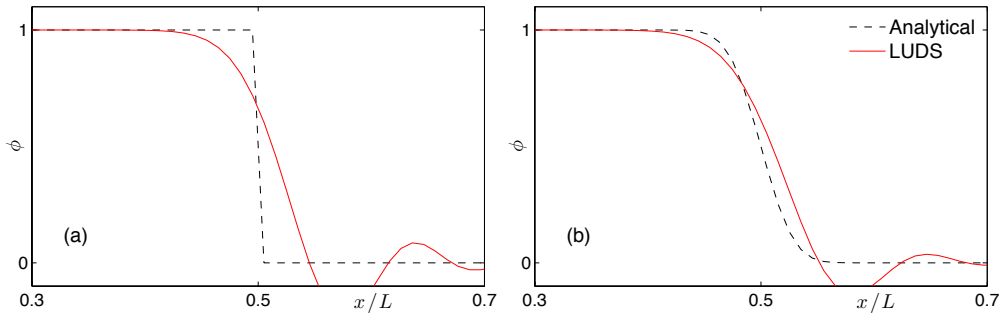


Figure 4.8: Two 1D functions transported by LUDS: (a) sharp, (b) smooth.

The undesirable properties of LUDS may be accounted for by considering its behaviour within a cell immediately downstream of a point of inflection. The cell $(i+1)$ in Fig. 4.7 is used as an example: the positive gradient across the two cells upstream results in an approximation of ϕ_e for cell i that is greater than $\phi(i)$. However, the actual value of ϕ_e can only lie between $\phi(i)$ and $\phi(i+1)$; the change in gradient around cell i causes an over-estimation of ϕ_e, and subsequently an over-estimated influx for cell $(i+1)$, so that the peak at the point of inflection will be amplified as it is transported. The same behaviour is observed for the cases in Fig. 4.8: in the cell at the bottom of the sharp gradient, ϕ_e will be under-predicted. This will cause a negative influx into the downstream cell, and subsequently, a negative scalar value. In the next time-step this negative value will cause an over-prediction of flux into the adjacent downstream cell, and therefore a positive peak. The oscillations thus propagate downstream as the number of simulated time-steps increases.

4.4 Higher-Order Schemes

Having considered the unsatisfactory behaviour of low order-of-accuracy schemes, some higher-order schemes will now be described and tested. These are the third-order Quadratic Upstream Interpolation for Convective Kinematics (QUICK) scheme, higher-order CDS, and the Essentially Non-Oscillatory (ENO) and Weighted Essentially Non-Oscillatory (WENO) schemes.

4.4.1 Third-Order QUICK

UDS, CDS2 and LUDS apply a constant value (zeroth-order polynomial) or constant gradient (first-order polynomial) assumption for the approximation of ϕ_F. To improve the accuracy of this approximation, the next logical step is to increase the order of the applied polynomial, which demands an increase in the number of points used for its evaluation. The QUICK scheme, proposed by Leonard [112], uses an upstream-weighted second-order fit, and may be written as:

$$\phi_e \approx \begin{cases} \frac{1}{8}\left(3\phi(i-1)+6\phi(i)-\phi(i+1)\right) & \text{for} \quad u<0 \\ \frac{1}{8}\left(3\phi(i)+6\phi(i-1)-\phi(i-2)\right) & \text{for} \quad u>0 \end{cases} \tag{4.14}$$

The dependence of the approximation in Eq. 4.14 on the direction of flow demonstrates the transportiveness of the scheme, while the accuracy may be shown to be third order in space (although Hirsch [113] argues that the scheme is second order for all practical purposes). Figure 4.9 provides a graphical interpretation of QUICK, with the quadratic fit across the three points shown by the red dashed line.

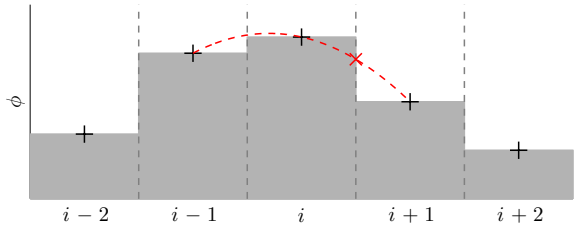


Figure 4.9: The third-order Quadratic Upwind approximation.

Applying the QUICK scheme to the two cases introduced in §4.3.1 results in the transported fields shown in Fig. 4.10. As can be seen, the QUICK scheme does not satisfy the condition of boundedness: oscillations are generated around the sharp scalar gradient, although they are smaller than those resulting from second-order CDS and LUDS. At the same time, it is apparent that QUICK introduces a degree of numerical diffusion, as the gradient of the sharp initial field is not captured. The gradient of the smooth field is well described, however.

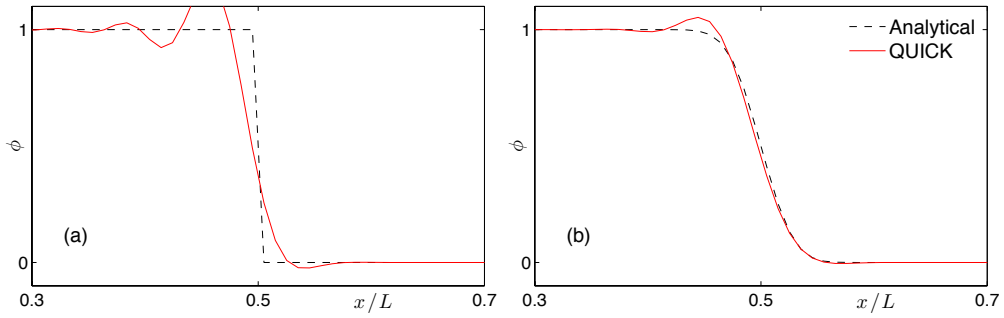


Figure 4.10: Two 1D functions transported by QUICK: (a) sharp, (b) smooth.

4.4.2 Higher-Order CDS

Higher-order Central Differencing Schemes may be formulated by extending the computational stencil. For example, the fourth-order accurate form of CDS (CDS4) may be written as:

$$\phi_e \approx -\frac{1}{16}\phi(i-2) + \frac{9}{16}\phi(i-1) + \frac{9}{16}\phi(i) - \frac{1}{16}\phi(i+1) \tag{4.15}$$

The scheme in Eq. 4.15 applies a cubic polynomial for the approximation of ϕ_F, as shown in Fig. 4.11. Furthermore, Fornberg [114] has formulated a generalised Central Scheme for any order of accuracy, and the resulting computational stencil coefficients for second- to eighth-order CDS are provided in Table 4.1.

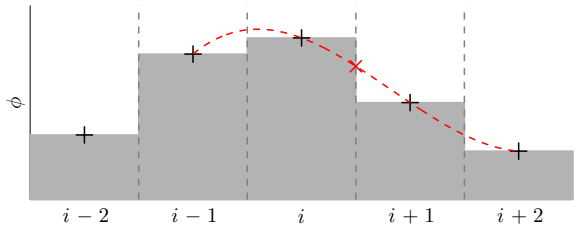


Figure 4.11: The fourth-order Central approximation.

Table 4.1: Coefficients for the approximation of ϕ_e up to eighth order-of-accuracy [114].

Order	Index of Neighbour							
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$
Second				$\frac{1}{2}$	$\frac{1}{2}$			
Fourth			$\frac{-1}{16}$	$\frac{9}{16}$	$\frac{9}{16}$	$\frac{-1}{16}$		
Sixth		$\frac{3}{256}$	$\frac{-25}{256}$	$\frac{150}{256}$	$\frac{150}{256}$	$\frac{-25}{256}$	$\frac{3}{256}$	
Eighth	$\frac{-5}{2048}$	$\frac{49}{2048}$	$\frac{-245}{2048}$	$\frac{1225}{2048}$	$\frac{1225}{2048}$	$\frac{-245}{2048}$	$\frac{49}{2048}$	$\frac{-5}{2048}$

Applying Fornberg's coefficients for an eighth-order CDS (CDS8), and subsequently transporting the one-dimensional cases introduced in §4.3.1, yields the results shown in Fig. 4.12. Although the spatial accuracy of the scheme has been increased by six orders, the transport of the sharp function yields a field which is not dissimilar to that from the second-order form of CDS. This similarity originates from the fact that a CDS of any order is independent of the flow direction, which is to say that it will always lack transportiveness.

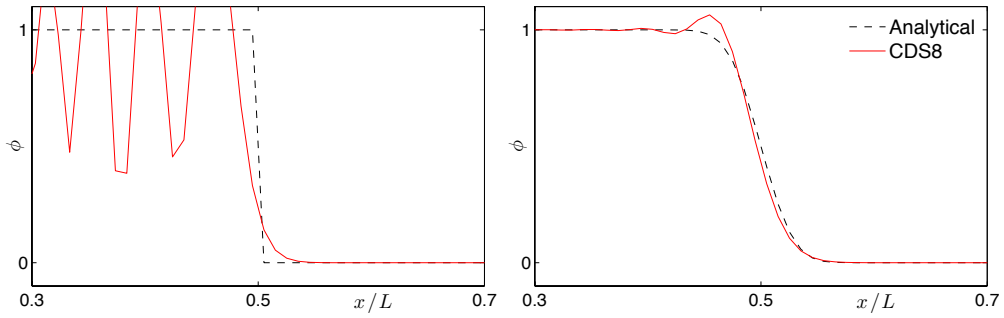


Figure 4.12: Two 1D functions transported by CDS8: (a) sharp, (b) smooth.

However, the recovered gradients for both the sharp and smooth transported functions have been improved, while the magnitude of the oscillations in the smooth case have been reduced. The improvement in the performance of CDS8 is made apparent by repeating the (qualitative) comparison of the minimum effective radius of the initial field to the oscillation magnitude of the

transported field, as previously described in §4.3.2 (Eqs. 4.11 and 4.12). Such an analysis is shown in Fig. 4.13, where, notably, the eighth-order scheme approaches stability much more rapidly as $R_{\min}$ increases (oscillations amplitudes fall below 0.01% for $R_{\min} > 67$). However, the magnitudes of oscillations for the sharpest cases ($R_{\min} \approx 10$) are actually the same for both ($\left|\phi^t - \phi^0\right| \approx 0.3$).

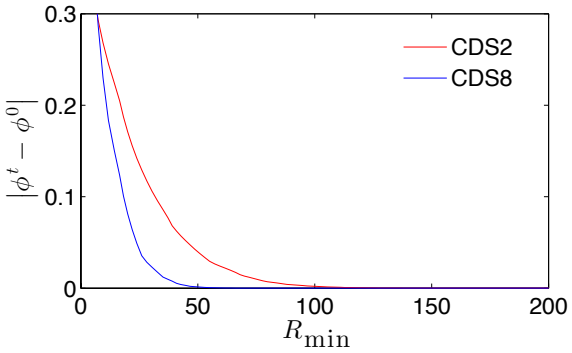


Figure 4.13: Oscillation reduction with increasing smoothness for CDS2 and CDS8.

4.4.3 ENO Schemes

All of the schemes that have been considered so far have used a fixed computational stencil (assuming a positive velocity), and are formulated such that they are independent of the local 'shape' of the transported function. However, it has been demonstrated that, for a Central Scheme at least, the behaviour of a convection scheme is strongly dependent on the topology of the transported scalar. To account for this, Harten et al. [115] introduced Essentially Non-Oscillatory (ENO) methods. In its simplest form, an ENO scheme may be written as:

$$\phi_e \approx \phi(i) + \frac{1}{2}\,\mathrm{amin}\left(\Delta\phi_w,\, \Delta\phi_e\right) \tag{4.16}$$

Where $\Delta\phi$ represents the local gradient of ϕ (with respect to the cell index), for example:

$$\Delta\phi_e = \phi(i+1) - \phi(i) \tag{4.17}$$

While the function $\mathrm{amin}(a,\, b)$ is defined as:

$$\mathrm{amin}(a,\, b) = \begin{cases} a, & |a| < |b| \\ b, & |b| \leq |a| \end{cases} \tag{4.18}$$

Considering the two possibilities for the value returned by $\mathrm{amin}(a,\, b)$, and substituting Eq. 4.17 into Eq. 4.16, this form of the ENO method may be written as:

$$\phi_e \approx \begin{cases} \phi(i) + \frac{1}{2}\left(\phi(i+1) - \phi(i)\right) = \frac{1}{2}\left(\phi(i+1) + \phi(i)\right), & |\Delta\phi_w| < |\Delta\phi_e| \\ \phi(i) + \frac{1}{2}\left(\phi(i) - \phi(i-1)\right) = \frac{1}{2}\left(3\,\phi(i) - \phi(i-1)\right), & |\Delta\phi_e| \leq |\Delta\phi_w| \end{cases} \tag{4.19}$$

Note that only the positive velocity approximation for ϕ_e is shown in Eqs. 4.16 and 4.19; for a negative velocity, the computational stencil is 'mirrored' about the position of ϕ_e (cf. the QUICK approximations for positive and negative velocities shown in Eq. 4.14).

A graphical representation of Eq. 4.19 is shown in Fig. 4.14. Examination of Eq. 4.19 reveals that in the first case, where the gradient downstream of the cell face is less than the gradient upstream, second-order CDS is used (black line); conversely, if the upstream gradient is the lesser of the two, LUDS is applied (red line).

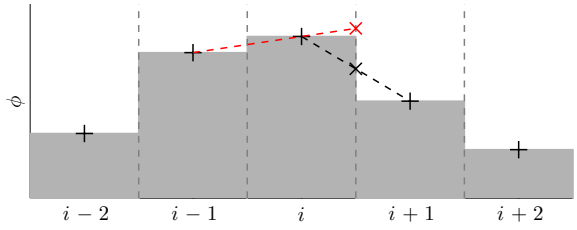


Figure 4.14: Two second-order polynomials used to construct an ENO approximation.

The use of the amin(a, b) function, which may be considered as a 'smoothness indicator', provides the ENO scheme with some sensitivity to the local shape of the scalar field. Second-order ENO is now applied for the test cases introduced in §4.3.1, and the results are shown in Fig. 4.15.

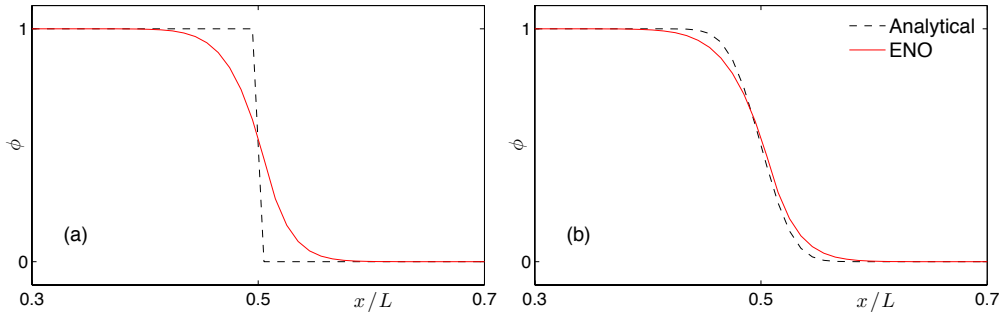


Figure 4.15: Two 1D functions transported by second-order ENO: (a) sharp, (b) smooth.

Even though this simple ENO scheme is built from two methods that individually display oscillatory behaviour, the ENO scheme itself is bounded, and is capable of delivering a relatively good description of the transported field for each case. However, both fields suffer from a degree of numerical diffusion, causing an under-estimation of the scalar gradient.

As the ENO scheme presented here selects from two second-order accurate methods, the ENO scheme itself is second-order accurate. Higher-order methods may be formulated in a similar fashion, as described by Harten et al. [115].

4.4.4 WENO Schemes

An extension to the ENO method has been proposed by Liu et al. [116], who developed the Weighted Essentially Non-Oscillatory (WENO) procedure. Rather than choosing one scheme based on some

function of the local scalar gradient, the WENO scheme applies a weighted combination of many schemes, where the weighting is a (continuous) function of the topology. The immediate advantage of this procedure is that, while an ENO method is limited to the accuracy of its constituent schemes, the WENO scheme may deliver an order of accuracy higher than its individual parts.

The formulation of the fifth-order WENO scheme is described here. Firstly, three third-order accurate approximations for ϕ_F are derived, for example:

$$\phi_e^{(1)} \approx \frac{1}{3}\phi(i-2) - \frac{7}{6}\phi(i-1) + \frac{11}{6}\phi(i) \tag{4.20}$$

$$\phi_e^{(2)} \approx -\frac{1}{6}\phi(i-1) + \frac{5}{6}\phi(i) + \frac{1}{3}\phi(i+1) \tag{4.21}$$

$$\phi_e^{(3)} \approx \frac{1}{3}\phi(i) + \frac{5}{6}\phi(i+1) - \frac{1}{6}\phi(i+2) \tag{4.22}$$

Each approximation is based on a different computational stencil, and therefore provides a different estimate for ϕ_F, as shown in Fig. 4.16.

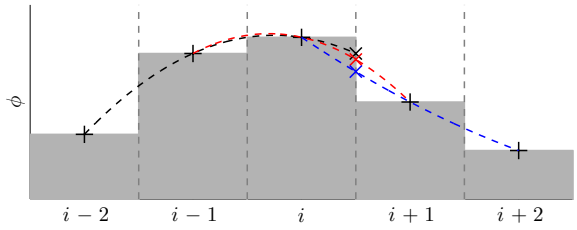


Figure 4.16: Three third-order polynomials used to construct a WENO approximation.

These three schemes are then combined to form a single approximation, as:

$$\phi_e \approx w_1\phi_e^{(1)} + w_2\phi_e^{(2)} + w_3\phi_e^{(3)} \tag{4.23}$$

To determine the weighting w_j for each approximation, Liu et al. define a smoothness indicator β_j for each of the three schemes in Eqs. 4.20 to 4.22:

$$\beta_1 = \frac{13}{12}\left(\phi(i-2) - 2\phi(i-1) + \phi(i)\right)^2 + \frac{1}{4}\left(\phi(i-2) - 4\phi(i-1) + 3\phi(i)\right)^2 \tag{4.24}$$

$$\beta_2 = \frac{13}{12}\left(\phi(i-1) - 2\phi(i) + \phi(i+1)\right)^2 + \frac{1}{4}\left(\phi(i-1) - \phi(i+1)\right)^2 \tag{4.25}$$

$$\beta_3 = \frac{13}{12}\left(\phi(i) - 2\phi(i+1) + \phi(i+2)\right)^2 + \frac{1}{4}\left(3\phi(i) - 4\phi(i+1) + \phi(i+2)\right)^2 \tag{4.26}$$

It can be seen that each of these indicators will be zero where the scalar gradient is zero, while the value of the indicator will increase as the variation between the three points within the stencil for each scheme increases. A weight $\hat{w}_j$ is then determined from:

$$\hat{w}_j = \frac{\gamma_j}{(\beta_j + \epsilon)^2} \qquad \text{with} \qquad \gamma_j = \left\{\frac{1}{10}, \frac{6}{10}, \frac{3}{10}\right\} \tag{4.27}$$

Where γ_j are the 'linear' WENO weights for each constituent scheme, and ϵ is a small number to avoid a division by zero[1]. Finally, each weight is normalised so that the sum of the weights w_j is unity:

$$w_j = \frac{\hat{w}_j}{\sum_{k=1}^{3} \hat{w}_k} \tag{4.28}$$

Even though this formulation of the WENO procedure is based on three third-order approximations for ϕ_F, it may be demonstrated that, in combination, the scheme is fifth-order accurate overall [94]. Applying the scheme to the one-dimensional cases introduced in §4.3.1 yields the transported fields shown in Fig. 4.17.

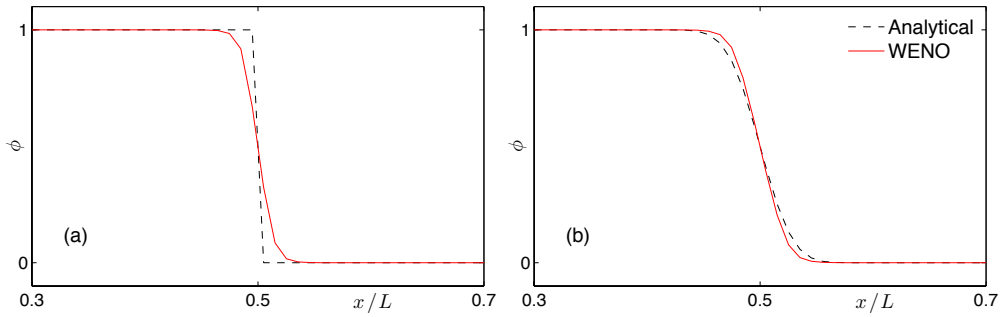


Figure 4.17: Two 1D functions transported by fifth-order WENO: (a) sharp, (b) smooth.

Figure 4.17(a) shows that the WENO scheme delivers the best approximation yet for the transport of a sharp change in gradient; the transported field is bounded, while experiencing only a limited amount of numerical diffusion. However, it is important to note that the boundedness of the scheme is not guaranteed, as it is built upon the assumption that the scalar field will be smooth in at least one of the three computational stencils appearing in Eqs. 4.20 to 4.22. Furthermore the numerical diffusivity is not zero, while Fig. 4.17(b) shows that the scalar gradient in the smooth case is over-predicted. This over-compression of shallow gradients may be expected, given that WENO methods were particularly formulated for the simulation of discontinuous flows (such as shock waves). A further potential disadvantage of the WENO procedure is the computational expense of the method, as, for each point in space, three different convection schemes must be applied, while a further number of operations must be performed in order to determine the weightings for each of the resulting approximations.

4.5 TVD Schemes

A final family of methods that finds wide use in CFD codes (amongst other areas) are the Total Variation Diminishing (TVD) schemes introduced by Harten [118]. There are an extension of the Monotone Upstream-centred Schemes for Conservation Laws (MUSCL) method developed by van

[1]Shu [117] uses a value of $\epsilon = 10^{-6}$. However, in the present case, the use of a relatively large 'small' number has a noticeable impact on the results. A value $\epsilon = 10^{-15}$ is therefore applied instead, to no obvious detrimental effect

Leer [119]. TVD schemes are designed to provide a non-oscillatory solution, while ensuring that the Total Variation (TV) of a function does not increase with time. Harten defines the TV of a one-dimensional function ϕ as:

$$\mathrm{TV}(\phi) \equiv \sum_{i=-\infty}^{\infty} |\Delta\phi_e| \tag{4.29}$$

Where the term $\Delta\phi$ represents the local scalar gradient, and takes the definition given in Eq. 4.17. Essentially, a scheme that possesses the property of being 'TVD' is guaranteed to preserve monotonicity, which is to say that local maxima and minima are not increased or decreased, respectively, and that no new maxima or minima will be created [110]. TVD schemes achieve this by the combination of UDS and CDS2, to form a second-order approximation for ϕ_F. The oscillatory nature of CDS is restricted by the addition of numerical diffusion from UDS. In general, the value of ϕ_e is approximated as:

$$\phi_e \approx \phi(i) + \frac{1}{2}B(r)\left(\phi(i+1) - \phi(i)\right) \tag{4.30}$$

Where $B(r)$ is the 'flux limiter', which is a function of the ratio r of the local scalar gradients:

$$r = \frac{\phi(i) - \phi(i-1)}{\phi(i+1) - \phi(i)} \tag{4.31}$$

Note that r is set to zero where the (absolute) value of the denominator of Eq. 4.31 approaches zero, while $B \geq 0$. Equations 4.30 and 4.31 are formulated for a positive velocity; the corresponding scheme for a negative velocity at the cell face is found by 'mirroring' the computational stencil around the position ϕ_e, as before. The action of the flux limiter function is to define the blend of UDS and CDS2, and because it is a function of the adjacent scalar gradients, it provides the scheme with some sensitivity to the local topology. Considering some typical possible values of B, a TVD scheme may return approximations of:

$$\phi_e \approx \begin{cases} \phi(i), & B = 0 \\ \frac{1}{2}\left(\phi(i+1) + \phi(i)\right), & B = 1 \\ \phi(i+1), & B = 2 \end{cases} \tag{4.32}$$

A TVD scheme may therefore be equivalent to UDS ($B = 0$), CDS2 ($B = 1$) or a 'downwind' scheme ($B = 2$); in general, the approximation of ϕ_F will be based on some linear combination of the two scalar values either side of it.

Finally a flux limiter, based on the gradient ratio r, must be constructed to determine how the values adjacent to ϕ_F are combined. As a starting point, it is possible to say that a function that has a locally constant, non-zero gradient (i.e. $r = 1$) may be exactly transported using CDS, as the second derivative in that region will be zero (i.e. R_{min} will be infinite). A flux limiter should therefore return a value $B = 1$ for an input $r = 1$. On the other hand, where the function experiences a sudden 'step', it is necessary to apply UDS to prevent the development of any oscillatory behaviour; in these regions B should therefore approach zero.

A further property that is useful (though not essential) for a flux limiter to possess is 'symmetric' behaviour, which is to say that it will operate identically on forward and backward gradients [110]. A flux limiter is symmetric in this sense if the following criterion is satisfied:

$$\frac{B(r)}{r} = B\left(\frac{1}{r}\right) \tag{4.33}$$

The 'shape' of a flux limiter function may otherwise be chosen with relative freedom. A wide variety of limiters are proposed in the literature, of which a selection are provided in Table 4.2; a number of other schemes are compared and contrasted in the review of Waterson and Deconinck [120]. Sweby [121] summarised the criteria for a scheme to be TVD, while adding a restriction to ensure that it possesses second-order accuracy. To visualise the response of different flux limiters he introduced what is now known as the Sweby diagram, showing the region that a flux limiter must remain within to ensure that the scheme is second-order accurate and TVD. A Sweby diagram showing the 14 limiters presented in Table 4.2, as well as the UDS, CDS2, LUDS and QUICK methods, is provided in Fig. 4.21, where the shaded area represents the second-order TVD region.

In order to assess the performance of these second-order TVD schemes, the one-dimensional test cases introduced in §4.3.1 are re-computed with a selection of the flux limiters appearing in Table 4.2.

MinMod As can be seen from Fig. 4.21, the MinMod limiter of Roe [123] follows the 'bottom edge' of the second-order TVD region, and so provides the lowest possible value of B for a given gradient ratio. This makes the scheme most biased towards the point upwind of the cell face for which ϕ_F is being estimated. As suggested by the transported fields in Fig. 4.18, the MinMod limiter is consequently the most numerically diffusive of all second-order TVD schemes.

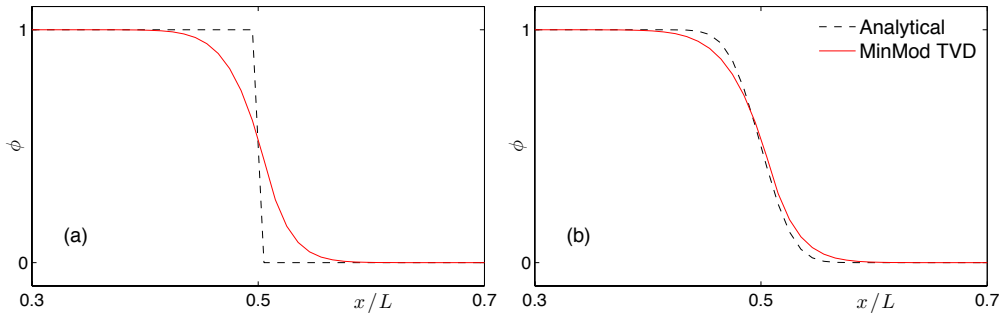


Figure 4.18: Two 1D functions transported by MinMod TVD: (a) sharp, (b) smooth.

SuperBee The SuperBee limiter [123], on the other hand, provides the highest possible value for B while remaining within the second-order TVD region. This gives the scheme maximum bias towards the downwind cell, which means that, of all second-order TVD schemes, SuperBee is the most capable of describing sharp changes in gradient (Fig. 4.19(a)). However, similar to the fifth-order WENO method, SuperBee is over-compressive when used to transport smooth functions, as it tends to make shallow gradients steeper (Fig. 4.19(b)).

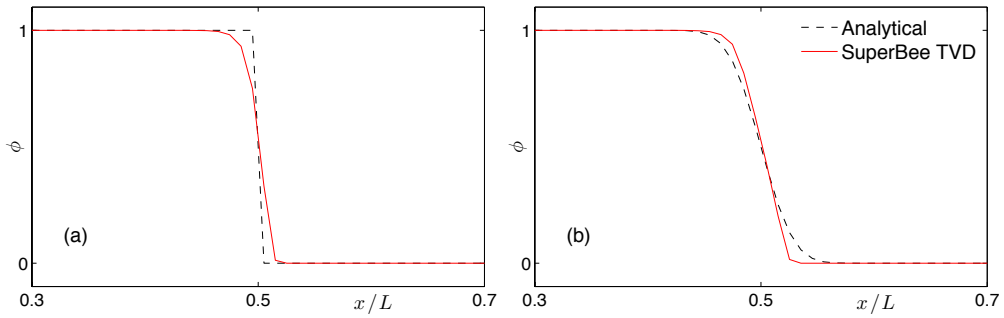


Figure 4.19: Two 1D functions transported by SuperBee TVD: (a) sharp, (b) smooth.

CHARM The Cubic-parabolic High Accuracy Resolution Method (CHARM) limiter is selected as a compromise between the two extremes of MinMod and SuperBee, and is currently the flux limiter of choice for the `PsiPhi` DNS/LES code (introduced in Chapter 3). However, the original formulation of Zhou [130] (provided in Table 4.2) tends to a value $B = 3$ as the gradient ratio r tends to infinity, and lies outside of the second-order TVD region above $r = \left(3 + \sqrt{17}\right)/2$. A straightforward solution to this is to limit the function $B(r)$ such that it returns a maximum value of $B = 2$, constraining the limiter to the upper edge of the second-order TVD region. Applying the original and modified versions of the CHARM TVD scheme results in the transported fields shown in Fig. 4.20, where the non-fulfilment of the TVD property for the original implementation (red) is apparent from the slight undershoot (around 1.5%) visible downstream of the maximum scalar gradient for each case. Although the gradient for the smooth case is relatively well captured, both versions of the scheme demonstrate a degree of numerical diffusion.

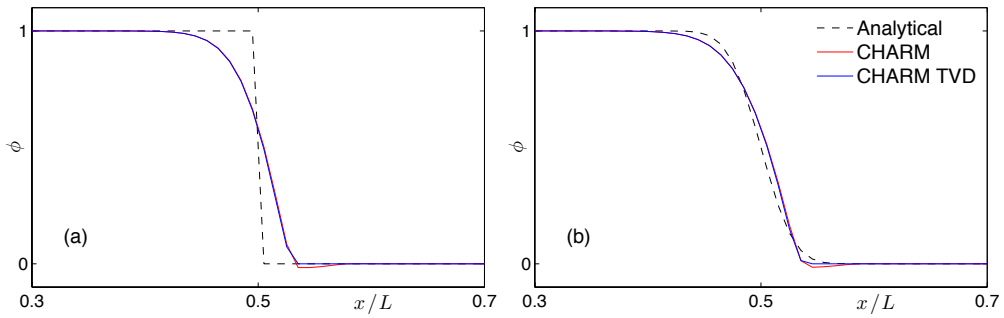


Figure 4.20: Two 1D functions transported by CHARM TVD: (a) sharp, (b) smooth.

Table 4.2: Flux limiters from the literature. Non-TVD limiters are shown in the lower part. Limiters marked (*) are 'symmetric' (Eq. 4.33).

Name	Author(s)	Flux Limiter Function
Koren	Koren [122]	$B(r) = \max\left[0, \min\left(2\,r, \frac{1}{3}(1+2\,r), 2\right)\right]$
MinMod*	Roe [123]	$B(r) = \max\left[0, \min\left(1, r\right)\right]$
Monotonised Central*	van Leer [124]	$B(r) = \max\left[0, \min\left(2\,r, \frac{1}{2}(1+r), 2\right)\right]$
Osher	Chatkravathy, Osher [125]	$B(r) = \max\left[0, \min\left(\beta, r\right)\right]; (1 \leq \beta \leq 2)$
Ospre*	Waterson, Deconinck [126]	$B(r) = 3\left(r^2+r\right) / 2\left(r^2+r+1\right)$
SuperBee*	Roe [123]	$B(r) = \max\left[0, \min(1, 2\,r), \min(2, r)\right]$
Sweby*	Sweby [121]	$B(r) = \max\left[0, \min\left(\beta\,r, 1\right), \min\left(\beta, r\right)\right]; (1 \leq \beta \leq 2)$
UMIST	Lien, Leschziner [127]	$B(r) = \max\left[0, \min\left(2\,r, \frac{1}{4}(1+3\,r), \frac{1}{4}(3+r), 2\right)\right]$
van Albada*	van Albada et al. [128]	$B(r) = \left(r^2+r\right) / \left(r^2+1\right)$
van Leer*	van Leer [129]	$B(r) = \left(\lvert r\rvert + r\right) / \left(\lvert r\rvert + 1\right)$
CHARM	Zhou [130]	$B(r) = \begin{cases} \left(r\left(3\,r+1\right)\right) / \left(r+1\right)^2 & \text{for} \quad r > 0, \\ 0 & \text{for} \quad r \leq 0 \end{cases}$
HCUS	Waterson, Deconinck [126]	$B(r) = 3\left(r + \lvert r\rvert\right) / 2\left(r+2\right)$
HQUICK	Waterson, Deconinck [126]	$B(r) = 2\left(r + \lvert r\rvert\right) / \left(r+3\right)$
SMART	Gaskell, Lau [131]	$B(r) = \max\left[0, \min\left(\frac{1}{4}(1+3\,r), 4\right)\right]$

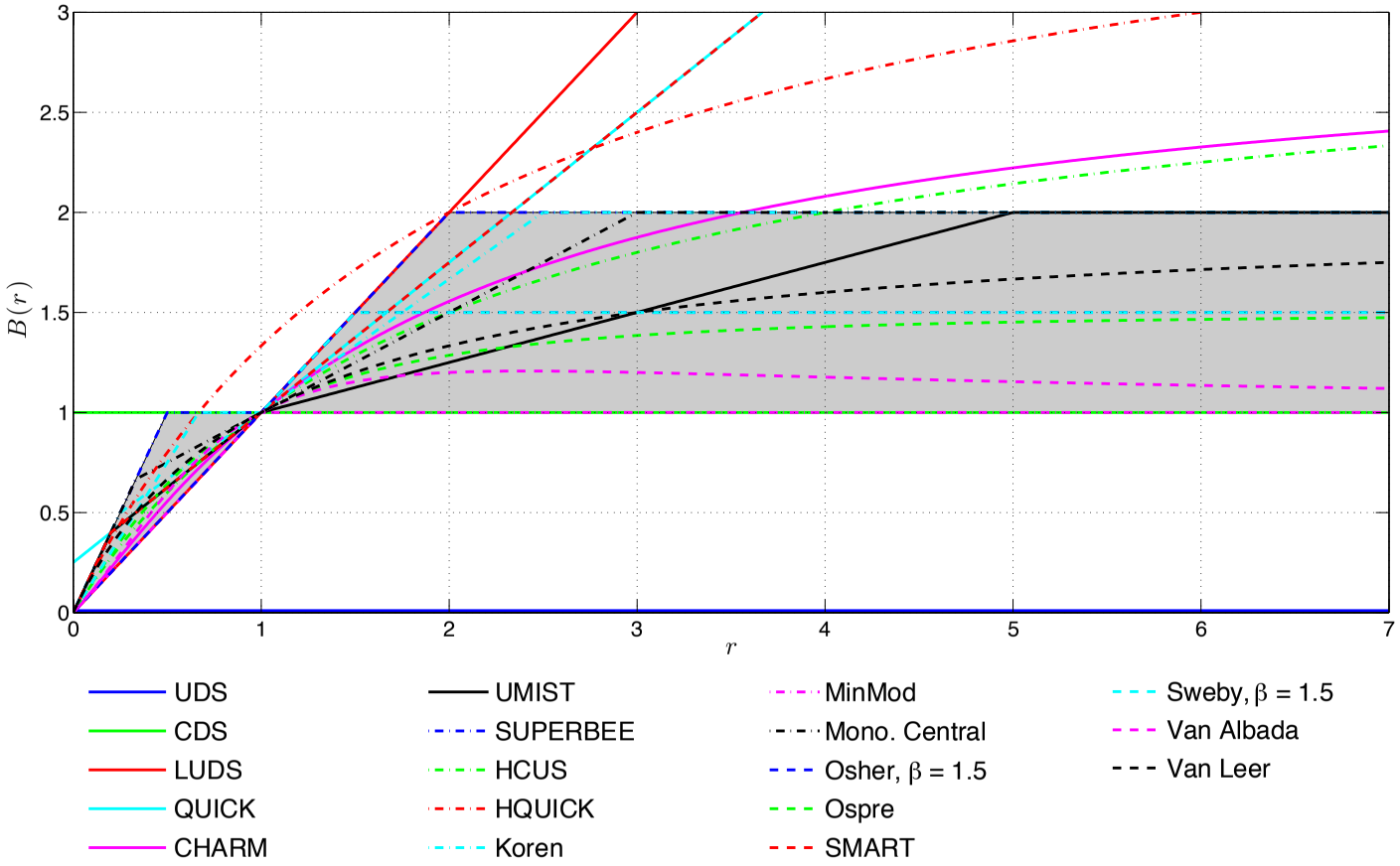


Figure 4.21: Sweby diagram of 18 schemes, including the 14 flux limiters in Table 4.2. The shaded area represents the bounds for second-order TVD schemes.

4.6 Summary of Convective Schemes

Discretisation of the governing equations of Fluid Mechanics by the Finite Volume Method demands an approximation of a scalar quantity at each face of a computational cell, so that the fluxes over the boundaries of that cell may be calculated. A wide range of approximation methods have been proposed, of which a selection have been examined in this chapter. The suitability of a convection scheme depends on the computational stencil from which an approximation is formulated, and the sensitivity of that scheme to the topology of the local scalar field. However, in general these schemes may only provide a compromise between the undesirable artefacts of numerical diffusion, from the influence of points upstream of the cell face, and oscillation, due to the influence of downstream positions.

The effects of numerical diffusion are most readily observed with the first-order Upwind Differencing Scheme, while oscillations are prevalent in all Central Schemes (although the magnitude of these oscillations is reduced as the smoothness of the scalar field is increased). Weighted Essentially Non-Oscillatory methods are capable of providing relatively accurate results by adapting to the local shape of the scalar field, but are not guaranteed to suppress oscillatory behaviour, and suffer from an increase in computational cost compared to other approaches. A Total Variation Diminishing scheme is guaranteed to deliver bounded results provided that its flux limiter remains within the second-order TVD region defined by Sweby, while the variety of available flux limiters permits a wide degree of flexibility in the TVD method. However, this is equally a disadvantage: flux limiters are usually developed for a particular problem, but there exists no single flux limiter capable of delivering accurate results for all cases.

4.7 Diffusion in the Finite Volume Method

The discretisation of diffusive flux terms appearing in the transport equation for a scalar quantity requires a different treatment to that of the convective fluxes. To understand how diffusive fluxes are approximated, the one-dimensional domain shown in Fig. 4.1 is reconsidered; however, in this case the velocity field is set to zero and a finite diffusion coefficient D_ϕ is applied to the scalar field. From the general transport equation (Eq. 2.9), the evolution in time of ϕ may be written as:

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x}\left(D_\phi \frac{\partial \phi}{\partial x}\right) \tag{4.34}$$

The domain is then assumed to be in a steady state as before:

$$\frac{d}{dx}\left(D_\phi \frac{d\phi}{dx}\right) = 0 \tag{4.35}$$

Integration of Eq. 4.35 across a single cell yields:

$$\int_A D_\phi \left(\frac{d\phi}{dx} \cdot n\right) dA = \left[D_\phi \left(\frac{d\phi}{dx} \cdot 1\right) A\right]_e + \left[D_\phi \left(\frac{d\phi}{dx} \cdot (-1)\right) A\right]_w = 0 \tag{4.36}$$

A general expression for the diffusive flux $F^d(i)$ across a cell i is then:

$$F^d(i) \approx \sum_F \left[D \left(\frac{d\phi}{dx} \cdot n \right) A \right]_F \qquad \text{with} \qquad F = \{e, w, \ldots\} \tag{4.37}$$

Where $F^d(i)$ is equal to zero for a steady state problem, or is equal to the rate of change of the scalar ϕ within cell i for a transient problem. Equation 4.37 may equally be applied to two- or three-dimensional cases. As for a convective flux estimation, the only unknown in this expression relates to the scalar field. For diffusion, however, it is necessary to approximate a scalar gradient, which is less problematic than the cell-faced scalar value estimate required for a convection scheme. The concepts of 'upwind' and 'downwind' do not apply in the context of a diffusion-only process, and there is an equal influence of the points each side of the cell face for which a gradient approximation is being made. The simplest estimate is thus found from the difference between the scalar values either side of the cell face, for example:

$$\left. \frac{d\phi}{dx} \right|_e = \frac{\phi(i+1) - \phi(i)}{\Delta} \tag{4.38}$$

A Taylor expansion about the point x_e for this approximation of the gradient of ϕ demonstrates that Eq. 4.38 is second-order accurate on a uniform grid.

In addition to the generalised Central Differencing Scheme described in §4.4.2, Fornberg [114] also provides a general expression for a central approximation of the scalar gradient at a cell face. The resulting computational stencil coefficients for second- to eighth-order accurate schemes are given in Table 4.3.

Table 4.3: Coefficients for the approximation of $\left. \frac{d\phi}{dx} \right|_e$ up to eighth order-of-accuracy [114].

Order	Index of Neighbour							
	$i-3$	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$
Second				-1	1			
Fourth			$\frac{1}{24}$	$\frac{-9}{8}$	$\frac{9}{8}$	$\frac{-1}{24}$		
Sixth		$\frac{-3}{640}$	$\frac{25}{384}$	$\frac{-75}{64}$	$\frac{75}{64}$	$\frac{-25}{384}$	$\frac{3}{640}$	
Eighth	$\frac{5}{7168}$	$\frac{-49}{5120}$	$\frac{245}{3072}$	$\frac{-1225}{1024}$	$\frac{1225}{1024}$	$\frac{-245}{3072}$	$\frac{49}{5120}$	$\frac{-5}{7168}$

5 Formulation of the Joint Eulerian-Lagrangian Method

In the previous chapter, the problems associated with the discretisation of convective fluxes in the Finite Volume framework were described. In this chapter a possible solution to these problems – referred to as the *joint Eulerian-Lagrangian method* – is introduced. Firstly, a brief overview of related work as described in the literature is presented. A theoretical formulation of the idea is then outlined, as the decomposition of a scalar into its high- and low-frequency components, and the formulation of a separate transport equation for each component. A description of the actual implementation of the method is then provided, with details of the main elements required for its application in a time-dependent Computational Fluid Dynamics simulation. Aspects of the implementation in the context of High-Performance Computing, where a simulation domain is shared between a number of computer processors, are also considered.

5.1 Review of the Literature

The combination of Eulerian and Lagrangian descriptions is not a novel technique in itself, and various facets of this combination are described by a large body of work in the literature. The following review is not an exhaustive summary of that work, but is included to provide an overview of the origins of the proposed method, and to highlight some similar techniques that are already in use.

Perhaps the earliest example of Lagrangian particles implemented in an underlying Eulerian framework is the Particle-in-Cell (PIC) method, proposed and developed by the research group of Harlow at the Los Alamos National Laboratory during the 1950s and 1960s; some examples of Harlow's early contributions may be found in [132, 133, 134], while a more extensive bibliography is provided in his later review of the method and its advances [135]. In the PIC method the momentum and energy properties of a fluid flow are stored and computed in the Eulerian framework, while Lagrangian particles are used to represent the mass of the fluid. Particle motion is described by interpolating the Eulerian velocities on to the Lagrangian particle positions, such that a calculation cycle is split into separate Eulerian and Lagrangian 'phases'. Applications for fluid simulations are numerous; for example, Amsden and Harlow used the method in the two-dimensional, compressible simulation of a supersonic wake [136], while Snider presented incompressible results in three dimensions for particle-laden flows [137]. Subsequent extensions of the PIC method include formulations for multiphase flow by Cook et al. [138], in addition to Snider's previously-mentioned work; the inclusion of relativistic effects [139]; simulations of plasma physics with charged particles by Bird-

sall [140] and Birdsall and Langdon [141]; and, beyond the context of fluids, for solid mechanics calculations [142]. A further notable development is the Fluid-Implicit-Particle (FLIP) method of Brackbill and Ruppel [143] and Brackbill et al. [144], where the information stored by each particle is extended to include mass, momentum, internal energy, and other fluid properties.

The Marker-and-Cell (MAC) method, originally proposed by Harlow and Welch [145] and expanded upon by Welch et al. [146], is distinct from the PIC method in that the transported particles act only as massless tracers and do not directly influence the dynamics of the flow. In the former contribution these 'markers' are used to indicate the configuration of an incompressible fluid with a free surface, where Eulerian cells containing one or more particles are filled and empty cells are external to the liquid phase. The physics of the flow is solved in the Eulerian framework, and the markers are advected by linear interpolation of the underlying velocity field on to the particle position (as for the PIC method). The markers may then be used to visualise the flow, but can also track the surface of the liquid, defined by the region in which filled cells and empty cells are adjacent. Harlow and Welch tested the MAC method by simulating partially- and fully-opened sluice gates, using the IBM 7030 Stretch computer – IBM's first computing hardware based on transistors, rather than thermionic valves – and claimed excellent agreement with the experimental results of Martin and Moyce [147]. The method was subsequently applied by the group of Harlow, for example in the simulation of the splash of a liquid drop [148], and continues to find applications in a range of fluid research topics such as rheology [149]. However, the MAC method is now most widely used in computer graphics, and a summary of advances may be found in [150]. The method is also cited as a precursor to the Volume of Fluid Sub-Mesh method (VOF-SM) of Vincent et al. [151].

The computational resources available for the research of fluid dynamics problems have improved significantly since the first calculations were made. This trend is famously quantified by Gordon Moore, co-founder of the Intel Corporation, who predicted in 1965 that transistor density on a computer processor (and correspondingly, processor speed) would approximately double every two years [152]. This trend in processor speed has continued to hold up to the present day, even as manufacturing technologies have encountered significant physical barriers: transistor junction gaps are now smaller than the wavelength of ultra-violet light. The development of computers therefore resulted in an increasing applicability of particle-based methods for fluid flow simulations, and the Direct Simulation Monte Carlo (DSMC) technique, originally proposed by Bird [153], was one such result. In DSMC the usual macroscopic, continuum-based approach (as used in the flow-solving methods of Reynolds-Averaged Navier Stokes (RANS) simulation, Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES)) is replaced with a microscopic model, in which fluid properties are described by considering the discrete particles comprising that flow; the 'Monte Carlo' aspect of the method stems from the stochastic treatment of inter-particle interactions (collisions), which would otherwise make the expense of such a simulation prohibitive. Computational cost is also mitigated by 'grouping' individual particles together, such that the total particle number may be reduced by two orders of magnitude or more. A review of the first implementations of DSMC is provided by Bird [154], while a more complete description of the method and its applica-

tions is given in the textbook by Bird [155]. Related to DSMC is the Lattice Boltzmann Method (LBM), proposed by Hardy et al. [156] and successfully applied for the Navier-Stokes equations in two dimensions by Frisch et al. [157], in which the discrete particle motion is restricted to an underlying lattice of possible locations. A review of LBM is provided by Chen and Doolen [158]. An advantage of both of these methods is that they remove any dependence of the solution on the continuum assumption (see §2.1.1), so that they may be applied in situations where the Knudsen number becomes non-negligible. (For example, a spacecraft in the near-vacuum of the Earth's outer atmosphere will pass through a fluid in which the mean free path of a molecule is comparable to the size of the craft). Although DSMC and LBM are usually based only on the motion of particles, it is typically convenient to generate an Eulerian description of the Lagrangian field for flow visualisation.

A significant application of particles in combustion Computational Fluid Dynamics is the Lagrangian joint Probability Distribution Function (PDF) method of Pope [24], which is an extension to the previous (Eulerian) PDF method proposed by Dopazo and O'Brien [21] and developed by Pope [22]. In this method Lagrangian particles are used to describe the properties of a flow (such as the chemical composition), and these properties are updated in time according to a stochastic model. Interactions between particles subsequently allow reaction terms, in the transport equations governing the evolution of the fluid properties, to be treated exactly. Lagrangian PDF methods have been applied extensively in the context of RANS simulations – for example, by the group of Lindstedt [159, 160] – and continue to be developed both in RANS, and more recently in LES [161, 162].

Combined Eulerian and Lagrangian descriptions are also used extensively for the simulation of particulate flows, where the particles assume more of a physical significance within the continuum. Particle motion must then account for drag, gravitational, and inter-particle forces, while it may also be necessary to include the momentum exchange from the particle to the surrounding flow. Such an approach has been developed as an extension to the PIC method by Andrews and O'Rourke [163] and Snider et al. [164], and has been applied by Patankar and Joseph [165, 166] and others. Similar methods find frequent use in the simulation of, for example, coal combustion, as suggested by Lockwood et al. [167] and used by Snider et al. [168] and Franchetti et al. [109], amongst others.

The techniques considered so far have largely separated aspects of a computation into distinct Eulerian and Lagrangian parts (although the PIC and Lagrangian PDF methods, and other approaches in which the particles have an active role in the flow development, require a close coupling between the two descriptions). Attention is now turned to procedures where the Eulerian and Lagrangian frameworks are more closely interlinked, perhaps the most notable example being the Arbitrary Lagrangian-Eulerian (ALE) method. Noh [169], Franck and Lazarus [170], Trulio [171] and Hirt et al. [172] introduced and developed the method for finite difference calculations in the 1960s and 1970s, while later works in finite element computations include the contributions of Hughes et al. [173], Belytschko et al. [174] and Huerta and Liu [175]. In the ALE method the vertices of the computational grid are allowed to move exactly with the fluid, as in a Lagrangian description; are fixed in position, as for an Eulerian description; or may deform, in some specified way, as a

compromise between the two. The benefit of this approach is that it combines the advantages of the Lagrangian and Eulerian methods: the Lagrangian mesh is well-suited to the tracking of free surfaces and interfaces, while the Eulerian mesh is more capable of handling significant distortions in the flow-field. The ALE method therefore finds particular application in multiphase flows [176, 177]. A comprehensive overview of the ALE method is provided by Donea et al. [178]. However, Fedkiw claims that the ALE approach may suffer from low accuracy, and therefore presents an alternative in which the Eulerian (fluid) and Lagrangian (solid) meshes are kept more distinct [179].

A more recent application combining the Eulerian and Lagrangian frameworks has been proposed by Enright et al. [180], who applied marker particles around the interface defined by a 'level set' to improve resolution in under-resolved regions. Briefly stated, the (Eulerian) level set method proposed by Osher and Sethian [181] describes an interface as a contour or surface of an implicit function (in two or three dimensions, respectively); the dynamics of the interface may then be determined by applying a convection equation to that function. A detailed account of the level set method is provided in the textbook of Osher and Fedkiw [182], while further examples of its application are given in the review of Sethian [183]. However, a limitation of the method is that it is incapable of capturing structural details of the interface below the resolution of the underlying Eulerian grid, and this forms the motivation for the addition of Lagrangian markers. In the modification of Enright et al. these marker particles are placed around the initial location of the interface defined by the level set, and are (passively) convected; because the particles and interface move with the same velocity, particles 'inside' the interface should not be able to cross to the 'outside'. Should such a crossing occur – indicating that the level set no longer accurately represents the interface – the implicit function is locally rebuilt, according to the positions of the markers. In addition, the marker particles are able to provide a local description of the interface at a resolution greater than that of the Eulerian mesh. Particles may be added in cells where the particle number has dropped, or deleted from cells that are beyond some maximum distance from the interface (as defined by the level set). The particle level set method has been applied favourably to a variety of test cases, including the solid body rotation of Zalesak [184] and the single vortex used by Rider and Kothe [185]. In a subsequent contribution Enright et al. demonstrated that the computational cost of the method may be reduced by applying low order-of-accuracy schemes for the convection of the implicit function [186]. A similar method was also presented by Hieber and Koumoutsakos [187], while further developments and applications were presented by Losasso et al. [188], Gaudlitz and Adams [189], Li et al. [190], and others.

5.2 Theoretical Formulation

The main concept of the joint Eulerian-Lagrangian (E-L) method is illustrated in Fig. 5.1, where a scalar quantity ϕ (black line) is described as the sum of a smooth, low-frequency field ϕ^E (blue line) and a high-frequency field ϕ^L (red line). The scalar 'frequencies' in this context refer to the constituents of the scalar function if it were to be expressed as a Fourier series (although no Fourier transformation will be performed in the implementation itself). The smooth field may be

transported with high accuracy by a traditional Eulerian description, using one of the methods described in Chapter 4; the high-frequency field, on the other hand, is described and transported using Lagrangian particles. The particle field can thus be considered to be a localised adjustment to a well-resolved Eulerian field, while the original scalar function may be recovered from the summation of the two fields. The potential of the proposed Eulerian-Lagrangian technique to improve the accuracy and resolution of convective scalar transport has not been exploited before.

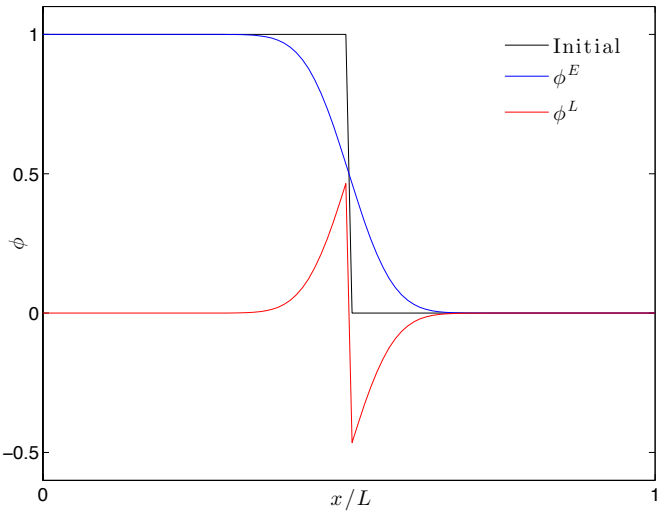


Figure 5.1: The Eulerian-Lagrangian decomposition.

The approach has potential for a wide range of problems, as the high-frequency scalar information is often constrained to a corrugated surface in space, for example across a mixing layer, or at an interface; it is then sufficient to describe only these regions using particles. In combustion problems, for example, such surfaces are found in both premixed flow, where there are thin flame fronts, and non-premixed flow, where there are steep gradients around shear and mixing layers. A limited distribution of particles should significantly reduce the computational cost of the method in comparison to previous techniques, such as the PIC method, that have treated the entirety of the scalar field with a Lagrangian description.

5.2.1 The Eulerian-Lagrangian Decomposition

The theoretical formulation of the proposed E-L method treats a conserved scalar ϕ as the summation of its low- and high-frequency parts:

$$\phi = \phi^E + \phi^L \tag{5.1}$$

Where the low-frequency component ϕ^E is determined by the convolution of the initial scalar ϕ by some filter function $h(x_i - x_i')$:

$$\phi^E(x_i,\, t) = \iiint_{-\infty}^{\infty} \phi(x_i,\, t)\, h(x_i - x_i')\, dx_i' \tag{5.2}$$

The high-frequency component ϕ^L is then determined from:

$$\phi^L = \phi - \phi^E \tag{5.3}$$

A constant density transport equation (comparable to Eq. 2.9) may now be written for each component of ϕ:

$$\frac{\partial \phi^E}{\partial t} + u_j \frac{\partial \phi^E}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu}{Sc}\frac{\partial \phi^E}{\partial x_j}\right) + S_{\phi^E} \tag{5.4}$$

$$\frac{\partial \phi^L}{\partial t} + u_j \frac{\partial \phi^L}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\nu}{Sc}\frac{\partial \phi^L}{\partial x_j}\right) + S_{\phi^L} \tag{5.5}$$

Where the diffusivity D_ϕ is expressed in terms of the Schmidt number (Eq. 2.11), and the source and sink terms appearing in Eqs. 5.4 and 5.5 must represent additions or subtractions to the (overall) scalar field, but will also account for any exchange of scalar information between the two phases that may be required to keep the low-frequency field suitably smooth. This exchange will take place during an operation of 're-initialisation', which will be explained in detail in §5.4.4. A corresponding set of equations for a variable density DNS may be formulated by multiplying the differentiated terms in Eqs. 5.4 and 5.5 by the local density ρ:

$$\rho\left(\frac{\partial \phi^E}{\partial t} + u_j \frac{\partial \phi^E}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\mu}{Sc}\frac{\partial \phi^E}{\partial x_j}\right) + S_{\phi^E} \tag{5.6}$$

$$\rho\left(\frac{\partial \phi^L}{\partial t} + u_j \frac{\partial \phi^L}{\partial x_j}\right) = \frac{\partial}{\partial x_j}\left(\frac{\mu}{Sc}\frac{\partial \phi^L}{\partial x_j}\right) + S_{\phi^L} \tag{5.7}$$

Where the LHS terms have been simplified by expansion with the chain rule and a subsequent substitution of the continuity equation (Eq. 2.3). In the context of a DNS these pairs of equations may be solved independently and directly; no further consideration of any sub-grid scale (SGS) terms in the velocity or scalar fields is required, as the velocities, and high- and low-frequency scalars, are well-resolved at the spatial resolution of the computational grid (assuming $Sc \leq 1$). For a constant density LES, however, corresponding filtered transport equations for $\bar{\phi}^E$ and $\bar{\phi}^L$ contain the unclosed terms $\overline{u_j\phi}^E$ and $\overline{u_j\phi}^L$ respectively, which, following the treatment of such terms that has been applied previously (§2.3.2), yields:

$$\frac{\partial \bar{\phi}^E}{\partial t} + \bar{u}_j \frac{\partial \bar{\phi}^E}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\bar{D}_\phi \frac{\partial \bar{\phi}^E}{\partial x_j} - \tau_j^{\phi^E}\right) + \bar{S}_{\phi^E} \tag{5.8}$$

$$\frac{\partial \bar{\phi}^L}{\partial t} + \bar{u}_j \frac{\partial \bar{\phi}^L}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\bar{D}_\phi \frac{\partial \bar{\phi}^L}{\partial x_j} - \tau_j^{\phi^L}\right) + \bar{S}_{\phi^L} \tag{5.9}$$

Applying the gradient diffusion model described in §3.6 for each of the SGS scalar flux terms,

substituting the turbulent (modelled) viscosity for the term $(C\Delta)^2 |S|$ appearing in Eq. 3.26, and expressing the diffusivity $\bar{D}_\phi$ in terms of the (laminar) Schmidt number (Eq. 2.11), results in:

$$\frac{\partial \bar{\phi}^E}{\partial t} + \bar{u}_j \frac{\partial \bar{\phi}^E}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\frac{\bar{\nu}}{Sc} + \frac{\nu_t}{Sc_t}\right)\frac{\partial \bar{\phi}^E}{\partial x_j}\right] + \bar{S}_{\phi^E} \tag{5.10}$$

$$\frac{\partial \bar{\phi}^L}{\partial t} + \bar{u}_j \frac{\partial \bar{\phi}^L}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\frac{\bar{\nu}}{Sc} + \frac{\nu_t}{Sc_t}\right)\frac{\partial \bar{\phi}^L}{\partial x_j}\right] + \bar{S}_{\phi^L} \tag{5.11}$$

The sum of the laminar and turbulent diffusivities appearing in Eqs. 5.10 and 5.11 is the same for each of the Eulerian and Lagrangian phases, although the treatment of each term will be very different. Finally the Favre-averaged LES-filtered transport equations for the two phases may be written as:

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{\phi}^E\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\phi}^E\right) = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\tilde{D}_\phi \frac{\partial \tilde{\phi}^E}{\partial x_j} + \hat{\tau}_j^{\phi^E}\right)\right] + \bar{S}_{\phi^E} \tag{5.12}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{\phi}^L\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\phi}^L\right) = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\tilde{D}_\phi \frac{\partial \tilde{\phi}^L}{\partial x_j} + \hat{\tau}_j^{\phi^L}\right)\right] + \bar{S}_{\phi^L} \tag{5.13}$$

While substitution of the gradient diffusion model as before yields:

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{\phi}^E\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\phi}^E\right) = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\frac{\tilde{\nu}}{Sc} + \frac{\nu_t}{Sc_t}\right)\frac{\partial \tilde{\phi}^E}{\partial x_j}\right] + \bar{S}_{\phi^E} \tag{5.14}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{\phi}^L\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{\phi}^L\right) = \frac{\partial}{\partial x_j}\left[\bar{\rho}\left(\frac{\tilde{\nu}}{Sc} + \frac{\nu_t}{Sc_t}\right)\frac{\partial \tilde{\phi}^L}{\partial x_j}\right] + \bar{S}_{\phi^L} \tag{5.15}$$

Four sets of transport equations have thus been presented for the four cases of: constant density DNS (Eqs. 5.4 and 5.5); variable density DNS (Eqs. 5.6 and 5.7); constant density LES (Eqs. 5.10 and 5.11), and variable density LES (Eqs. 5.14 and 5.15). From these equations it can be seen that the only practical difference between a constant density simulation and a Favre-filtered one is whether the transported quantity is multiplied by the local (filtered) density. However, for a variable density simulation this causes an additional complication, as the total diffusive flux of the density-weighted transported quantity $\bar{\rho}\tilde{\phi}$ is proportional to the gradient of the unweighted scalar $\tilde{\phi}$. For the Eulerian phase $\tilde{\phi}^E$ may be determined directly by dividing the cell-centred transported scalar values by the local density, from which a gradient may be approximated. For the Lagrangian phase, however, the representation of this term is not so obvious; a detailed explanation will be provided in §5.4.2.

Throughout the rest of this chapter it is otherwise assumed that the formulation and implementation of the E-L method may equally be applied for constant density or variable density cases.

5.2.2 Lagrangian Representation

Having determined the magnitudes of the low- and high-frequency parts of the initial scalar field from Eqs. 5.2 and 5.3, it is now necessary to represent the high-frequency field with Lagrangian particles. (The representation of the low-frequency field is trivial, as it is assumed that the initial scalar has already been discretised in the Eulerian framework). At this stage one of the key parameters of the E-L method is introduced, which is the *Lagrangian particle density*, L^{ϱ}. This quantity defines the 'number density' of particles N_p that is used to describe the local distribution of the high-frequency scalar contribution, as:

$$N_p = \left\lceil \mathrm{L}^{\varrho} \left| \phi^L \right| \right\rceil \tag{5.16}$$

Where the $\lceil \ldots \rceil$ operator denotes rounding up to the nearest integer, and the absolute value of ϕ^L is taken to reflect the fact that the local value of the high-frequency scalar field may be positive or negative. The Lagrangian particle density is the single most significant parameter in terms of the accuracy and computational cost of the E-L method. As will be shown in §5.3.2, a higher particle density will generally deliver a more accurate result; however, a greater number of particles will also slow down the calculation, and their storage will demand a larger amount of memory.

From this particle density it is useful to define the scalar contribution of a particle, w_{ϕ}^{p}:

$$w_{\phi}^{p} = \frac{\alpha^p}{\mathrm{L}^{\varrho}} \tag{5.17}$$

Where α^p is the individual 'weight' of a particle p, taking values in the range $0 < |\alpha^p| \leq 1$. Allowing for 'partial' particles ($|\alpha^p| < 1$) means that the high-frequency scalar field may be represented exactly, rather than being 'rounded' to the nearest increment of $1/\mathrm{L}^{\varrho}$. For regions where ϕ^L is negative, α_p will also be negative.

5.2.3 Treatment of Transport Terms

The transport equation for the smoothed contribution ϕ^E may be solved independently from the equation for the contribution of ϕ^L. As the low-frequency field is described using a traditional Eulerian approach, each of the (resolved) transport terms in Eqs. 5.4, 5.6, 5.10 and 5.14 may be treated using any of the spatial convection and diffusion schemes that have previously been described in Chapter 4. Transport of the Lagrangian particles, however, will require a different approach.

Particle Convection Convective particle transport may be described using an equation of motion, of the form:

$$\frac{d}{dt}\left(s_i^p(t)\right) = u_i(s_i^p(t), t) \tag{5.18}$$

Where $s_i^p(t)$ is the position of a particle p at time t, and u_i is the underlying Eulerian velocity. Equation 5.18 therefore provides a connection between the Lagrangian and Eulerian descriptions,

in terms of the Eulerian flow velocity and the Lagrangian particle position. Integrating Eq. 5.18 in time yields:

$$s_i^p(T) = s_i^p(0) + \int_0^T u_i(s_i^p(t), t)\, dt \tag{5.19}$$

Where T is the total physical time. For implementation in a time-dependent simulation, Eq. 5.19 must be discretised in time:

$$s_i^p(T) = s_i^p(0) + \sum_{t=0}^{T} u_i(s_i^p(t), t)\, dt \tag{5.20}$$

Where the initial position $s_i^p(0)$ is prescribed. Alternatively, for an incremental update over dt:

$$s_i^p(t + dt) = s_i^p(t) + u_i(s_i^p(t), t)\, dt \tag{5.21}$$

To some extent Eq. 5.21 is analogous to convective flux in Eulerian space, in that the magnitude of the update to s_i^p is proportional to the flow velocity. Particle convection is therefore subject to the same restriction (from the CFL condition, Eq. 3.9, §3.3) that the product $u_i\, dt$ must not exceed the local cell size Δ. The fact that the same (Eulerian) velocity field is used to calculate both the Eulerian and Lagrangian convection terms ensures that the two fields remain 'in step', which is vital for the success of the E-L method.

Although Eq. 5.21 is shown in Euler-explicit form, its actual implementation applies the third-order accurate low-storage Runge-Kutta time integration scheme described in §3.2. It is essential that the same order of accuracy in time is maintained between the Eulerian and Lagrangian phases, as a significant discrepancy in temporal errors may otherwise accumulate over the course of a simulation. The only practical difference in applying the Runge-Kutta method is that the particle velocity (i.e. the update to the particle position) must be stored from the previous sub-time-step, and that the time-step width dt must be scaled by the Runge-Kutta weights h_j provided in §3.2.

In order to equate a Lagrangian description of convection to the Eulerian one, the derivation of Press and Rybicki [191] is followed: for the Eulerian spatial and temporal co-ordinates x_i and t, respectively, corresponding Lagrangian co-ordinates ξ_i and τ are defined. The relationship between the spatial derivatives in x_i and ξ_i may then be written using the chain rule:

$$\frac{\partial}{\partial \xi_i} = \frac{\partial x_j}{\partial \xi_i} \frac{\partial}{\partial x_j} \tag{5.22}$$

While a corresponding relationship between the temporal derivatives, for a particular particle position ξ_i, reads:

$$\left.\frac{\partial}{\partial \tau}\right|_{\xi_i} = \frac{\partial t}{\partial \tau}\frac{\partial}{\partial t} + \frac{\partial x_j}{\partial \tau}\frac{\partial}{\partial x_j} = \frac{\partial}{\partial t} + u_j \frac{\partial}{\partial x_j} \tag{5.23}$$

Where the RHS of Eq. 5.23 is often referred to as the Lagrangian (or total) derivative. The convective transport of a scalar quantity in the Lagrangian frame of reference may therefore be

equated to the corresponding Eulerian description:

$$\left.\frac{\partial \phi}{\partial \tau}\right|_{\xi_i} = \frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} \tag{5.24}$$

Where the RHS of Eq. 5.24 is equivalent to the LHS terms of the transport equation (Eq. 2.9).

Particle Diffusion The diffusion of Lagrangian particles requires a more complex treatment: the magnitude of the diffusive flux is a function of the local scalar gradient, which is not readily available in the Lagrangian context. The Lagrangian diffusion term could be accounted for by considering the total number of particles in each cell, and subsequently applying a net diffusive transport (drift) of particles from regions of higher particle numbers to regions of lower particle numbers. In practice this would be equivalent to calculating the scalar gradient from an Eulerian description of the Lagrangian field, and applying a drift velocity to the Lagrangian particles to recover the corresponding (Eulerian) diffusive flux.

However, a simpler approach is presented if a stochastic description of particle motion is considered, and how this may be related to a physical diffusion process. For a collection of real particles within a flow the action of Brownian motion will cause those particles to diffuse amongst each other. Einstein [192] formulated a relationship for the rate of change of the number of particles f within a given volume due to this motion, according to the diffusion equation:

$$\frac{\partial f}{\partial t} = D \frac{\partial^2 f}{\partial x^2} \tag{5.25}$$

Equation 5.25 is closely related to the Fokker-Planck equation (introduced some years after Einstein's work by Kolmogorov [193]) where in this case the drift term (mean velocity) is zero. The solution to Eq. 5.25 may be written as:

$$f(x,t) = \frac{N_p}{\sqrt{4\pi D t}} \exp\left(\frac{-x^2}{4Dt}\right) \tag{5.26}$$

Where N_p is now the total number of particles. A normalised, stationary Gaussian probability distribution function (PDF) has the form:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{5.27}$$

Where σ is the standard deviation of the distribution, and μ is the mean. Comparing Eqs. 5.26 and 5.27 it is can be seen that Einstein's PDF of local particle number (normalised by N_p) is equivalent to a (non-stationary) Gaussian distribution, with zero mean ($\mu = 0$) and a standard deviation of $\sigma = \sqrt{2Dt}$.

The next step is to describe the motion of a single particle as a stochastic process; more specifically, as a Lévy process, which is defined as having: a known (typically zero) initial condition; independence between steps of the process, so that the motion at one step is uncorrelated to the motion at any other step; and a probability distribution (for example, of the particle position in

space) that is a function only of the length of each step. In particular, we would like to select a Lévy process that satisfies Eq. 5.26. Such a process – known as the Wiener process – is used to study the Brownian motion of an individual particle, and is defined in one dimension as:

$$s^p(t) = \sigma W \tag{5.28}$$

Where W represents the Wiener step, or (continuous) random walk, which is a Gaussian-distributed random variable. Figure 5.2 shows the random walks of a collection of 20 particles, initialised at a position $x = 0$ at time $t = 0$ (at the top of the plot), transported at a fixed time-step width over a total time T, and with a Wiener step in space at each time-step.

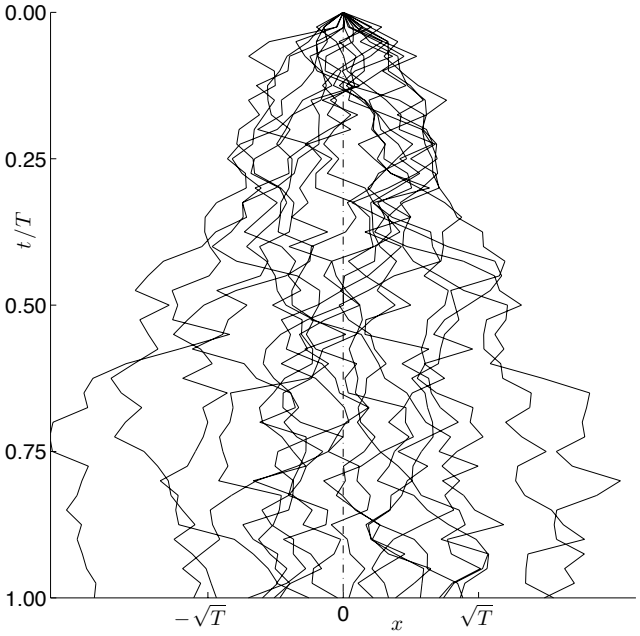


Figure 5.2: Random walks of 20 particles.

The PDF of a generalised, one-dimensional Wiener process may be written as:

$$f_{\mathrm{W}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{5.29}$$

Comparing Eqs. 5.27 and 5.29 the Wiener process can be seen to have a zero first moment, meaning that the average displacement of a particle is zero. To demonstrate this visually the random walks shown in Fig. 5.2 are repeated for a total of one million particles, and PDFs (histograms) of particle position are generated at times $t = \frac{1}{4}[T,\, 2T,\, 3T,\, 4T]$, as shown in Fig. 5.3. The red lines indicate the expected standard deviation of each PDF based on $\sigma^2 = t$ in Eqs. 5.28 and 5.29.

A one-dimensional Wiener process specifically satisfies Einstein's description of Brownian motion if the variance σ^2 is equal to $2Dt$, and the diffusion terms for the Lagrangian and Eulerian phases will remain coupled as long as the diffusivity of each phase is the same.

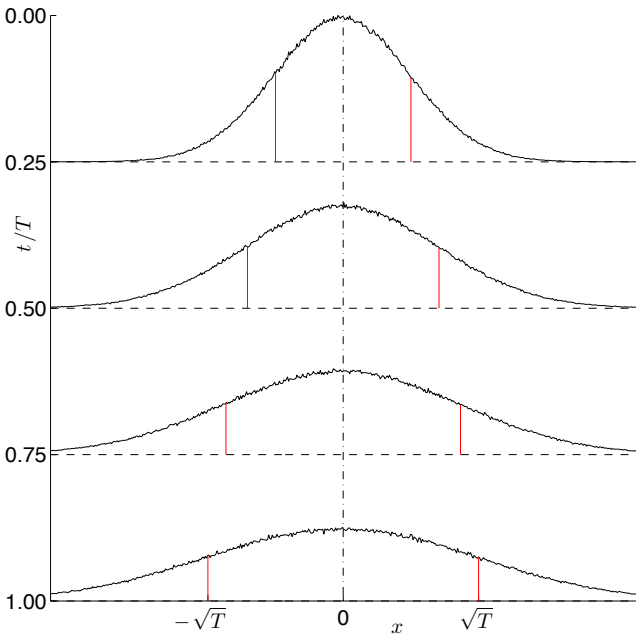


Figure 5.3: PDFs generated from 10^6 random walks.

Discretisation of Eq. 5.28 in time, and substitution of the correct σ to satisfy Eq. 5.26, results in:

$$s^p(t+dt) = s^p(t) + dW\sqrt{2Ddt} \tag{5.30}$$

Where dW is an incremental Wiener step, which is a Gaussian-distributed random variable as before. It should be noted that particle diffusion must be advanced in time using an Euler-explicit scheme (although the dt appearing in Eq. 5.30 will still be multiplied by the Runge-Kutta sub-time-step widths h_j provided in §3.2), as the particle does not follow a smooth trajectory while diffusing; the application of a high-order time integration scheme does not affect the accuracy of such a stochastic process, and no temporal correlation in this process should be introduced.

The Wiener process defined by Eq. 5.30 is applicable in a continuous domain, but does not hold once the domain is discretised in space. In order to maintain coupling of diffusion terms between a discretised Eulerian field on a uniform grid of cell size Δ and a corresponding Lagrangian field, it is necessary to scale the variance of the Wiener step by that cell size:

$$s^p(t+dt) = s^p(t) + dW\sqrt{2D\Delta dt} \tag{5.31}$$

Consideration should also be given to the implementation of the Wiener process described here in three dimensions. To ensure that the correct rate of diffusion is maintained a Wiener process is applied in each Cartesian direction of particle motion, with the same scaling for the length of the Wiener step for each direction:

$$s_i^p(t+dt) = s_i^p(t) + dW_i\sqrt{2D\Delta dt} \tag{5.32}$$

Where dW_i represents a vector composed of three independent, Gaussian-distributed random numbers. Equations 5.21 and 5.32 may be combined into the form of a Langevin stochastic differential equation, to describe the total motion due to convection and diffusion, as:

$$s_i^p(t+dt) = s_i^p(t) + u_i(s_i^p(t), t)\, dt + dW_i\sqrt{2D\Delta dt} \tag{5.33}$$

5.2.4 Lagrangian Reconstruction

Once the Lagrangian particles have been transported using the particle convection-diffusion equation given in Eq. 5.33, it is necessary to 'reconstruct' an Eulerian description of the scalar contribution of those particles so that a summation of the Eulerian and Lagrangian components may be performed. As a particle is considered to be point-like in space, a probability distribution of its scalar contribution would resemble the product of a Dirac peak centred on the particle position with the individual particle contribution w_ϕ^p. In order for this contribution to be represented in the Eulerian framework, it is necessary to convolute w_ϕ^p with some localised filter function F:

$$\phi^L = \sum_{p=1}^{N_p} w_\phi^p * \mathit{F}(s_i^p - s_i') \tag{5.34}$$

Where N_p is the total number of Lagrangian particles, and the filter F is a function of the particle position s_i^p and the local co-ordinates s_i' (with $s_i' = 0$ at the position of s_i^p). In practice it is convenient to set the width of the filter F to be equal to the computational cell size Δ; the exact implementation of Eq. 5.34 will be described fully in §5.4.3.

The frequency of the reconstruction described here will depend on the type of simulation being carried out, and may not be required after every computational time-step. For cases involving passive scalars it may only be necessary to reconstruct the Lagrangian field when its contribution to the overall scalar field is required (for the recording of statistical samples, for example). However, for cases where the transported scalar plays an active role in the simulation, reconstruction may be required at every time-step.

5.3 Implementation – Initialisation

Having considered the theoretical formulation of the E-L method the actual implementation in the `PsiPhi` DNS/LES code introduced in Chapter 3 will be described, beginning with the steps required during initialisation. This comprises two main parts: the decomposition of some prescribed initial scalar field into its high- and low-frequency components, and the distribution and assignment of the Lagrangian particles used to represent the high-frequency field.

5.3.1 Decomposition

The Eulerian-Lagrangian decomposition relies upon the selection of a suitable filter kernel $h(x_i - x_i')$ (from Eq. 5.2). The action of the filter should be to remove (attenuate) the high frequencies from

the scalar field; the chosen filter will therefore be of the 'low-pass' form, in that it preserves low-frequency information. Filter design forms a substantial part of the topic of Digital Signal Processing (DSP), and finds a wide range of applications in many commercial and scientific disciplines; only a limited discussion will be presented here. The interested reader is referred to the textbooks of Parks and Burrus [194] or Loy [195] for a more detailed overview.

A number of choices exist for the filter function that may be applied in this context; for example, the use of a Gaussian (exponential) filter has already been discussed in §3.7.1. Other notable options are the iterative application of a top-hat filter and the theoretically ideal sinc filter. These three options are described and compared below.

Sinc Filter The sinc function in theory constitutes a perfect low-pass filter, in that the high frequencies (above a certain cut-off frequency) are completely removed, while the low frequencies are unaffected; in LES this is alternatively referred to as a spectral cut-off filter, as applied by Vreman et al. [196] and others. In its simplest form the sinc function may be expressed as:

$$h(\delta x) = \frac{\sin(\beta \delta x)}{\beta \delta x} = \mathrm{sinc}(\beta \delta x) \tag{5.35}$$

Where δx is used as short-hand notation for the local separation $(x - x')$, and β is a parameter controlling the width (cut-off) of the function. The impulse response of a (discretised) sinc filter with $\beta = 1$ is shown in Fig. 5.4(a), where $h(\delta x)$ has been normalised such that $h(0) = 1$. To understand the behaviour of this filter in the frequency domain a times series of normally-distributed random numbers, sampled at $f_s = 1/\Delta_t = 1$ kHz for a total time of 100 seconds, are filtered with the function $h(\delta t) = \mathrm{sinc}(\delta t)$. The Discrete Fourier Transforms (DFT)[1] of the initial and filtered data sets are then compared, as shown in Fig. 5.4(b). Also shown in Fig. 5.4(b) is the analytical filter response, which may be determined from a DFT of the filter kernel itself.

The magnitude of the frequency response H for this filter is shown to drop abruptly above a cut-off frequency of $f_s/2\pi$ and rapidly decays towards zero at increasing frequencies, while any information below this cut-off is unaffected. This suggests that the sinc filter may be an attractive choice for the present work, assuming that some relationship exists between the frequencies in a scalar field and its second derivative (which has been shown to play a key role in the development of oscillations in a transported Eulerian field in §4.3.2 and §4.4.2).

However, before any further consideration is given to the implementation of the sinc filter, two practicalities must be addressed. Firstly, the filter kernel shown in Fig. 5.4(a) extends far beyond the range shown (it has $n_f = 1000$ support points in each direction), which is essential for its exact cut-off behaviour in the frequency domain. Unfortunately, requiring such an extensive kernel has practical implications for the use of the filter: a very large 'buffer' region[2] (of length n_f) is required at each end of the data set to allow the filter to be applied across the entire series. The computational cost associated with the large number of multiplications and additions required in

[1] Performed with the 'Fastest Fourier Transform in the West' (FFTW) library [197], as implemented in the `MatLab` programming environment (2012a release)

[2] Sometimes referred to as a 'ghost' or 'halo' region

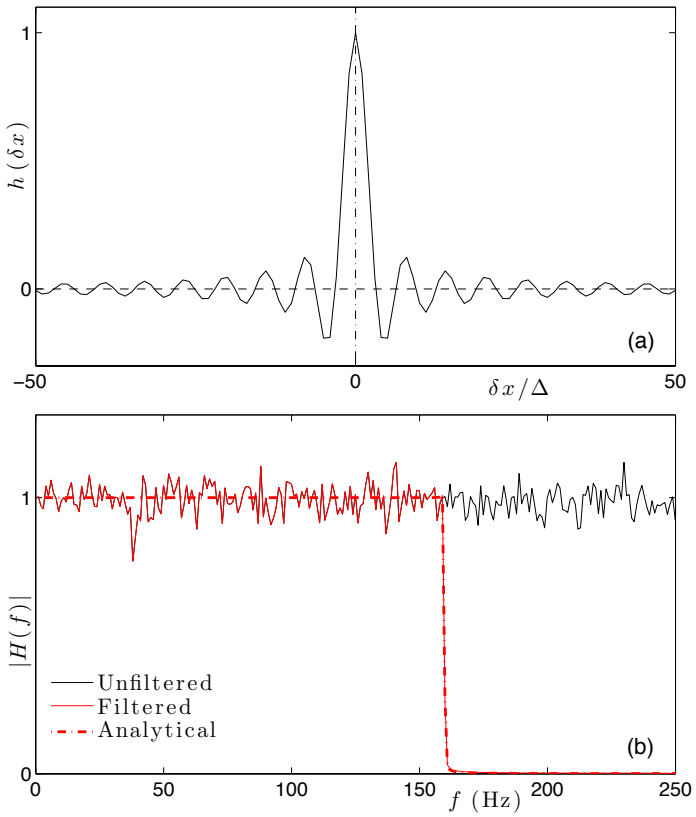


Figure 5.4: Sinc function filter kernel (a); typical filter response (b).

the application of such a kernel must be considered, as well as the corresponding communication overhead in a parallel computing environment.

A solution to this problem is to truncate the kernel to a more manageable size, as shown in Fig. 5.5(a), where n_f has been reduced to 25 points in each direction. However, this truncation has a distinct disadvantage: as can be seen in Fig. 5.5(b) the frequency response of this filter is no longer ideal, as some undesirable artefacts have been introduced. Most notably the filter now amplifies frequencies just below the cut-off, which may actually promote the development of oscillations in the Eulerian field.

The artefacts introduced by the truncation of the filter kernel may be reduced by applying a 'window' function to the sinc function filter kernel. Figure 5.6(a) shows a truncated sinc filter kernel that has been multiplied by an exponential kernel of the same extent, i.e.:

$$h(\delta x) = \frac{\sin(\beta \delta x)}{\beta \delta x} \exp\left(-\frac{\delta x^2}{2\,\sigma_f^2}\right) \tag{5.36}$$

Where σ_f is the standard deviation of the exponential kernel, in this case taken to be $0.4\,n_f$. Applying a window function to the sinc filter yields the impulse response shown in Fig. 5.6(a), where the abrupt truncation of the filter weights apparent in Fig. 5.5(a) has been replaced by a gradual decay to zero. The corresponding frequency response shown in Fig. 5.6(b) displays similar characteristics to that of the un-windowed filter, but without the amplification around the cut-off

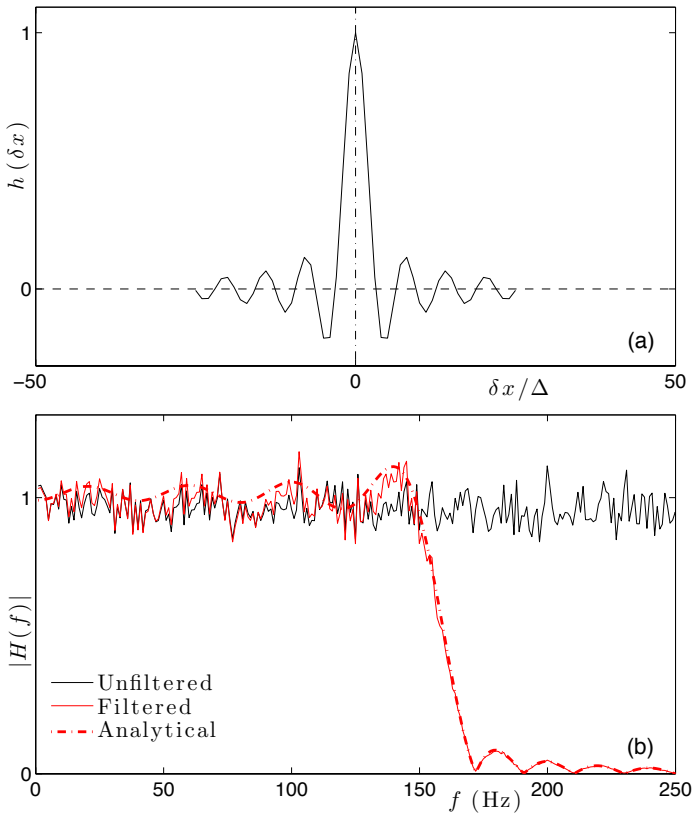


Figure 5.5: Sinc function filter kernel with reduced kernel size (a); typical filter response (b).

frequency and other associated artefacts.

The application of a windowed sinc filter in three dimensions raises a further practical concern. In §3.7.1 it has been demonstrated that a Gaussian filter kernel may be expressed as the combination of three one-dimensional filters, so that a scalar field may be filtered independently in each direction at a considerable reduction in computational cost. However, such a transformation is not possible for the sinc filter. The process required to filter a two-dimensional field with a kernel equivalent to that shown in Fig. 5.6(a) is considered; such a kernel may be visualised as the surface formed by a rotation of the function in Fig. 5.6(a) about the centreline (at $\delta x = 0$). The filtered value at a point $(i,\, j)$ is then a weighted sum of every point within the radius of the kernel (i.e. within a distance of 25 cells, or nearly 2000 points in total), where, for each point, it is necessary to calculate the filter weight as a function of the distance from that point to $(i,\, j)$. It is estimated that the computational cost of this method would scale as $(2\, n_f + 1)^d$, where d is the dimensionality of the filtered field, which is typically far greater than the cost of a Gaussian filtering procedure.

Top-hat and Gaussian Filters A top-hat filter is the simplest kernel that may be constructed for the filtering of a set of data. The filtered value at a point in a continuous function is calculated as the (linear) average of the local values of that function, such that the impulse response has a rectangular appearance. A discretised, normalised top-hat filter kernel of width $w_{\mathrm{TH}} = 3\Delta$ is shown in Fig. 5.7(a), where the filter weights are equal over the three points $\delta x = [-\Delta,\, 0,\, \Delta]$ and are zero

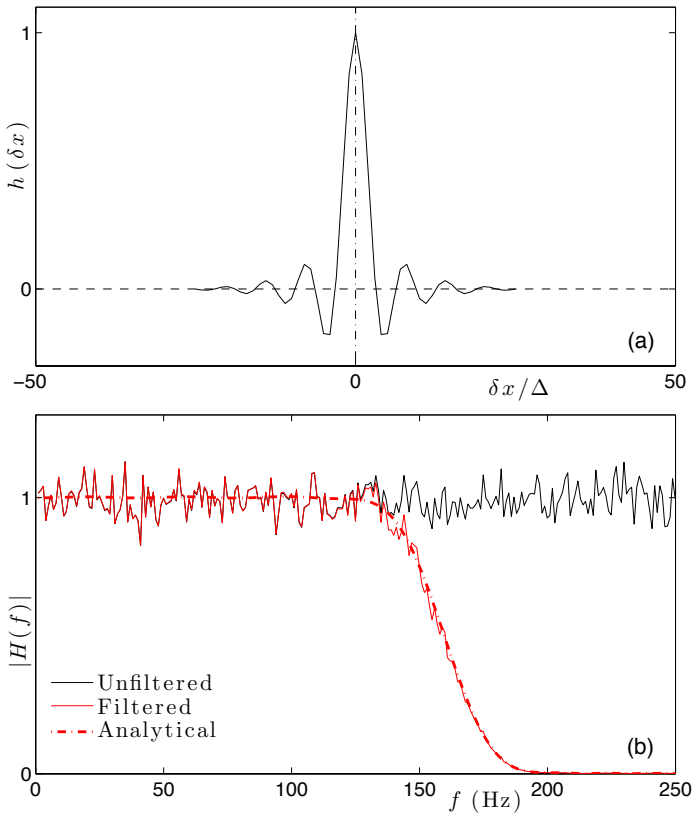


Figure 5.6: Sinc filter windowed with an exponential function (a); typical filter response (b).

elsewhere. However, filter widths that are not odd multiples of Δ are not rectangular once they are discretised; therefore the kernel weights are not equal, but may be determined as the proportion of the kernel that lies within $\pm\frac{1}{2}\Delta$ of each support point. If the filter weights across the three support points $\delta x = [-\Delta,\, 0,\, \Delta]$ are denoted $h = [b_1,\, a_1,\, b_1]$, it may be shown that:

$$a_1 = \frac{\Delta}{w_{\mathrm{TH}}} \tag{5.37}$$

$$b_1 = \frac{1}{2}\left(1 - a_1\right) \tag{5.38}$$

A top-hat filter with $w_{\mathrm{TH}} = 2\Delta$ therefore has a discrete kernel of $h = \frac{1}{4}\,[1,\, 2,\, 1]$ over the points $\delta x = [-\Delta,\, 0,\, \Delta]$, and is typically referred to as a 1–2–1 filter.

Top-hat filter kernels in higher dimensions may similarly be constructed as the proportion of the area or volume of the kernel within the extent of the computational cell at each support point; for example, in two dimensions:

$$h = \begin{bmatrix} c_2 & b_2 & c_2 \\ b_2 & a_2 & b_2 \\ c_2 & b_2 & c_2 \end{bmatrix} \quad \text{with} \quad \begin{aligned} a_2 &= a_1 a_1 \\ b_2 &= a_1 b_1 \\ c_2 &= b_1 b_1 \end{aligned} \tag{5.39}$$

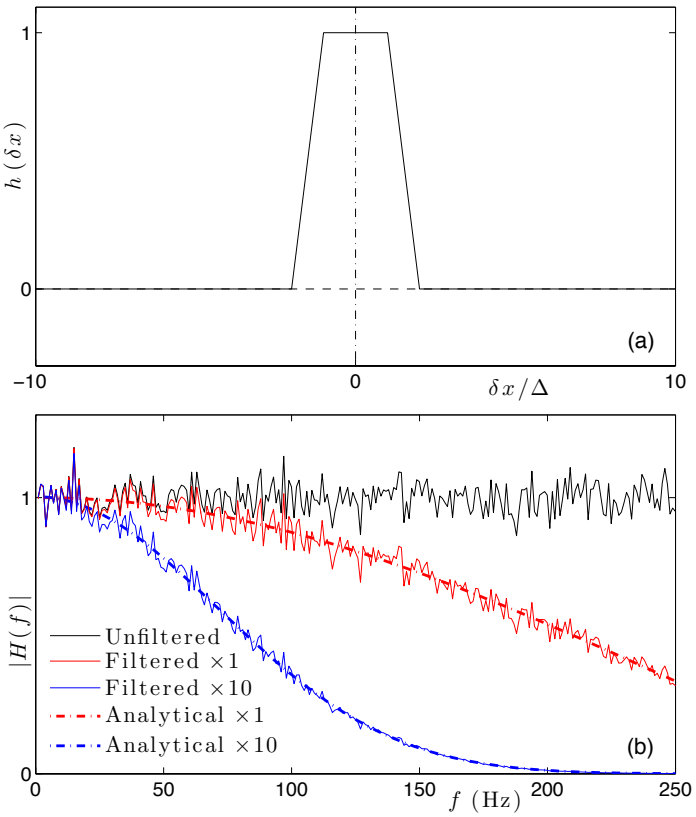


Figure 5.7: Top-hat filter kernel of width 3Δ (a); typical filter response (b).

Because kernel weights for a two-dimensional top-hat filter may be calculated as the products of the one-dimensional weights, a one-dimensional top-hat filter may be applied independently for each direction in higher dimensions (similar to a Gaussian filter, as discussed in §3.7.1).

The frequency response of a top-hat filter of width $w_{\mathrm{TH}} = 3\Delta$ (Fig 5.7(b), red line), suggests that the use of such a narrow kernel does not yield a substantial attenuation of the highest frequencies. A top-hat filter may therefore by applied to a data set iteratively, such that the filtering has a cumulative effect. Applying the present filter 10 times results in the frequency response shown by the blue line in Fig. 5.7(b), where the filter can be seen to have removed the high frequencies much more effectively.

The computational cost of an iterative top-hat filter is approximately $dN_f\,(2\,n_f + 1)$, where N_f is the number of filter operations and d is the dimensionality of the filtered field. The filter has few support points ($n_f = 1$ for $w_{\mathrm{TH}} \leq 3\Delta$) so that the expense of its application in two or three dimensions is small compared to that of the sinc function described previously, and is comparable to a Gaussian filter for small N_f (i.e. less than around 10 operations).

The relationship between an iteratively-applied top-hat filter and an exponential (Gaussian) filter is now examined. A normalised Gaussian filter may be expressed in the form:

$$h(\delta x) = \exp\left(-\frac{\delta x^2}{2\,\sigma_f^2}\right) \tag{5.40}$$

Where the kernel extent n_f (i.e. the number of support points) should be at least four times the chosen standard deviation σ_f. This ensures that that 'tails' of the distribution are captured sufficiently to prevent any truncation artefacts, while restricting the kernel size to reduce computational (and, for simulations performed on parallel architectures, communication) overheads. Figure 5.8(a) shows a Gaussian filter kernel of width $\sigma_f = 2\Delta$, while the corresponding frequency response is provided in Fig. 5.8(b).

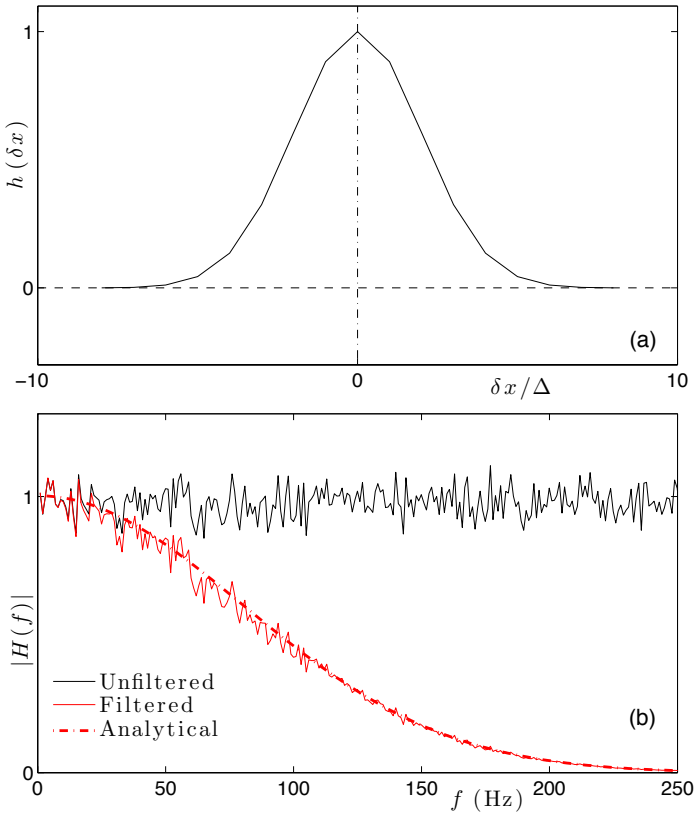


Figure 5.8: Gaussian filter kernel (a); typical filter response (b).

The frequency response of a top-hat filter applied $N_f = 10$ times, shown by the blue line of Fig. 5.7(b), is similar to that of the Gaussian filter shown in Fig. 5.8(b). In fact, the impulse and frequency response of a top-hat filter of width $w_{\mathrm{TH}} = 3\Delta$ applied N_f times can be exactly matched to that of a Gaussian filter of standard deviation σ_f using the relationship:

$$N_f = \frac{3}{2}\sigma_f^2 \tag{5.41}$$

More generally, for a top-hat filter where $\Delta < w_{\mathrm{TH}} \leq 3\Delta$:

$$N_f = \frac{w_{\mathrm{TH}}}{(w_{\mathrm{TH}} - \Delta)}\sigma_f^2 \tag{5.42}$$

Considering that a d-dimensional top-hat filter may be expressed as the product of d one-dimensional filters, it is apparent that the relationship expressed in Eq. 5.42 holds true in any

number of dimensions. Note that in practice Eq. 5.42 must be rounded to the nearest integer number of filter operations.

The choice between applying a top-hat filter several times, or a Gaussian filter once, becomes a question only of the ease-of-implementation or suitability of that filter in a given context. For example: to filter some regions of a data set more than others, it may be more efficient to vary N_f for a narrow top-hat filter than it would be to apply a different Gaussian filter kernel for each point in the data set (varying σ_f). Conversely, considering that the number of top-hat filter operations is proportional to σ_f^2 (from Eq. 5.42), for uniform filtering it is typically more efficient to filter with a Gaussian kernel once than to filter with a top-hat filter kernel N_f times (particularly for larger values of σ_f).

In practice the Gaussian filter weights must be normalised so that the kernel has a unit integral (to ensure that the sum of a field remains equal after filtering). Equation 5.40 is therefore modified by a suitable normalising factor:

$$h(\delta x) = \frac{1}{\sigma_f \sqrt{2\pi}} \exp\left(-\frac{\delta x^2}{2\,\sigma_f^2}\right) \tag{5.43}$$

Thus far it has been assumed that the removal of high-frequency information from a series of data results in a corresponding reduction in its second derivative, which is approximately equal to its curvature. In order to demonstrate this relationship a series of noisy data y is filtered over a range of Gaussian filter kernel widths (standard deviations), with $\sigma = [0.5,\, 1.0, \ldots, 5.0]\Delta$, and the impulse and frequency responses for each kernel are determined as before. In addition the maximum absolute value of the second derivative of the filtered data series y_f (with respect to the separation Δ of the data points) is also determined for each filter kernel width, as an indication of the smoothness of that series. The second derivative of y_f at a point i is approximated using a second-order accurate estimate:

$$\nabla^2 y_f(i) = y_f(i-1) - 2y_f(i) + y_f(i+1) \tag{5.44}$$

The results from this analysis are shown in Figs. 5.9(a–c), which show the filter impulse response, frequency response and maximum second derivative, respectively. The coloration of each line or marker corresponds to the width of the filter, from green ($\sigma = 0.5\Delta$) to magenta ($\sigma = 5.0\Delta$), while values for the initial field are shown in black in Figs. 5.9(b–c). The analysis demonstrates that as the width of the applied Gaussian filter (or, equivalently, the number of iterations for a top-hat filter) is increased, the smoothness of the resulting filtered field is also increased. In addition the analysis shows that the increase in smoothness of the field is asymptotic with increasing filter width, suggesting that, for a given data set, there is an optimal filter width beyond which additional filtering has little effect on smoothness.

The selection of a filter width for the Eulerian-Lagrangian decomposition will also depend on the duration of a simulation, the expected nature of the flow-field in that simulation, and the choice of re-initialisation scheme (which will be discussed in §5.4.4); as a guide, however, a Gaussian filter width of $\sigma_f = 5\Delta$ is suggested (corresponding to $N_f = 38$ for a top-hat filter of $w_{\mathrm{TH}} = 3\Delta$).

The chosen filter width should not depend on the resolution of the computational grid: numerical oscillations resulting from convective transport by a Central Differencing Scheme depend on the smoothness of the field with respect to the cell co-ordinates (as shown in §4.3.2 and §4.4.2), and are independent of the physical co-ordinates. In addition, as convective fluxes in the Eulerian framework are calculated independently for each component of velocity, the dimensionality of a simulation should not affect the filter width that is applied.

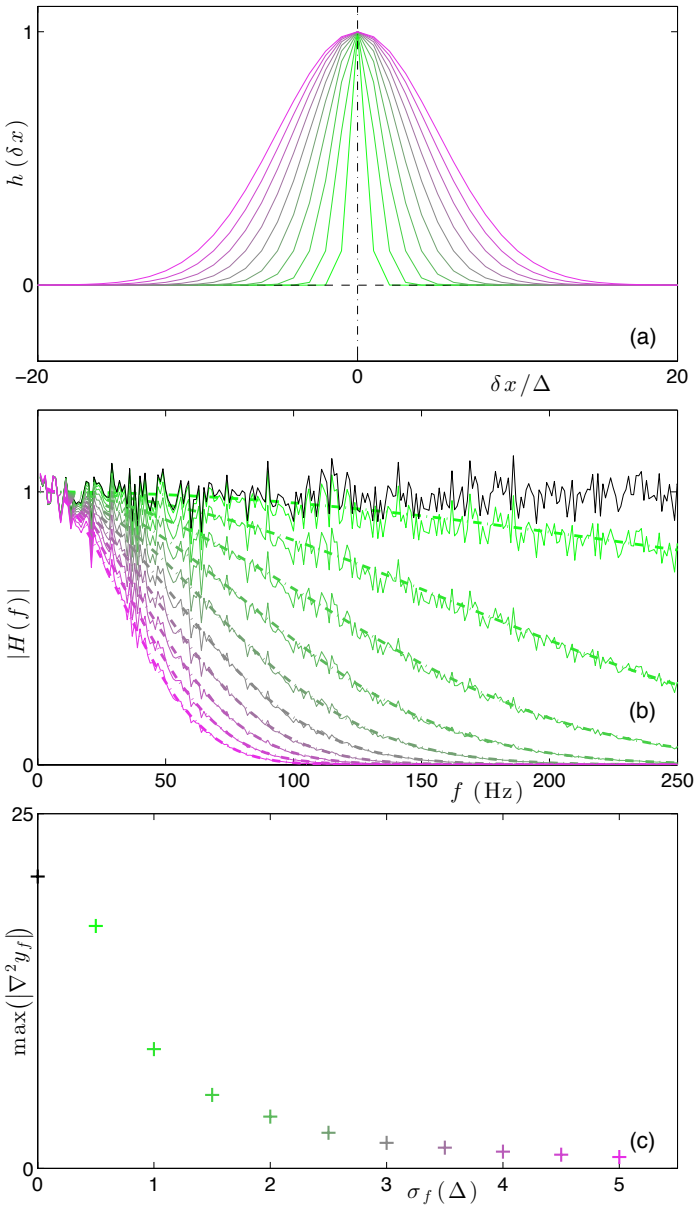


Figure 5.9: For various Gaussian filter widths: (a) kernel, (b) response, (c) smoothness.

5.3.2 Particle Distribution

Once the low- and high-frequency fields have been determined from Eqs. 5.2 and 5.3, the Lagrangian particles used to describe the high-frequency field are distributed throughout the simulation domain.

To provide guidance for an appropriate value of the Lagrangian particle density L^{ϱ} in a given case, the relationship between the error in a reconstructed particle field and L^{ϱ} is considered. To quantify this relationship tests are performed in one, two and three dimensions, on domains of size $N_{1\mathrm{D}} = 729$ cells, $N_{2\mathrm{D}} = 27 \times 27$ cells and $N_{3\mathrm{D}} = 9 \times 9 \times 9$ cells, respectively, such that the total number of particles is constant for each test. Each domain is initialised with a unit scalar field throughout, and Lagrangian particles are used to represent each field with particle densities in the range $1 \leq \mathrm{L}^{\varrho} \leq 20000$.

Figure 5.10 shows the two measures of standard (blue) and maximum (black) deviation of the total error E, between the analytical and reconstructed scalar fields. Also shown for comparison is the inverse root of the Lagrangian particle density (red). The different line styles represent errors in one (solid), two (dashed) and three (dot-dashed) dimensions. From this analysis it is apparent that the dimensionality of a problem does not influence the accuracy of the Lagrangian representation of a scalar field, as each error measure is approximately collinear in one, two or three dimensions. The maximum error is observed to be approximately equal to the inverse root of L^{ϱ}, while the standard deviation of the error in each cell is directly proportional to it.

A reduction in error in the Lagrangian field may therefore be achieved by increasing the particle density. However, such an increase comes with an associated increase in computational cost. Figure 5.11 shows the relationship between the execution time t and the particle density, where the execution time is found to be proportional to L^{ϱ} (with a constant $\gamma \ll 1$ in Fig. 5.11). The test is performed in one, two and three dimensions – shown by the solid, dashed and dot-dashed lines, respectively – where the tested domain sizes are increased to $N_{1\mathrm{D}} = 4096$ cells, $N_{2\mathrm{D}} = 64 \times 64$ cells and $N_{3\mathrm{D}} = 16 \times 16 \times 16$ cells, to ensure that execution times are of a meaningful duration for the lower values of L^{ϱ}. It can be seen that the computational cost is also a function of the dimensionality of the problem, as the Lagrangian reconstruction (which will be described in detail in §5.4.3) requires more operations per particle for higher dimensions.

It should be stressed that this analysis is not intended to be indicative of the computational cost of the E-L method as a whole, but is performed to highlight the compromise that should be made between accuracy and cost when a value for the Lagrangian particle density is selected.

For a scalar with permissible values in the range $\phi = [0-1]$, it is suggested that a particle density of the order 10^2 to 10^3 is appropriate for a two-dimensional case. In three dimensions a reduction in L^{ϱ} will usually be required, due to the increase in computational cells in the region of a change in scalar gradient. For example, considering a shear layer in a two-dimensional domain measuring 200×50 cells, the shear layer will initially occupy a total of 200 cells; extending the case into the third dimension, to a depth equal to the height, means that the shear layer will now occupy 10000 cells. To maintain a practicable computational cost (neglecting any restrictions placed on the total particle number by the amount of Random Access Memory (RAM) available to a processor) the particle density may need to be reduced by an order of magnitude from the two-dimensional case,

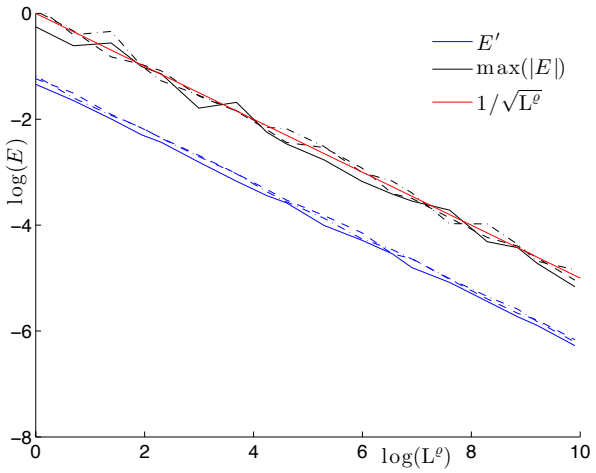


Figure 5.10: Reducing standard deviation and maximum error with increasing L^{ϱ}.

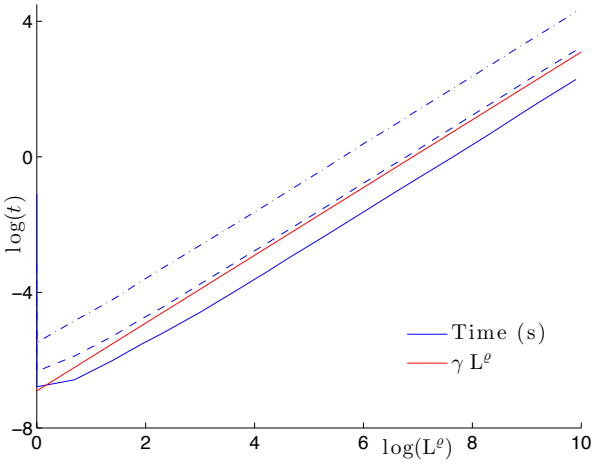


Figure 5.11: Increasing computational cost with increasing L^{ϱ}.

for which a corresponding inverse-root increase in E should be expected.

Having chosen a value for the particle density L^{ϱ}, it is possible to determine the number of particles N_p in a cell by applying Eq. 5.16. The next consideration is how those particles should be distributed within that cell. The simplest approach would be to place all of the particles at the cell centre; however, such an approach results in an undesirable artefact when a reconstruction of the transported Lagrangian field is performed. Figure 5.12 shows the effect for a rectangular field of particles that has been deformed by a vortical flow: as the Cartesian arrangement of those particles becomes distorted, an interference (moiré) pattern is created. The lighter lines radiating from the edges of the field represent higher reconstructed Lagrangian scalar values, although these higher values only exist in the reconstruction, and not in the underlying particle field itself.

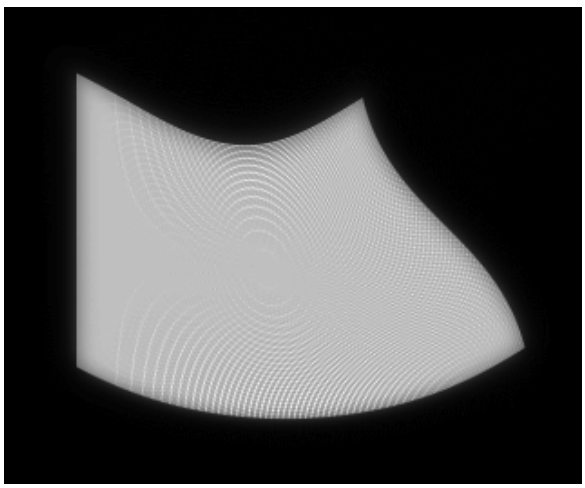


Figure 5.12: Moiré pattern resulting from cell-centred particle distributions.

These artefacts in the reconstruction may be avoided by positioning the Lagrangian particles with a uniform random distribution throughout each cell, so that there is no possibility of interference

between the grid arrangement and the particle arrangement. However, this method will only recover the mean value of the high-frequency scalar field in each cell, while there is potential to improve the description of the high-frequency field by also considering the scalar gradients. Figure 5.1 (§5.2) provides an example of where this may be particularly significant: at the step in the initial scalar field the high-frequency contribution passes from positive to negative. This means that at the position of the step there is possibly a cell where the cell-centred value of the high-frequency field is zero, which (from Eq. 5.16) will lead to no particles within that cell. It is apparent that there will in fact be a steep scalar gradient across that cell, so that there should be positive Lagrangian particles in the left-hand half of the cell and negative particles on the right.

In one dimension it would be possible to recover this scalar gradient by, firstly, approximating the values of ϕ^L at each face of the cell i:

$$\phi^L_w = \frac{1}{2}\left(\phi^L_{i-1} + \phi^L_i\right) \tag{5.45}$$

$$\phi^L_e = \frac{1}{2}\left(\phi^L_i + \phi^L_{i+1}\right) \tag{5.46}$$

Where the subscripts w and e denotes the 'west' and 'east' cell faces, respectively. A linear probability density function for the particle distribution, matching the variation of ϕ^L across the cell, is then assumed:

$$\phi^L(x) = mx + c \tag{5.47}$$

Where the gradient m and intercept c are determined from the cell-face values ϕ^L_w and ϕ^L_e. A change in sign of ϕ^L within the cell is detected where the product of ϕ^L_w and ϕ^L_e is negative. With no change in sign the number of particles may be determined from Eq. 5.16; if there is a change of sign the number of 'positive' and 'negative' particles must be calculated separately, by integration of Eq. 5.47 over the region bounded by the cell face and the x-intercept (found from setting Eq. 5.47 to zero, i.e. at $x = -c/m$).

Particles may then be distributed throughout the cell using a Monte Carlo method to integrate the probability distribution in Eq. 5.47. Each of the (positive or negative) particles is given a random position and a random (positive or negative) value within the cell, while the scalar value corresponding to the position of the particle is determined from Eq. 5.47. The particle is initialised at that point on the condition that the (absolute) particle value is less than the (absolute) scalar value at that position; otherwise a new random particle position and value is chosen, until the condition is satisfied. This process is repeated until the required number of particles has been placed within the cell. Such a distribution provides a coarse approximation to the variation of ϕ^L throughout the cell.

Although this method is straightforward to implement in one dimension, it does not extend well into two- or three-dimensional space. The variation of ϕ^L linearly approximated by Eq. 5.47 is well-defined, as the expression has two unknowns (m and c) which may be determined by the evaluation of Eq. 5.47 at two points (the west and east cell faces). In two dimensions, however, a linear approximation would be over-defined, as there would be three unknowns (two gradients and an intercept), but four points (corners) at which to evaluate it. As a higher-order approximation in x

and y would be under-defined (for example, the quadratic form $\phi_i^L(x,\, y) = ax^2+bxy+cy^2+dx+ey+f$ has six unknowns) the only option for the two-dimensional case would be to fit two linear planes intersecting along a diagonal of the cell. In three dimensions, where there are eight known corner values, the problem becomes even worse; a linear approximation has four unknowns, while a quadric surface has ten.

The significant additional complexity of this gradient method in two or three dimensions advocates the application of the simpler uniform particle distribution, for which an implementation in three dimensions is no more complex than an implementation in one dimension.

5.3.3 Particle Assignment

For a three-dimensional simulation each Lagrangian particle will have a total of eight properties, consisting of: three components of position; three components of velocity, required for the third-order accurate Runge-Kutta update in time; a scalar 'weight' α_p (introduced in §5.2.2); and an 'activity', which will be discussed here.

Particle positions are stored in physical co-ordinates, rather than cell co-ordinates. Conceptually this provides a clearer distinction between the Eulerian and Lagrangian frameworks, but is also easier to implement for simulations performed on multiple processes (where particles will be communicated from one process sub-domain to another). However, this requires a transformation from (global) physical space to (local) cell space, through the cell size Δ, for operations where Eulerian quantities stored on the cell-centred Cartesian grid are needed for calculations in the Lagrangian frame (such as particle convection, requiring a velocity; or diffusion, requiring a viscosity).

The weight α_p of a particle represents its contribution to the scalar field, where L^ϱ particles in a cell, all with a weight of $\alpha_p = 1$, equate in summation to a unit scalar value in that cell.

Particle activity indicates the status of a particle, where each particle may be active, inactive, a 'ghost' particle, or an 'inflow' particle. Every Lagrangian particle is stored with its associated properties in a single array on each process, where the size of the array is chosen to be large enough to hold slightly more than the maximum expected number of particles on that process. The activity property may then be used to recycle particles that have left the simulation domain, as inactive particles are made available to be re-assigned with new properties during re-initialisation events (as described in §5.4.4), or at the domain inflow (if there is a need to add particles here).

The role of the particle activity property is illustrated by considering a two-dimensional domain of 8×4 cells divided across two processes A and B, as shown in Fig. 5.13. The top part of Fig. 5.13 shows the entire domain, where an interface in a scalar ϕ has been filtered to yield a smooth part ϕ^E, and a number of Lagrangian particles have been included to represent the high-frequency part ϕ^L. Particles are indicated by filled markers, where a red particle has a positive scalar contribution and a blue particle has a negative one. The inflow plane is located at the left-hand side of process A. The lower part of Fig. 5.13 shows the sub-domain of process A, where the 'real' part (in white) is bounded by two regions of computational cells that are used for the setting of boundary conditions on that process. To the left (in the darker blue) is the inflow plane, where conditions are prescribed, while to the right are the 'halo' cells (light blue) that are copied from process B.

Ghost particles (black markers in Fig. 5.13) are required in simulations performed on multiple

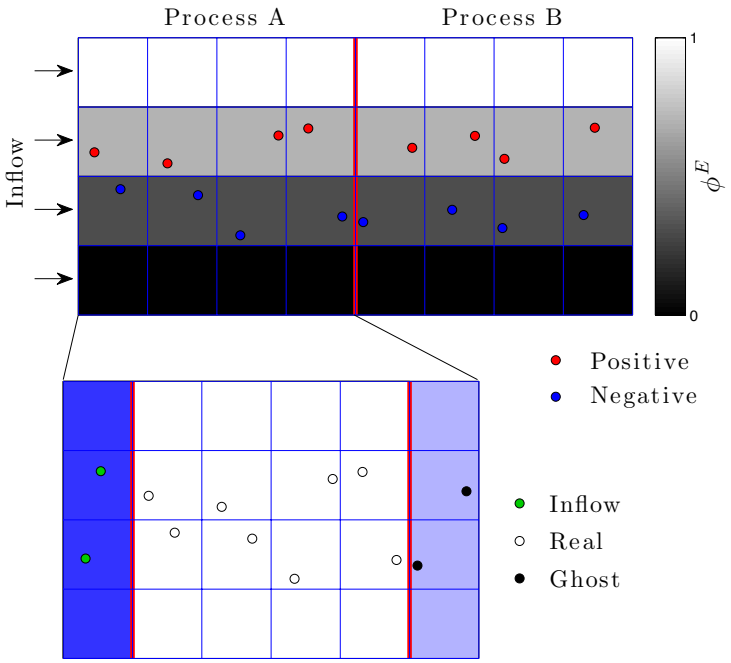


Figure 5.13: Generation of ghost particles on neighbour processes, and inflow particles.

processes for the correct reconstruction of the Lagrangian field around process boundaries. To perform a reconstruction on process A (using the method that will be described in §5.4.3), some of the high-frequency scalar contribution in the leftmost real cells will come from particles in the rightmost cells of process B (and vice versa). The influence of the particles on B must therefore be copied into the halo cells of A. However, these ghost particles must be treated with care, to ensure that they do not become full duplicates of the 'real' particles; they are therefore treated passively by process A, which is to say that their physical properties (position and velocity) are not modified by A. Ghost particles are deactivated as soon as their influence has been taken into account in the Lagrangian reconstruction, although they may be recreated during the next time-step once their real counterparts have been moved on process B.

An inflow particle (green marker in Fig. 5.13) also lies outside of the real domain of process A, but in this case it is treated actively (moved or modified) by A. Once an inflow particle has been transported from the inflow plane into the real domain, its activity is changed from being an inflow particle to an active particle.

In an effort to reduce the computational cost of the particle initialisation (and subsequent) routines, the index of the last active (or ghost, or inflow) particle is stored, and any computations that involve an iterative loop over the particle indices are only performed up to this last active index (rather than to the end of the particle storage array, of which only a fraction may be used at times). This index is updated periodically in case the last active particle has been deactivated, or is extended as necessary where new particles are added (although new particles will be added at any inactive indices, below the last active index, first).

5.4 Implementation – Time-Stepping

Once the Eulerian phase and the Lagrangian particles have been initialised, the simulation may progress into the main time-stepping loop.

5.4.1 Eulerian Convection and Diffusion

Convective transport of the Eulerian phase is performed with an eighth-order accurate Central Differencing Scheme (CDS8; see §4.4.2), while the Eulerian diffusive flux is estimated using an eighth-order central approximation for the scalar gradient (§4.7). Eulerian transport terms are therefore treated with eighth order-of-accuracy overall.

CDS8 is chosen for the convection term as it demonstrates a number of advantages over alternative schemes, provided that the field to be transported is smooth. This smoothness results from the filtering of the initial scalar field, as described in §5.3.1, and is maintained by the use of a re-initialisation routine (to be described in §5.4.4). A high-order CDS is capable of transporting a scalar gradient accurately, where that gradient experiences a minimal amount of numerical diffusion. CDS will also avoid any undesirable artefacts such as the steepening of gradients observed with the WENO and SuperBee TVD schemes (as demonstrated in §4.4.4 and §4.5, respectively). In terms of its implementation CDS8 is straightforward to apply, as the computational stencil is independent of the direction of flow. Although this means that CDS lacks transportiveness (as discussed in §4.2), the smoothness of the transported field in this case ensures that this is not an issue.

The use of CDS offsets some of the additional computational cost incurred by the storage and transport of a large number of Lagrangian particles. Firstly it removes the requirement of a conditional (‘`IF...`’) statement to distinguish between positive and negative velocities at a cell face, due to its lack of transportiveness. Although the cost of a single conditional statement is small, the total cost of applying a conditional statement for the flux calculation across each of the faces of every cell in a domain, at every time-step in a simulation, becomes significant.

Secondly the symmetry of the computational stencil applied for CDS means that flux calculations may be expressed in array notation (rather than in a structure of nested iterative (‘`DO...`’) loops). The use of array notation allows a `FORTRAN` compiler to apply some in-built optimisations, which may speed up the execution of the code significantly. Instruction pipelining allows multiple instructions, such as instruction fetching, decoding, execution, and memory reads and writes, to be performed in parallel for a sequence of identical arithmetic operations. The compiler may also direct the processor to perform prefetching, where a prefetch is made on the assumption that an access at a given memory address will be immediately followed by an access at the next memory address (i.e. memory accesses are contiguous). The value in the next address is loaded into the cache before operations on the value from the present address have been completed, so that the processor does not have to wait for the value in the next memory address to be loaded. The method may also be applied to the instructions themselves, for example where the program repeats a short set of instructions many times.

5.4.2 Lagrangian Convection and Diffusion

Particles are convected according to Eq. 5.21, where a cell-centred velocity vector u_i is determined from the Eulerian phase by dividing the components of momentum in each cell by the local density. The flow velocity at the current particle position s_i^p is then determined by interpolation from the values of u_i nearest to that position. The method applied here is known as trilinear interpolation, which comprises a weighted average of values from the eight cells (in three dimensions) in the locality of the particle for which a velocity is being determined. The weights are based on the distance of the particle to each of the eight surrounding cell centres. Higher-order interpolation schemes such as the tricubic, Catmull-Rom [198] and Hermite methods have been tested in the context of the present work, but are found to result in significant increases in computational cost (as these interpolations use an increased stencil requiring the nearest 64 cell-centred values) while providing inconsequential improvements in accuracy. Particle positions are updated in time using the third-order accurate low-storage Runge-Kutta scheme described in §3.2.

A trilinear interpolation must also be carried out to determine the viscosity of the flow at the current particle position, so that the diffusivity D at that point may be determined. This diffusivity, combined with the cell size and current time-step width, is used to scale the length of the incremental Wiener step (according to Eq. 5.31) applied to replicate diffusion in the Lagrangian phase (as described in §5.2.3).

The incremental Wiener step itself must be a Gaussian-distributed random number, with zero mean and unit variance. In the present work a straightforward method known as the Box-Muller transformation [199] is used to generate Gaussian-distributed random numbers from an available uniform distribution. Given two uniformly-distributed random numbers u_1 and u_2, in the range $[0-1]$, it is possible to create a pair of independent random numbers g_1 and g_2 satisfying a Gaussian distribution:

$$g_1 = R\cos\theta \tag{5.48}$$

$$g_2 = R\sin\theta \tag{5.49}$$

Where the radius R and angle θ are defined as:

$$R = \sqrt{-2\ln u_1} \tag{5.50}$$

$$\theta = 2\pi u_2 \tag{5.51}$$

Once scaled appropriately, d transformed random numbers may be used as the components of a d-dimensional particle Wiener step.

A further consideration regarding particle diffusion is the viscosity that should be used to determine the diffusivity of the scalar quantity at the particle position. In a DNS the only viscosity experienced by the scalar is the laminar viscosity due to molecular interactions; in an LES, however, there is the additional turbulent (modelled) viscosity designed to account for the effect of the unresolved turbulent scales. In an LES the scalar diffusivity should be calculated from the sum of the laminar and turbulent viscosities (divided by their respective laminar and turbulent Schmidt numbers), so that the diffusive fluxes experienced by the Eulerian and Lagrangian phases remain

coupled. This summation is apparent from the bracketed terms on the RHS of Eqs. 5.10, 5.11, 5.14 and 5.15.

Three assumptions are made for the treatment of particle convection and diffusion. The first is that the velocity or viscosity of the flow varies smoothly in space, so that any change in these quantities over the distance that the particle moves within one time-step is negligible. The purpose of the Runge-Kutta time integration scheme, for the particle convection term at least, is to minimise any error resulting from this assumption. Secondly the variation of velocity or viscosity in space is assumed to be gradual enough that the convection and diffusion terms may be considered independently, even though the particle will be transported by convection and diffusion processes concurrently. Finally it is assumed that the turbulent viscosity at the particle position may be interpolated from the surrounding cell-centred values, rather than re-computing a turbulent viscosity at the particle position from interpolated velocities. This is particularly significant where a dynamic turbulence model (such as the Germano model presented in §3.5.2) is applied, as the additional computational cost of directly calculating the model contribution at every particle position would be considerable.

At this point the problem of particle diffusion in variable density flows, as mentioned in §5.2.1, is addressed. A one-dimensional domain measuring 10 mm in length is discretised into 60 computational cells, and is initialised with a smooth scalar interface at the mid-point. This scalar represents the mixture fraction in an interfacial region of fuel and air, and is multiplied by a density that is initially taken to be constant ($\bar{\rho} = 1.4$ kg/m^3). The density-weighted mixture fraction is now represented both with an Eulerian field, and separately as Lagrangian particles (as shown in Fig. 5.14(a)) and is subjected to a constant diffusivity[3] ($\bar{\mu} = 2.0 \times 10^{-5}$ kg/ms; $Sc = 1.0 \times 10^{-3}$). The Eulerian and Lagrangian descriptions of this density-weighted mixture fraction field $\bar{\rho}\tilde{\phi}$ are transported by diffusion with an eighth-order accurate approximation for the Eulerian term, and a Wiener process (Eq. 5.31) for the particles. The diffusivity used to scale the length of the Wiener step is:

$$D_p = \frac{\bar{\mu}_p}{\bar{\rho}_p Sc} = \frac{\bar{\nu}_p}{Sc} \tag{5.52}$$

Where the subscript p indicates that the quantity has been interpolated on to the particle position. The mixture fraction $\tilde{\phi}$ after transport may be derived by dividing the transported field $\bar{\rho}\tilde{\phi}$ by the density $\bar{\rho}$. The resulting transported and derived scalars for a constant density, constant viscosity case are shown in Fig. 5.14(a), where initial fields are labelled with the subscript 0, and final fields are labelled with the subscript t. The good agreement between the Eulerian and Lagrangian descriptions suggests that the current particle approach describes the diffusion process satisfactorily.

Next a variable density is defined throughout the simulated domain, where the fuel/air interface recreates a reactive, counter-flowing 'opposed jet' arrangement (described in detail in Chapter 11). The fuel is a typical diluted methane/nitrogen mixture, and the steady flamelet approach is used to describe the physical properties of the flow: the local density and temperature are determined from the mixture fraction according to the pre-computed values in a look-up table. A full description of

[3] A very low Schmidt number is chosen to reduce the number of time-steps required for the scalar to have experienced a significant net diffusive flux

the steady flamelet approach is omitted here; a detailed explanation is provided by, for example, Bilger [200]. A variable molecular viscosity field is also derived by relating the local viscosity to the temperature, through Sutherland's law [201]:

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + T_s} \tag{5.53}$$

Where T is the local temperature and the mixture properties are assumed to be similar to air, so that $C_1 = 1.458 \times 10^{-6}$ kg/ms$\sqrt{\text{K}}$ and $T_s = 110.4$ K. As the present case is for demonstrative purposes only the initial conditions are held constant (i.e. the density and temperature are not updated as the mixture fraction field is transported), while convective fluxes are also ignored.

Applying the present formulation for particle diffusion under these conditions results in the transported and derived fields shown in Fig. 5.14(b). In this case a significant disparity exists between the transported Eulerian and Lagrangian density-weighted mixture fractions. The order-of-magnitude variation in the density around the mid-point of the domain ($x = 5$ mm) results in a deviation of up to 30% between the two derived mixture fraction fields, while the Lagrangian description also over-predicts the mixture fraction on the fuel (left-hand) side. The mixture fraction exceeds unity in this region, which strongly suggests that the present expression for particle diffusion is inadequate for this configuration.

Further investigation indicates that this disparity results from the non-uniform diffusivity around the mid-point of the domain. A solution is proposed by Fyrillas and Nomura [202] (citing Kinzelbach and Uffink [203]) who suggests that an additional drift term equal to the divergence of the diffusivity should be added to the particle equation of motion. Diffusive particle transport is then described in one dimension as:

$$s^p(t + dt) = s^p(t) + \left.\frac{dD}{dx}\right|_p dt + dW\sqrt{2D_p \Delta dt} \tag{5.54}$$

Considering the definition of the particle diffusivity given by Eq. 5.52 (based on the kinematic viscosity), this expression results in a non-zero drift term for the case of constant dynamic viscosity and variable density. This adjustment to the particle equation of motion therefore causes an accumulation of particles around the local density minimum, as shown in Fig. 5.14(c), and subsequently a significant over-prediction of the mixture fraction in this region. The drift term is therefore modified so that it is equal to zero for a constant viscosity, but is still a function of density, and maintains the correct units of length:

$$s^p(t + dt) = s^p(t) + \left.\frac{1}{\bar{\rho}_p Sc}\frac{d\bar{\mu}}{dx}\right|_p dt + dW\sqrt{2D_p \Delta dt} \tag{5.55}$$

For the case of constant dynamic viscosity and variable density this modified term yields the transported fields shown in Fig. 5.14(d), where the erroneous accumulation of particles around the local density minimum has been corrected. Furthermore, a re-computation of the case with varying density and varying viscosity applying Eq. 5.55 shows that the desired agreement between the Eulerian and Lagrangian descriptions or diffusive flux has been restored, as illustrated by Fig. 5.14(e).

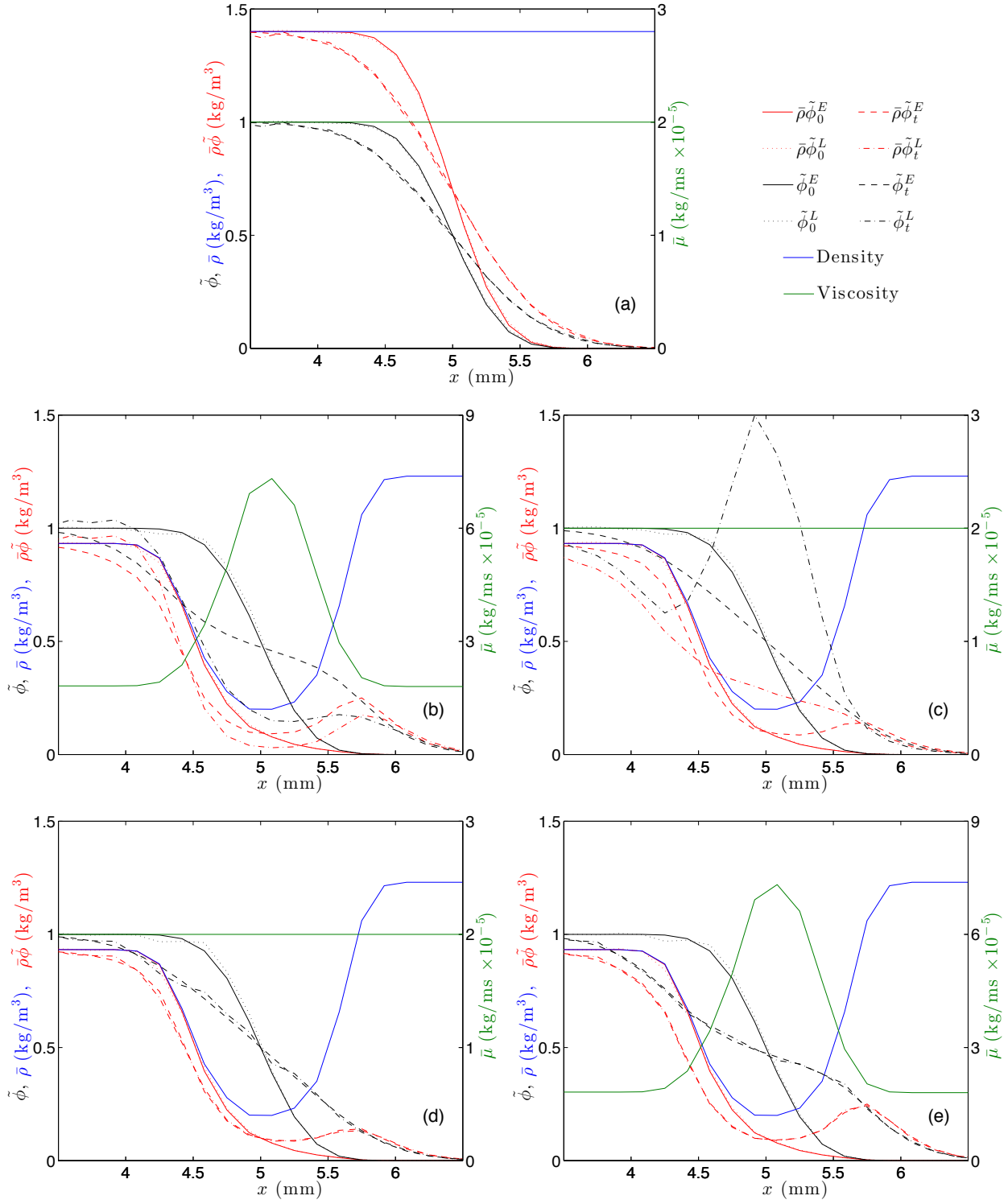


Figure 5.14: Transported and derived scalars described by Eulerian and Lagrangian methods, at constant or variable density and viscosity.

5.4.3 Lagrangian Reconstruction

To perform a summation of the Eulerian and Lagrangian phases a representation of the Lagrangian field must be constructed in the Eulerian framework, as described in §5.2.4. According to Eq. 5.34 a spatial filter $\mathcal{F}$ is applied to the scalar contribution of each particle; in practice, it is convenient to use the same top-hat filter (of width Δ) that is (implicitly) applied for the discretisation of scalar quantities in the Eulerian framework. The reconstruction of the Lagrangian field therefore becomes the summation of the weighted averages over Δ from the position of each particle on to the surrounding grid points.

Figure 5.15 demonstrates the reconstruction of four Lagrangian particles in two dimensions, where the position of each particle is marked with a blue point. The red box around each particle p denotes the extent of the filter that is applied to that particle, where the contribution from p for a given cell (i, j) is equal to the area (or, in three dimensions, the volume) of the p-th box within that cell. The overall value of the Lagrangian field ϕ^L in each cell is the sum of the individual contributions (scaled by the corresponding scalar weight w_ϕ^p for each particle) from all of the particles.

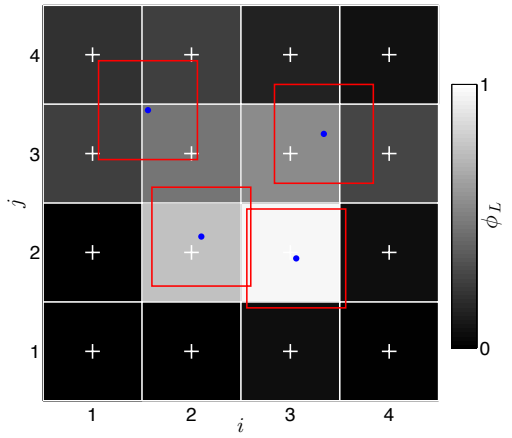


Figure 5.15: Scalar contributions of four Lagrangian particles in a 2D Eulerian frame.

Figure 5.15 also highlights a limitation of the E-L method: the particle in cell (3, 2) lies near to the cell centre, meaning that almost all of the scalar contribution from that particle goes to that cell. However, in cell (2, 3), the particle has been (randomly) placed far from the cell centre, meaning that its contribution is distributed almost equally across the nearest four neighbours. In this case, therefore, the reconstructed Lagrangian field bears little resemblance to the original field (which would have consisted of the four central cells having a unit scalar value). This random error has already been shown in §5.3.2 to reduce as the Lagrangian particle density L^ϱ is increased. However, the 'noise' in the reconstruction may still cause the scalar value in a cell to exceed its bounds (for example, where mixture fractions outside the range $0\% \leq Z \leq 100\%$ may be recorded) at any Lagrangian particle density.

The method of particle reconstruction described here has been found to be more expensive as the dimensionality of a simulation is increased, as previously shown in Fig. 5.11. The reason for this increase in expense now becomes apparent: in one dimension the contribution of one particle

to two cells is determined from one length (i.e. the distance from the particle to the cell centre); in two dimensions, one particle contributes to four cells as a function of two lengths (the areas of the red boxes in Fig. 5.15), and in three dimensions the contribution to eight cells is a function of three lengths (a corresponding volume). There are therefore more arithmetic and memory operations required to determine the scalar contribution for a particle in three dimensions than there are in two. The computational cost of the reconstruction for a fixed number of particles may be expected to scale as 2^d, where d is the dimensionality of the domain.

It should be noted that the present method has the advantage of being conservative, i.e. the sum of the reconstructed Eulerian field is equal to the sum of the individual particle contributions throughout the domain (assuming that an appropriate boundary treatment is applied). However, the use of a top-hat filter of width Δ is not the only conservative option for the spatial basis of the Lagrangian reconstruction. Wider top-hat filters would be applicable, for example, but would require a more complex calculation to be performed for the summation of weighted averages across a larger number of cells; they may also demand a more complicated boundary treatment to ensure their conservativeness. Other spatial bases, such as the delta function or Fourier basis, are also commonly applied, as described by Kucukelbir et al. [204]; the interested reader is referred to the book of Snieder for more information on these methods [205]. Kucukelbir et al. also describe an adaptive algorithm which varies the basis with the topology of the data [204]. These latter methods would all be more computationally expensive than the simple, cell-sized top-hat filter, however.

5.4.4 Re-initialisation

In §4.4.2 it has been demonstrated that there exists a relationship between the smoothness of an Eulerian field and the magnitude of the oscillations that are generated within that field when it is transported using CDS. It has also been shown that these oscillations may be restricted to a negligible size if a sufficient smoothness in the transported field is sustained (Fig. 4.13). The stability of the E-L method may therefore be ensured through the process of 're-initialisation', where information is transferred from the Eulerian phase to the Lagrangian phase, and this should specifically be performed in regions where the smoothness of the field is becoming reduced (by the steepening of gradients due to shear or strain within the flow, for example).

For regions of the flow that contain Lagrangian particles, but that have been transported away from sharp changes in scalar gradient (where they would previously have been needed), it is desirable to transfer information in the opposite direction, i.e. from the Lagrangian phase to the Eulerian phase. The removal of redundant particles in these regions (and a corresponding addition of their contribution into the Eulerian field) would have the benefit of reducing the computational cost of the method, while accuracy may also be improved as the noise in the Lagrangian field (resulting from the randomness of the particle distribution, as shown in Fig. 5.15) becomes less wide-spread.

Two different methods for the re-initialisation process are proposed and compared here.

Global Re-initialisation The transferral of information between the Lagrangian and Eulerian fields may be achieved by calculating an intermediate Eulerian-Lagrangian summation, and initialising the Eulerian and Lagrangian fields (using the decomposition, particle distribution and

assignment described in §5.3) from this intermediate solution. This approach therefore requires a reconstruction of the Lagrangian field to be performed prior to re-initialisation. The global method appears to have a number of benefits: it should exactly preserve the scalar information, while re-distributing it between the two phases as necessary; and it should account for the transfer of information from the particles to the Eulerian field, as a smooth region containing particles should not be re-assigned any high-frequency contribution upon re-initialisation.

However, global re-initialisation is found to have a distinct disadvantage. Figure 5.16 shows a stationary, one-dimensional scalar field that has been decomposed into low- and high-frequency parts, which has then been re-initialised 10 times using the global method described here. The re-initialised fields should exactly match the initial fields, which can be seen to be true for the Eulerian phase; however, the Lagrangian phase shows a significant deviation after re-initialisation. Any errors that enter into the solution during any of the steps of decomposition, particle distribution, particle assignment, and reconstruction become incorporated into the summation from which the re-initialised fields are determined, and this results in an accumulation of errors in the Lagrangian field with each re-initialisation event.

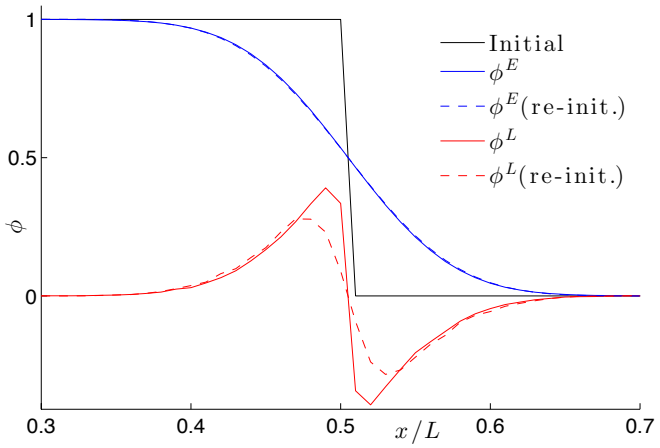


Figure 5.16: Global re-initialisation, showing accumulating Lagrangian errors.

Local Re-initialisation To prevent this accumulation of error an alternative method of re-initialisation is proposed, where the transfer of information between the low- and high-frequency fields is performed only in specific regions of the simulation domain. Considering the relationship between the scalar second derivative and the development of oscillations in the Eulerian field during transport, it makes sense to identify these regions based on the local (absolute) value of that second derivative. The procedure for a local re-initialisation is as follows. Firstly the second derivative of the Eulerian field is estimated using a second-order accurate approximation, which may be written in one dimension as:

$$\nabla^2\phi^E(i) \approx \phi^E(i-1) - 2\phi^E(i) + \phi^E(i+1) \tag{5.56}$$

Where $\nabla^2\phi^E(i)$ is the second derivative of ϕ^E in cell i with respect to the cell co-ordinates

(not the physical co-ordinates). Higher-order approximations would be straightforward to apply, where stencil weights are provided by Fornberg [114], but second-order accuracy is sufficient in the present context: the expression in Eq. 5.56 is only used to identify regions requiring re-initialisation, meaning that it does not affect the accuracy of the scalar field itself.

A 'cut-off' value of the local second derivative $\phi''_{\max}$ is defined, below which the Eulerian field is smooth enough that transfer of any information to the Lagrangian phase is not required. In general $\phi''_{\max}$ should be set to the maximum value of the second derivative of the Eulerian field at initialisation, assuming that the Gaussian filter width σ_f has been selected to yield a sufficiently smooth Eulerian field. If $\phi''_{\max}$ is set to be lower than this value, the re-initialisation scheme will transfer scalar information the first time it is applied; if it is set much higher, the Eulerian field may start to exhibit undesirable oscillatory behaviour.

Figure 5.17 illustrates a step function (black line) decomposed with an arbitrary Gaussian filter of width σ_f. The resulting filtered field (blue line) is equal to the cumulative distribution function of the Gaussian filter, otherwise known as the Gauss error function, while the first derivative of the filtered field (red line) is equal to the filter kernel (marked with '+' symbols) centred around the original step. The second derivative (magenta line) is equal to the gradient of the filter kernel.

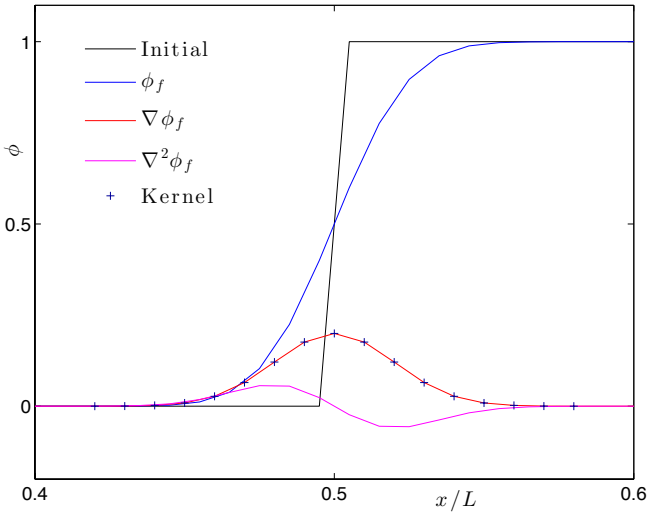


Figure 5.17: Step function: initial and filtered, with derivatives, compared to the filter kernel.

The maximum second derivative of the filtered field is therefore equal to the maximum gradient of the filter kernel. Differentiation of the expression defining the filter kernel (Eq. 5.43) yields:

$$h'(\delta x) = \frac{-\delta x}{\sigma_f^3\sqrt{2\pi}} \exp\left(-\frac{\delta x^2}{2\,\sigma_f^2}\right) \tag{5.57}$$

While differentiating a second time gives:

$$h''(\delta x) = \left(\frac{\delta x^2}{\sigma_f^5\sqrt{2\pi}} - \frac{1}{\sigma_f^3\sqrt{2\pi}}\right) \exp\left(-\frac{\delta x^2}{2\,\sigma_f^2}\right) \tag{5.58}$$

Setting Eq. 5.58 to zero indicates that the point of inflection (maximum gradient) of the kernel will be located at $\delta x = \sigma_f$, while subsequently evaluating Eq. 5.57 at $\delta x = \sigma_f$ yields:

$$\left|\nabla^2 \phi^E\right|_{\max} = \frac{1}{\sigma_f^2 \sqrt{2\pi e}} \tag{5.59}$$

The gradient of the discretised kernel will be slightly different to that of the continuous function given in Eq. 5.57; however, the maximum deviation is typically small (7% for a filter width $\sigma_f = 2\Delta$, dropping to 1.3% at $\sigma_f = 5\Delta$). Finally it is suggested that the value of $\phi''_{\max}$ should be based on the value determined from Eq. 5.59 as:

$$\phi''_{\max} = \gamma \left|\nabla^2 \phi^E\right|_{\max} \tag{5.60}$$

Where γ is a constant for which $\gamma = 1$ should be satisfactory for most cases, assuming that the initial filtered field is sufficiently smooth. Applying Eq. 5.59 for the Gaussian filter width $\sigma_f = 5\Delta$ suggested in §5.3.1 leads to a maximum second derivative in the initial Eulerian phase of 0.0097, which is equivalent to $R_{\min} \approx 100$ in Fig. 4.13. Applying a value $\gamma = 1$ in Eq. 5.60, any regions of the Eulerian field in which $\left|\nabla^2 \phi^E\right| > 0.0097$ will therefore be filtered to increase their smoothness.

A suitable filtering process must now be selected. An iterative top-hat filter has been shown to be equivalent to applying a Gaussian filter once (in §5.3.1), and as a range of filter widths will be required throughout the domain, a top-hat filter is the more suitable choice. It is then necessary to define a relationship between the second derivative $\nabla^2 \phi^E$ and the number of filtering operations N_f (which may be related to an effective Gaussian filter width) that should be applied in each cell. A linear trend is assumed, so that N_f is defined as a function of $\phi''_{\max}$ as:

$$N_f(i) = \max\left(0,\ \frac{\left|\nabla^2 \phi^E(i)\right| - \phi''_{\max}}{\phi''_{\max}}\right) \tag{5.61}$$

Having calculated N_f for each cell, localised iterative top-hat filtering may be performed on the Eulerian field. The additional contribution S_{ϕ^L} to the Lagrangian field is determined as the difference between the pre- and post-filtered Eulerian fields:

$$S_{\phi^L} = -S_{\phi^E} = \phi^E - \phi^E_f \tag{5.62}$$

Where Eq. 5.62 reflects the fact that, in the transport equations presented in §5.2.1 (Eqs. 5.4–5.15), the Lagrangian source is equivalent to an Eulerian sink.

This approach ensures that a repeated application of the re-initialisation algorithm does not cause any accumulation of error in the solution, while satisfying the aim of maintaining a suitably smooth Eulerian field. However, it will tend to generate a large number of 'partial' particles (for which the scalar weight α_p appearing in Eq. 5.17 will be less than one). A partial particle will probably be created in every cell in which re-initialisation filtering is performed, as the value of S_{ϕ^L} in a given cell is unlikely to be exactly equal to the contribution of a whole number of particles. Over numerous re-initialisation events these additional partial particles may start to have an impact on

the computational efficiency of the simulation.

It would be possible to formulate an algorithm to identify any cells containing multiple partial particles, and to re-combine the partial particles into a number a 'whole' ($\alpha_p = 1$) particles with a further, single, partial particle for any remainder. This algorithm could prove to be inefficient, however, as all of the particles within a given cell must be explicitly identified before any particle re-combination may be performed. This would require a search to be performed throughout the entire particle array, for every cell in the domain, at every re-initialisation event.

An alternative is therefore proposed, which is to avoid the need for partial particles entirely. Wherever a Lagrangian contribution is determined – either during the initial decomposition, or in a re-initialisation – the value of that contribution is quantised so that it is an exact multiple of $1/\mathrm{L}^{\varrho}$, and the error in the Lagrangian contribution due to this quantisation is added back into the Eulerian phase. Considering that the quantisation error will be less than the scalar contribution of a single particle, the enforcement of whole particles is not expected to disrupt the smoothness of the Eulerian field.

For testing of the E-L method re-initialisation will be performed at a fixed time-step interval, where it is assumed that oscillations will develop over a larger number of time-steps than the frequency of the re-initialisation events. The chosen frequency depends on the required level of accuracy, and perhaps on the configuration of the simulation, but a value of four time-steps between each re-initialisation event is suggested. This represents a balance between the need to maintain stability and the additional computational cost incurred by the re-initialisation routine itself; more frequent re-initialisation may be considered for more complex cases. Particles will not necessarily be added every time re-initialisation is performed: if the smoothness of the field changes slowly particles may be added quite infrequently, as it may take many time-steps for the second derivative of the Eulerian field to exceed $\phi''_{\max}$. Even then, any re-initialisation filtering may be highly localised, meaning that only a few particles would be added.

A disadvantage of the local re-initialisation method proposed here is the fact that it cannot account for the transferral of scalar information from the particle field to the Eulerian field. An algorithm could be designed to identify particles for removal; however, the requirement for such a scheme may be mitigated as regions of redundant particles will eventually be convected out of the domain and removed from the calculation.

5.5 Summary

In this chapter the concept of the joint Eulerian-Lagrangian method has been introduced, starting with the decomposition of a scalar field into its low- and high-frequency components. This decomposition is performed with a Gaussian filter kernel, which may be implemented more efficiently than an equivalent iterative top-hat filtering process. A Gaussian filter width of $\sigma_f = 5\Delta$ is suggested to provide a compromise between the stability and computational cost of the method.

Transport equations for each of the Eulerian and Lagrangian components have been derived, for constant and variable density cases, in the context of Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES). For the (low-frequency) Eulerian phase the convection and diffusion

terms are estimated using central approximations of eighth order-of-accuracy. Advancement in time for both phases is performed using a third-order accurate, low-storage Runge-Kutta method.

The high-frequency phase is represented with massless Lagrangian particles. A key parameter of the method is the Lagrangian particle density L^{ϱ}: errors in the high-frequency field are proportional to $1/\sqrt{\mathrm{L}^{\varrho}}$, while the computational cost of transporting and reconstructing the particle phase is linearly proportional to L^{ϱ}. Generally it is suggested that a particle density $\mathrm{L}^{\varrho} = 1000$ is suitable for two-dimensional cases, while L^{ϱ} should be reduced by an order of magnitude in three dimensions. However, the chosen value of L^{ϱ} may depend on the size, resolution and duration of a simulation; on the topology of the transported scalar field; or on the available computational resources.

Particles are positioned with a uniform random distribution in each computational cell, where the location of each particle is stored in physical co-ordinates. Other particle properties include its velocity and scalar contribution, while a particle 'activity' is also defined to facilitate the implementation of the E-L method on parallel computing architectures, and to allow for an efficient recycling of deactivated particles.

Particle convection is performed by interpolating the underlying (Eulerian) velocity field on to the location of each particle, and then updating the particle position using the same Runge-Kutta time integration scheme applied for the Eulerian phase. Particle diffusion is approximated via a stochastic Wiener process, with an additional drift term for flows involving non-constant density or viscosity. An Eulerian representation of the Lagrangian field is reconstructed by applying a top-hat filter of width Δ to the scalar contribution of each particle, and by subsequently calculating the summation of all particle contributions in each cell.

The stability of the E-L method is preserved by an intermittent transferral of information from the Eulerian phase to the Lagrangian phase. A fixed frequency of four time-steps is recommended for this transferral, although the mechanism will not necessarily transfer information at every application. The maximum second derivative of the low-frequency field allowed by the re-initialisation algorithm is a function of the second derivative of the low-frequency field after the initial decomposition, but is ultimately controlled by a re-initialisation parameter γ. This parameter typically takes a value $\gamma = 1$, but may be reduced to make the re-initialisation routine more 'aggressive' in its transfer of information away from the Eulerian phase.

6 Test Case 1: The Zalesak Disc

To test and validate the different aspects of the joint Eulerian-Lagrangian (E-L) method proposed in Chapter 5, and to highlight possible improvements, a simple, two-dimensional test case is studied. The selected case was originally proposed by Zalesak, and consists of a 'solid body' rotation in a forced flow-field [184]; this avoids any complications associated with the solution of the flow-field itself. In this chapter the criteria considered in the selection of the test case are discussed; the design and numerics of the test simulations are described; and a framework of metrics are introduced and applied to provide a quantitative assessment of the performance of the E-L method in comparison to traditional Eulerian convective schemes.

6.1 Case Objectives

In the present study a range of existing (Eulerian) convective transport schemes will be tested to demonstrate that none provide a satisfactory description of scalar transport for the selected two-dimensional case. The performance of the proposed E-L method will be investigated to verify that it is capable of delivering a higher level of accuracy than these existing Eulerian schemes. This study will also be used to identify the most effective parameters for the E-L method in terms of accuracy and computational efficiency, while suggesting avenues for potential improvement and development.

6.2 Case Description

The selected case was originally designed and used for the analysis of the 'Fully Multidimensional Flux-Corrected Transport' (FCT) algorithms proposed by Zalesak [184]. The Zalesak case, as it will be referred to here, is selected as it satisfies a number of the criteria that are considered to constitute an effective test for a numerical convective scheme. These criteria are described below.

Diffusivity To test the numerical diffusivity of a scheme, it is advantageous to enforce a zero-diffusion (infinite Schmidt number) condition. The inclusion of physical diffusion would reduce the impact of numerical diffusion and would complicate the assessment of the accuracy of a given scheme. The Zalesak case implements zero diffusivity.

Velocity Field The action of a turbulent flow on a scalar field is to translate, rotate, and deform that field, due to the bulk motion of the flow and any shear stresses within it. However, to simplify the quantitative analysis of the resulting transported fields it is desirable to consider a pure translation or pure rotation case, where the scalar field does not become deformed. In this way the

accuracy of the method is more easily measured, as the initial field may be compared to the final field by moving the frame of reference accordingly (assuming that no diffusive fluxes are applied). In the Zalesak case the scalar is transported by pure rotation at a constant rotational velocity.

Boundary Conditions Calculations starting from a set of initial values require the prescription of boundary conditions at the edges of the computational domain. For this study it is important that these conditions do not have any influence on the scalar field, as this could be mistaken for an artefact of the numerical scheme under consideration. Ideally the domain would extend infinitely in all directions, which may be effectively achieved by applying periodic boundary conditions; here, however, the region of interest is removed from the boundaries by a suitable number of computational cells.

Symmetry An initially-symmetric scalar field subjected to a constant (translational or rotational) velocity, at constant density and viscosity, should maintain its symmetry. Specifying such a field therefore allows a measure of that symmetry to be applied as an additional metric. This may be an important factor as numerical schemes often behave differently for 'uphill' and 'downhill' gradients, which tends to cause a symmetric shape to lose its symmetry. The initial scalar field used in the Zalesak case possesses one plane of symmetry.

Feature Size The shape of the scalar field to be transported should be chosen carefully; ideally the scalar topology should include small features, as these are most susceptible to (physical and numerical) diffusive fluxes. Although the initial scalar topology of the Zalesak case is relatively simple, it contains sharp corners and points that are sufficient to satisfy this requirement.

Maximised Fluxes The shape of the scalar field should vary such that the contours of the field lie normal to the streamlines of the flow. This ensures that convective fluxes are as large as possible, so that any distortion of the scalar topology due to the chosen numerical scheme is accentuated. A poor combination of topology and flow-field would make any scheme appear to perform well (for example, where the principal flow direction is parallel to the scalar contours). Considering the expression for the scalar flux across a cell (Eq. 4.5), it can be seen that fluxes are proportional to the velocities u_F normal to the faces of the cell, i.e. $u_F = (u_i \cdot n_i)_F$. For a symmetrical scalar field the plane of symmetry should therefore be normal to the flow direction. In the Zalesak case this condition is satisfied as the initial plane of symmetry lies normal to the vortical streamlines, and subsequently rotates with the flow.

A further benefit of the Zalesak case is that the analytical transported solution is identical to the initial solution. This permits a straightforward quantification of the degree of error introduced by each method tested. Finally, the Zalesak case has previously had extensive use for the testing of convective transport methods and level set algorithms (for example by Scardovelli and Zaleski [206]; Enright et al. [180, 186]; and Hartmann et al. [207], amongst others).

6.2.1 Velocity Field

The velocity components u and v (in the Cartesian directions i and j, respectively) of the rotational flow-field are described as:

$$u = \omega(y_0 - y) \tag{6.1}$$

$$v = \omega(x - x_0) \tag{6.2}$$

Where ω is the (constant) angular velocity, x and y are the Cartesian co-ordinates in the i- and j-directions, and x_0 and y_0 are the co-ordinates of the centre of rotation (located at the centre of the domain). The angular velocity is chosen such that one revolution about the rotational axis at (x_0, y_0) is completed in 157 seconds; the reason for this will be explained in §6.3. The flow-field yields no shear at any point within the domain, meaning that the topology of a scalar quantity convected by these velocity components will not be distorted.

6.2.2 Solid Body

The solid body consists of a disc with a slot removed, as shown in Fig. 6.1. The two-dimensional domain measures 100×100 cells, with the rotational axis located at co-ordinates $(50, 50)$. The disc indicated by the shaded region in Fig. 6.1 has a radius $r = 15$ cells, and is centred at co-ordinates $(50, 75)$ so that the radius of rotation $R = 25$ cells. The slot is cut from the point on the disc closest to the centre of rotation, and extends across the disc to leave a 'bridge' of five cells remaining to connect the two halves. The slot itself is symmetric about the disc centre, and measures five cells across. In Zalesak's original work grid points within the disc were given a 'weight' $w = 3$, while all other points had $w = 1$; in the present work, however, it is more appropriate to consider the disc as representing a scalar value $\phi = 1$, where all other points constitute $\phi = 0$. Reducing the total range of the transported field reduces the maximum convective flux by a factor of two, which changes the numerics of the problem in comparison to the original case. No direct comparison with Zalesak's work is intended, however; and as the Eulerian-Lagrangian description will typically be aimed at bounded scalars such as mixture fraction, it makes more sense to define the field in this way.

Figure 6.2 shows the solid body described above discretised on to the computational grid, where the height of each square above the ij-plane represents the (cell-centred) scalar value. Figure 6.3 shows the summation of that initial scalar field after it has been decomposed into its Eulerian and Lagrangian components. The views shown in Figs. 6.2 and 6.3 (and in subsequent figures throughout this chapter) are limited to 50×50 cells centred on the mid-point of the disc.

6.3 Simulation Numerics

Computations are performed using the `PsiPhi` DNS/LES code introduced in Chapter 3, where the dimensionality of the code is reduced to two; the flow-solving capability is removed; and the diffusivity of the scalar is set to zero. The program computes only the convective fluxes for a single scalar, and, where applicable, the convective transport of Lagrangian particles, within a constant prescribed flow-field. The domain is discretised on to cells of unit size ($\Delta = 1.0$ m).

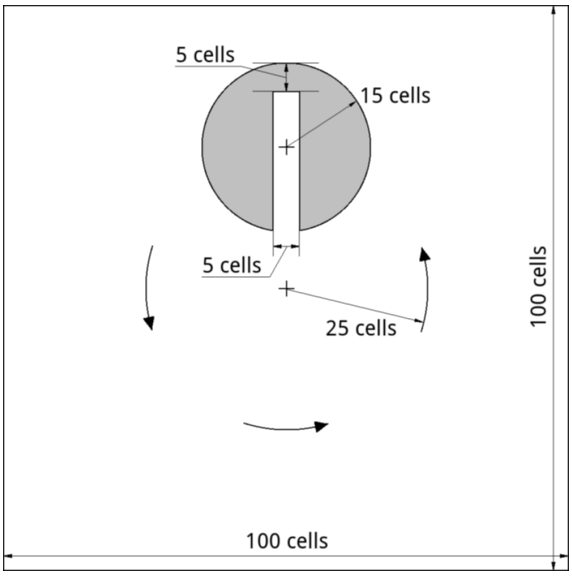


Figure 6.1: Zalesak disc: Schematic representation. The shaded area represents $\phi = 1$.

Time integration is performed using the low-storage, three-step, third-order accurate Runge-Kutta scheme described in §3.2. The angular velocity is set to $\omega = \pi/(3.14R)$, which, for a time-step width of $dt = 0.25$ s, corresponds to one revolution in 628 computational steps[1]. The maximum value of the CFL number (§3.3) for this flow-field and dt is CFL = 0.5.

In addition to various formulations of the E-L method a number of the traditional spatial schemes introduced in Chapter 4 are used for comparison, including the Upwind Differencing Scheme (UDS), Quadratic Upwind Interpolation for Convective Kinematics (QUICK), and a selection of TVD schemes. The E-L method itself is tested with both second and eighth orders of accuracy for the Central Differencing Scheme (CDS) used to transport the Eulerian phase, as well as differing Lagrangian particle densities. The cases presented here are summarised in Table 6.1.

Table 6.1: Numerical schemes to be tested with the Zalesak disc.

#	Method	Eulerian Scheme	Particle Density, L^{ϱ}
Z1	Eulerian	UDS	-
Z2	Eulerian	QUICK	-
Z3	Eulerian	Osher TVD ($\beta = 1.0$)	-
Z4	Eulerian	Osher TVD ($\beta = 2.0$)	-
Z5	Eulerian	CHARM TVD	-
Z6	Eulerian	SuperBee TVD	-
Z7	E-L	CDS2	1000
Z8	E-L	CDS8	1000
Z9	E-L	CDS8	100

Other parameters relating to the E-L method remain fixed throughout these tests. The width of

[1]The factor $\pi/3.14$ in the expression for ω ensures that a revolution takes 628 steps exactly, rather than the 628.3185... steps that would be required from using $\omega = 1/R$

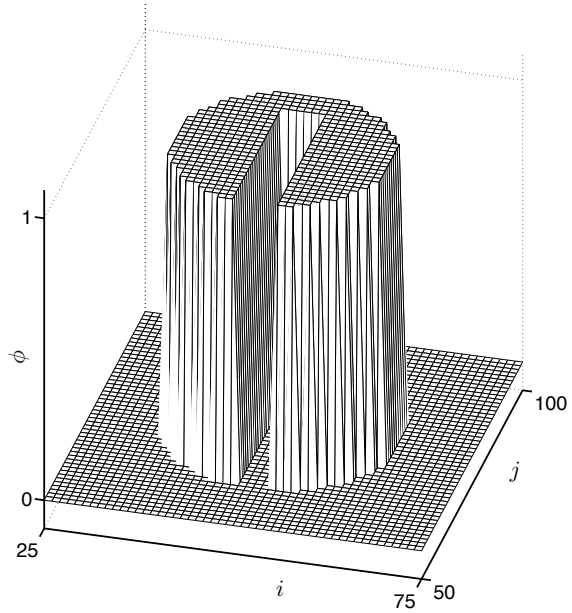


Figure 6.2: Zalesak disc: Initial conditions after discretisation.

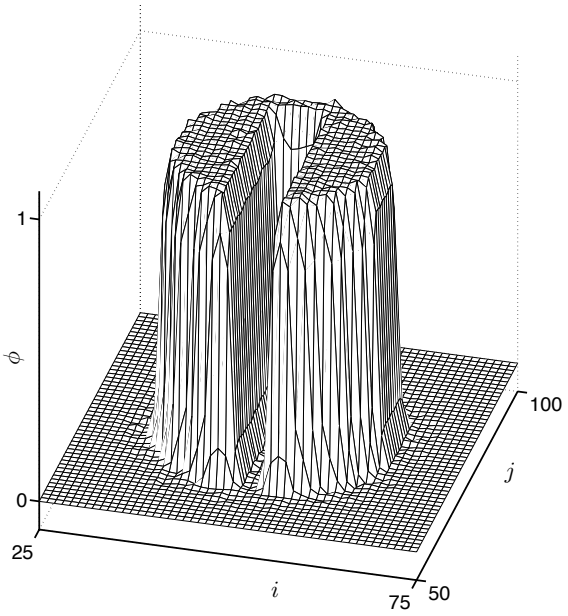


Figure 6.3: Zalesak disc: E-L reconstruction.

the Gaussian filter kernel (with which the original field is decomposed) is limited by the distance from the perimeter of the solid body to the computational domain boundary. It would be inconvenient to prescribe boundary conditions for either the Eulerian or Lagrangian phases in this simple test, so there should be no contribution at the boundary from either phase during the simulation. The filter width is thus set to $\sigma_f = 2.5\Delta$ (corresponding to $n_f = 10$ filter support points in each direction), which is less than the value of $\sigma_f = 5\Delta$ suggested in §5.3.1. To account for this the re-initialisation parameter γ in Eq. 5.60 is reduced to $\gamma = 0.25$. Such a value of γ will result in a transfer of information to the Lagrangian phase at the first re-initialisation event, as the value of $\phi''_{\max}$ will be lower than the maximum second derivative of the initial Eulerian phase. This is necessary to maintain a compromise between numerical accuracy (in limiting the development of oscillations in ϕ^E) and computational efficiency, but also helps to provide a thorough test of the re-initialisation algorithm. Re-initialisation is performed every four time-steps, as suggested in §5.4.4.

6.4 Results and Discussion

A selection of metrics are employed to provide a quantitative assessment for each case. The boundedness of a scheme is considered by measuring the amount by which the upper (ΔB_U) and lower (ΔB_L) limits of the transported field ϕ^t exceed (or have dropped below) the range of the initial field ϕ^0:

$$\Delta B_U(\phi) = B_U^0(\phi) - \max_{1 \le i \le N} \left[\max_{1 \le j \le N} \left(\phi^t(i,\, j) \right) \right] \tag{6.3}$$

$$\Delta B_L(\phi) = \min_{1 \le i \le N} \left[\min_{1 \le j \le N} \left(\phi^t(i,\, j) \right) \right] - B_L^0(\phi) \tag{6.4}$$

Where:

$$B_U^0(\phi) = \max_{1 \le i \le N} \left[\max_{1 \le j \le N} \left(\phi^0(i,\, j) \right) \right] \tag{6.5}$$

$$B_L^0(\phi) = \min_{1 \le i \le N} \left[\min_{1 \le j \le N} \left(\phi^0(i,\, j) \right) \right] \tag{6.6}$$

Boundedness is expressed as a percentage of the initial (unit) range, and should remain as close to zero as possible. The definitions of ΔB_U and ΔB_L are formulated so that negative values indicate unbounded results for both measures.

The total error E is determined as the sum of the absolute difference in each cell between the initial and transported fields:

$$E(\phi) = \sum_{i=1}^{N} \sum_{j=1}^{N} \left| \phi^0(i,\, j) - \phi^t(i,\, j) \right| \tag{6.7}$$

Where N is the size of the domain. The total error should be as close to zero as possible.

The total variation TV of the original field is compared to that of the final field. The definition of TV for a one-dimensional function is provided in Eq. 4.29 (§4.5); in the present case, however, this expression must be extended into two dimensions, and TV is thus defined as:

$$\mathrm{TV}(\phi) \equiv \sum_{i=1}^{N} \sum_{j=1}^{N} \left| \phi(i+1,\, j) - \phi(i,\, j) \right| + \left| \phi(i,\, j+1) - \phi(i,\, j) \right| \tag{6.8}$$

The measured quantity is then the change in total variation, ΔTV:

$$\Delta\mathrm{TV}(\phi) = \mathrm{TV}^t(\phi) - \mathrm{TV}^0(\phi) \tag{6.9}$$

The TV of the transported field should be equal to the TV of the initial field (i.e. ΔTV$= 0$), as the scalar is transported by convection only. The total variation of the initial scalar field shown in Fig. 6.2 is 164.

The initial field possesses left-to-right symmetry, and this should be maintained at the end of each full revolution. The symmetry error E_{sym} of the transported field is calculated as the absolute difference between the left-hand and mirrored right-hand sides:

$$E_{sym}(\phi) = \sum_{i=1}^{N/2} \sum_{j=1}^{N} \left| \phi^t(i,\, j) - \phi^t(N - i + 1,\, j) \right| \tag{6.10}$$

This measure of asymmetry is zero for the initial Zalesak disc field (indicating perfect symmetry), and should be as close to zero as possible after transport. Note that for a zero symmetry error in the discretised initial scalar field it is necessary to increase the width of the disc slot to six cells.

The boundedness, total error, change in total variation and symmetry error metrics are also used in E-L simulations to determine the accuracy of the initial Eulerian-Lagrangian field (i.e. the original scalar field after decomposition, Lagrangian reconstruction and summation, as shown in Fig. 6.3),

while the boundedness and total error measures are used to compare the final transported E-L field to the initial E-L field. The total error in the E-L method may then be considered to consist of an initialisation error and a transport error; it is important to distinguish between the two, as the initial decomposition and reconstruction introduces a degree of error into the representation of the scalar field even before any transport has occurred.

A further useful measure is the computational cost of each of the schemes used. This cost is determined by recording the time taken for each case to execute on a single 2.4GHz AMD Opteron^{TM} 280 processor. For the E-L simulations the total number of Lagrangian particles at the end of the simulation is also recorded, as this directly influences the computational cost.

Results are presented in the manner of Zalesak [184], where each transported field may be compared directly with the initial field shown in Fig. 6.2. Each of the associated plots also shows the scalar profile through the centre of the solid body, where the transported field (shown as a dashed black line) is compared to the initial field (solid black line). In addition the plane of symmetry (dotted black line) and a reflection in this plane of the left-hand side of the transported field (dashed red line) are plotted. Figures 6.4 to 6.14 for Cases Z1 to Z9 depict the scalar field after 628 computational time-steps, i.e. after one rotation.

6.4.1 Eulerian Simulations

Case Z1: UDS, Fig. 6.4. The use of a low order-of-accuracy, upwind-biased convection scheme results in a large degree of numerical diffusion. This has smoothed out the topology of the initial scalar field, and has caused the peak scalar value to drop to around 0.5. The solution remains bounded. The scalar has been spread over a wide area, and the maximum scalar gradient is significantly reduced. It can be seen that the solution is nearly symmetrical, and this suggests that the action of numerical diffusion in UDS is the same for both 'uphill' and 'downhill' scalar gradients.

Case Z2: QUICK scheme [112], Fig. 6.5. The QUICK scheme delivers a transported field that at least has some recognisable features. The field shows significant oscillatory behaviour and unboundedness, however, resulting from the 'downwinding' introduced by the extension of the computational stencil in the downstream direction. In addition numerical diffusion has caused a reduction in the maximum scalar gradient, has started to fill in the slot through the centre of the disc, and has attenuated the bridge connecting the two halves of the disc. The solution is asymmetric: the point of inflection in the location of the central slot is removed by several cells from the centreline, and a prominent trough has developed on the upwind side of the scalar field.

Case Z3: TVD scheme with Osher limiter [125] ($\beta = 1.0$), Fig 6.6. The application of the Osher flux limiter with $\beta = 1.0$, which is equivalent to the MinMod limiter, produces the most dissipative behaviour of all TVD schemes. This flux limiter follows the lower edge of the second-order TVD region (Fig. 4.21, §4.5), and is therefore most heavily biased towards the upwind cell of the computational stencil. Although some evidence of the central slot remains it has been almost entirely filled in, with a scalar value at the centreline of around 0.6. In addition the maximum scalar value has dropped to approximately 0.7, so that the slot is now just a shallow depression. The scalar

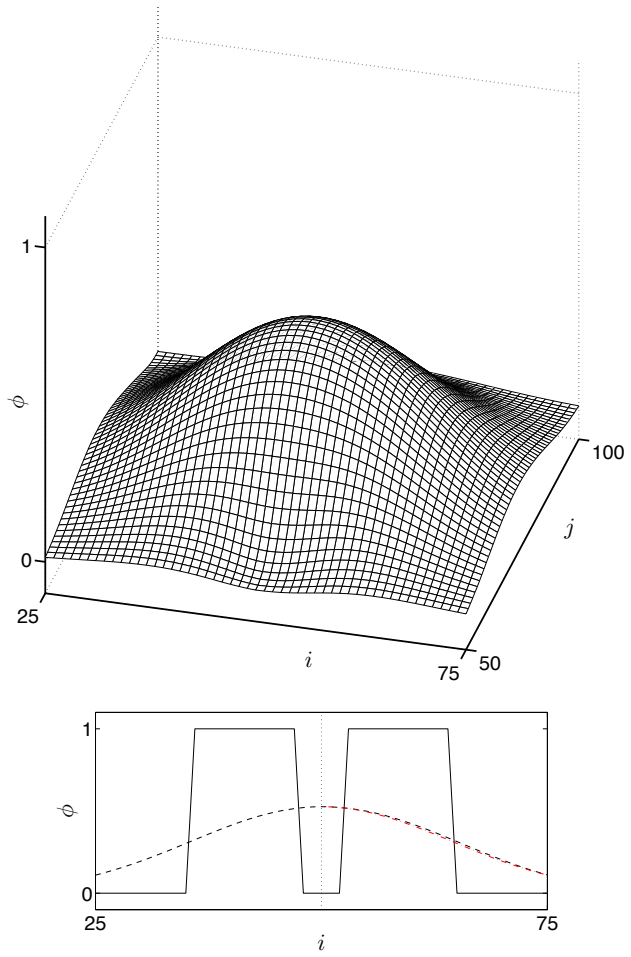


Figure 6.4: Case Z1: UDS.

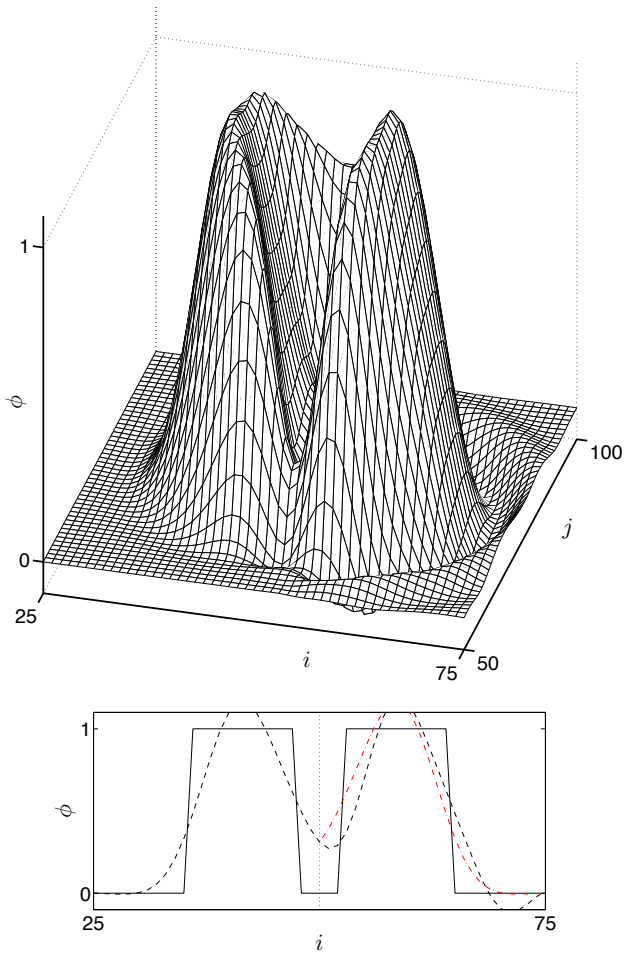


Figure 6.5: Case Z2: QUICK scheme.

gradient around the outer edge of the disc has decreased significantly, although the transported field is largely symmetric.

Case Z4: TVD scheme with Osher limiter [125] ($\beta = 2.0$), Fig 6.7. The Osher limiter with $\beta = 2.0$ exhibits similar properties to the previous case ($\beta = 1.0$), where evidence of the original shape may be observed, but with an overall attenuation of peak scalar values and a filling-in of the central slot. The effect of following the upper edge of the second-order TVD region beyond $r = 1$ is apparent; however the increased bias towards the downwind cell results in less numerical diffusion, where the peak value has dropped to around 0.8, and a steeper scalar gradient at the edges of the disc. The increased downwinding has also caused the transported field to become asymmetric, with the right-hand (upwind) half of the disc having a lower peak and steeper face than the left-hand half.

Case Z5: TVD scheme with CHARM limiter [130], Fig. 6.8. The modified version of the CHARM limiter (described in §4.5) is used here, where B is limited to a maximum $B(r) = 2$. The CHARM limiter appears to combine the advantages of the Osher limiters shown previously; numer-

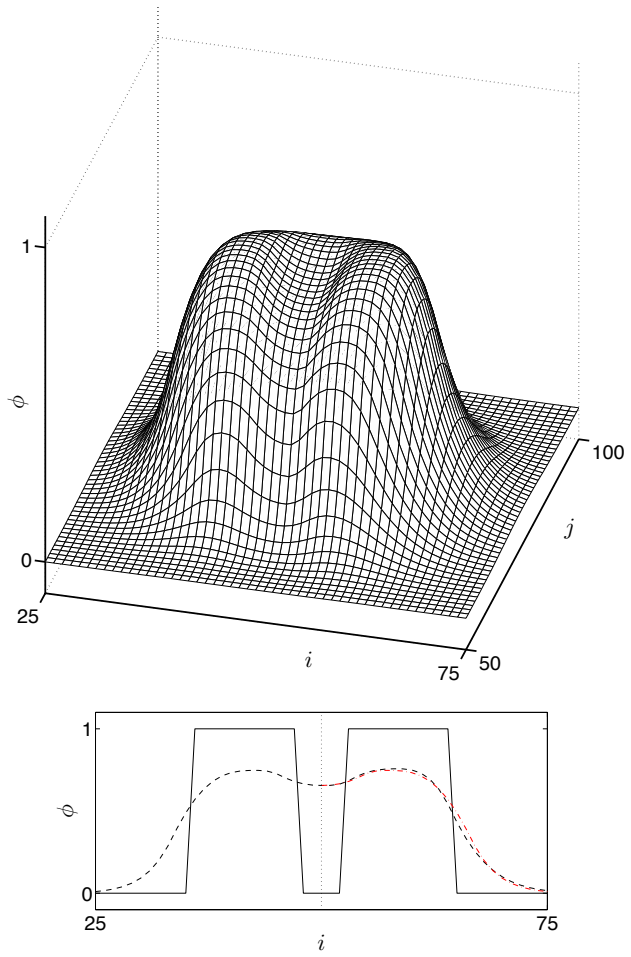


Figure 6.6: Case Z3: Osher TVD ($\beta = 1.0$).

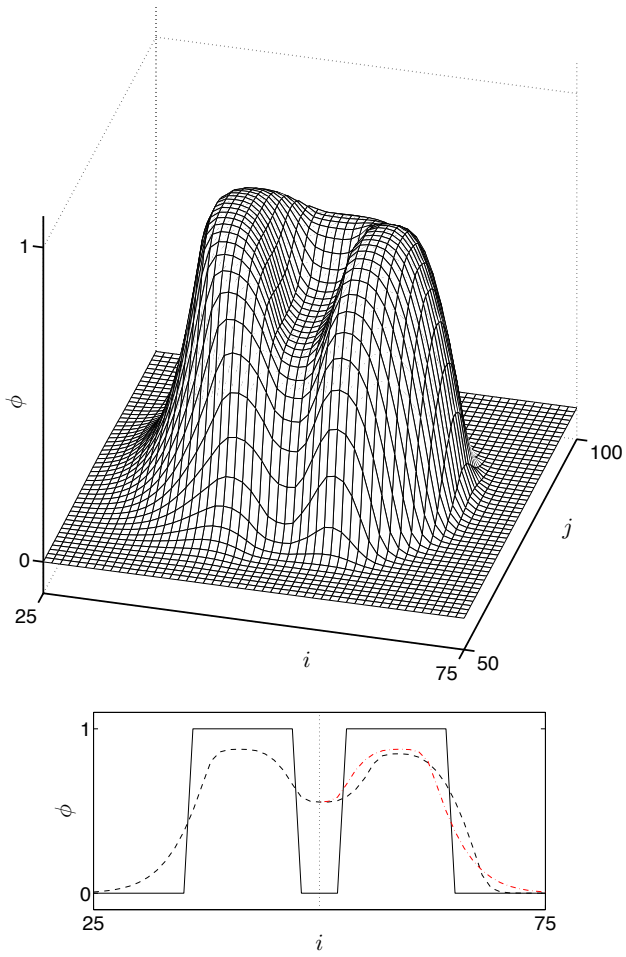


Figure 6.7: Case Z4: Osher TVD ($\beta = 2.0$).

ical diffusion is reduced, while the symmetry of the transported function is maintained. However, numerical diffusion is still significant: the peak scalar value has dropped to around 0.9, the central slot is filled to around 0.6, and the gradient of the scalar field at the edge of the disc is reduced.

Case Z6: TVD scheme with SuperBee limiter [123], Fig. 6.9. SuperBee represents the best choice of the second-order TVD schemes for a convection-only problem such as the Zalesak disc, because numerical diffusion is limited as much as possible by following the upper edge of the second-order TVD region (for any value of r). Inspection of the transported field presented in Fig. 6.9 shows that the overall shape of the disc is well-preserved. Most notably the maximum scalar value has not dropped below 1.0, although the central slot has been filled in up to a value of around 0.3 and the bridge connecting the two halves of the disc shows some attenuation. The solution is also asymmetric: the leading (left-hand) face has a relatively sharp step at its base, but is smoother at the top; the trailing face, however, has a sharp top edge and a smoother base. Compared to the results presented previously the scalar gradient around the edge of the disc is most accurately recovered by this scheme. However, this gradient is still reduced compared to that of the original field.

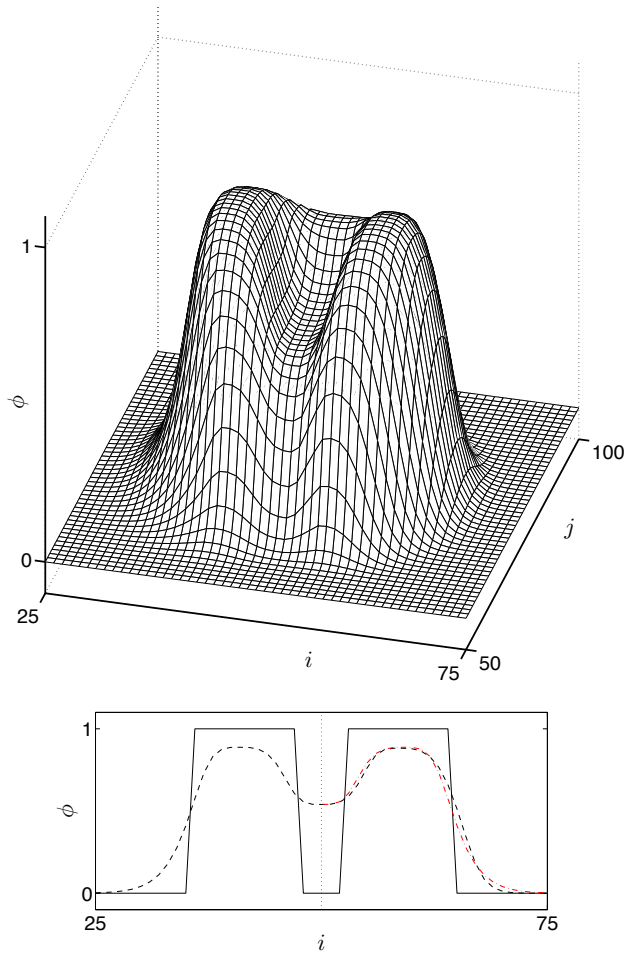


Figure 6.8: Case Z5: CHARM TVD.

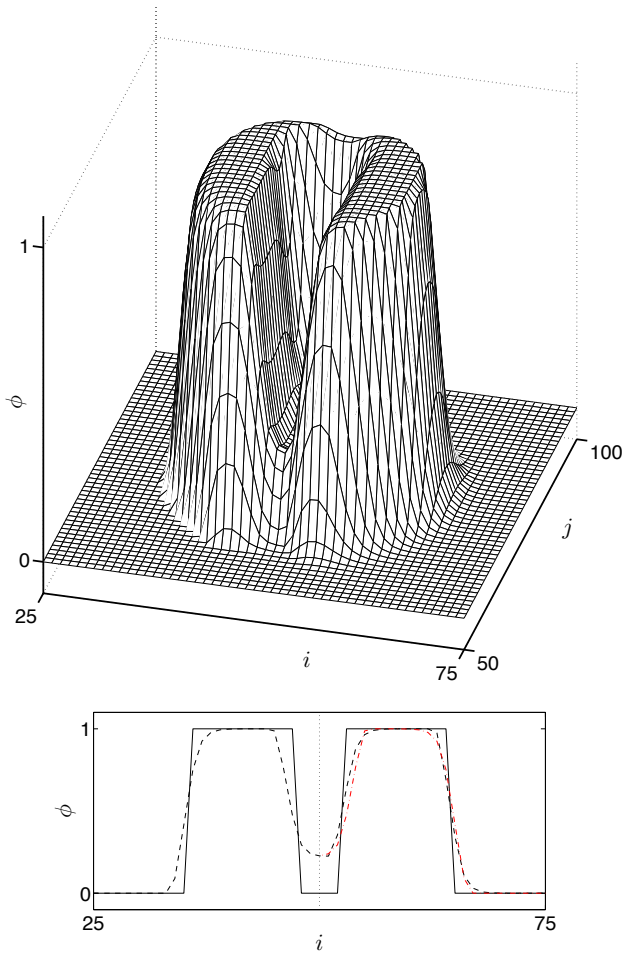


Figure 6.9: Case Z6: SuperBee TVD.

6.4.2 Eulerian-Lagrangian Simulations

Case Z7: E-L method with CDS2, $\mathbf{L^{\varrho} = 1000}$, Fig. 6.10. The most notable feature of the transported field for this case is the significant asymmetry in the solution. Given that the sharp gradient around the edge of the disc is well-described, and that the Lagrangian contribution to the solution will be highest here, it may be surmised that the error causing this asymmetry comes from the transport of the Eulerian phase. This is supported by the observation of oscillations around the base of the disc, which, firstly, extend beyond the region within which Lagrangian particles are added, and secondly, represent a behaviour expected from a Central Differencing Scheme.

This suggestion is verified by inspection of the Eulerian and Lagrangian phases individually, as shown in Figs. 6.11 and 6.12. Although the shape and magnitude of each field is modified by re-initialisation over the course of the simulation (as apparent from the decreased low-frequency scalar values, with corresponding increases in the Lagrangian field) the low-frequency field has developed a more prominent peak on the downwind side, while the particle reconstruction remains symmetric.

The sharp edges of the disc are well recovered, and artefacts of numerical diffusion appear to have been mostly removed. The central slot in the disc is preserved, and the bridge connecting the two

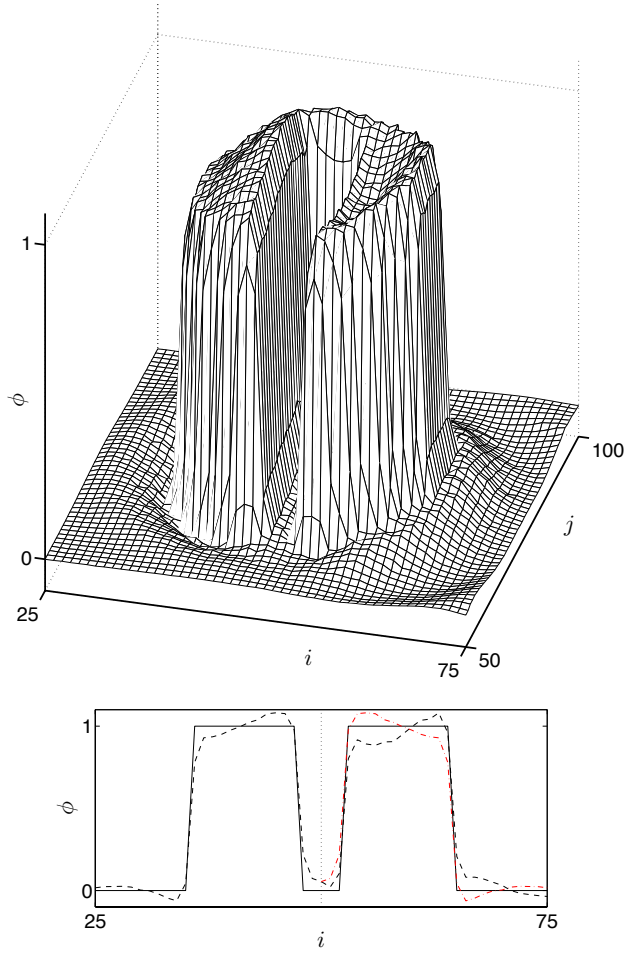


Figure 6.10: Case Z7: E-L solution applying CDS2 for the Eulerian phase.

halves of the disc shows no attenuation. However, the solution is unbounded by 10% of the initial range, i.e. the local minima and maxima are around -0.1 and 1.1 respectively. Some noise in the field is also apparent, and by comparing Figs. 6.11 and 6.12 this can be seen to derive from the Lagrangian particle reconstruction. The source of this noise is related to the point-like nature of the particles, and has previously been discussed in §5.4.3.

Case Z8: E-L method with CDS8, $\mathbf{L^{\varrho} = 1000}$, Fig. 6.13. The use of eighth-order CDS for the Eulerian phase proves to be considerably more accurate, where the asymmetry prevalent in Case Z7 is almost completely eliminated. The description of the disc edges is good, and the central slot and connecting bridge are well recovered as before. However, the solution is still unbounded by around 4%, while the noise from the particle phase remains. A comparison of Figs. 6.10 and 6.13 indicates that the selection of a high-order CDS is essential for the accuracy of the E-L method.

A feature of the solution that becomes apparent here is the 'chamfer' around the top and bottom edges of the disc face. By inspection of Fig. 6.12 it would appear that this artefact is introduced into the Lagrangian phase, and it is suggested that this occurs during the reconstruction of the high-frequency field from the particle distribution. The fact that the chamfer is also apparent in

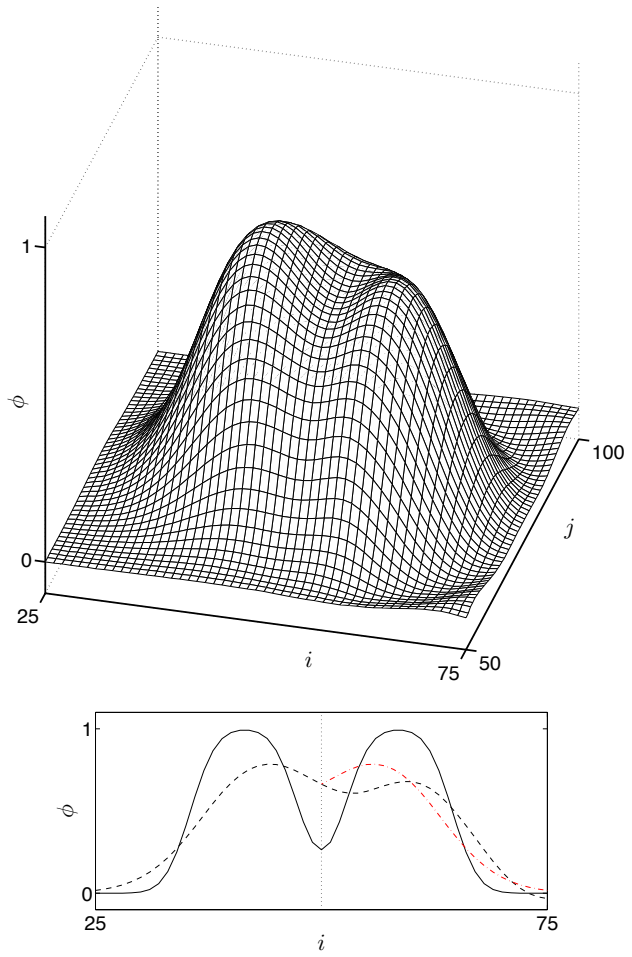


Figure 6.11: Case Z7: Eulerian component.

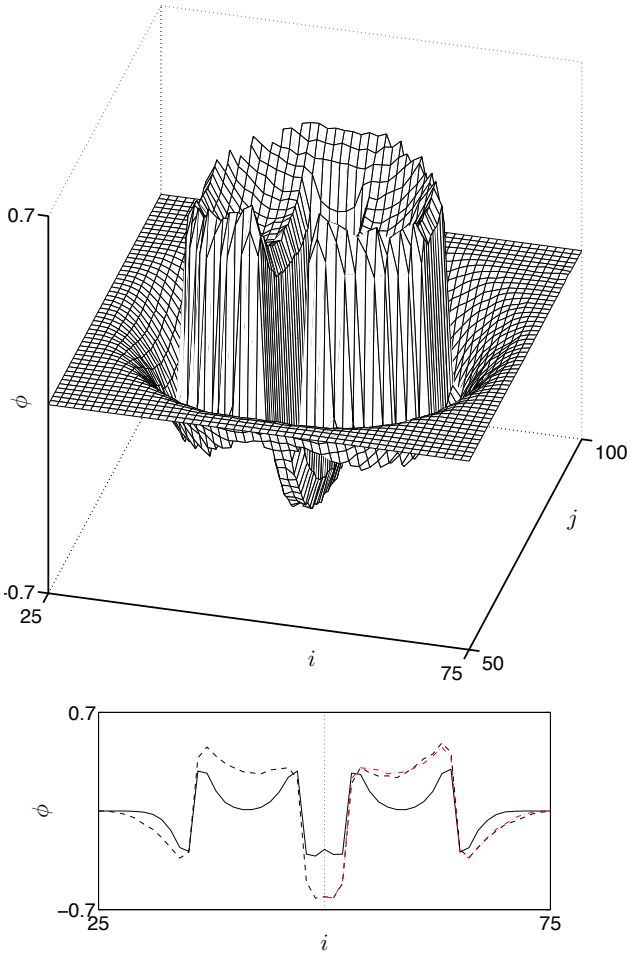


Figure 6.12: Case Z7: Lagrangian component.

the initial decomposition (Fig. 6.3) supports this suggestion, as it shows that the artefact is not related to the transport of the particle field.

Case Z9: E-L method with CDS8, $L^{\varrho} = 100$, Fig. 6.14. Finally the effect of reducing the Lagrangian particle density L^{ϱ} is considered, where the number of particles per cell is reduced by a factor of 10. These results indicate that a reduced particle density translates into a poorer description of the high-frequency field, as shown by the increase in the amplitude of the noise in the transported field. However, the edges of the disc are still well-recovered. This test is significant as it illustrates the reduction in accuracy that may be expected when the E-L method is applied for a three-dimensional case, where the particle density will need to be reduced to maintain a comparable computational cost between the Eulerian and E-L methods for convective transport.

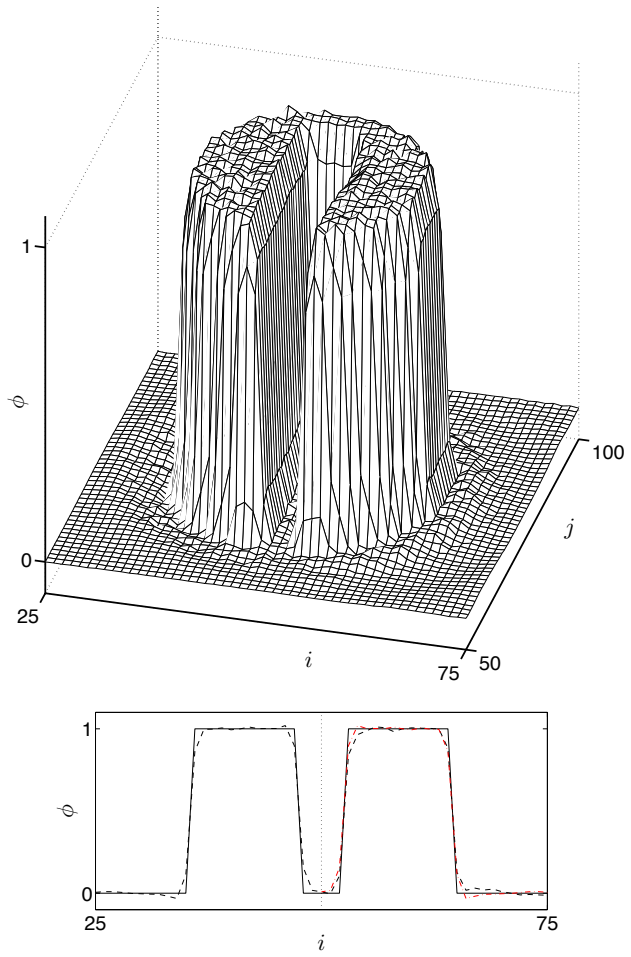


Figure 6.13: Case Z8: E-L method with CDS8, $\mathrm{L}^{\varrho} = 1000$.

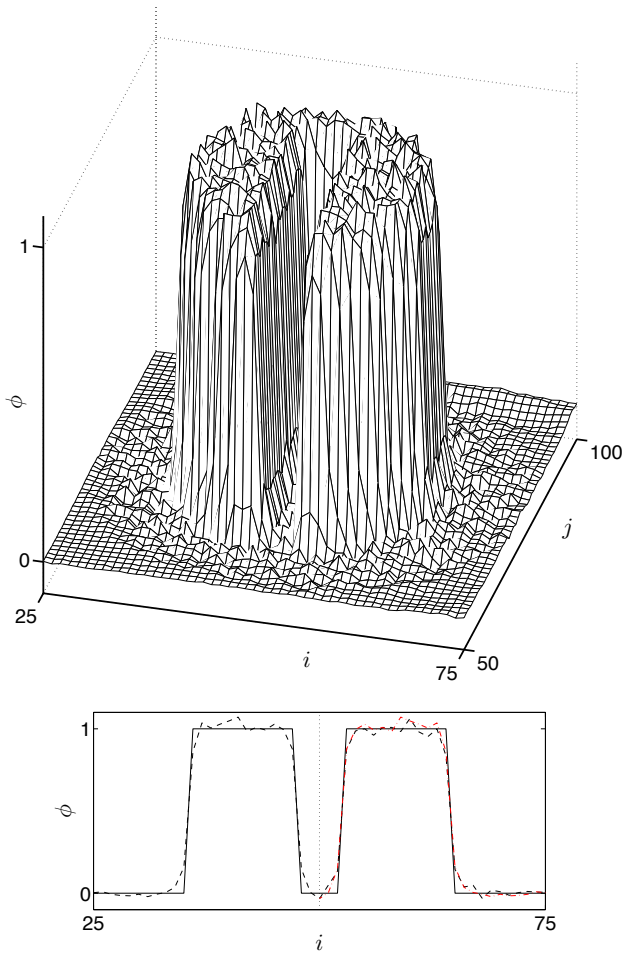


Figure 6.14: Case Z9: E-L method with CDS8, $\mathrm{L}^{\varrho} = 100$.

6.4.3 Tabulated Results

Table 6.2 presents the metrics of boundedness (ΔB_L and ΔB_U, expressed as percentages), total error E, change in total variation ΔTV, asymmetry E_{sym} and computational cost (execution time, in seconds), for Cases Z1 to Z9. The number of Lagrangian particles at the end of each E-L simulation is also included.

Two sets of data are provided for the boundedness and total error measurements from Cases Z7 to Z9. The first set of data for each E-L case compares the transported solution to the initial solution of Fig. 6.2, and may be taken as the true measurement of accuracy. The second set, in italics, compares the transported solution to the initial E-L summation for the corresponding configuration, and therefore represents how much additional transport error has been introduced into the scalar field. The initial summations are labelled 'EL1K', corresponding to a particle density $\mathrm{L}^{\varrho} = 1000$, and 'EL1C' for the lower particle density $\mathrm{L}^{\varrho} = 100$.

Measurements of boundedness and total error are presented as bar charts in Fig. 6.15, to provide a visual indication of the performance of each method.

Table 6.2: Measurements of accuracy and cost for Cases Z1 to Z9.

Case	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)	Time (s)
Z1	0.00	46.72	683.90	-94.54	9.39	-	7.79
Z2	-14.44	-19.72	278.21	27.21	97.53	-	7.11
Z3	0.00	24.25	421.92	-70.47	11.42	-	7.56
Z4	0.00	12.39	336.72	-54.91	44.19	-	7.14
Z5	0.00	11.15	317.88	-52.87	16.45	-	7.62
Z6	0.00	0.01	143.46	-11.46	20.88	-	7.10
EL1K	-2.22	-2.26	43.09	9.39	2.73	2.742	-
Z7	-7.09 *-4.87*	-10.32 *-8.06*	103.71 *78.81*	41.12	70.87	4.940	100.92
Z8	-4.52 *-1.87*	-2.40 *-0.42*	60.65 *25.96*	37.46	21.37	4.813	106.89
EL1C	-5.01	-4.65	47.89	27.20	9.26	0.274	-
Z9	-4.99 *0.03*	-7.12 *-2.47*	76.52 *39.92*	90.15	32.05	0.485	15.52

Boundedness The measures ΔB_U and ΔB_L show that the UDS and TVD schemes remain bounded, where a TVD scheme with the SuperBee flux limiter exhibits the best performance. The solutions for the QUICK scheme, and for all of the E-L simulations, are unbounded; however, for the E-L method, unboundedness decreases when the order of accuracy of the CDS used to transport the Eulerian field is increased. Reducing the Lagrangian particle density causes increased unboundedness from 6.9% at $\mathrm{L}^{\varrho} = 1000$ to 12.1% at $\mathrm{L}^{\varrho} = 100$ (over the entire range of scalar values, i.e. adding ΔB_U and ΔB_L), due to the increased amplitude of noise in the particle reconstruction. The initial decomposition of the scalar into its low- and high-frequency components causes some unboundedness in itself (4.5% at $\mathrm{L}^{\varrho} = 1000$, and 9.7% at $\mathrm{L}^{\varrho} = 100$). For the E-L implementations with eighth-order CDS, this initial amount is more significant than the increase in unboundedness caused by transport (2.3% at $\mathrm{L}^{\varrho} = 1000$, and 2.4% at $\mathrm{L}^{\varrho} = 100$).

Total Error Total error measures show that, typically, the more numerically-diffusive convective schemes suffer from greater discrepancies between the initial and transported fields, and that the SuperBee flux-limited TVD scheme (Case Z6) performs best out of the Eulerian schemes. However, in terms of total error all of the E-L implementations perform better than the SuperBee scheme. Increasing the order of accuracy of the CDS used for the low-frequency field reduces total error; with CDS8 the error is less than 40% of that from Case Z6, and remains lower even at a reduced Lagrangian particle density. It can be seen that the initialisation error for the E-L implementations is typically larger than the transport error. It is suggested that the major contributors to the total error in Cases Z7 to Z9 are the noise in the particle reconstruction and the 'chamfer' around the edge of the disc, neither of which are likely to be significantly changed during transport.

Change in Total Variation All of the TVD schemes live up to their name, as total variation diminishes for Cases Z3 to Z6. The same may be said for UDS, which is essentially a first-order

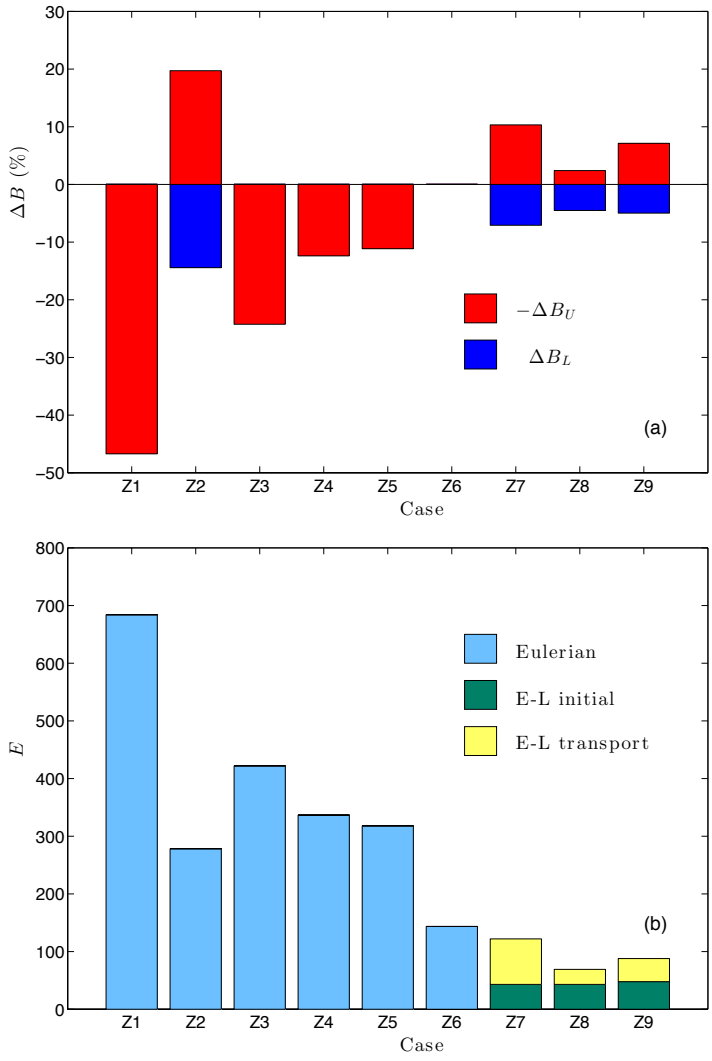


Figure 6.15: A visual representation of boundedness (a) and total error (b) for Cases Z1 to Z9.

TVD, but not for QUICK, whose path across the Sweby diagram shown in Fig. 4.21 only falls inside the second-order TVD region in the range $0.2 \leq r \leq 7/3$. All of the E-L solutions suffer from an increase in total variation. It would seem that the measure is particularly sensitive to noise in the particle field, as the noisier solution (where the Lagrangian particle density is reduced) has the greatest ΔTV.

Symmetry Numerical diffusion is found to correlate with the symmetry of a transported solution, as the more diffusive Eulerian schemes exhibit the lowest asymmetry. The SuperBee scheme also performs relatively well due to its reasonable recovery of the correct scalar shape. The symmetry of the E-L solutions is improved when the accuracy of the Eulerian convective transport scheme is increased, but is limited by the inherent noise in the Lagrangian phase. The asymmetry increases as the Lagrangian particle density is reduced.

Computational Expense The most significant characteristic of the execution times for Cases Z1 to Z9 is the order of magnitude difference between the Eulerian schemes and the E-L implementations with $L^{\varrho} = 1000$. This is accounted for by the additional computational cost of transporting the Lagrangian particles, which require a bilinear interpolation of the velocity field for every particle at every time-step (or, more specifically for the Runge-Kutta scheme used here, every sub-time-step). The effect of decreasing the particle density is clearly to reduce the total number of Lagrangian particles required, and thus the computational cost; Case Z9 is therefore only twice as expensive as an Eulerian simulation. Using the data from Cases Z8 and Z9 and assuming a linear dependence of computational cost on the number of particles (as shown in Fig. 6.16), the additional particle cost is found to be around 0.33 CPU-seconds per time-step per million particles with an intercept for zero particles of 5.47 seconds. This intercept agrees well with the Eulerian-only simulation run-times, considering that the CDS applied for the Eulerian field in an E-L simulation will be computationally cheaper than the TVD schemes used in the Eulerian-only cases. Furthermore this trend indicates that the fixed overheads (i.e. initialisation routines) associated with the E-L method are of minimal cost, although the expense of the initial particle distribution routine will scale with the number of particles being distributed. Finally it can be seen that the application of CDS8 in an E-L simulation increases cost by around 6% compared to an E-L simulation with CDS2.

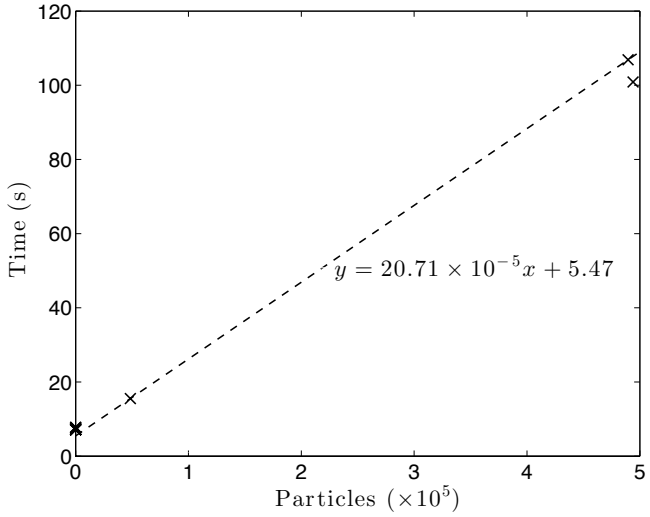


Figure 6.16: The additional computational cost of Lagrangian particles.

The standard deviation in timings is determined to provide an indication of the uncertainty in the computational cost measure. A number of repeated simulations are performed over different simulation durations, from which an appreciable fluctuation in the execution times is found. For E-L simulations the variation is found to be between 2% and 4% of the mean execution time, where larger fluctuations are observed for shorter executions; for Eulerian simulations, however, more significant fluctuations of 10% to 11% are recorded. To account for this discrepancy it is suggested that the Lagrangian part of the computation, which consists of a few simple calculations repeated for a large number of particles, is more consistent in timing than the Eulerian part, which constitutes a more complex operation repeated over a relatively small number of cells.

6.5 Conclusions

The Zalesak disc has been used in this part of the study to examine the strengths and weaknesses of the proposed joint Eulerian-Lagrangian method. A framework of metrics has been suggested and used to quantify the accuracy of different convective transport schemes. Graphical results have been included to illustrate how each scheme behaves, and this qualitative analysis supports the quantitative metrics.

The well-known poor behaviour of existing Eulerian convective schemes has been highlighted in this investigation. First-order accurate UDS introduces a large amount of numerical diffusion, which renders it useless for most practical problems. The third-order QUICK scheme causes oscillatory behaviour, potentially resulting in numerical instabilities. Second-order TVD schemes tend to fare better; however, the Osher limiter is numerically diffusive and will cause increasing asymmetry in the results as β is increased; the CHARM limiter is better than the Osher limiter (for any β) but is still diffusive; and the SuperBee limiter provides the best possible results for the Zalesak disc (of the Eulerian schemes), but still exhibits some undesirable behaviour.

The same analysis applied to the E-L method demonstrates the importance of using a high-order CDS for the transport of the Eulerian field: CDS2 tends to induce oscillations in the low-frequency scalar field, which advocates the use of an eighth-order scheme. Computational cost is only slightly affected by the associated increase in the required computational stencil (although some additional communication overheads would also be expected for simulations performed on a parallel computing architecture). Higher-order CDS would be simple enough to implement, but the eighth-order scheme has been accurate enough for the purposes of the present work. In general it is beneficial to keep the second derivative of the low-frequency scalar field as low as possible to prevent the development of oscillations. However, this typically demands the addition of more Lagrangian particles, which increases the computational cost of the method and exacerbates the error introduced by noise in the reconstructed particle field.

The E-L method at a high Lagrangian particle density $\mathrm{L}^{\varrho} = 1000$ proves to be around an order of magnitude more expensive than a traditional Eulerian scheme, although this increased computational cost may be mitigated by reducing L^{ϱ} at the cost of an associated reduction in accuracy.

The present study has provided useful insight into the performance of the proposed joint Eulerian-Lagrangian method, although it only tests the ability of the method to deal with the translation of a pre-defined scalar topology. In the next stage of the study some modifications are made to the Zalesak case to provide a more rigorous test of the strengths and limitations of the method, as described in Chapter 7.

7 Test Case 2: Zalesak Modified

The study of the Zalesak disc presented in Chapter 6 has provided an indication of the potential of the joint Eulerian-Lagrangian (E-L) method for the improved convective transport of a scalar. However, the Zalesak case applies a very specific set of conditions, which are unlikely to be found in real turbulent flows. The next stage of the investigation is to apply a more generalised test configuration: the performance of the new method will now be considered under a variety of physical (varying Schmidt number) and computational (varying time-steps to solution) conditions, to assess its applicability to practical Direct Numerical Simulation (DNS) and Large-Eddy Simulation (LES) problems.

7.1 Case Modification

Previous work on numerical schemes, including that of Shu and Osher [68, 208] and Zalesak [184], has often focused on the convective transport of interfaces and discontinuities (e.g. shocks). This has resulted in convection schemes tailored to the transport of sudden ('step'-like) variations in a scalar field. It has been shown in Chapter 4 that some schemes (for example, the SuperBee Total Variation Diminishing (TVD) scheme) perform poorly for smooth cases, where gradients become artificially sharpened (over-compressed). The E-L method, however, should be capable of transporting scalars that vary rapidly or gradually, and a new test case is formulated to represent this in the present study.

A modified version of the Zalesak case is therefore used: the sharp outer edge of the Zalesak disc is replaced with a Gaussian error function profile, while the 'slot' is retained; a finite Schmidt number is applied, to yield a non-zero (physical) diffusive flux; and the overall dimensions of the simulated domain are extended to 200×200 cells. The domain size is increased to ensure that the region of interest (which remains centred on a position 25 cells from the mid-point) is still sufficiently removed from the boundaries, given that the scalar field is now expected to spread due to diffusion. The velocity field is unchanged. The scalar field for this new case is shown in Fig. 7.1, where the view shown is limited to 50×50 cells centred on the mid-point of the disc.

The investigation of the modified Zalesak case is divided into three configurations. Initially the scalar field is transported for one revolution with a small amount of physical diffusion (high Schmidt number), to assess the performance of different convective schemes for this new topology. This is followed by a test of the same schemes for a single revolution but with a lower Schmidt number, to verify the applicability of the E-L method for a wider range of physical conditions. Finally the modified disc is transported for 100 revolutions, to reveal the performance of each scheme over a longer period of time. In this latter case a high Schmidt number is applied as the extended residence

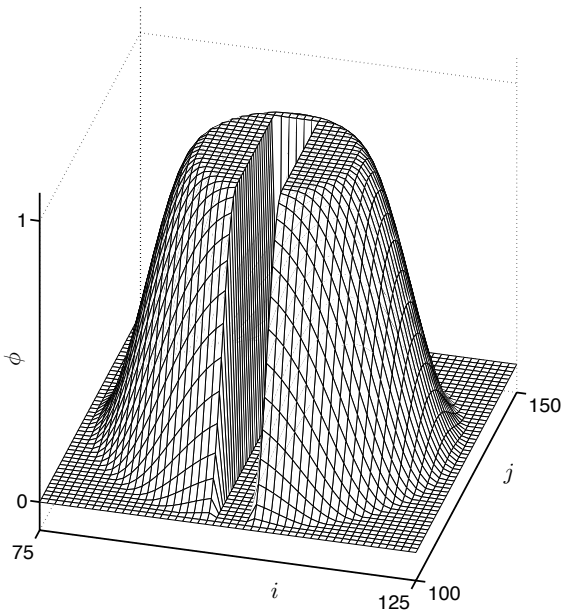


Figure 7.1: Modified Zalesak disc: Initial conditions

time of the scalar would otherwise result in the transported field approaching homogeneity. Each configuration is computed using one of two Eulerian TVD schemes, or the E-L method at high or low particle density. The cases studied here are summarised in Table 7.1.

Table 7.1: Numerical schemes tested with the modified Zalesak disc.

Case	Method	Eulerian Scheme	Sc	No. of Revs.	Particle Density, L^{ϱ}
ZM1	Eulerian	CHARM TVD	10	1	-
ZM2	Eulerian	SuperBee TVD	10	1	-
ZM3	E-L	CDS8	10	1	1000
ZM4	E-L	CDS8	10	1	100
ZM5	Eulerian	CHARM TVD	1	1	-
ZM6	Eulerian	SuperBee TVD	1	1	-
ZM7	E-L	CDS8	1	1	1000
ZM8	E-L	CDS8	1	1	100
ZM9	Eulerian	CHARM TVD	100	100	-
ZM10	Eulerian	SuperBee TVD	100	100	-
ZM11	E-L	CDS8	100	100	1000
ZM12	E-L	CDS8	100	100	100

The extended size of the computational domain means that there is no longer a need to restrict the width of the Gaussian filter kernel used for the initial Eulerian-Lagrangian decomposition. The filter width is therefore increased to the value of $\sigma_f = 5\Delta$ suggested in §5.3.1, while the re-initialisation constant γ appearing in Eq. 5.60 is set to the recommended value of $\gamma = 1$. As for the previous study of the Zalesak case, re-initialisation is performed after every four computational steps. The time-step width for the new case is reduced to $dt = 0.125$ s to maintain a CFL number of CFL = 0.5,

resulting in one revolution requiring 1256 computational steps.

Simulation numerics are otherwise the same as for the original Zalesak case, described in §6.3. Eulerian diffusive fluxes are added to convective fluxes using an eighth-order accurate approximation of scalar gradient, while Lagrangian diffusion is approximated using a Wiener process for each particle (see Chapter 5). As a unit cell size is assumed for this case, an arbitrary laminar viscosity of $\mu = 0.1$ kg/m/s is chosen so that the applied Schmidt numbers are comparable to those expected for realistic gaseous ($Sc \approx 1$) and liquid ($Sc \approx 100$) mixing problems.

7.2 Results and Discussion

An analysis of the performance of each numerical convective scheme is carried out using the same methods employed for the original Zalesak disc, as described in §6.4.

7.2.1 One Revolution, High Schmidt Number

Figure 7.2 shows the modified Zalesak disc transported for 1256 time-steps by diffusion only. This provides a reference for visual comparison with the results presented in the rest of §7.2.1, as it represents a good approximation of the analytical solution for the rotated case. The largest diffusive fluxes are found around the sharp edges of the central slot in the disc, resulting in a slight attenuation of the bridge connecting the two halves of the disc, and a small amount of in-filling of the slot itself. It is important to note that diffusive fluxes have had much less of an effect on the smoothed outer edge of the disc because of the reduced scalar gradients, which would not have been the case with Zalesak's original topology. The expected peak scalar value after one revolution is 1.0; although diffusive fluxes act to reduce local maxima, the low diffusivity of this configuration means that the initial maximum value is not reduced over the relatively short duration of the simulation. The local minimum within the central slot of the disc has been increased from zero to 0.15.

Graphical results from cases ZM1 to ZM4 are presented in Figs. 7.3 to 7.6. The scalar profile through the centre of the diffused disc illustrated in Fig. 7.2 is included (solid line) for comparison with the profile from the transported solution (dashed line).

Case ZM1: TVD scheme with CHARM limiter [130], Fig. 7.3. The behaviour of the CHARM scheme is reminiscent of its performance for Case Z5 (§6.4.1); numerical diffusion has had a significant impact on the accuracy of the description of the sharper edges of the disc, resulting in lower gradients and some in-filling of the central slot. The peak scalar value has dropped to around 0.9 (from the expected value of 1.0), while the local minimum at the centre of the disc has risen from 0.15 (expected) to around 0.55. An important observation is that the CHARM scheme has performed relatively well around the smoothed outer edges of the disc, where the overall shape is well-recovered.

Case ZM2: TVD scheme with SuperBee limiter [123], Fig. 7.4. This case reiterates the disadvantage of using the SuperBee scheme for the transport of anything other than step functions: the outer edges of the disc have become compressed, meaning that the outer face has acquired a

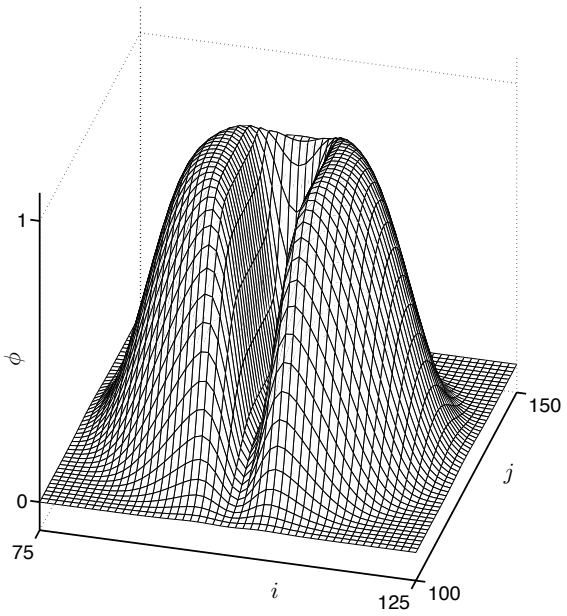


Figure 7.2: Modified Zalesak disc: The Eulerian solution applying diffusion only, $Sc = 10$.

constant (instead of a smoothly varying) gradient, and a sharper change in gradient now exists at the upper and lower scalar limits. Overall, the outer edge has gained a more stepped appearance. Numerical diffusion is still apparent, however, as the scalar value inside the central slot has increased to 0.3. An asymmetry has also developed in the transported solution, as the left-hand half of the disc is now narrower than the right-hand half.

Case ZM3: E-L method with CDS8, $\mathbf{L^{\varrho} = 1000}$, Fig. 7.5. The performance of the E-L method is now considered, where it can be seen that the scalar profile through the centreline of the disc closely matches the expected result. The regions in which particles have a significant contribution to the scalar summation, for example throughout the central slot region, are discernible from the noise introduced by the Lagrangian reconstruction. However, the overall shape of the modified disc is well preserved.

Case ZM4: E-L method with CDS8, $\mathbf{L^{\varrho} = 100}$, Fig. 7.6. A reduction in the Lagrangian particle density only appears to cause an increase in the noise in the solution, while the overall profile of the transported disc generally matches the analytical solution as well as that of Case ZM3. This perhaps advocates the addition of some method of noise removal, as the computational cost of Case ZM4 may be expected to be significantly less than that of ZM3 due to the lower particle density.

Tabulated Results Measurements of the accuracy and computational cost for Cases ZM1 to ZM4 are presented in Table 7.2, applying the boundedness, total error, change in total variation and symmetry error metrics introduced in §6.4. The most notable result to be drawn from the boundedness measures is that the noise from the Lagrangian particle reconstruction in Cases ZM3

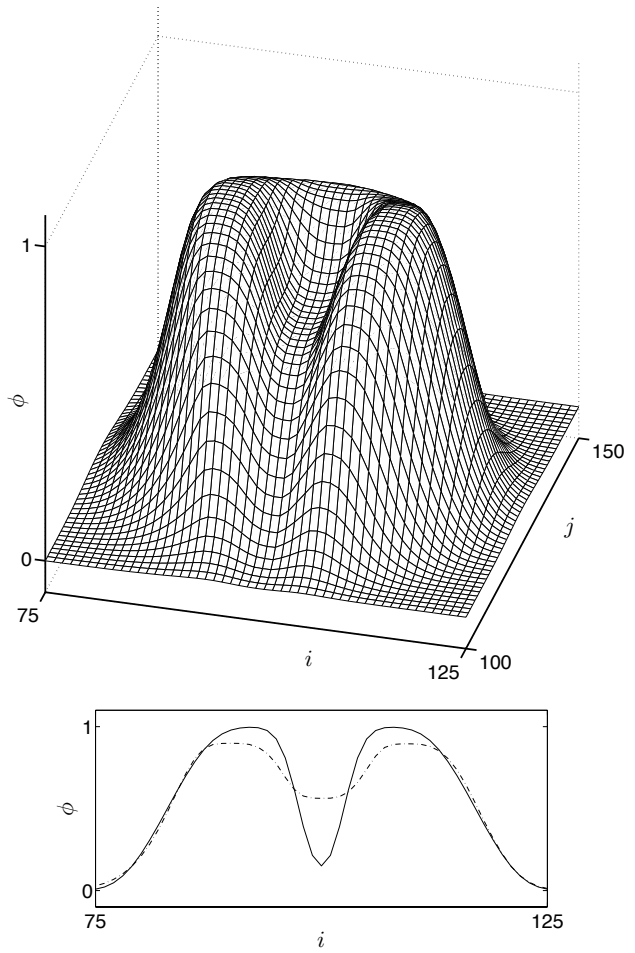


Figure 7.3: Case ZM1: CHARM TVD; $Sc = 10$.

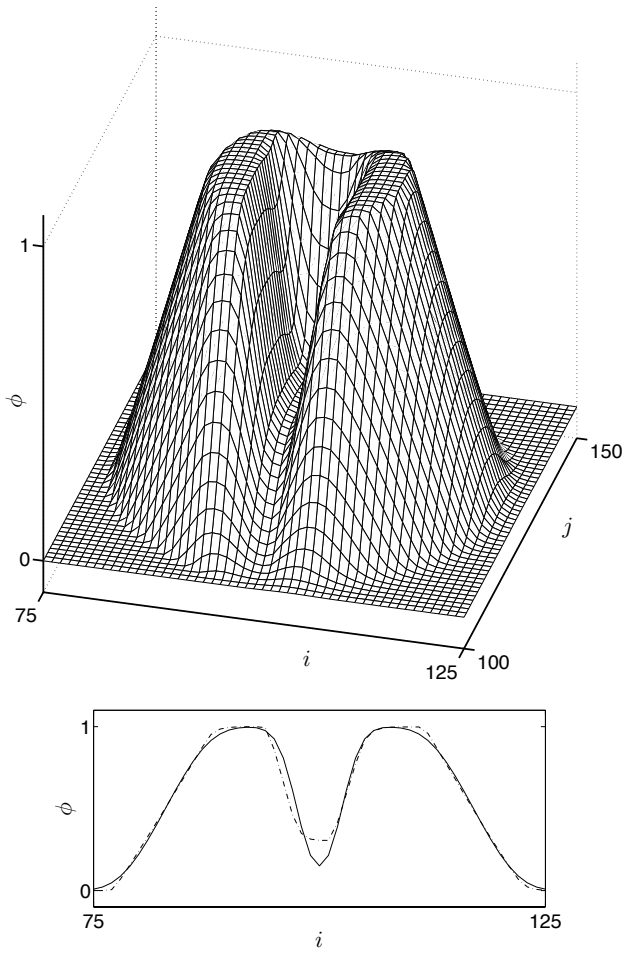


Figure 7.4: Case ZM2: SuperBee TVD; $Sc = 10$.

and ZM4 causes unboundedness in the scalar field. In terms of total error the CHARM TVD scheme applied in ZM1 performs worst due to the numerical diffusion it introduces into the solution, while the best scheme is the E-L method at the higher particle density in ZM3; notably, Case ZM4 is almost as good as the SuperBee scheme. Changes in total variation reveal that the CHARM and SuperBee schemes experience a decrease, while the noise from the particle reconstruction causes an increase for the E-L cases[1]. The symmetry error is lowest for the CHARM scheme, while the E-L method in Case ZM3 shows lower symmetry error than the Eulerian SuperBee scheme.

In terms of computational cost all cases are approximately six times more expensive than the equivalent cases for the original Zalesak disc. An increase of a factor of eight for the Eulerian-only cases, due to the increased number of cells ($\times 4$) and time-steps ($\times 2$), would be expected; however, a lesser proportional increase would be expected for the E-L method as the number of particles is similar. The observed improvement in efficiency for the Eulerian-only modified Zalesak cases

[1]The total variation of the analytical transported field is now lower than that of the original field, due to the action of diffusion. However, the change in total variation quoted here is the difference between the TV of the transported field and the TV of the analytical solution, such that any change in total variation due to physical diffusion is not included

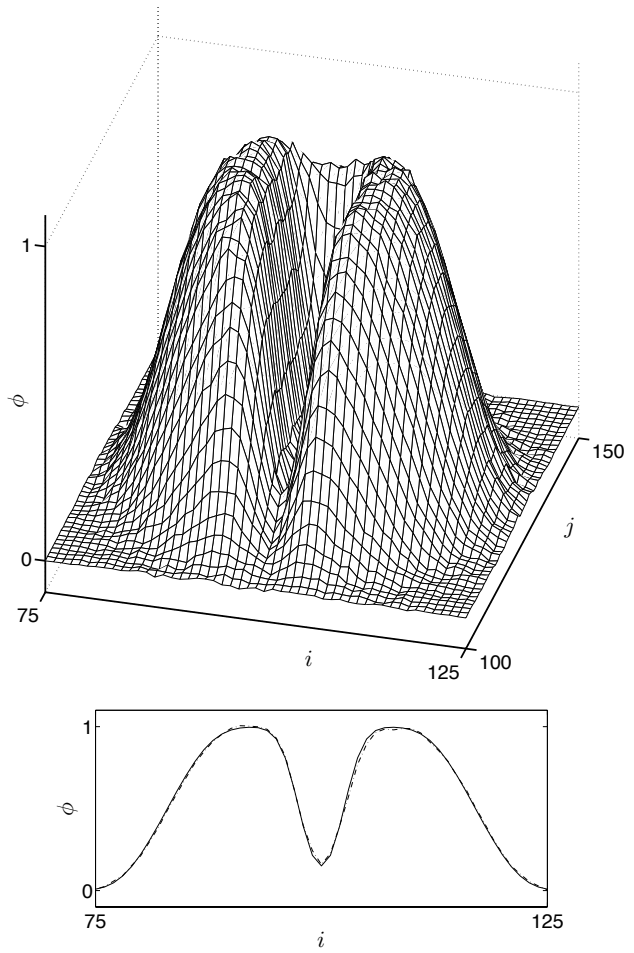


Figure 7.5: Case ZM3: E-L method with CDS8, $\mathrm{L}^{\varrho} = 1000$; $Sc = 10$.

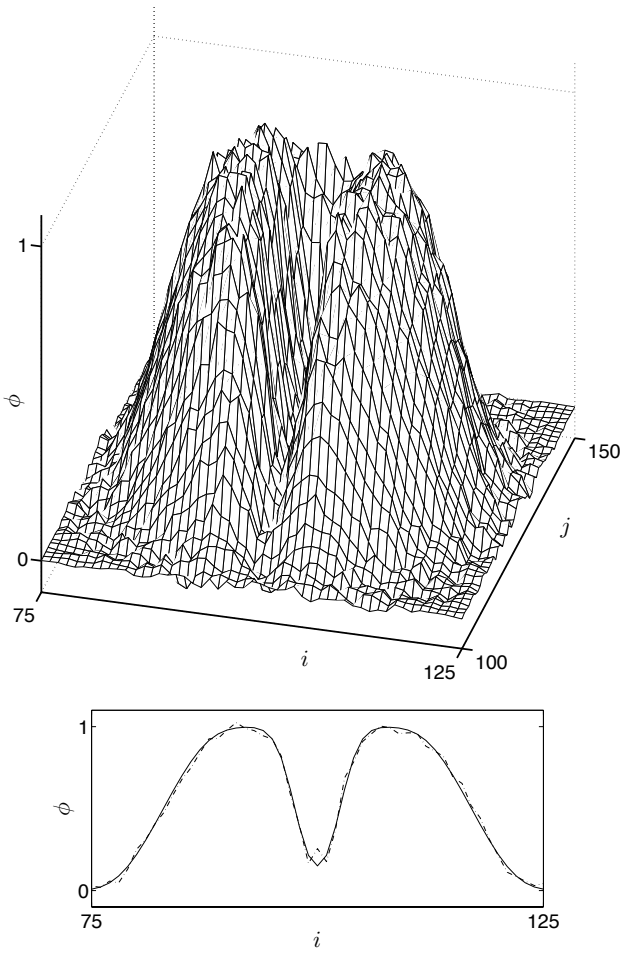


Figure 7.6: Case ZM4: E-L method with CDS8, $\mathrm{L}^{\varrho} = 100$; $Sc = 10$.

suggests that optimisation features built into the compilation or execution of the code may be playing an important part. For example, the larger Cartesian array sizes used here would be more amenable to the optimising methods of instruction pipelining, and instruction or memory prefetching (described in §5.4.1), for the Eulerian parts of the computation. However, the same advantages would not occur in the Lagrangian computations: in particular, the bilinear interpolation used to determine particle velocities is unlikely to benefit from memory prefetching at all (as memory accesses are not contiguous). Calculations associated with Lagrangian particles have become more expensive overall, as the additional cost of particles is now found to be 1.11 CPU-seconds per time-step per million particles (compared to a previous value of 0.33), assuming a linear dependence of run-time on particle numbers between Cases ZM3 and ZM4. The increased particle cost is due to the calculation of the diffusion terms appearing in Eq. 5.55, which require the density, viscosity and viscosity gradient to be interpolated on to the particle position, as well as the generation and transformation (by the Box-Muller method described in §5.4.2) of a set of random numbers for the incremental Wiener step.

A further point to note is that the specifications of the processor used for these tests state that

Table 7.2: Measurements of accuracy and cost for Cases ZM1 to ZM4.

Case	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)	Time (s)
ZM1	0.00	9.77	100.50	-36.45	15.50	-	46.73
ZM2	0.00	-0.33	48.26	-6.23	18.85	-	45.19
ZM3	-1.21	-2.38	21.37	12.02	16.00	3.778	573.17
ZM4	-5.20	-6.60	54.24	59.67	37.04	0.379	99.93

the highest level (L1) on-chip memory cache has a size of 128KB. Considering the data that must be accessed in each case: the majority of computations are performed from the two components of velocity and the transported scalar, stored in single precision (i.e. requiring 4 bytes per floating point number), meaning that the original Zalesak disc uses a total of around 120KB. This may therefore be stored entirely in the L1 cache. For the larger modified Zalesak case, however, the three scalar fields require 480KB of storage, so that the majority of data must reside in the (slower, off-chip) L2 memory cache. This would imply that the Eulerian and Lagrangian parts of the computation should both be more expensive for the latter case, due to the added cost of reading from the L2 cache.

7.2.2 One Revolution, Low Schmidt Number

Figure 7.7 shows the modified Zalesak disc transported by diffusion only, for 1256 time-steps, at the lower Schmidt number of $Sc = 1$. The effect of the increased diffusive flux is clearly apparent, as the sharp features of the disc have been largely dissipated. Only a minor depression remains to show the position of the central slot, where the minimum scalar value is now 0.65, and the outer edges of the disc have become less steep. In addition, the peak scalar value has dropped to around 0.78.

Case ZM5: TVD scheme with CHARM limiter [130], Fig. 7.8. It can be seen that the performance of the CHARM scheme is consistent with that of the previous Cases Z5 and ZM1; generally, the total diffusive flux has been over-estimated due to a significant amount of numerical diffusion, resulting in an under-prediction of the left- and right-hand side peaks (by about 3%), and an over-prediction (also approximately 3%) of the scalar value at the centre of the disc. The degree of error in the description of the transported scalar field is reduced, but this is due to the effects of numerical diffusion being obfuscated by physical diffusion. As before, however, the outer edges of the disc are well recovered using the CHARM scheme.

Case ZM6: TVD scheme with SuperBee limiter [123], Fig. 7.9. A behaviour consistent with previous SuperBee tests is observed, although the adverse properties seen in Case ZM2 are mitigated by the increased physical diffusive flux; the compression of the gradient at the outer edge of the disc is no longer apparent. Here, however, the scheme over-predicts the peak scalar value by around 4%. It can also be seen that the transported field is asymmetric, with the local maxima and (well-predicted) minimum at the centre of the disc shifted by a few computational cells to the left.

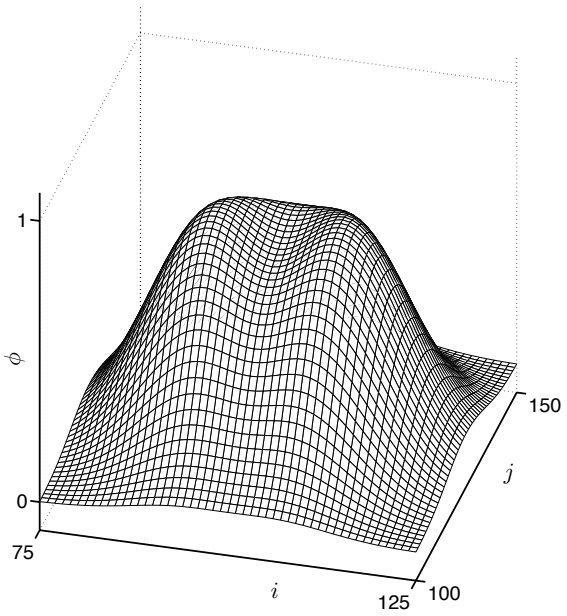


Figure 7.7: Modified Zalesak disc: The Eulerian solution applying diffusion only, $Sc = 1$.

Case ZM7: E-L method with CDS8, $\mathbf{L}^{\varrho} = 1000$, Fig. 7.10. The performance of the E-L method continues to show an improvement in comparison to that of the Eulerian schemes presented in Cases ZM5 and ZM6, as the shape of the transported disc agrees almost exactly with the expected solution. The only criticism of the method is the noise apparent in the Lagrangian phase. However, these results indicate that the E-L method is consistently capable of dealing with both low and high rates of diffusive flux.

Case ZM8: E-L method with CDS8, $\mathbf{L}^{\varrho} = 100$, Fig. 7.11. As for the high Schmidt number configuration (Case ZM4) a reduction in the number of particles seems only to have increased the amplitude of the noise in the solution, while the overall profile of the transported disc is well-maintained. This result again suggests that some form of noise reduction may permit the E-L method to deliver more accurate results at a computational cost comparable to that of the Eulerian schemes.

Tabulated Results Table 7.3 shows accuracy and cost metrics for Cases ZM5 to ZM8. Interpretation of these results largely follows that from the previous configuration discussed in §7.2.1, although the effect of the increased physical diffusion is apparent. The E-L implementation of Case ZM7 now yields the best results for boundedness, as the CHARM TVD scheme delivers excessive diffusion and the SuperBee scheme sharpens the solution. Total error is relatively low for all cases, although the E-L method is slightly less accurate than the SuperBee and CHARM schemes due to Lagrangian noise. Changes in total variation and symmetry errors are also reduced by the action of physical diffusion; however, ΔTV for the SuperBee scheme is now positive, meaning that the total variation has not decreased as much as it should have in this case (although it has diminished overall). The computational cost for these four cases follows the trends identified for the previous

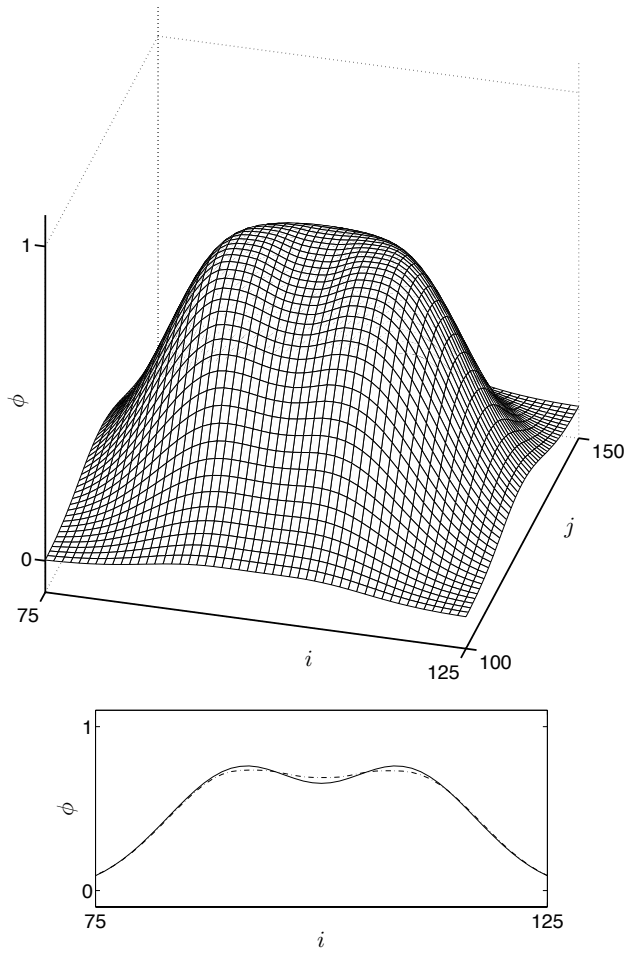


Figure 7.8: Case ZM5: CHARM TVD; $Sc = 1$.

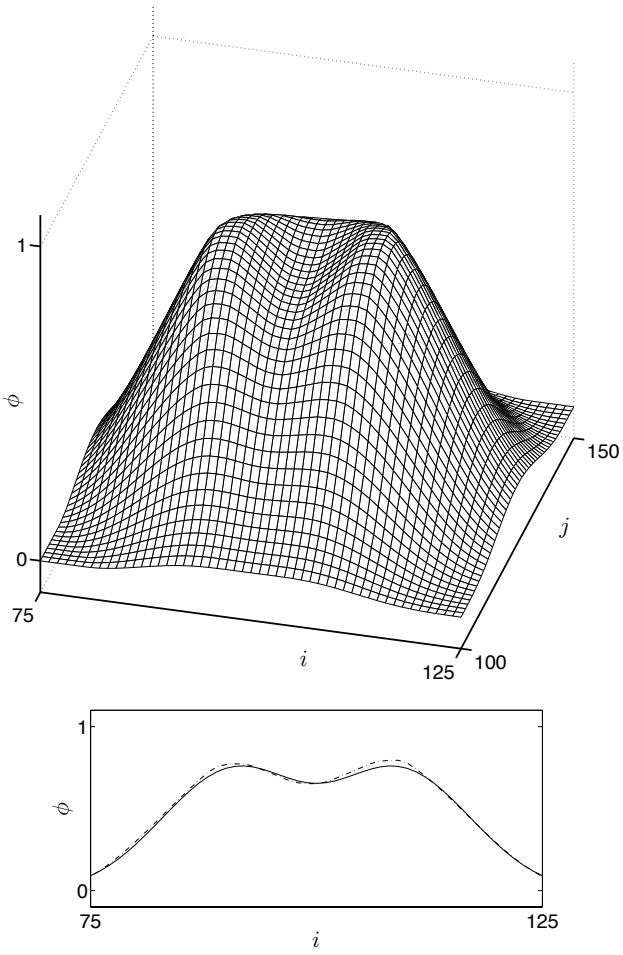


Figure 7.9: Case ZM6: SuperBee TVD; $Sc = 1$.

configuration (as evident from the data in Table 7.2), where the E-L implementation is slightly quicker than before due to a reduction in the total number of particles required. The additional cost of particles for this configuration is found to be 1.10 CPU-seconds per million particles per time-step, which is consistent with the value from the high Schmidt number configuration.

Table 7.3: Measurements of accuracy and cost for Cases ZM5 to ZM8.

Case	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)	Time (s)
ZM5	0.00	2.51	14.05	-3.09	5.89	-	43.38
ZM6	0.00	-3.36	31.36	2.58	12.97	-	42.92
ZM7	-0.42	-2.42	19.94	14.61	14.11	3.729	563.13
ZM8	-2.79	-7.50	61.23	81.96	42.64	0.369	99.25

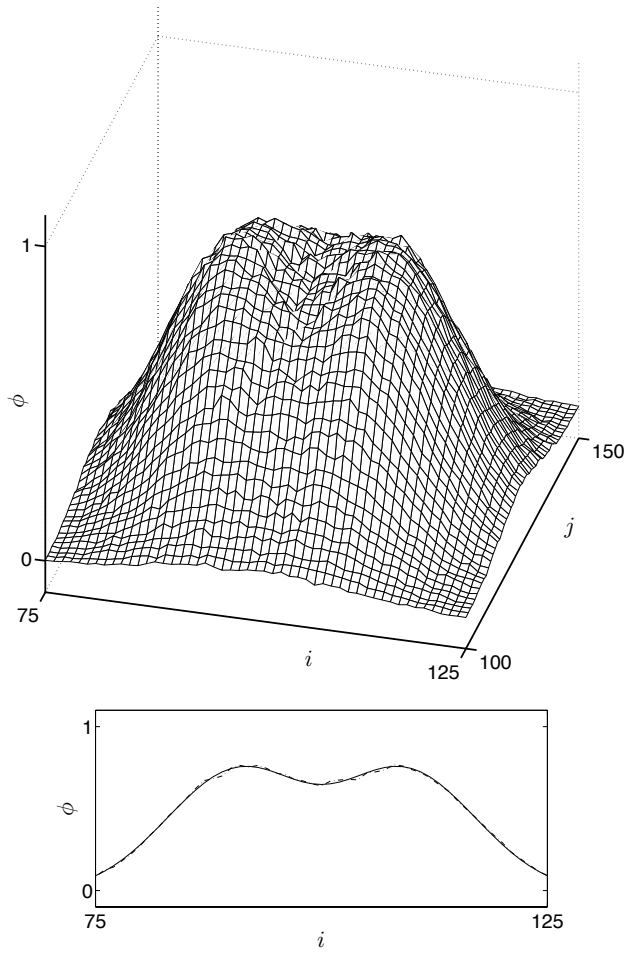


Figure 7.10: Case ZM7: E-L method with CDS8, $\mathrm{L}^{\varrho} = 1000$; $Sc = 1$.

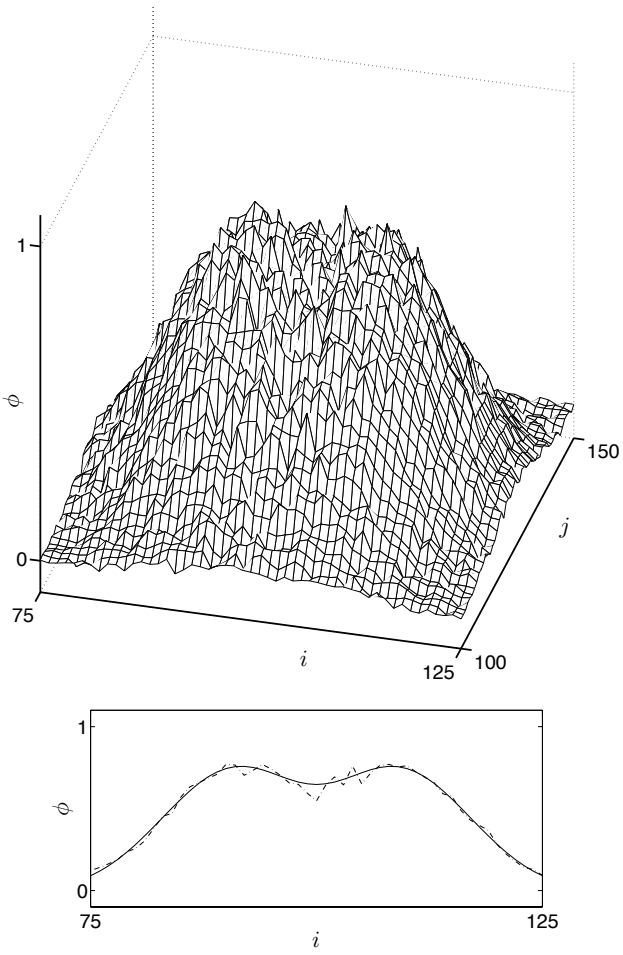


Figure 7.11: Case ZM8: E-L method with CDS8, $\mathrm{L}^{\varrho} = 100$; $Sc = 1$.

7.2.3 100 Revolutions, High Schmidt Number

The final configuration to be tested provides insight into the performance of each scheme over an extended time. This is important as a typical DNS or LES will typically require up to the order of 10^6 computational steps. In addition recirculation zones within a simulated domain may cause a scalar quantity to become trapped, and thus to have a residence time much longer than the expected flow-through time. In these cases the scalar may be affected by the numerical artefacts of the convection scheme more than fluid that passes straight through the domain with the bulk of the flow. Simulating 100 revolutions of the modified Zalesak disc will require a total of 125600 time-steps, and therefore provides a far more representative test case for practical applications.

In addition to spatial errors, the errors due to temporal discretisation may now become important. A third-order accurate Runge-Kutta time integration scheme (described in §3.2) is applied throughout this investigation, so that any such error will at least be consistent between test cases. Furthermore it should be possible to investigate the degree by which any temporal error affects the results, by inspecting the motion of the Lagrangian particles and comparing it to the transport

of the Eulerian phase. Because the high- and low-frequency phases are transported by fundamentally different approaches, errors from spatial discretisation are unlikely to be the same; the time integration, however, should affect each phase equally.

For this final configuration the transported field is expected to exactly match that shown in Fig. 7.7: while the diffusive flux coefficient has been reduced by two orders of magnitude, the number of computational steps has been increased by an equal degree, so that the total effect of diffusive fluxes should be the same. In practice some slight differences are observed; this is possibly due to the higher effective diffusive CFL condition in the previous case, and the increased effect of cumulative rounding errors in the present case. The solid lines shown on the following plots are from the analytical solution for this configuration.

Case ZM9: TVD scheme with CHARM limiter [130], Fig. 7.12. The CHARM scheme now exhibits some asymmetry, where the leading (left-hand) face of the disc has acquired a more concave shape compared to the smoother right-hand face. The central region of the disc has reached a constant scalar value of around 0.52, which is much lower than the expected peak of 0.78. The disc has also become elongated in the stream-wise direction, while no evidence of the central slot remains. The overall position of the disc seems to have drifted upstream of the expected final position, so that the transported solution is shifted a number of cells to the right; however, it is not clear whether this is due to the transport scheme or the time integration scheme.

Case ZM10: TVD scheme with SuperBee limiter [123], Fig. 7.13. More pronounced behaviour is observed when the SuperBee flux limiter is applied. The compressive property previously observed in Cases ZM2 and ZM6 has entirely removed any resemblance of the transported scalar field to the expected solution. The relatively weak physical diffusive fluxes have been overtaken by a stronger (numerical) compressive term, so that the faces of the scalar field have attained a stepped appearance. Furthermore the field has acquired a highly asymmetric shape, where distortion has occurred in both the stream-wise and stream-normal directions, and the 'base' of the transported disc now covers a smaller area than that of the initial field. Rather than the expected decrease in the maximum value of the scalar, the peak is found to have maintained a consistent value of 1.0.

Case ZM11: E-L method with CDS8, $\mathbf{L^{\varrho} = 1000}$, Fig. 7.14. In this case the E-L method shows some significant shortcomings: while the shape of the solution is reminiscent of the expected scalar field, the upwind (right-hand) side of the disc has a considerably higher value than the downwind side. The solution does, however, show some evidence of the central slot, and appears to recover the overall shape of the field better than the Eulerian schemes of Cases ZM9 and ZM10.

To locate the source of the error, and to shed light on the significance of errors due to the chosen temporal discretisation, the Eulerian and Lagrangian components of the solution presented in Fig. 7.14 are considered individually. Figure 7.15 shows the initial and final low-frequency fields, while Fig. 7.16 shows corresponding plots for the particle phase. While the shape of each transported field is not comparable to the associated initial field due to the action of diffusion, the plane of symmetry should, at least, lie in the same position following transport. The results for the La-

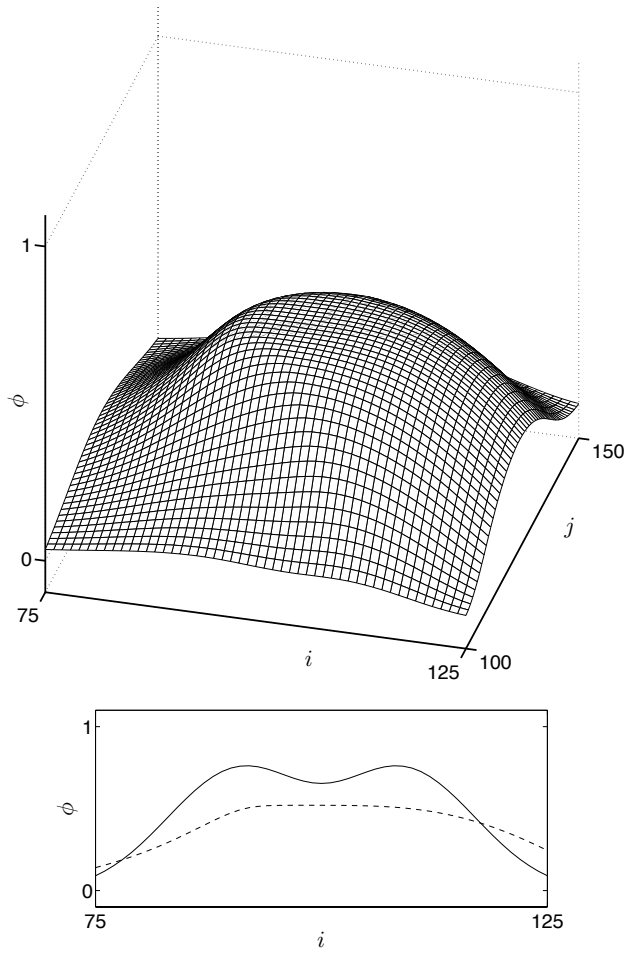


Figure 7.12: Case ZM9: CHARM TVD; 100 revolutions, $Sc = 100$.

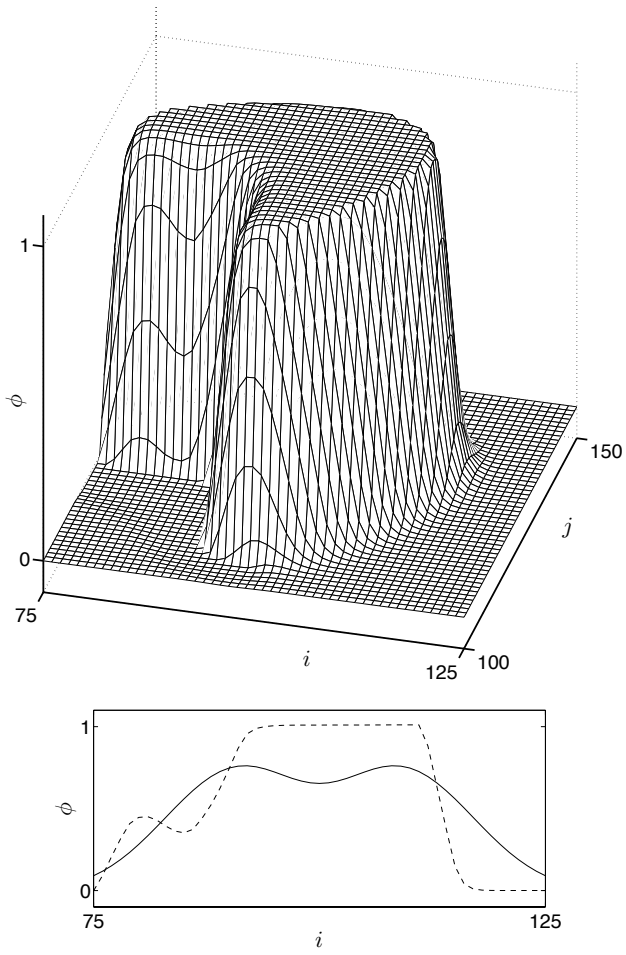


Figure 7.13: Case ZM10: SuperBee TVD; 100 revolutions, $Sc = 100$.

grangian field show that the plane of symmetry is positioned correctly, suggesting that the particles have been accurately transported; the Eulerian field, however, shows a significant disparity between the initial and final fields, and is clearly the cause of the skewed appearance in the scalar summation shown in Fig. 7.14. The accuracy in the position of the transported Lagrangian field suggests that errors due to the time integration scheme are small compared to those from the chosen spatial discretisation.

Case ZM11 Revisited As a result of the above analysis Case ZM11 is reconsidered, and modifications designed to reduce the error introduced in the transport of the Eulerian phase are suggested. The problem relates to the gradual accumulation of oscillatory error; possible solutions, therefore, may be to increase the order of accuracy of the applied CDS (considering the improvement gained when comparing Cases Z7 and Z8 – transported with CDS2 and CDS8, respectively – presented in Chapter 6), or to reduce the second derivative of the low-frequency field.

For the first of these options a twelfth-order CDS (CDS12) is implemented. This scheme requires an extension of the computational stencil by two cells in each direction; in a parallel computation,

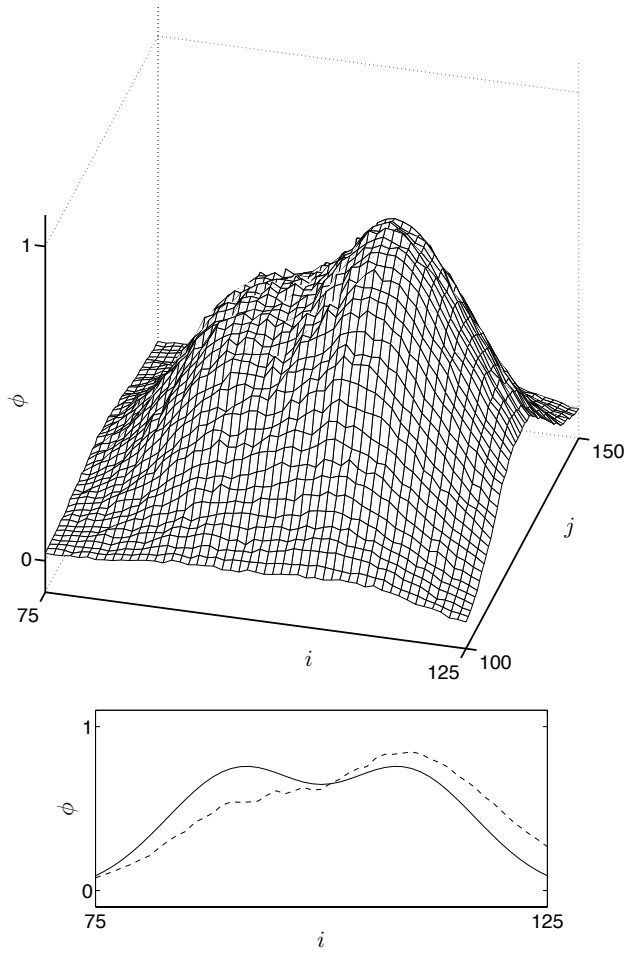


Figure 7.14: Case ZM11: E-L method with CDS8, $L^{\varrho} = 1000$; 100 revolutions, $Sc = 100$.

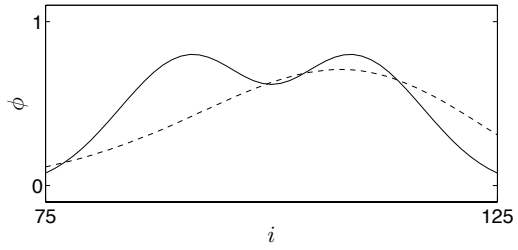


Figure 7.15: Case ZM11: Eulerian phase.

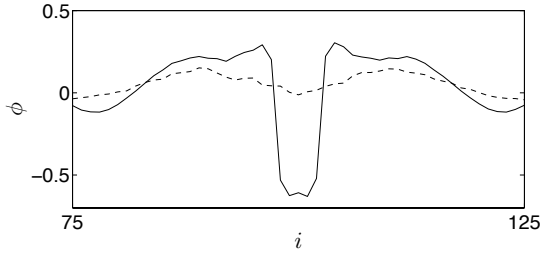


Figure 7.16: Case ZM11: Lagrangian phase.

therefore, the cost of communication would be increased, as more information would need to be transferred between process domain boundaries. Applying the algorithm of Fornberg [114], a twelfth-order approximation for the value of ϕ^E at the right-hand cell face may be written, for a one-dimensional case, as:

$$\phi^E\left(i+\frac{1}{2}\right) \approx \frac{1}{524288}\Big[320166\left(\phi^E(i)+\phi^E(i+1)\right) - 76230\left(\phi^E(i-1)+\phi^E(i+2)\right) + 22869\left(\phi^E(i-2)+\phi^E(i+3)\right) - 5445\left(\phi^E(i-3)+\phi^E(i+4)\right) + 847\left(\phi^E(i-4)+\phi^E(i+5)\right) - 63\left(\phi^E(i-5)+\phi^E(i+6)\right)\Big] \quad (7.1)$$

All other simulation parameters are kept the same. The order of accuracy of the approximation of scalar gradient at the cell face (for diffusive fluxes) will not have any significant effect on the behaviour of the convective scheme, and is therefore unchanged. Results from this test are referred

to as Case ZM11(2).

The second option utilises a reduction in the second derivative of the initial Eulerian field. This reduction is achieved by increasing the width of the Gaussian filter used for the initial Eulerian-Lagrangian decomposition, from $\sigma_f = 5\Delta$ to $\sigma_f = 15\Delta$. The re-initialisation parameter γ remains unchanged, and as a result (according to Eq. 5.59) the cut-off value $\phi''_{\max}$ for re-initialisation is reduced by an order of magnitude. Increasing the value of σ_f in this manner will result in a higher number of particles at initialisation, as more of the scalar information is stored in the high-frequency phase. However, the additional computational cost of these extra particles may be mitigated by reducing the frequency of re-initialisation events: given that the shape of the Eulerian field is not expected to change over the course of the simulation (as there are no shear stresses acting upon it) and that the Eulerian field is now very smooth, it is assumed that re-initialisation will not need to be performed at the same frequency as in previous tests. The interval between re-initialisation events in this test is therefore increased by an order of magnitude (i.e. to once every 40 time-steps). It should be stressed that this increase in the re-initialisation interval would not be suitable for cases with shear, where the second derivative of the Eulerian phase may be expected to change rapidly in time. This test is referred to as Case ZM11(3).

Figures 7.17 and 7.18 show the impact of these modifications on the transported field after 100 revolutions. In comparison with the results from Case ZM11 (shown in Fig. 7.14), it can be seen that the application of CDS12 for the Eulerian phase in Case ZM11(2) has improved the solution to some degree, but still results in a distinctly asymmetric final field. Although the order of accuracy of the CDS could be increased further, it is suggested that any additional improvement would be asymptotic. The impact on communication overheads for simulations performed on parallel architectures would also tend to out-weigh any improvement in accuracy as the computational stencil is extended. Far more effective, however, is the use of an increased Gaussian filter width for the initial decomposition in Case ZM11(3): although there is still a discernible asymmetry to the transported solution, the field shows a significant improvement over that displayed in Fig. 7.14.

Case ZM12: E-L method with CDS8, $\mathbf{L^{\varrho} = 100}$, Fig. 7.19. Consistent with the results from the previous configurations, a reduction in particle number yields a noticeable increase in the level of noise in the solution. The overall shape of the transported scalar field exhibits the same asymmetric features observed in Case ZM11, due to a distortion of the Eulerian phase. The shape of the transported field is now slightly more shifted towards the upwind side, in comparison to ZM11, while the downwind peak is less pronounced.

Case ZM12 is therefore repeated with the Gaussian filter width increased to $\sigma_f = 15\Delta$. The results from this test, referred to as Case ZM12(2), are shown in Fig. 7.20. As observed in Case ZM11(3), it can be seen that increasing the initial smoothness of the Eulerian phase has improved the overall shape of the transported solution significantly.

Tabulated Results Measurements of accuracy and computational cost for Cases ZM8 to ZM12, including results from the modifications to ZM11 and ZM12, are presented in Table 7.4. The data augments the observations made from the graphical results: the CHARM and SuperBee schemes

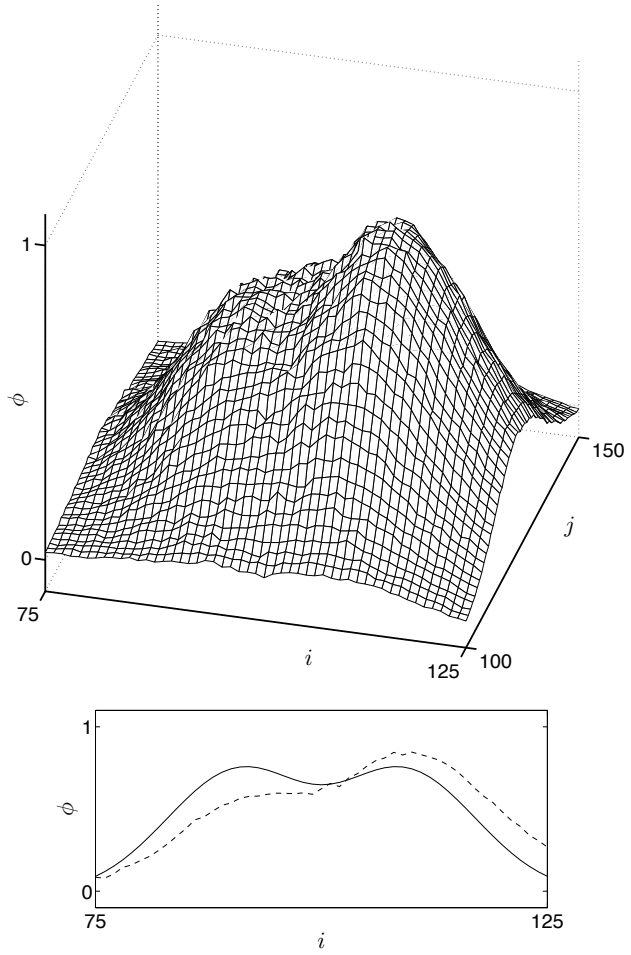


Figure 7.17: Case ZM11(2): With CDS12 for the Eulerian phase.

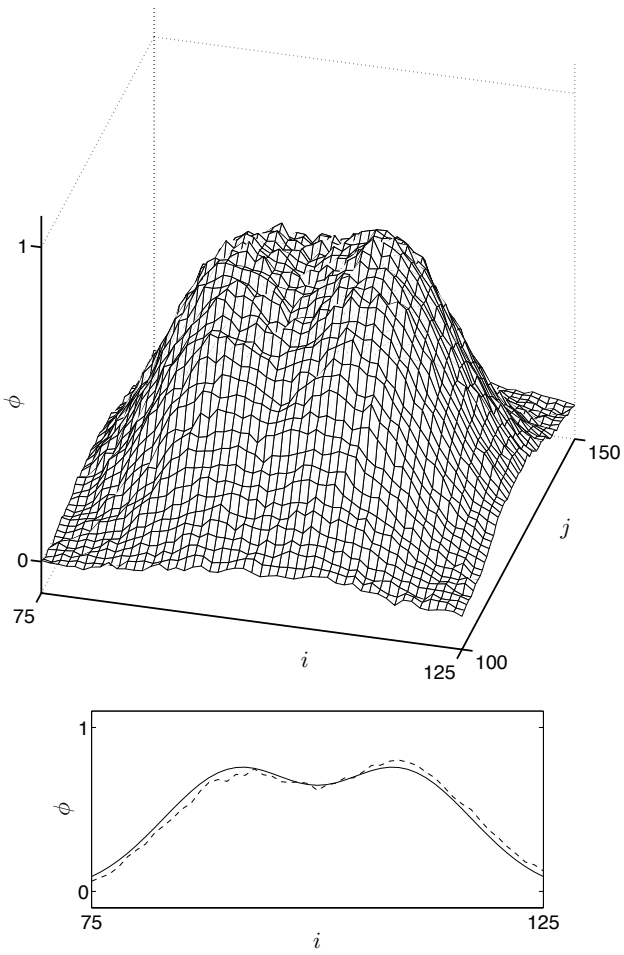


Figure 7.18: Case ZM11(3): With $\sigma_f = 15\Delta$ for the initial decomposition.

show under- and over-boundedness, respectively; the E-L simulations in Cases ZM11 and ZM12 show some improvement relative to the Eulerian-only cases; and the increased Gaussian filter widths used for the initial decomposition in ZM11(3) and ZM12(2) deliver significant additional improvement. The total error E is relatively large for all cases, but is minimised for ZM11(3) and ZM12(2). The significant lag in the Eulerian phase of the E-L solutions to Cases ZM11 and ZM12 (apparent in Fig. 7.15) causes a pronounced increase in total error, change in total variation ΔTV, and symmetry error E_{sym}. The change in total variation is consistently higher for the E-L cases than for the Eulerian-only cases, due to the sensitivity of this measure to noise in the Lagrangian phase. More noise is introduced into the solutions for ZM11(3) and ZM12(2) as a wider region of the scalar field includes some contribution from Lagrangian particles, due to the widened initial decomposition.

In terms of computational cost the E-L method continues to be significantly more expensive. Furthermore, the three-fold increase in the number of particles in Case ZM11(3) causes it to be almost three times as expensive as ZM11. The additional cost of the Lagrangian particles is found to be 1.12 CPU-seconds per million particles per time-step based on the computational times of Cases ZM11, ZM11(3), ZM12 and ZM12(2), which agrees with previous values for the modified Zalesak

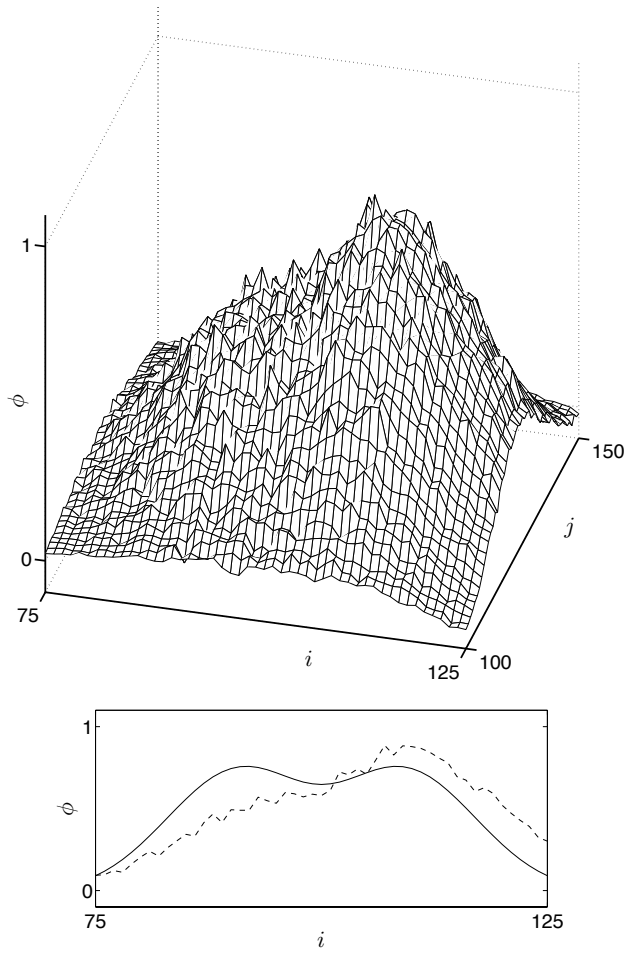


Figure 7.19: Case ZM12: E-L method with CDS8, $L^{\varrho} = 100$; 100 revolutions, $Sc = 100$.

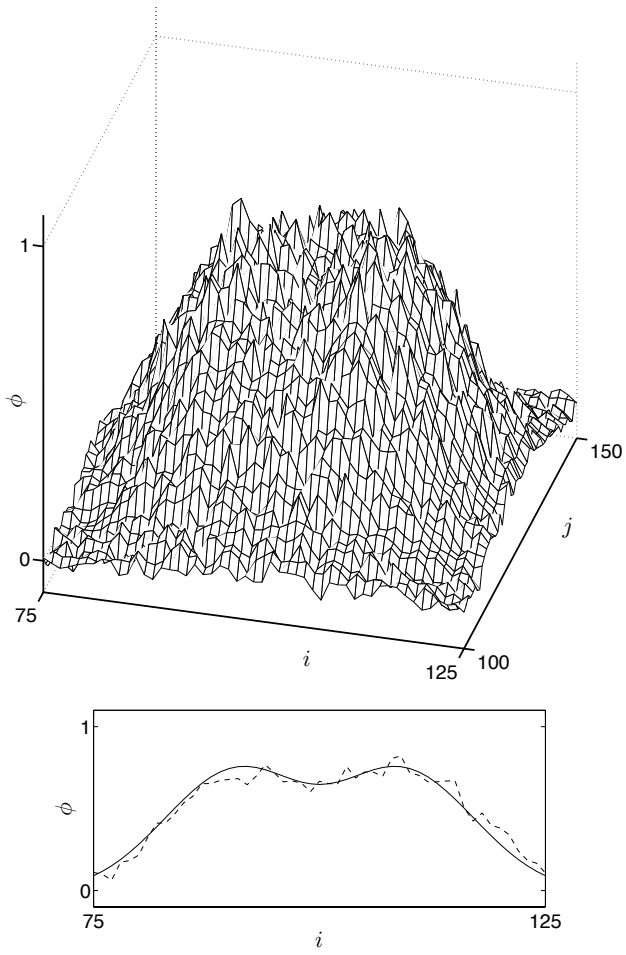


Figure 7.20: Case ZM12(2): With $\sigma_f = 15\Delta$ for the initial decomposition.

case. Modification of the initial decomposition parameters has little effect on the computational cost per particle, suggesting that the cost of the re-initialisation routine (which is performed less frequently for Cases ZM11(3) and ZM12(2)) is relatively low.

The overall CPU time for each case is two orders of magnitude greater than for the previous configurations, as the number of time-steps to solution has been increased by a factor of 100.

7.3 Conclusions

To gain a deeper insight into the performance of the joint Eulerian-Lagrangian method, the case originally studied by Zalesak (and applied in Chapter 6) has been modified. The topology of the scalar field now includes sharp steps and smoothly-varying gradients, so that the transported field is more representative of that which may be found in a real turbulent flow. This modified case has provided a more rigorous test of each of the applied convective schemes, and has highlighted aspects in the behaviour of each scheme that were not apparent from the previous study of the Zalesak disc.

Table 7.4: Measurements of accuracy and cost for Cases ZM9 to ZM12.

Case	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)	Time (s)
ZM9	0.00	24.24	335.14	-21.94	115.25	-	4423
ZM10	-1.17	-24.81	616.81	43.07	166.55	-	4619
ZM11	-7.76	-10.09	448.25	97.93	383.79	3.928	59769
ZM11(2)	-7.74	-9.14	443.57	93.85	380.17	3.919	60290
ZM11(3)	-1.63	-6.53	114.84	66.60	106.20	9.229	132812
ZM12	-9.68	-21.68	552.36	263.67	447.95	0.382	10578
ZM12(2)	-7.28	-11.43	175.72	250.62	142.64	0.910	17309

Testing has been performed using three configurations, to cover a wider variety of physical and computational conditions. In the first configuration a single revolution of the modified Zalesak disc is simulated at a low (physical) diffusivity; then, a single revolution is performed at a relatively high diffusivity, to verify the applicability of the E-L method for low Schmidt numbers; and finally, 100 revolutions are simulated at a low diffusivity, so that the total number of simulated time-steps is comparable to that of a typical DNS or LES. For each configuration the CHARM and SuperBee TVD schemes are compared to the E-L method at high and low Lagrangian particle densities.

Results for the first two configurations are consistent with the results and analysis provided in Chapter 6, in that both of the Eulerian schemes exhibit some adverse behaviour. The E-L method is capable of delivering a similar or reduced level of total error, but at considerable computational cost. The E-L solutions also contain an element of noise from the Lagrangian phase, the amplitude of which increases as the particle density is reduced. Significantly, however, the E-L method appears to show an improved ability to accurately recover the gradients of the transported field, while the Eulerian schemes tend to distort its shape. In particular the SuperBee TVD scheme demonstrates unphysical compressive behaviour, where the smooth edges of the modified Zalesak disc become artificially sharpened during transport.

The final configuration shows how extreme the behaviour of the Eulerian schemes tested here can become, as the CHARM and SuperBee TVD methods yield transported fields that exhibit little resemblance to the analytical solution. Although the E-L method also shows some adverse behaviour, it has been demonstrated that modification of the Gaussian filter width applied for the initial Eulerian-Lagrangian decomposition can permit the method to deliver a relatively high level of accuracy at the cost of an increase in computational expense.

The E-L method continues to be much more computationally expensive than the Eulerian approach. However, for the final configuration such a comparison is arbitrary, as the Eulerian simulation results have accumulated significant spatial discretisation errors. The most notable result that may be drawn from this part of the investigation is that the cost of Case ZM12(2) is no more than four times that of the corresponding Eulerian cases, which is encouraging given its ability to perform well over long simulation durations.

At this stage the practicalities of the joint Eulerian-Lagrangian method have been demonstrated. It would now be useful to consider any potential improvements to the method that have been highlighted by the studies presented in Chapters 6 and 7. These points will be addressed in Chapter 8.

8 Further Development

The work presented in Chapters 6 and 7 has shown the ability of the Eulerian-Lagrangian (E-L) method to deliver an improved description for the convective transport of a scalar, for a selection of two-dimensional problems. However, it has also revealed some shortcomings of the method, namely: its computational expense; the noise introduced by the Lagrangian reconstruction; and the chamfering of sharp edges. These problems are now addressed, before proceeding with the more complex two- and three-dimensional test cases presented in Chapters 9–11.

The cost of the E-L method may be moderated by reducing the total particle number, and a mechanism for the removal of Lagrangian particles – where such a removal will not jeopardise the stability of the Eulerian field – is therefore developed. The second and third points mentioned above constitute additional post-processing operations that may be performed once the Lagrangian field has been reconstructed. Following a description of their development the three modifications presented in this chapter are applied to a selection of the Zalesak and modified Zalesak cases studied previously, to assess their practicality for later test simulations.

8.1 Particle Removal

The action of the existing re-initialisation routine described in §5.4.4 is to improve the stability of the Eulerian field by the addition of Lagrangian particles; conversely, the removal of those particles will only serve to make the solution less stable. The concept of a particle removal mechanism to reduce the computational cost of the E-L method must therefore be treated with some care.

It is possible, however, to envisage situations in which particle removal could play an important role, both in reducing the computational cost, and by improving the accuracy of the results through a reduction in the noisiness of a solution. In simulations of flows at finite Schmidt numbers the action of diffusion will tend to make a solution more stable, such that particles that were required to ensure stability at the beginning of a simulation may become redundant over time. The low Schmidt number modified Zalesak disc configuration presented in §7.2.2 represents such a situation; however, similar conditions may also be found in, for example, a bluff body jet, where steep scalar gradients around the jet nozzle become reduced by molecular and turbulent mixing as the scalar is transported downstream. At the very least, an adjustable particle removal routine potentially provides a greater level of control in the trade-off between the accuracy of the E-L method and its computational cost.

8.1.1 Formulation

A mechanism to identify and remove Lagrangian particles in a manner that does not compromise the stability of the Eulerian component is required. The first step is to identify a suitable method of localising the particle removal so that it only reduces stability in regions that are already very stable. The second derivative of the Eulerian phase is currently used as an indicator for regions in which particle addition is required, where particles are added above some value $\phi''_{\max}$; it would thus seem reasonable to remove them below some value $\phi''_{\min}$, where $\phi''_{\min} \ll \phi''_{\max}$. Conceptually, a scalar field may then be considered to contain regions that are 'too sharp' (where $\phi''_{\max}$ is exceeded, and particles are added), and 'too smooth' (where the second derivative is less than $\phi''_{\min}$, and particles are removed). A lower limit for particle removal is therefore defined as:

$$\phi''_{\min} = \beta\, \phi''_{\max} \tag{8.1}$$

Where β is a constant in the range $0 < \beta \ll 1$. Once the local second derivative of the Eulerian field has been determined for each computational cell (using the expression given in Eq. 5.56), a particle removal probability for that cell is defined according to:

$$P_{\mathrm{MC}}\,(i) = 1 - \min\left(1,\, \frac{\left|\nabla^2 \phi^E(i)\right|}{\phi''_{\min}}\right) \tag{8.2}$$

Where P_{MC} is the proportion of particles in cell i that will be removed, which will be zero for $\left|\nabla^2 \phi^E\right| \geq \phi''_{\min}$, will be unity for regions of constant scalar gradient where $\left|\nabla^2 \phi^E\right| = 0$, and will be linearly dependent on $\left|\nabla^2 \phi^E\right|$ otherwise. A linear function is implemented to prevent any step change in the Lagrangian field (and subsequently, the Eulerian field) which may otherwise occur if the mechanism simply removed all of the particles in regions where $\left|\nabla^2 \phi^E\right| < \phi''_{\min}$.

To satisfy this calculated particle removal probability for cell i exactly it would be necessary to determine how many particles are in that cell, so that the number to be removed may be found directly. For example, in a cell with 20 particles and $P_{\mathrm{MC}} = 0.6$, it would be a case of selecting 12 particles at random for deactivation. However, such an operation would prove to be relatively expensive, as it would require the generation of a list of the indices of particles residing in each cell. A more attractive alternative, therefore, is to apply a computationally-cheap Monte Carlo approximation for P_{MC}. A removal probability P^p_{MC} is found for each active particle by linearly interpolating the adjacent (cell-centred) values of P_{MC} on to the particle position (similar to the interpolation of the Eulerian velocity field to determine the particle velocity, as described in §5.4.2). Each particle is also assigned a single, uniformly-distributed random number u_1, and particles are removed where the condition $P^p_{\mathrm{MC}} > u_1$ is satisfied. This approach has the advantage of requiring the properties (i.e. position) of each particle to be accessed only once for each application of the algorithm. Although some error in the Monte Carlo approximation of P_{MC} should be expected for each cell, in practice this will typically translate into only one or two more particles removed or retained than intended.

Once a particle has been identified for removal its activity property is set to 'inactive', so that it is made available for re-use later on. For the actual transferral of scalar information from the

particle field to the Eulerian phase the reconstruction algorithm described in §5.4.3 is applied, so that the scalar contribution of the particle is shared proportionally between the adjacent cells. The cost of this operation is reduced by the fact that the weightings for the adjacent cells have already been calculated for the interpolation of P_{MC} on to the particle position.

Particle removal is implemented so that it occurs immediately before the existing re-initialisation routine, and at the same frequency. This ensures the stability of the Eulerian phase, as any particles that are removed erroneously (for example, as a result of the coarseness of the Monte Carlo approximation for P_{MC}) will be immediately replaced by the particle addition routine.

8.1.2 Testing

The procedure outlined in §8.1.1 is implemented for a one-dimensional test case, to investigate the behaviour of the mechanism and to modify it if necessary. A domain of length $l = 1$ m is discretised at a uniform cell size $\Delta = 1$ mm, with a step in the scalar field located at $x = 0.5$ m. The initial decomposition is performed at a Gaussian filter width $\sigma_f = 5\Delta$, with a re-initialisation parameter of $\gamma = 1$ resulting in $\phi''_{\max} = 0.0097$. The initial conditions for the test are shown in Fig. 8.1. The Lagrangian field is represented at a particle density $\mathrm{L}^{\varrho} = 1000$, and re-initialisation is performed every four time-steps. The constant β appearing in Eq. 8.1 is taken to be 0.2 for testing. After decomposition the solution will be transported by diffusion only for 6000 time-steps, at a fixed time-step width $dt = 0.1$ s, under conditions of i) zero and ii) finite diffusivity.

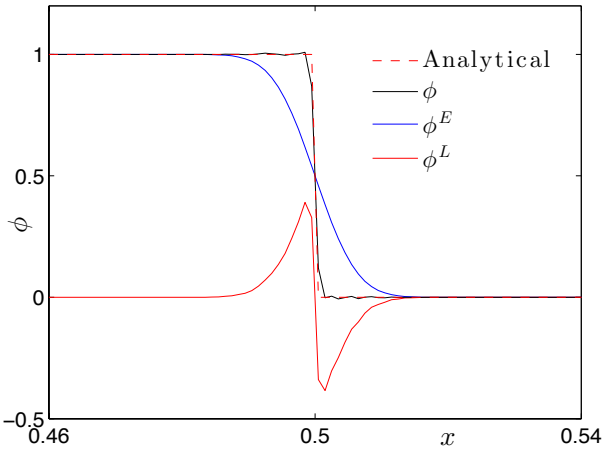


Figure 8.1: Initial decomposition.

With zero diffusivity it is apparent that nothing at all should happen to the decomposed field, and this is the case for the E-L solution with no particle removal applied. Once the particle removal mechanism is added, however, some undesired behaviour becomes apparent, as observed in Fig. 8.2. Around the point of inflection at $x = 0.5$ m (where the second derivative of the Eulerian field changes sign) the particle removal probability is approximately 40%, and the routine has removed a number of particles that should have been retained. The E-L summation now exhibits a 'kink' in the step, and a small undershoot and overshoot to either side of that kink.

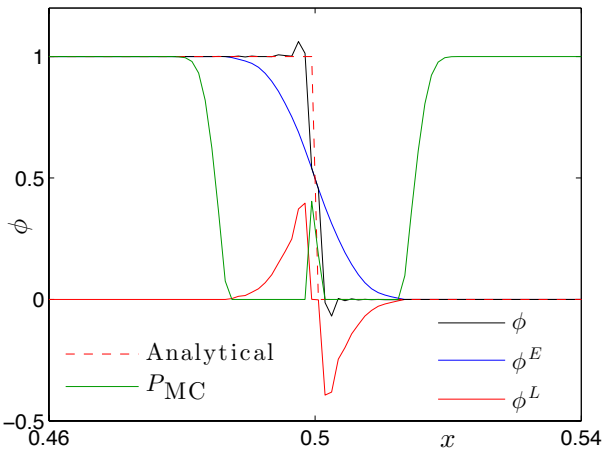


Figure 8.2: Particle removal: Without transport.

In Chapter 4 it has been shown that stability is linked to the smoothness of the transported field (see Fig. 4.13), and that a field with zero second derivative should therefore be unconditionally stable. In theory, then, the removal of particles at the point of inflection ($x = 0.5$ m) should not be prevented. However, it is suggested that the mechanism might be discouraged from removing particles in this region, at least where the low-frequency scalar field is at its steepest, to prevent the error apparent in Fig. 8.2 from occurring. This is achieved by including an additional criterion for the calculation of P_{MC}, so that Eq. 8.2 is modified to become a function of both the first and second derivatives of the Eulerian field:

$$P_{\mathrm{MC}}(i) = 1 - \min\left[1,\ \max\left(\frac{\left|\nabla\phi^E(i)\right|}{\phi'_{\mathrm{min}}},\ \frac{\left|\nabla^2\phi^E(i)\right|}{\phi''_{\mathrm{min}}}\right)\right] \tag{8.3}$$

Where the gradient of the Eulerian field is approximated as:

$$\nabla\phi^E(i) = \frac{1}{2}\left[\phi^E(i+1) - \phi^E(i-1)\right] \tag{8.4}$$

This expression provides a second-order estimate; higher orders of accuracy may be achieved by extending the computational stencil and applying the weights provided by Fornberg [114], although the expression has no impact on the accuracy of the E-L method as a whole. This modification means that particles will not be removed where the gradient of the Eulerian phase exceeds some value ϕ'_{min}. To obtain a reasonable value for this limit, the analysis used in §5.4.4 is repeated: after decomposition, the Eulerian phase in this case will take the form of a Gauss error function, such that the gradient of the Eulerian field at each point will be equal to the corresponding value of the Gaussian filter kernel at that point. The maximum gradient of the field may then be found from the maximum value of the Gaussian kernel, which is determined by setting δx in Eq. 5.43 to zero. This results in:

$$\left|\nabla\phi^E\right|_{\mathrm{max}} = \frac{1}{\sigma_f\sqrt{2\pi}} \tag{8.5}$$

The gradient limit for particle removal is then defined to be some proportion of this maximum gradient:

$$\phi'_{\text{min}} = \lambda \left| \nabla \phi^E \right|_{\text{max}} \tag{8.6}$$

Where λ is a constant in the range $0 < \lambda \leq 1$, and is taken to be 0.7 for further testing. Applying the new expression for P_{MC} (Eq. 8.3) results in the desired behaviour, such that the error observed in Fig. 8.2 is corrected; the field after 6000 steps at zero diffusion is now identical to that shown in Fig. 8.1.

The next step is to determine whether the particle removal mechanism is effective in reducing the total number of particles for a case with finite diffusivity. The test conditions are therefore modified so that the kinematic viscosity and Schmidt number are $\nu = 1.0 \times 10^{-4}$ m^2/s and $Sc = 1$, respectively. In the first instance the scalar field is transported without any particle removal mechanism implemented, and Fig. 8.3 show the components and summation of the E-L solution after transport with the analytical field for comparison.

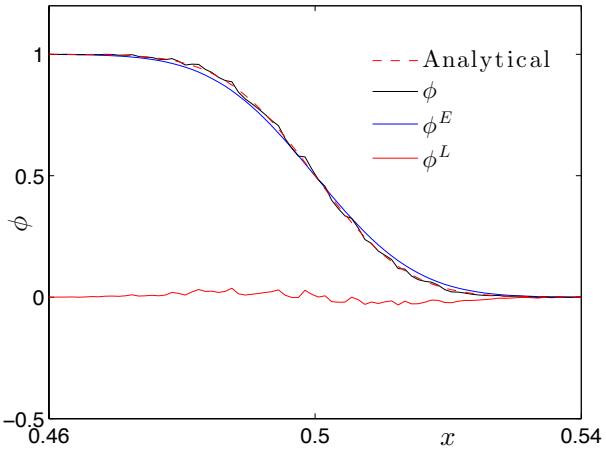


Figure 8.3: Diffused field, without particle removal applied.

The particle removal mechanism is then added, and the evolution of the Lagrangian field is considered. Figure 8.4 shows the E-L and analytical solutions after (a) 1000, (b) 2000, (c) 4000 and (d) 6000 steps.

The modified expression for P_{MC} (Eq. 8.3) limits particle removal near to the step in the scalar field (i.e. over the range of x for which $0 < \phi < 1$) for at least 1000 time-steps, after which the particle removal probability in this region gradually increases. This increase occurs as the Eulerian phase becomes more smooth, and the mid-point gradient drops below ϕ'_{min}. At the final time-step there is a non-zero probability of particle removal throughout most of the domain.

Table 8.1 compares statistics for the tests performed with and without the particle removal mechanism implemented. It can be seen that the mechanism yields a significant reduction in the number of particles by the end of the simulation, which results in a 35% reduction in computational time and an order of magnitude decrease in the total error between the analytical and simulated fields.

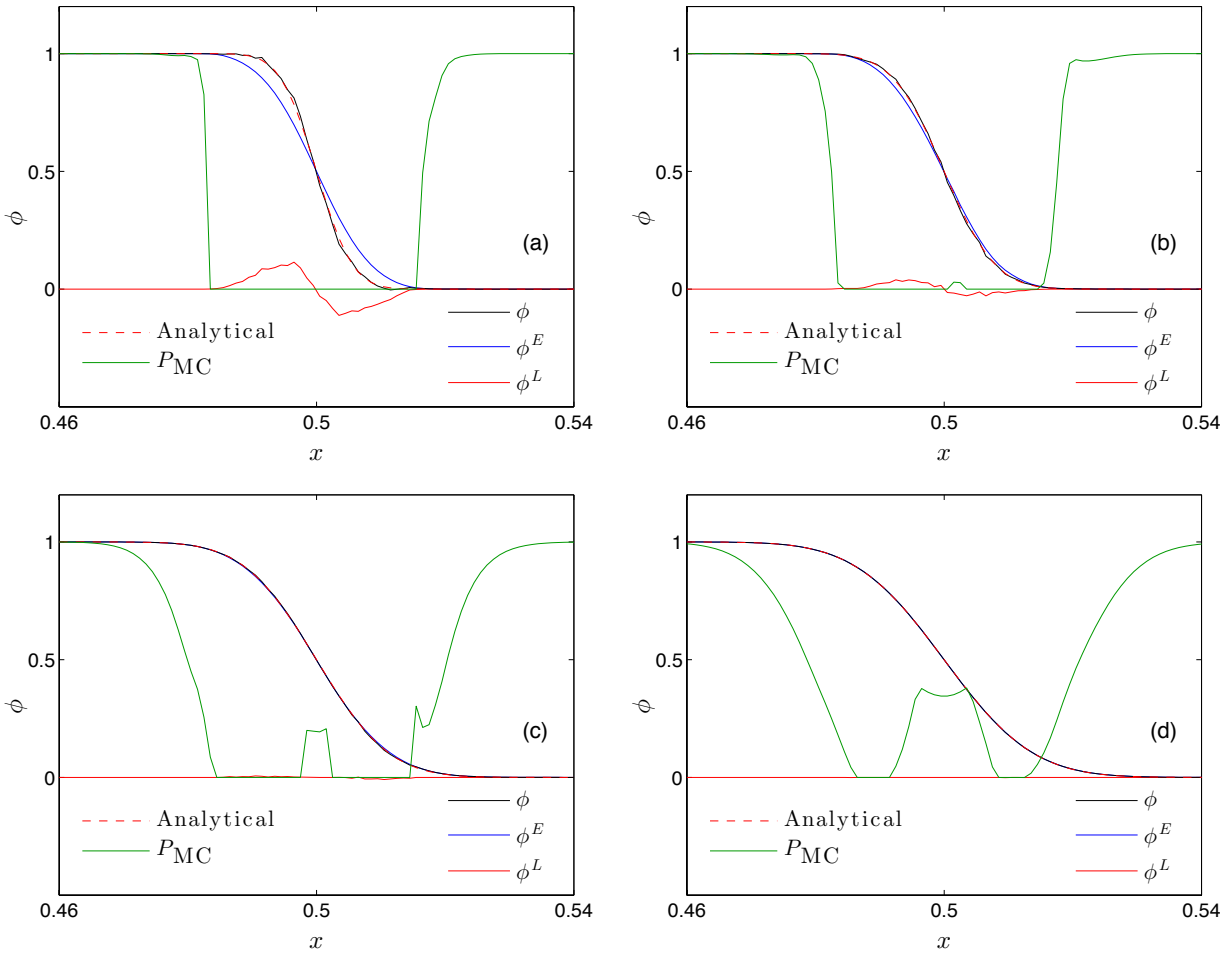


Figure 8.4: Particle removal: After (a) 1000, (b) 2000, (c) 4000 and (d) 6000 steps.

Table 8.1: Particle removal test results.

Case	Particles (after 6000 steps)	Time (s)	Error
Without removal	4042	7.9	0.329
With removal	3	5.1	0.023

8.1.3 Application in Higher Dimensions

The results presented in §8.1.2 suggest that the particle removal mechanism is capable of correctly identifying and removing redundant particles in a one-dimensional case. The extension to higher dimensions now needs to be considered.

The particle addition routine described in §5.4.4 applies operations independently in each direction, i.e. the one-dimensional formulation is performed separately two or three times (for cases in two or three dimensions, respectively). This is only possible as the stability of the convective transport scheme is direction-dependent: if the contours of the Eulerian field and the streamlines of the flow-field are parallel no oscillations will develop as the Eulerian field is convected, even if the gradient of that field varies rapidly in a direction normal to the flow. Particles, however, do not possess a sense of direction; when their scalar contribution is added back into the Eulerian phase, they will alter the smoothness of that phase in all directions. Applying the particle addition routine independently in each direction also provides an improvement in computational efficiency, as the cost of the filtering operation required by that routine is reduced; no such improvement may be gained here, however, as the particle removal routine requires no filtering.

The particle removal mechanism is therefore applied in all directions at once. The second derivative of the Eulerian field (which would be a tensor in higher dimensions) is replaced with the Laplacian of the Eulerian field:

$$\nabla^2 \phi^E = \sum_{i=1}^{d} \frac{\partial^2 \phi^E}{\partial x_i^2} \tag{8.7}$$

Where d is the dimensionality of the simulation, and the derivative is taken with respect to cell co-ordinates (not physical co-ordinates). In three dimensions, this is (locally) estimated as the sum of three one-dimensional approximations for the (unmixed) second derivative in each direction:

$$\begin{aligned}\nabla^2 \phi^E(i,\, j,\, k) \approx \phi^E(i-1,\, j,\, k) + \phi^E(i,\, j-1,\, k) + \phi^E(i,\, j,\, k-1) - 6\phi^E(i,\, j,\, k) + \\ \phi^E(i+1,\, j,\, k) + \phi^E(i,\, j+1,\, k) + \phi^E(i,\, j,\, k+1)\end{aligned} \tag{8.8}$$

Similarly the gradient of the Eulerian phase becomes a vector in two or more dimensions, and is therefore replaced with an approximation for the magnitude of the gradient. In three dimensions:

$$\begin{aligned}\left|\nabla \phi^E(i,\, j,\, k)\right| \approx \Bigg[\left(\frac{\phi^E(i+1, j,\, k) - \phi^E(i-1, j,\, k)}{2} \right)^2 + \\ \left(\frac{\phi^E(i,\, j+1,\, k) - \phi^E(i,\, j-1,\, k)}{2} \right)^2 + \\ \left(\frac{\phi^E(i,\, j,\, k+1) - \phi^E(i,\, j,\, k-1)}{2} \right)^2 \Bigg]^{1/2}\end{aligned} \tag{8.9}$$

Implementation of the mechanism in higher dimensions is otherwise identical to the procedure described in §8.1.1 and §8.1.2.

8.2 Low-Pass Filtering

The removal of background noise from a source has been widely researched and developed, and finds application in a variety of areas from image processing to seismology and the design of audio equipment. The most widely-used method of noise reduction in Digital Signal Processing (DSP) is the convolution of a signal with a low-pass filter. In an audio signal this filtering will occur in the time domain (between samples), whereas in a digital photograph, filtering will be performed spatially (between pixels). In the present work the use of a spatial low-pass filter may deliver an improvement in the description of the transported scalar field, by reducing the amount of noise introduced into the solution by the Lagrangian particle reconstruction. As the wavenumber of the noise may be assumed to be approximately equal to $1/\Delta$ (where Δ is the cell size), the width of the filter should only need to be of the order of Δ.

Such a filter must be carefully designed, as one of the main aims of the E-L method is to eliminate the smoothing associated with traditional convective schemes. The filter width should be limited so that information is not lost in regions of the scalar field with a rapidly-changing gradient; the formulation of a localising algorithm, permitting a variable filter width to be applied depending on the local topology of the field, would be ideal.

A discussion of the various filters that may be used has already been provided in Chapter 5; in this context a Gaussian filter kernel (applied once), or a narrow top-hat kernel (applied iteratively), would be appropriate. The choice between these filtering methods is one of computational efficiency. In addition a choice must be made between filtering the reconstructed particle field on its own, or filtering the E-L summation. The impact of filtering must also be assessed; the set of metrics applied to measure the performance of each convective scheme, introduced in §6.4 and defined by Eqs. 6.3 to 6.10, will be used again here.

To assist in the design of the low-pass filter a selection of scalar fields of various topologies are required for testing and development. The results from the modified Zalesak study presented in Chapter 7 could provide such information; however, for the configuration described in §7.2.3 (where the scalar field is transported for 100 revolutions) the E-L solutions are heavily influenced by discretisation errors in the Eulerian field. As the present filtering method is only concerned with the noise from the particle field the single-revolution Cases ZM3 and ZM7 from Table 7.1 are re-computed with the velocity field set to zero, so that only diffusive fluxes act upon each scalar component. This ensures that the only error in the test fields comes from the particle reconstruction. The computations are performed for 1256 time-steps, so that the final fields for each test should match the approximations to the analytical solutions shown in Figs. 7.2 and 7.7.

Each of these tests is also performed with a increased Gaussian filter width applied for the initial decomposition of the scalar field. This replicates the effect of a large number of re-initialisation events causing the magnitude of the Lagrangian component to increase (as in the 100-revolution configuration described in §7.2.3). The cases used for testing are summarised in Table 8.2, where σ_f refers to the width of the filter used for the initial decomposition. E-L computations from both the original and modified Zalesak studies will also be repeated with low-pass filtering applied, where the results will be presented in §8.4.

Table 8.2: Simulation parameters for filter localisation tests.

Case	σ_f (Δ)	Steps	Sc	Particles ($\times 10^5$)
Initial	5	-	-	4.109
F05LS	5	1256	1	4.118
F10LS	10	1256	1	7.388
F05HS	5	1256	10	4.123
F10HS	10	1256	10	7.388

8.2.1 Metrics

To assist in the design of a low-pass filter algorithm the usefulness of each of the metrics introduced in §6.4 is assessed. This assessment is achieved by filtering the Lagrangian component of the solutions for the cases listed in Table 8.2 at a range of Gaussian filter widths, from $\sigma_f = 0$ to $\sigma_f = 2\Delta$, and comparing the boundedness, total error, change in total variation and symmetry error for each resulting E-L summation. Absolute data is shown in the left-hand side plots of Fig. 8.5, while the right-hand side shows the same data expressed as a percentage of the initial (unfiltered) value for each measure.

Figure 8.5 shows that the unboundedness is generally reduced as the filter width is increased, up to an optimal filter width for each test case (where $|\Delta B_L|$ approaches its minimum, or $|\Delta B_U|$ reaches zero). These measures are of little use here, however, because the bounds of the filtered solution will either tend to the bounds of the Eulerian phase (if filtering is applied to the Lagrangian field only), or to the global mean of the scalar field (where the E-L summation is filtered), rather than to the bounds of the analytical field.

The total error measure E provides a more valuable indication of the effectiveness of filtering, as the normalised plots indicate that, for each set of data, there exists an optimal filter width that varies with the smoothness (Schmidt number) of the solution. This optimal width is zero in the case of the initial field, as any filtering will degrade the good description of the sharp edges around the disc slot. At a lower Schmidt number (with higher diffusive fluxes) more of the high frequency information in a solution will be due to noise, rather than due to the shape of the scalar field; the action of the low-pass filter in removing these high frequencies will therefore tend to improve a smooth solution more than for a sharp field, where more of the high frequency information should be retained.

The change in total variation ΔTV and the symmetry error E_{sym} reduce monotonically as the filter width is increased. However, the trend in symmetry error is asymptotic, indicating that (based on this measure alone) the best solution is achieved at an infinite filter width. Change in total variation, on the other hand, reaches zero at a different filter width for each case, but the non-zero optimal value of $\sigma_f \approx 0.8\Delta$ for the initial field (at which the total error in the initial field has more than doubled) suggests that this measure could be misleading in the present context.

From this analysis the total error is considered to be the most useful measure for the design of a low-pass filter to be applied to an E-L solution. (However, the other metrics will still be applied for the results presented in §8.4).

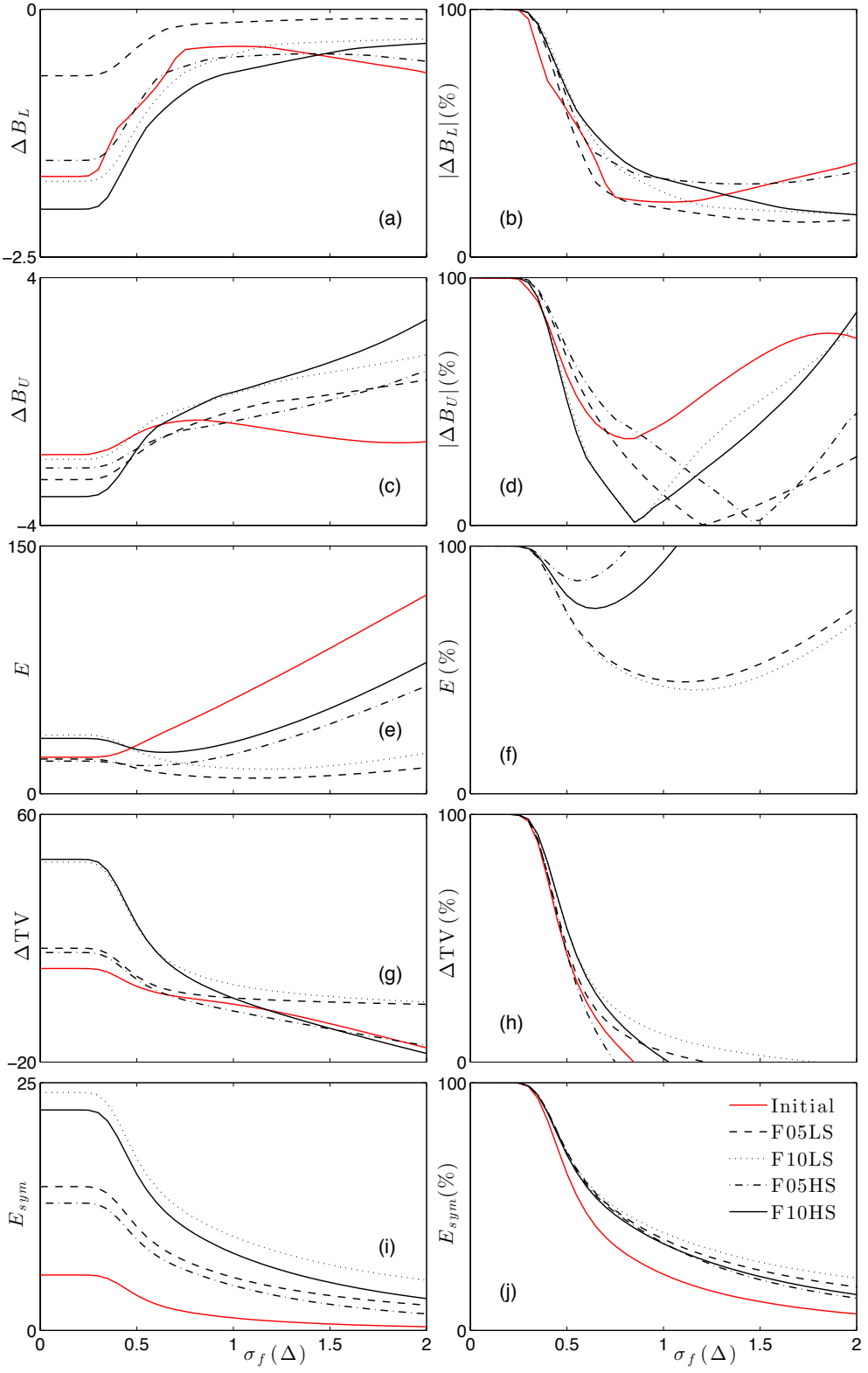


Figure 8.5: Effect of varying filter width on: lower boundedness (a) ΔB_L and (b) $|\Delta B_L|$ (%); upper boundedness (c) ΔB_U and (d) $|\Delta B_U|$ (%); total error (e) E and (f) E (%); change in total variation (g) ΔTV and (h) ΔTV (%); symmetry error (i) E_{sym} and (j) E_{sym} (%).

8.2.2 Filtered Field

The choice of whether to filter the Lagrangian field only, or to filter the E-L summation, is made by comparing the effect of filtering on the total error for each case. Figure 8.6 shows such comparisons for the cases listed in Table 8.2, where the red lines denote filtering of the Lagrangian field only, and the black lines show filtering of the summation. For each case the error is expressed as a percentage of the total error with no filtering applied. Filtering is performed over a range of Gaussian filter kernel widths $0 < \sigma_f < 2\Delta$.

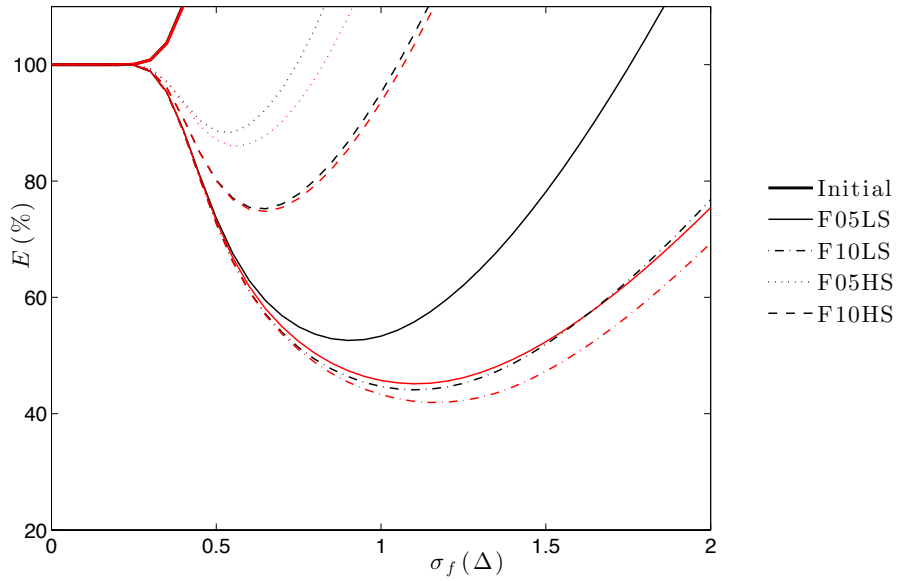


Figure 8.6: Effect of filtering the Lagrangian field (red) or the E-L summation (black).

Figure 8.6 shows that filtering of the initial field is detrimental regardless of whether the filter is applied to the summation or to the high-frequency component only. This is because much of the high-frequency information in the field is supposed to be there, constituting the sharp edge of the slot in the disc. Any degree of filtering will tend to remove some of the information that should be retained, as well as the noise that should be discarded.

Filtering does, however, provide a varying degree of improvement for the other cases, and it can be seen that filtering of the Lagrangian field alone is more beneficial than filtering the E-L summation. Notably this difference only becomes apparent above a particular value of σ_f, prior to which the trends are concurrent for each case. The Lagrangian field contains both undesirable (noise) and desirable (sharp changes in scalar gradient) high-frequency information, whereas the Eulerian field should only contain desirable frequencies; however, as σ_f is increased, filtering of the E-L summation may remove information from both the Eulerian and the Lagrangian fields (as the frequency response of the applied filter begins to affect the spectrum of the Eulerian field). This also explains the more pronounced difference between the red and black lines for Cases F05LS and F05HS: as the Eulerian field will be less smooth in these cases (because the filter used for the initial decomposition is narrower) the Eulerian field will contain scalar information across a

wider frequency spectrum than the more widely-filtered initial Eulerian fields of Cases F10LS and F10HS. This will therefore make the Eulerian fields of F05LS and F05HS more susceptible to further filtering.

Based on the evidence presented above, only the reconstructed Lagrangian field is filtered.

8.2.3 Filter Localisation

Figures 8.5 and 8.6 suggest that the optimal filter width that should be applied for noise reduction will vary with the topology of the scalar, and that a relationship between the shape of the scalar field and the applied Gaussian kernel width σ_f should therefore be established. The shape of the scalar field will vary in space; therefore the filter width should also vary in space. This advocates the development of a filter localisation algorithm, similar in some respects to the localised re-initialisation described in §5.4.4, although in this case it is necessary to identify local regions that should *not* be filtered.

To develop such an algorithm a suitable quantity should be selected to describe the local topology of the scalar field. Given that filtering should not be performed in regions with sharp changes in gradient, the second derivative of the scalar field constitutes such a quantity.

The process of filter localisation described here, and of filtering itself, is performed independently in each Cartesian direction. This improves the efficiency of the technique by reducing the computational cost of the filtering operation, but also increases its effectiveness by allowing filtering to be performed in Cartesian directions that are parallel to scalar contours. (Considering the example of a step in a scalar field, it is apparent that filtering must be avoided normal to the step; however, it is still useful to filter along to the top and bottom edges of, i.e. parallel to, the step). The implementation of the method is also simplified, as the extension from a one-dimensional development code into higher dimensions is trivial (although sufficient testing must also be performed in higher dimensions).

The approximation of the second derivative is calculated for the Lagrangian component, as it is the 'edges' of this field that should not be filtered. However, the Lagrangian phase also contains the noise that we intend to remove by filtering; an approximation similar to Eq. 5.56 may therefore lead to spurious high values when applied to ϕ^L directly. To avoid this a copy of ϕ^L is firstly convoluted with a Gaussian filter of (uniform) width $\sigma_f = 2\Delta$ to remove the noise, and the second derivative (with respect to the cell co-ordinates) is then determined for this intermediate filtered field ϕ^L_f:

$$\nabla^2\phi^L_f(i) \approx \phi^L_f(i-1) - 2\phi^L_f(i) + \phi^L_f(i+1) \tag{8.10}$$

The unfiltered Lagrangian field is retained, as this is the field to which the localised filtering will be applied. Note that applying a filter to ϕ^L will cause the local maxima and minima of $\nabla^2\phi^L_f$ to be shifted by a distance of $\sigma_f = 2\Delta$ as the field is smoothed. (This has previously been illustrated in Fig. 5.17, where the minima and maxima of the magenta line can be seen to be removed from the corners of the original step function). However, the subsequent formulation will prevent the localising algorithm from applying filtering in the wrong location.

In §5.4.4 it has been discussed that an iteratively-applied top-hat filter is a more suitable choice

for a localised filtering process, as it is easier to set a number of filter operations N_f in each cell than it would be to calculate and apply a different Gaussian filter kernel at each point. A relationship is therefore defined to locally relate N_f to the approximation of the second derivative of the Lagrangian field. Considering that a constant gradient scalar field may be filtered unconditionally, while no filtering should be applied around a sharp change in gradient, such a relationship is proposed as:

$$N_f = \lceil \alpha_{\mathrm{TH}} \, \sigma_f^2 \rfloor \tag{8.11}$$

Where the $\lceil \ldots \rfloor$ operator denotes rounding to the nearest integer and the Gaussian filter width σ_f must be determined. The factor α_{TH} scales N_f according to the chosen top-hat filter width w_{TH} (from Eq. 5.42, assuming $\Delta < w_{\mathrm{TH}} \leq 3\Delta$), as:

$$\alpha_{\mathrm{TH}} = \frac{w_{\mathrm{TH}}}{(w_{\mathrm{TH}} - \Delta)} \tag{8.12}$$

The filter width σ_f is calculated as a function of the second derivative of ϕ_f^L:

$$\sigma_f \left(\nabla^2 \phi_f^L \right) = \max \left[0, \, \min \left(\frac{a - \left| \nabla^2 \phi_f^L \right|}{\left| \nabla^2 \phi_f^L \right|}, b \right) \right] \tag{8.13}$$

Where the constants a and b must be selected.

Considering the behaviour of Eqs. 8.11 and 8.13, it is apparent that σ_f will equal zero where the absolute value of the local second derivative of ϕ_f^L is greater than a (resulting in $N_f = 0$), while it will be limited to b as the second derivative approaches zero. The value of a therefore represents the value of $\left| \nabla^2 \phi_f^L \right|$ above which no filtering is performed, while b represents the maximum value of σ_f applied in the smoothest regions of the domain.

The variation of σ_f with $\left| \nabla^2 \phi_f^L \right|$ (described by Eq. 8.13) is shown in Fig. 8.7, where the constants a and b are given typical values of $a = 0.01$ and $b = \Delta$.

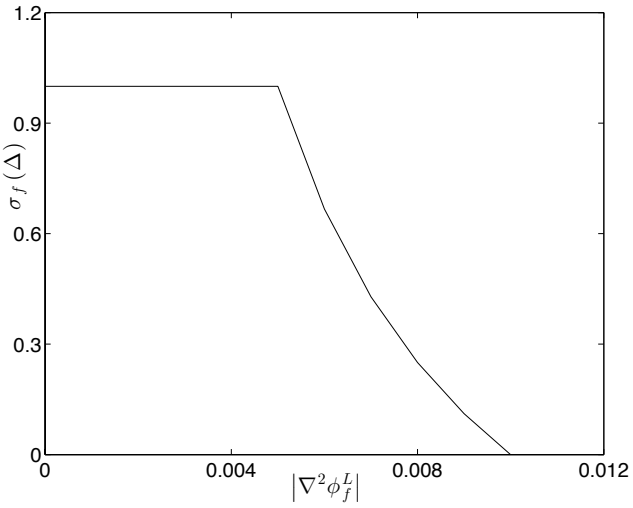


Figure 8.7: Decreasing σ_f with an increasing rate of change in scalar gradient.

A value of $a = 0.01$ is selected based on the maximum expected value of $\nabla^2\phi_f^L$ for a unit scalar field that has been filtered with a Gaussian kernel of width $\sigma_f = 2\Delta$. Following the approach used in §5.4.4, there exists a relationship between the applied filter width and the maximum second derivative of the filtered field (Eq. 5.59); in the present context:

$$\left|\nabla^2\phi_f^L\right|_{\max} = \frac{1}{\sigma_f^2\sqrt{2\pi e}} \tag{8.14}$$

Evaluating Eq. 8.14 for $\sigma_f = 2\Delta$ yields a maximum second derivative of 0.06. The value of a is then chosen to be well below this value, to ensure that filtering is removed from sharp changes in the scalar field. This also allows for the shift in the maxima and minima of $\nabla^2\phi_f^L$ resulting from the pre-filtering of the Lagrangian reconstruction: the effect of the shift will be most significant at those maxima and minima, but becomes negligible (i.e. less than Δ) for lower absolute values.

It is also necessary to select a suitable top-hat filter width w_{TH} for the actual filtering of the Lagrangian field. For this it is useful to consider how the number of filter operations N_f varies with that top-hat filter width; Fig. 8.8 presents such a trend, for the example of $a = 0.01$ and $b = \Delta$, with a selection of values for w_{TH}. Immediately apparent is the increase in N_f as w_{TH} is reduced, which is due to the factor α_{TH} (Eq. 8.12) tending to infinity as w_{TH} approaches Δ.

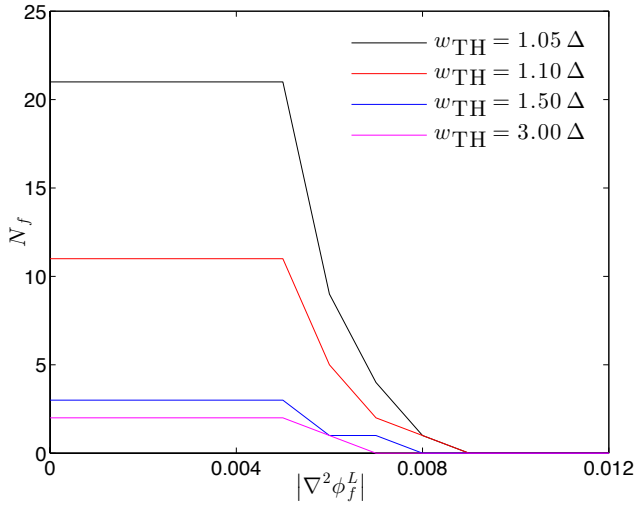


Figure 8.8: Variation in N_f for different top-hat filter widths.

Figure 8.8 suggests that the rounding operation in Eq. 8.11, which ensures an integer number of filter operations, causes wider top-hat filters to deliver a relatively inaccurate description of the relationship between $\left|\nabla^2\phi_f^L\right|$ and σ_f. Most notably the widest filter ($w_{\mathrm{TH}} = 3\Delta$) does not perform any filtering beyond $\left|\nabla^2\phi_f^L\right| = 0.007$, rather than the intended limit of $a = 0.01$. This is highlighted if Eq. 8.11 is re-arranged to determine an effective width σ_{eff} as a function of N_f and w_{TH}:

$$\sigma_{\mathrm{eff}} = \sqrt{\frac{N_f}{\alpha_{\mathrm{TH}}}} \tag{8.15}$$

Applying Eq. 8.15 to the trends presented in Fig. 8.8 allows the effective Gaussian filter width to be compared to the intended width determined from Eq. 8.13, and this is shown in Fig. 8.9.

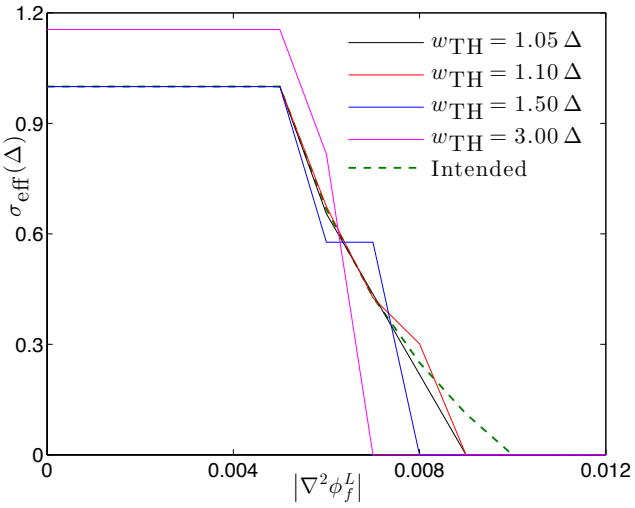


Figure 8.9: Effective Gaussian filter width for different top-hat filter widths.

It can be seen that the relationship between $\left|\nabla^2\phi_f^L\right|$ and σ_{eff} more closely matches the intended trend as w_{TH} is reduced. However, as stated previously, the number of filter operations will also increase rapidly as w_{TH} approaches Δ, which will result in an increase in computational cost. A minimum top-hat filter width of $w_{\text{TH}} = 1.05\Delta$ is therefore suggested.

A final consideration is the effect of a step change in the calculated value of σ_f from one cell to another, particularly as σ_f approaches zero in the region of a sharp change in scalar gradient. In such a case it is possible that N_f may change from zero in cell i to a large number in an adjacent cell $(i+1)$, particularly where w_{TH} is small (i.e. the scaling factor α_{TH} is large). The scalar value in i thus becomes an edge condition for the filtering process in cell $(i+1)$; however, that scalar value will include some noise from the Lagrangian reconstruction (as it must be near to a sharp change in scalar gradient). As the filtering operations in $(i+1)$ are completed, the value in $(i+1)$ will thus tend towards the (possibly incorrect) value in i, while the adjacent cells $(i+2)$, $(i+3)$, $\ldots(i+N_f)$ will also be affected. Although the error in i is relatively small in terms of the overall accuracy of the solution, the effect of the noisy value in i becoming an edge condition to the adjacent filtered cells may have a more significant impact.

To mitigate this effect the number of filter operations in each cell is modified so that the difference in N_f from one cell to the next is no more than one. This is achieved by setting the value of N_f in a cell to be no greater than one operation more than in the adjacent cells, i.e.:

$$N_f^{\text{lim}}(i) = \min\left[N_f(i-1)+1,\ N_f(i),\ N_f(i+1)+1\right] \tag{8.16}$$

Equation 8.16 is applied repeatedly until the values in N_f are unchanged by any further applications. Although noise in the unfiltered cells will still influence the filtered values, this is essentially unavoidable; however, the application of Eq. 8.16 should limit this influence as much as possible.

8.2.4 Testing: Two Dimensions

The filter localisation algorithm described in §8.2.3 is now tested with the cases listed in Table 8.2. This is done to verify that the formulation of the algorithm is correct, that the selected values of a, b and w_{TH} are reasonable, and that the algorithm is capable of delivering an improvement in error reduction compared to the (more simple) application of a uniform filter width at every point.

To assess the performance and behaviour of the algorithm the local values of $\left|\nabla^2\phi_f^L\right|$ and σ_f are inspected for the (sharp) initial and (smooth) final fields of Case F05LS. In each of Figs. 8.10 and 8.11 parts (a) and (b) show the calculated values of $\left|\nabla^2\phi_f^L\right|$ for each direction, while (c) and (d) show the corresponding values of σ_f as determined from Eq. 8.13. Also included are the pre-filtering (e) and post-filtering (f) error fields, to provide an indication of the effectiveness of the algorithm in removing the noise from the solution. Each plot shows only the 60×60 computational cells centred around the mid-point of the initial scalar field. Plots of the scalar fields themselves are not included here; the initial and final conditions shown in Figs. 7.1, 7.2 and 7.7 may be referred to as necessary.

Figures 8.10(a–b) show that the localisation algorithm accurately identifies the edges of the initial scalar field, where the second derivative is at its maximum. The highest values of $\left|\nabla^2\phi_f^L\right|$ in Figs. 8.10(a–b) – around the sharp edges at the centre of the disc – are greater than the scale indicates, but the range of values displayed is limited to $a = 0.01$ to ensure that the rest of the field is clearly visible on that scale. Figs. 8.10(c–d) show that the relationship between $\left|\nabla^2\phi_f^L\right|$ and σ_f (Eq. 8.13) correctly prevents filtering from being applied near to these sharp edges. A comparison of Figs. 8.10(e) and 8.10(f) reveals that the localisation algorithm has successfully reduced the error in regions around the outside edge of the modified Zalesak disc, and to some extent across the interior, although it appears that errors may have been slightly increased around the disc circumference. Such an increase is probably due to a degree of 'over-filtering' within this region, where some useful high-frequency information has been removed in addition to the undesirable noise. This issue may be resolved by a modification (reduction) of the parameter a.

There is a narrow strip of very high error (much larger than the limits of the chosen scale) around the edge of the central disc slot, in a region of the domain where no localised filtering is applied. This high error is representative of the 'chamfer' in the Lagrangian reconstruction, originally identified in the analysis of Case Z8 in Chapter 6, for which a solution will be proposed in §8.3.

The second derivative fields shown in Figs. 8.11(a–b) for the relatively smooth final field of Case F05LS indicate that the pre-filtering of ϕ^L adequately reduces the noise in the Lagrangian field, thus keeping $\left|\nabla^2\phi_f^L\right|$ free from any spurious high values. Although the effect of the noise on the second derivative is still visible, it has been attenuated sufficiently to prevent any influence on the local values of σ_f shown in Figs. 8.11(c–d). Case F05LS is sufficiently smooth that filter localisation is almost entirely unnecessary (with the exception of some small regions around the disc slot), and this is exactly the behaviour required of the algorithm for such smooth fields. Comparing the pre- and post-filtering error fields in Figs. 8.11(e) and 8.11(f), the benefit of filtering is more apparent than in the previous case; peak error values have been reduced across the entirety of the disc, while the error in some regions has been reduced almost to zero.

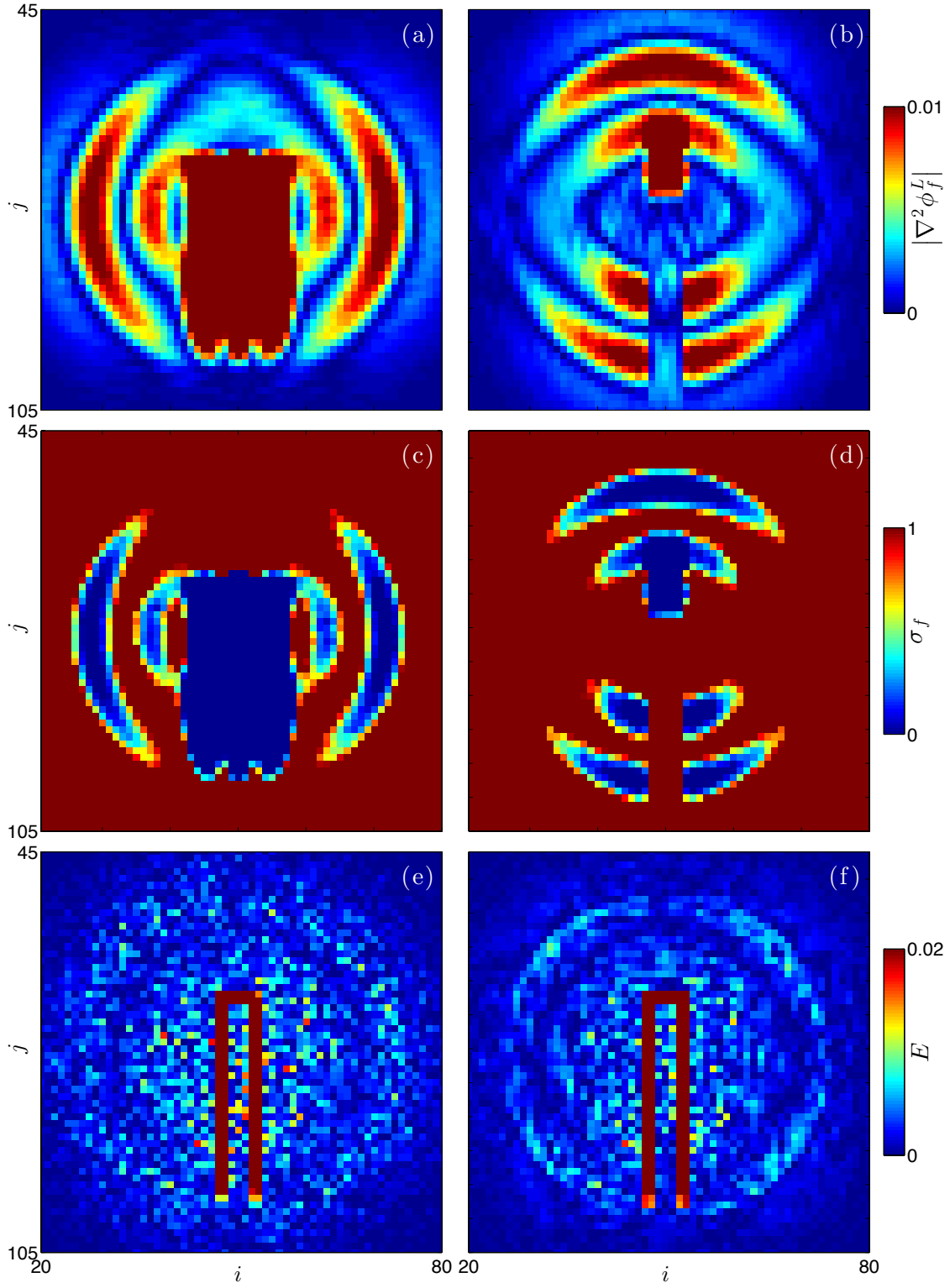


Figure 8.10: Initial field: Second derivative of ϕ_f^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding σ_f; (e) pre- and (f) post-filtered error field.

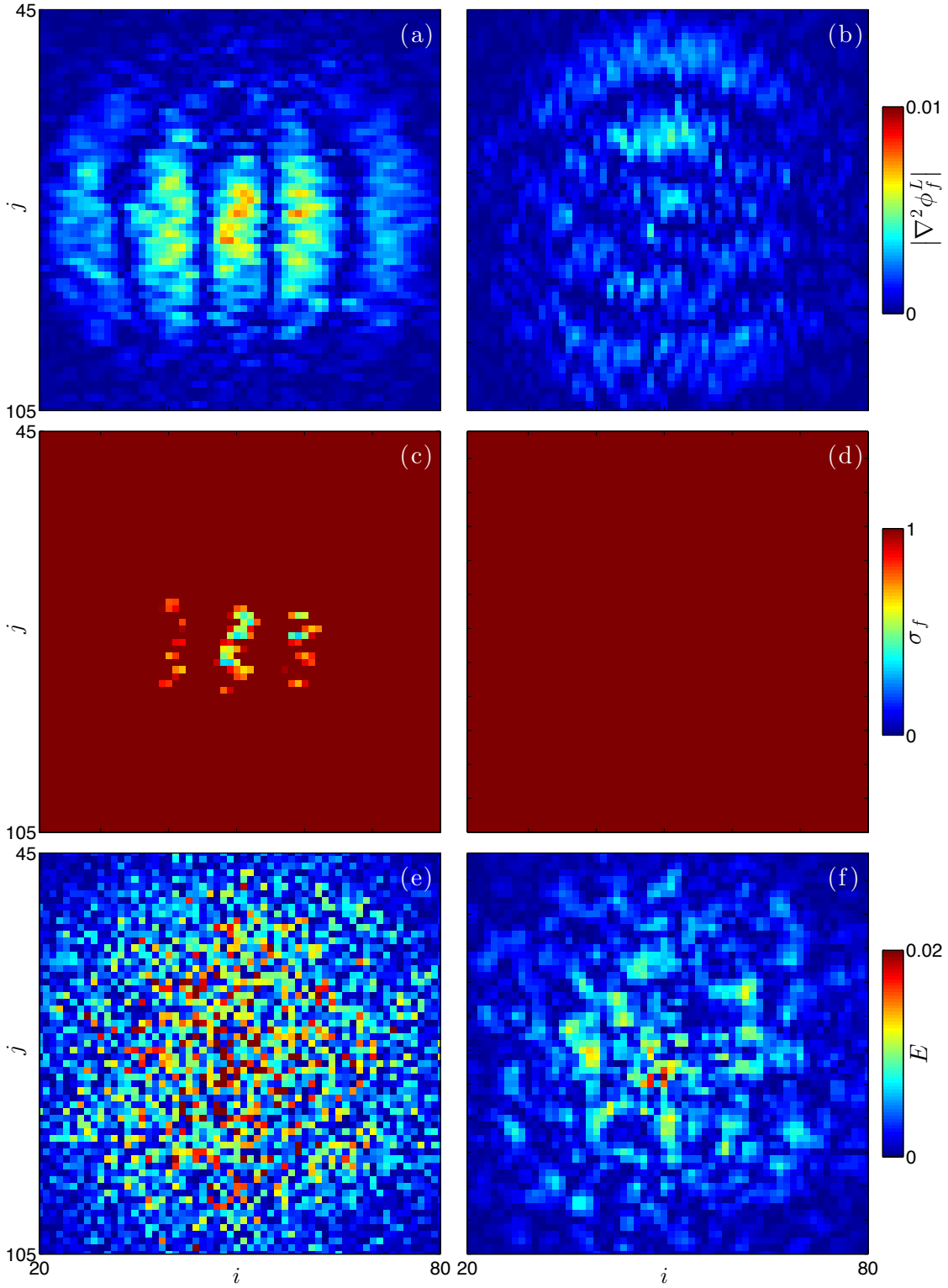


Figure 8.11: Case F05LS: Second derivative of ϕ_f^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding σ_f; (e) pre- and (f) post-filtered error field.

The performance of filter localisation for all five cases is assessed by comparing the total absolute error between the actual and analytical solutions of each case, for both the uniform filter width approach (applying Gaussian filter widths in the range $0 \leq \sigma_f \leq 1.5\Delta$) and the localising algorithm proposed here. Figure 8.12 displays the total errors expressed as a percentage of the initial (unfiltered) error for each case, where the black lines show the change in error as the filter width for uniform filtering is increased, and the red lines indicate the error resulting from localised filtering. The uniform and localised filtering performance is also compared in Table 8.3, which shows: for the uniform approach, the maximum percentage reduction in total error and the filter width at which that reduction occurs; and for the localising algorithm, the overall percentage reduction in total error.

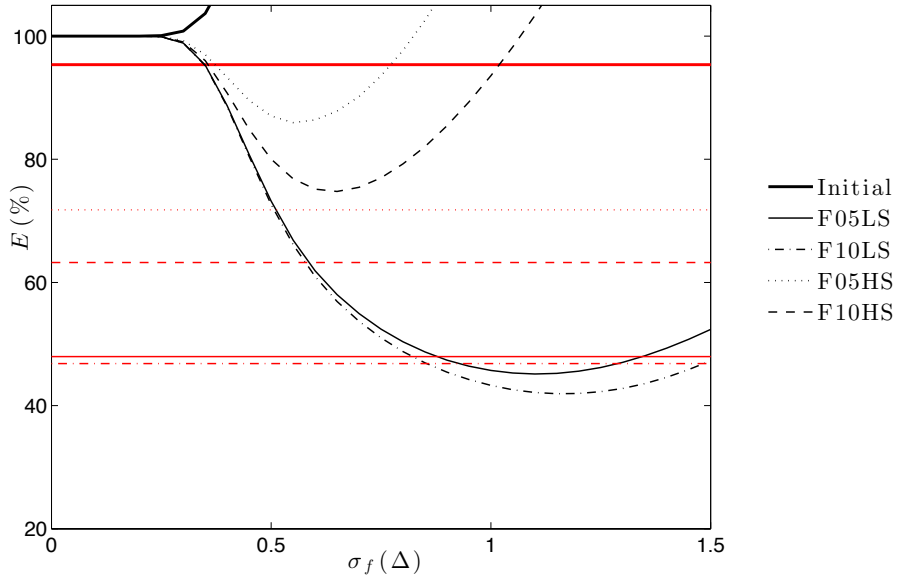


Figure 8.12: Post-filtering total error: uniform (black) and localised (red).

Table 8.3: Maximum change in total error, uniform vs. localised filtering.

Case	Uniform		Localised
	σ_f (Δ)	E (%)	E (%)
Initial	-	0.00	-4.64
F05LS	1.10	-54.87	-52.04
F10LS	1.17	-58.08	-53.18
F05HS	0.56	-14.06	-28.21
F10HS	0.64	-25.25	-36.77

This analysis demonstrates that the filter localisation algorithm performs well for all five cases tested: it is capable of reducing errors due to noise in the Lagrangian field more effectively than a uniform filtering approach for the sharper (initial and high Schmidt number) cases, while being almost as effective for the smooth (low Schmidt number) cases. Notably the localisation algorithm has successfully reduced the total error for the initial field by 5%, which no amount of uniform

filtering can achieve. A large proportion of the total error in that initial field will be due to the large local errors around the disc slot (due to the 'chamfer' in the Lagrangian reconstruction) such that the degree of noise reduction for the initial field may be higher than that shown in Fig. 8.12.

8.2.5 Testing: Three Dimensions

To ensure that the filter localisation algorithm exhibits a similar performance in three dimensions, the cases listed in Table 8.2 are expanded accordingly: the modified Zalesak disc now becomes a sphere, with the central slot extended through its entirety. The outer surface of the sphere is smoothed, as for the two-dimensional configuration, such that a 'slice' through the mid-plane of the sphere has the same appearance as the equivalent two-dimensional modified Zalesak case.

The filter localisation algorithm is extended into three dimensions by applying the method in the new (lateral) direction, since the algorithm has been formulated such that its implementation is independent for each dimension. Correspondingly the simplified `PsiPhi` code that has been used thus far also has a third dimension added, and the various routines specific to the E-L method are modified as necessary.

The analysis presented in Fig. 8.12 for the two-dimensional cases is repeated here, as shown in Fig. 8.13. Table 8.4 presents, for the uniform approach, the maximum percentage reduction in total error and the filter width at which that reduction occurs; and for the localising algorithm, the overall percentage reduction in total error. From this data it can be seen the performance of the localisation algorithm in three dimensions is generally better than that of the two-dimensional equivalent, although the uniform filtering method does provide a slightly increased improvement in comparison. However, there is no single uniformly-applied filter width that would out-perform the localised filtering algorithm, for all five cases, in both two and three dimensions. This advocates the application of the algorithm for the more complex test cases presented in the remaining chapters.

Table 8.4: Maximum change in total error in 3D, uniform vs. localised filtering.

Case	Uniform		Localised
	σ_f (Δ)	E (%)	E (%)
Initial	0.48	-6.83	-27.06
F05LS	1.35	-77.39	-66.17
F10LS	1.38	-77.84	-64.77
F05HS	0.81	-56.05	-46.54
F10HS	0.93	-61.69	-54.84

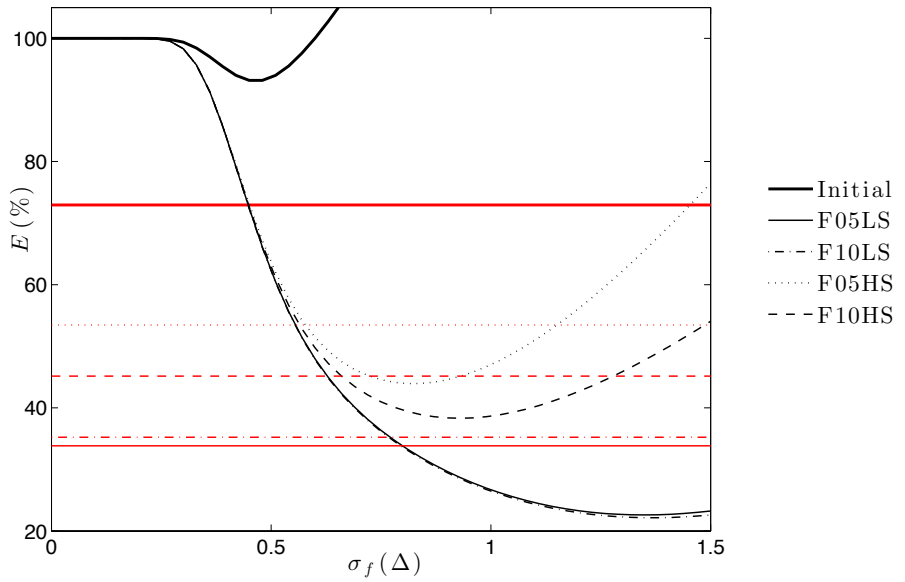


Figure 8.13: Post-filtering total error, in 3D: uniform (black) and localised (red).

8.3 Inverse Filtering

An artefact of the E-L method discussed in §6.4.2, and identified again in §8.2.4, will now be considered. E-L solutions containing very sharp 'edges' – such as the transported scalar fields for Cases Z7 to Z10 in Chapter 6 – exhibit a 'chamfer' around those sharp edges. Further investigation indicates that the chamfer originates within the Lagrangian field, and this leads to the suggestion that it is caused by a small degree of filtering introduced by the weighted averaging calculation used in the particle reconstruction (shown in Fig. 5.15).

To gain further insight into this aspect of the method, it is instructive to re-examine the way in which an Eulerian description of the Lagrangian field is constructed. Consider a one-dimensional computational cell i of unit scalar value, centred at $x = 1.5$, with 'empty' neighbours $(i-1)$ at $x = 0.5$ and $(i+1)$ at $x = 2.5$. Cell i is populated with 80 Lagrangian particles, each particle having a scalar weight of 0.0125. Initially all of the particles are placed at the centre of cell i, and an Eulerian description of this particle field is subsequently reconstructed by applying a top-hat filter of unit width to the scalar contribution of each particle. A visual representation of the reconstruction is shown in Fig. 8.14, where the filtered contribution of each particle is displayed as 20 'boxes'. The boxes for each particle are given a unique colour, and are 'stacked up' as each particle is added to the reconstruction.

The reconstructed field is exactly $\phi = [0,\ 1,\ 0]$ across the three cells as the particles are all located at the centre of cell i, so that all of the boxes stack directly on top of one another. However, due to the creation of 'moiré' interference patterns (as shown in Fig. 5.12) it is unwise to initialise all of the particles at the cell centre. The procedure is therefore repeated with a uniform random particle distribution: in this case a particle at (for example) position $x = 1.65$ – near to the centre of the populated cell – will contribute 85% of its scalar weight to cell i, and the remaining 15% to cell $(i+1)$. The resulting reconstruction now resembles a pyramid, as shown in Fig. 8.15.

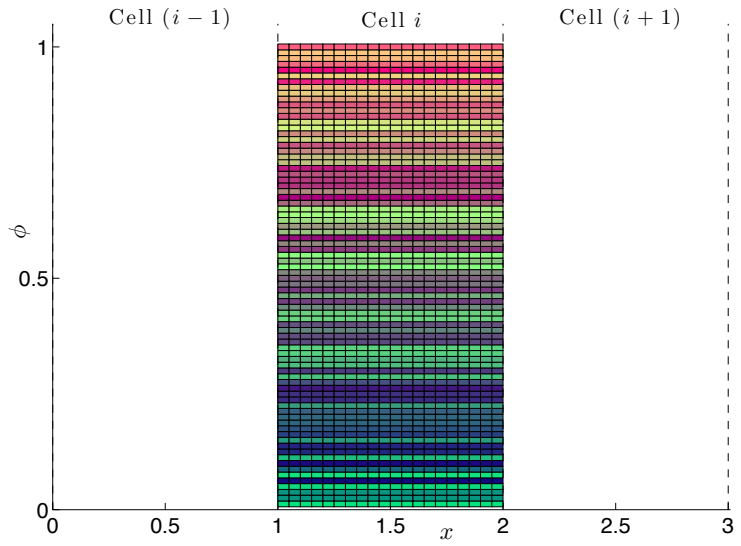


Figure 8.14: Lagrangian reconstruction: centred particle distribution.

The scalar values in the cells $(i-1)$, i, and $(i+1)$, determined by counting the number of boxes in each cell and multiplying by their individual scalar weight, are now $\phi = [0.123, 0.746, 0.131]$, respectively. In the limit, as the number of particles and the number of 'boxes' per particle tends to infinity, it can be shown that the unit scalar becomes spread over its neighbours in the proportions $[b_1,\, a_1,\, b_1] = \frac{1}{8}\,[1, 6, 1]$, using the nomenclature to describe a top-hat filter kernel introduced in §5.3.1. The effect of the reconstruction is therefore equivalent to applying a top-hat filter of width $a_1^{-1} = \frac{4}{3}$ to the initial field.

If this exercise is repeated in two dimensions, where a single cell $(i,\, j)$ is initialised with a unit scalar value at the centre of a a 3×3 array of cells, the resulting scalar field has the values:

$$\phi = \begin{bmatrix} c_2 & b_2 & c_2 \\ b_2 & a_2 & b_2 \\ c_2 & b_2 & c_2 \end{bmatrix} = \frac{1}{64} \begin{bmatrix} 1 & 6 & 1 \\ 6 & 36 & 6 \\ 1 & 6 & 1 \end{bmatrix} \tag{8.17}$$

In this case, the effect of the reconstruction is equivalent to applying a top-hat filter of width $a_2^{-1/2} = \frac{4}{3}$ – in other words, the same filter as in one dimension. Finally, in three dimensions, the resulting scalar field has values of $a_3 = \frac{216}{512}$ in the centre cell $(i,\, j,\, k)$; $b_3 = \frac{36}{512}$ in the directly-adjacent neighbours (shared faces); $c_3 = \frac{6}{512}$ in the in-plane diagonal neighbours (shared edges), and $d_3 = \frac{1}{512}$ in the diagonal-diagonal neighbours (shared corners). The top-hat filter equivalent to the effect of the reconstruction again has a width of $a_3^{-1/3} = \frac{4}{3}$.

8.3.1 Deconvolution

Having demonstrated that the chamfer problem is caused by a filtering inherent to the Lagrangian reconstruction, a remedy may now be formulated. If we consider the problem to be the convolution of the field with a (known) filter function, the solution is to perform a *deconvolution* of the resulting

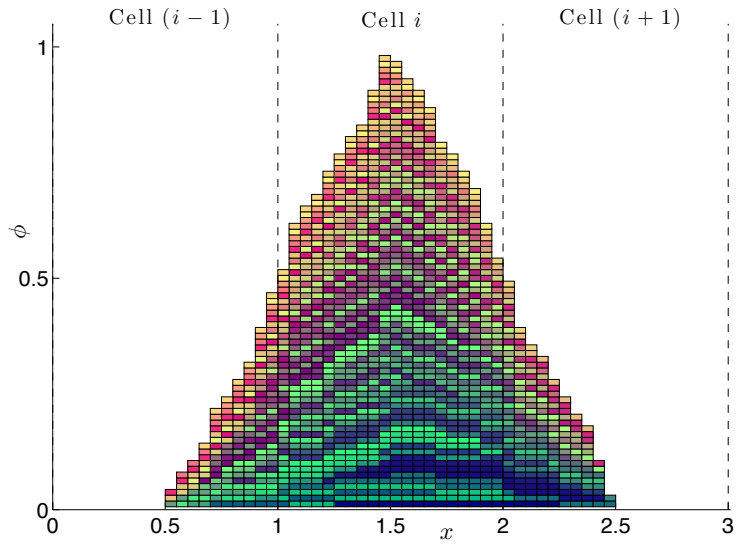


Figure 8.15: Lagrangian reconstruction: uniform random particle distribution.

field. The initial filtering may be expressed in a compact form, analogous to the filtering operation described by Eq. 5.2, as:

$$\phi_f(x) = \phi(x) * b(\delta x) \tag{8.18}$$

Where ϕ_f is the filtered field, b is the low-pass (top-hat) filter function, and $*$ represents the convolution operation. Alternatively, this filtering may be written in the frequency domain as:

$$\Phi_f(\omega) = \Phi(\omega)\mathcal{B}(\delta\omega) \tag{8.19}$$

Where Φ, Φ_f, and $\mathcal{B}$ are the Fourier transforms of the original and filtered fields and the low-pass filter kernel, respectively, and ω is the frequency. The high-pass filter $\mathcal{J}$ required to undo the action of $\mathcal{B}$ on Φ may then be determined simply as:

$$\mathcal{J}(\delta\omega) = \frac{1}{\mathcal{B}(\delta\omega)} \tag{8.20}$$

While the deconvoluted field (still in the frequency domain) may be calculated as:

$$\Phi(\omega) = \Phi_f(\omega)\mathcal{J}(\delta\omega) \tag{8.21}$$

An inverse Fourier transform of the field Φ would thus yield the desired deconvoluted field ϕ in the spatial domain. However, Fourier transforms are computationally expensive, and, for simulations performed on parallel computer architectures, demand a large amount of communication between processes; it would therefore be preferable to avoid having to determine Φ_f. An alternative, then, would be to take the inverse Fourier transform of the high-pass filter $\mathcal{J}$ to obtain its spatial equivalent $j(\delta x)$, considering that this operation would only need to be performed once (as the low-pass filter

b is known and does not change). The deconvolution may subsequently be performed in space, as:

$$\phi(x) = \phi_f(x) * j(\delta x) \tag{8.22}$$

Unfortunately, the kernel of the resulting high-pass filter j requires a number of support points of at least $n_f = 3$ in each direction – as shown in Fig. 8.16 – to ensure that it is sufficiently represented. This means that its implementation in two or three dimensions will tend to incur a relatively large computational cost (similar to the sinc function, as discussed in §5.3.1).

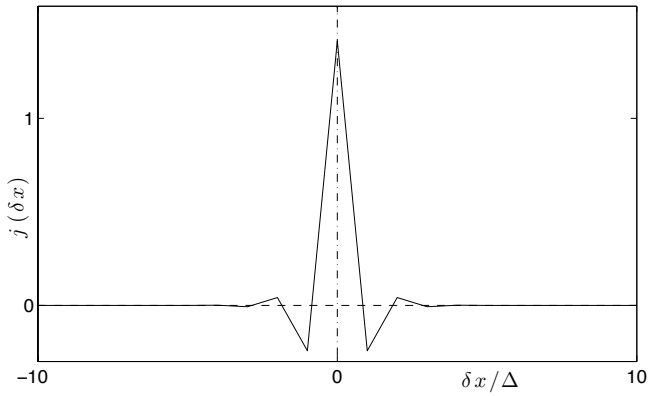


Figure 8.16: The high-pass filter kernel j required for the deconvolution of ϕ_f.

A more practical solution has been proposed by Jansson et al. [209], where the high-pass filter and deconvolution operation may be replaced with an iterative procedure applying only the original (known) low-pass filter b. An initial guess for the deconvoluted solution ϕ is made, based on the present filtered field ϕ_f:

$$\phi^0 = \phi_f \tag{8.23}$$

The m-th solution for ϕ is then calculated as:

$$\phi^{m+1} = \phi^m + \lambda \left(\phi^0 - \phi^m * b\right) \tag{8.24}$$

Where λ is a convergence factor. As the convoluted field ϕ_f and low-pass filter b are known, the only decisions to be made are to choose a suitable value for λ, and to estimate the total number of deconvolution operations N_d that must be performed to reach a satisfactory solution. Testing of the deconvolution algorithm indicates that the procedure remains stable with a convergence factor of $\lambda = 1$, due to the narrowness of the filter b (over-relaxation ($\lambda > 1$) is likely to result in a divergence in ϕ, however). In terms of N_d, a compromise should be found between the 'convolution error' – i.e. the degree by which the solution remains filtered after a given number of deconvolution operations – and the computational cost of the iterative procedure.

In order to provide some guidance for a suitable value of N_d, the following test is performed: a one-dimensional scalar field ϕ^0, discretised over N cells of unit size, is initialised with values $\phi^0(i) = 1$ in the range $1 \leq i \leq N/2$ and $\phi^0(i) = 0$ otherwise. A top-hat filter of width $w_{\mathrm{TH}} = \frac{4}{3}$

is then applied, generating a field ϕ_f to represent a reconstruction of the equivalent Lagrangian representation of ϕ^0 (where the particle density is assumed to be large, so that the noise in the Lagrangian field is negligible). This filtered field is then deconvoluted using the iterative procedure described above, for an increasing number of operations, and the total error E (i.e. the sum of the absolute difference) between the initial field ϕ^0 and the filtered/deconvoluted field ϕ is calculated. The resulting dependence of E on N_d is shown graphically in Fig. 8.17.

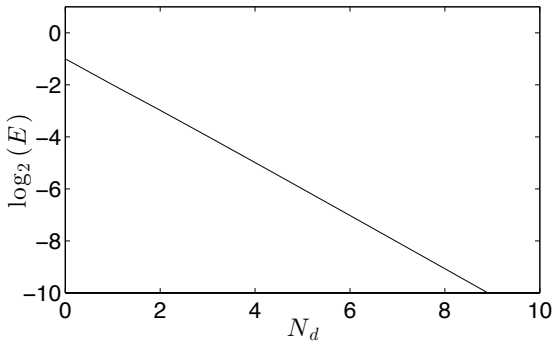


Figure 8.17: Reducing deconvolution error with increasing deconvolution operations.

Given that the gradient of $\log_2(E)$ in Fig. 8.17 is -1, it can be inferred that:

$$E(N_d) \propto \frac{1}{2^{N_d}} \tag{8.25}$$

This test indicates that the total error in the deconvoluted field is halved with every iteration, for this particular topology. Furthermore, by filtering ϕ^0 with a variety of Gaussian filter kernel widths prior to the application of the top-hat filter, it is possible to determine how the number of iterations required to attain a given value of E varies with the smoothness of ϕ_f. This smoothness is estimated using an approximation of the second derivative of ϕ_f, of the form in Eq. 8.10, and deconvolution iterations are performed until E has dropped to a negligible value (chosen to be 0.01% of the initial total error). The Gaussian kernel width is varied over the range $0 \leq \sigma_f \leq 2$. The relationship between the maximum absolute value of the second derivative of ϕ_f, and the number of iterations required to attain the target error, is shown in Fig. 8.18.

This analysis shows that the total error for smoother fields (where $\left|\nabla^2\phi_f\right|_{\max} < 0.25$) drops more rapidly than for sharp fields, as fewer iterations are required to achieve the target error, while the sharpest fields require between 12 and 14 deconvolution operations. Based on this, for a scalar field in the range $0 \leq \phi \leq 1$, a maximum of 14 iterations will be applied.

8.3.2 Deconvolution Localisation

A disadvantage of applying inverse filtering to the reconstructed Lagrangian field is that, in addition to correcting any attenuation of the desirable high-frequency information, the procedure will also amplify any (undesirable) noise. However, as for the low-pass filtering algorithm described in §8.2, it is possible to formulate a modification to the procedure that localises its implementation, depending

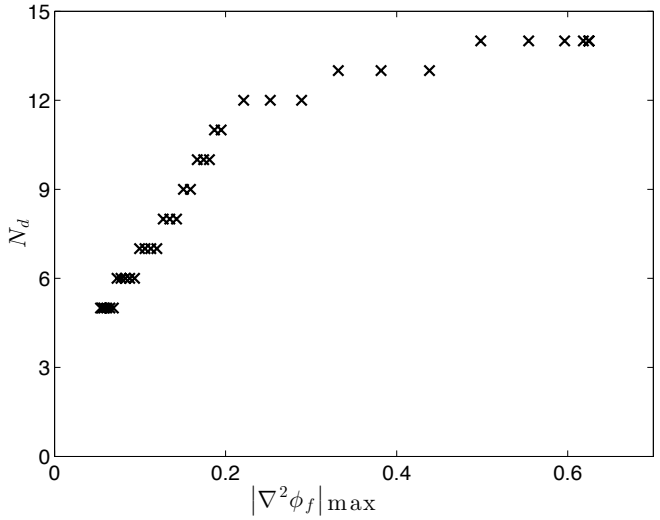


Figure 8.18: Variation of N_d required to attain 0.01% deconvolution error with $\left|\nabla^2\phi_f\right|_{\max}$.

on the shape of the underlying field.

Considering the trend shown in Fig. 8.18, it would seem reasonable to locally vary the number of deconvolution operations as a function of the second derivative $\left|\nabla^2\phi^L\right|$; however, it should also be considered that in regions of the scalar field that vary smoothly, deconvolution is more likely to increase error by amplifying noise in the Lagrangian field than it is to reduce it by undoing the filtering in the reconstruction. The localisation is thus performed as follows: once the Lagrangian field has been reconstructed the absolute value of $\left|\nabla^2\phi^L\right|$ is determined independently in each direction, using an approximation similar to that given in Eq. 8.10. For each computational cell i, the local value $N_d(i)$ is then set as:

$$N_d(i) = \begin{cases} 14 & \text{where} \quad \left|\nabla^2\phi^L(i)\right| > 0.25 \\ 0 & \text{otherwise} \end{cases} \tag{8.26}$$

During deconvolution an update to the deconvoluted field $\phi(i)$ is performed only if any further iterations are required in that cell:

$$\phi^{m+1}(i) = \begin{cases} \phi^m(i) + \lambda\mathcal{I}(i) & \text{if} \quad N_d^m(i) > 0 \\ \phi^m(i) & \text{if} \quad N_d^m(i) = 0 \end{cases} \tag{8.27}$$

Where the deconvolution update $\mathcal{I}$ in the cell (i) is:

$$\mathcal{I}(i) = \phi^0(i) - \phi^m(i) * b \tag{8.28}$$

Once one iteration of the update has been performed according to Eq. 8.27 across all cells the non-zero values in N_d are reduced by one, and the procedure is repeated until all of the values in N_d are zero.

As for the localised filtering algorithm presented is §8.2 the implementation of the deconvolu-

tion algorithm in three dimensions is relatively straightforward, as it has been formulated to be independent for each direction.

8.3.3 Testing

Testing of the proposed inverse filtering algorithm is performed using a similar strategy to that employed in §8.2. The five cases summarised in Table 8.2 are revisited, and localised deconvolution is applied to each case. Local values of the second derivative field $\left|\nabla^2\phi^L\right|$ for the initial case, in the horizontal and vertical directions, are shown in Figs. 8.19(a) and 8.19(b) respectively. The resulting values of N_d are displayed in Figs. 8.19(c) and 8.19(d), and the pre- and post-deconvolution error fields are presented in Figs. 8.19(e) and 8.19(f).

Figures 8.19(a) and 8.19(b) suggest that the algorithm correctly identifies the sharpest gradients in the initial scalar field, where the highest values $\left|\nabla^2\phi^L\right|$ are found around the edges of the slot in the modified Zalesak disc. From Figs. 8.19(c) and 8.19(d) it can be seen that the algorithm subsequently applies the iterative inverse filtering method only in those regions (i.e. where $N_d > 0$). A comparison of the two local error plots shown in Figs. 8.19(e) and 8.19(f) shows that the deconvolution algorithm has successfully removed the error around the edge of the disc slot, without adversely affecting the rest of the scalar field. Within the region of $N_d > 0$ (for either direction), the mean error in each cell is reduced by an order of magnitude (from 0.121 to 0.011).

More generally, the performance of the algorithm may be assessed by considering the change in total error pre- and post-deconvolution for each of the cases listed in Table 8.2, and these results are summarised in Table 8.5.

Table 8.5: Total error before and after localised deconvolution.

Case	E (pre)	E (post)	$\pm E(\%)$
Initial	22.04	9.72	-55.91
F05LS	20.97	20.97	0.00
F10LS	35.52	35.52	-0.00
F05HS	19.71	19.71	-0.00
F10HS	33.46	33.46	-0.00

The values shown in Table 8.5 reveal that the deconvolution algorithm achieves a significant reduction in the total error E for the initial field (containing the sharpest 'edges'). All of the other fields are unaffected by the algorithm, however, as it correctly identifies that each of these scalar fields contains no very sharp changes in gradient.

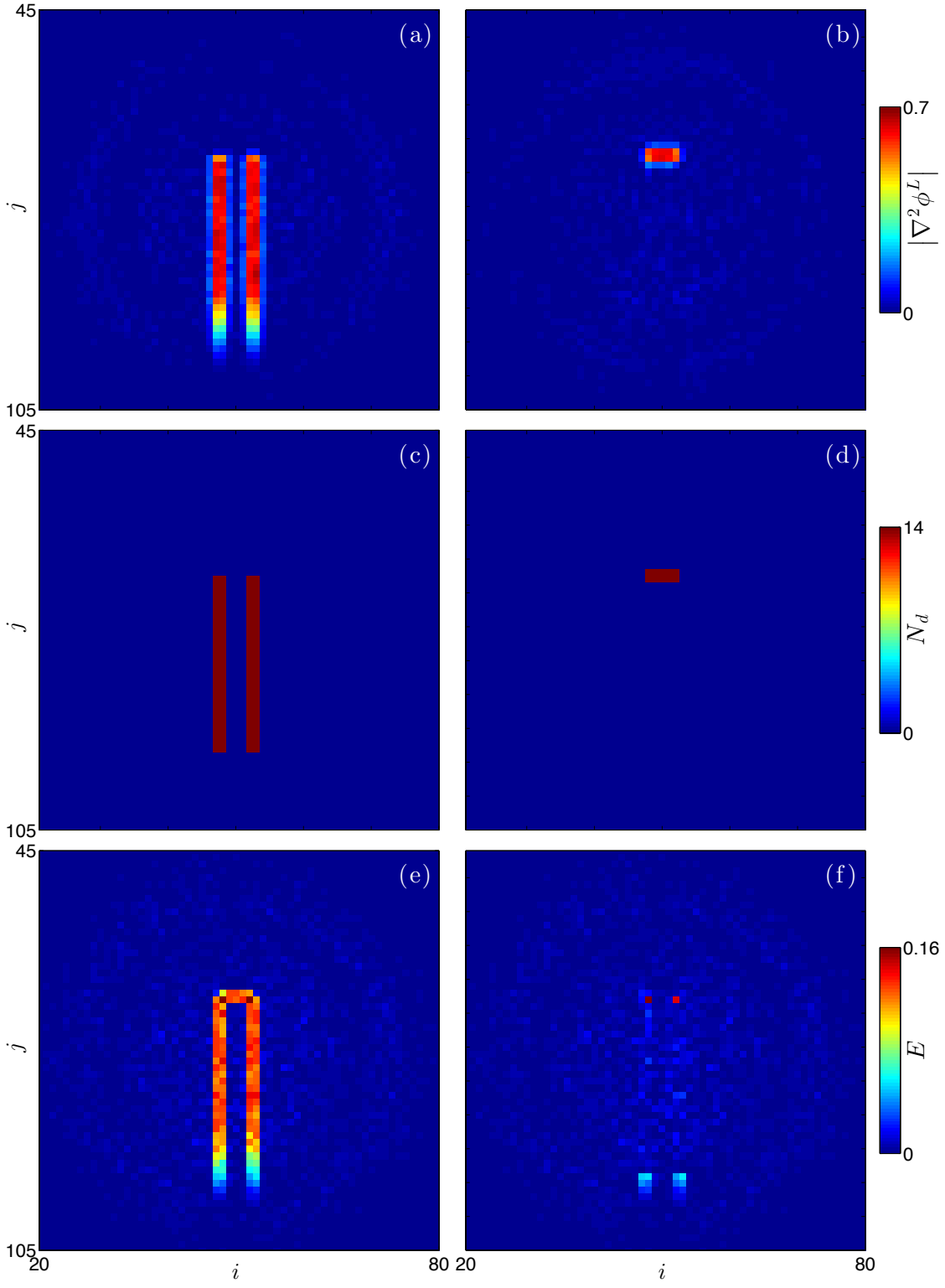


Figure 8.19: Initial field: Second derivative of ϕ^L in (a) horizontal and (b) vertical directions; (c) and (d), corresponding N_d; (e) pre- and (f) post-deconvolution error field.

Testing is also performed in three dimensions with the modified Zalesak disc configurations used for three-dimensional testing of the filter localisation algorithm in §8.2. Results for total error pre- and post-deconvolution, and the percentage change in error, are presented in Table 8.6.

Table 8.6: Total error before and after localised deconvolution, in three dimensions.

Case	$E \times 10^3$ (pre)	$E \times 10^3$ (post)	$\pm E(\%)$
Initial	3.02	2.32	-23.02
F05LS	5.01	5.01	0.01
F10LS	8.33	8.33	0.05
F05HS	4.84	4.85	0.25
F10HS	7.50	7.51	0.20

The three-dimensional performance of the deconvolution algorithm is found to be similar to the performance in two dimensions, although the results are now slightly degraded for the smoother fields. In particular, for the high Schmidt number cases F05HS and F10HS, it is suggested that the algorithm has applied deconvolution in limited regions around the disc slot (where the particle noise, combined with the underlying shape of the scalar field, has caused local values of $\left|\nabla^2\phi^L\right| > 0.25$). The detrimental effect of the amplification of the noise in these regions has then out-weighed any improvement in the solution gained from the deconvolution. The increase in error is relatively small (less than 0.25%), however, and the considerable improvement that the algorithm is able to deliver at the sharpest edges in a scalar field should not be overlooked. As for the algorithm in two dimensions, for the initial case the mean error in each cell within the region $N_d > 0$ is found to have dropped by an order of magnitude (from 0.113 to 0.022). The percentage improvement in results for the initial case is less than for the same case in two dimensions; however, the proportion of the 'surface area' of the three-dimensional scalar field in which inverse filtering is applied – i.e. where the error reduction occurs – is less than the proportion of the 'perimeter' of the scalar field for the corresponding case in two dimensions.

8.3.4 Other Methods

Compensating for the degradation that a digital signal is prone to – either through unwanted low-pass filtering, or from the addition of noise – forms a significant part of the topic of Digital Signal Processing, and a great deal of research has been carried out in the field. It is worth noting that the inverse filtering method described here is relatively simplistic; many other methods exist, some of which combine deconvolution with noise reduction in a single process. Two notable examples of such alternatives are Wiener filtering [210], which is designed to implement an optimal balance of inverse and low-pass filtering in the frequency domain; and the wavelet restoration technique of Donoho [211], where inverse and low-pass filtering is instead performed in the wavelet domain.

8.4 Application to the Zalesak Cases

To conclude this chapter the particle removal, low-pass filter, and deconvolution algorithms that have been described thus far will now be implemented and tested using a selection of the Zalesak and modified Zalesak test cases (presented in Chapters 6 and 7, respectively). In order to assess the performance of the algorithms for each case, the metrics introduced in §6.4 are employed with and without each of the modifications introduced in this chapter applied.

8.4.1 Particle Removal

Firstly the particle removal mechanism is added to the Eulerian-Lagrangian Cases Z8 and Z9 for the original Zalesak disc, and the effect of its application is summarised in Table 8.7. To account for the fact that a Gaussian filter width of $\sigma_f = 2.5\Delta$ is used for the initial decomposition – rather than the suggested value of $\sigma_f = 5\Delta$ – the particle removal parameters β and λ, appearing in Eqs. 8.1 and 8.6 respectively, are set to $\beta = 0.05$ (reduced from the suggested value by a factor of four) and $\lambda = 0.35$ (reduced by a factor of two). This makes the mechanism less 'aggressive' in identifying particles for removal. To provide an at-a-glance indication of the impact of applying particle removal measures that have improved are shown in a bold green font, while measures that have deteriorated are in red.

Table 8.7: Impact of particle removal, Cases Z8 and Z9.

Case	Particle Removal	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)			Time (s)
							Add.	Rem.	Final	
Z8	×	-4.52	-2.40	60.65	37.46	21.37	2.119	-	4.813	106.89
	✓	-10.07	-5.20	75.64	50.85	27.40	3.609	1.561	4.742	**106.09**
Z9	×	-4.99	-7.12	76.52	90.15	32.05	0.207	-	0.485	15.52
	✓	-13.98	-10.37	89.27	106.41	40.10	0.411	0.197	0.484	15.62

From the information in Table 8.7 it is apparent that the particle removal mechanism in this instance has a negative impact on the performance of the E-L method: all of the measured quantities have become worse, while the final particle numbers and computational times are only slightly affected. The only positive aspect that may be drawn from these results is the fact that the computational cost of the mechanism would seem to be relatively small, as the overall cost per particle per time-step is unchanged. Figure 8.20 compares 'slices' through the Eulerian fields from the solutions for Case Z8 with and without particle removal, and from this it may be inferred that, in this instance, the removal mechanism causes an increase in the susceptibility of the Eulerian field to oscillations during transport. These oscillations may be exacerbated by the small degree of noise that will be introduced into the Eulerian phase via the reconstruction of the transferred particle contribution.

Considering these results it is straightforward to conclude that particle removal is not beneficial for an infinite Schmidt number, translation-only simulation. This is not unreasonable, given that there should be no change in the shape of the Lagrangian field over the course of the simulation. Generally,

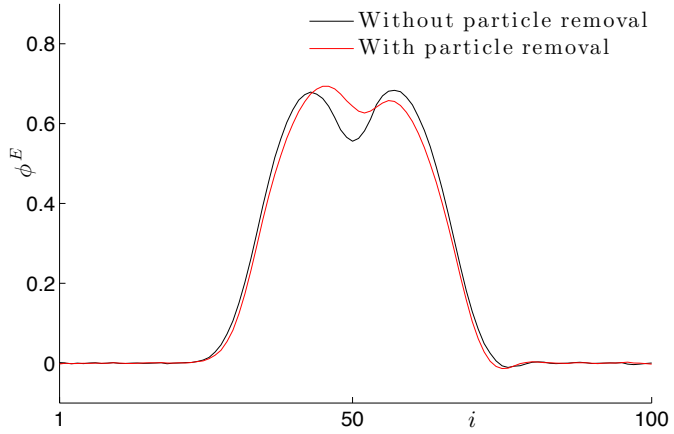


Figure 8.20: Case Z8: Transported Eulerian phase with and without particle removal.

therefore, particles that are de-activated in each removal operation are immediately replaced by the particle addition routine; without diffusion, there is no mechanism by which particles may end up in a region where they are not needed. The accumulation of error in the low-frequency field thus indicates that the particle removal routine is too 'aggressive', even for the current values of β and λ, suggesting that a further reduction in these parameters may improve the results.

It should be noted that the significant numbers of particles added are a result of the re-initialisation parameter γ being set to 0.25, instead of the typical unit value, to account for the restricted width of the Gaussian filter used for the initial decomposition.

The use of particle removal is included in repeated computations of the modified Zalesak cases ZM3 and ZM4, where the scalar field is rotated once at a relatively high Schmidt number ($Sc = 10$). For these tests the removal parameters β and λ are set to their suggested values of 0.2 and 0.7, respectively, to reflect the fact that the initial Eulerian-Lagrangian decomposition is now performed at the recommended filter width $\sigma_f = 5\Delta$. Table 8.8 summarises the results for these tests, where entries are again colour-coded to indicate the overall impact of the mechanism.

Table 8.8: Impact of particle removal, Cases ZM3 and ZM4.

Case	Particle Removal	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)			Time (s)
							Add.	Rem.	Final	
ZM3	×	-1.21	-2.38	21.37	12.02	16.00	0.082	-	3.778	573.17
	✓	-1.21	-3.28	31.12	**5.95**	20.36	5.186	6.544	2.339	**506.31**
ZM4	×	-5.20	-6.60	54.24	59.67	37.04	0.010	-	0.379	99.93
	✓	**-2.32**	**-5.06**	**54.13**	**30.15**	**36.31**	0.851	0.990	0.229	**94.79**

The inclusion of particle removal is generally beneficial for this configuration: the final particle number is reduced by around 40% for each case, resulting in computational savings of 12% and 5% respectively, the latter being less significant due to a smaller proportion of time being spent in the particle routines. A lower number of particles results in less noise in the final transported field, as

indicated by the general reduction in ΔTV (by around 50% in each case). Figure 8.21 compares slices through the Eulerian fields from the solutions for Case ZM3 with and without particle removal, and suggests that any gains from this noise reduction are balanced by an increased accumulation of error in the Eulerian phase, similar to (but less pronounced than) that observed for Cases Z8 and Z9. The physical diffusion in this configuration may be reducing the impact of Lagrangian noise introduced into the Eulerian phase; although the amplitude of that noise will be small, the rapid changes in gradient that it contains will make it more susceptible to the diffusivity of the scalar, meaning that it is quickly smoothed out.

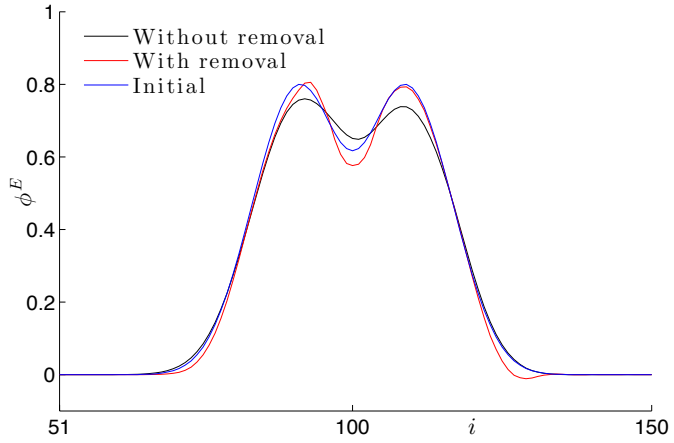


Figure 8.21: Case ZM3: Transported Eulerian phase with and without particle removal; initial Eulerian component for comparison.

Figure 8.21 also shows that the final Eulerian field with particle removal applied is quite similar to the initial Eulerian field. This suggests that the mechanism is operating as intended: as the scalar becomes smoothed by (physical) diffusion information is transferred to the Eulerian phase, which therefore maintains a fairly constant shape. With particle removal, physical diffusion is essentially represented only by the dispersion and removal of particles – rather than by a balanced dissipation of the Eulerian and Lagrangian phases – resulting in a more computationally efficient simulation.

Table 8.9 shows the effect of the particle removal mechanism on the low Schmidt number configuration in Cases ZM7 and ZM8, by comparing measures of accuracy with and without the mechanism applied.

It can be seen that the removal mechanism is highly effective: a significant proportion (up to 90%) of Lagrangian particles have been removed by the end of each simulation. This results in a marked improvement in the overall accuracy of both cases, as the noise in the reconstructed Lagrangian field is largely mitigated. In addition the decreasing number of particles over the course of the simulation results in a 19% reduction in computational cost for ZM7, and a 11% reduction for ZM8.

However, for this low Schmidt number configuration, the performance of an Eulerian-only simulation applying an eighth-order Central Differencing Scheme (CDS8) should also be considered: the high physical diffusivity in this case is effective in attenuating any oscillations that might develop

Table 8.9: Impact of particle removal, Cases ZM7 and ZM8; CDS8 for comparison.

Case	Particle Removal	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)			Time (s)
							Add.	Rem.	Final	
ZM7	×	-0.42	-2.42	19.94	14.61	14.11	0.033	-	3.729	563.13
	✓	**-0.01**	**-1.34**	**9.85**	**1.34**	**8.13**	1.111	4.114	0.693	**458.16**
ZM8	×	-2.79	-7.50	61.23	81.96	42.64	0.001	-	0.369	99.25
	✓	**-0.01**	**-3.73**	**15.66**	**6.05**	**12.62**	0.268	0.606	0.030	**88.02**
CDS8	-	-0.00	-0.75	10.63	-0.03	10.58	-	-	-	41.71

within the field, meaning that CDS8 is much more stable than it would be for a high or infinite Schmidt number flow. Measures of accuracy and cost for such a simulation are shown in the last row of Table 8.9, where it can be seen that the scheme exhibits a similar performance to the E-L method with particle removal (and is better than the CHARM and SuperBee TVD schemes of Cases ZM5 and ZM6, shown in Table 7.3), while completing the simulation in much less time. One might conclude from this result that CDS8 should always be used in lieu of the E-L method for cases of high diffusivity; however, the current study does not accommodate for flows where strain and shear act to increase the gradient of the transported scalar field. In these cases CDS8 would be very likely to cause oscillatory behaviour, while the E-L method – particularly with the particle removal mechanism implemented – should continue to be stable and more accurate.

The particle removal mechanism is finally applied to the E-L method for the long-running high Schmidt number configuration. The implementations previously delivering the best accuracy – in Cases ZM11(3) and ZM12(2), with the modified initial decomposition – are repeated here, and results are shown in Table 8.10. The particle removal parameters β and λ are set to their suggested values of 0.2 and 0.7, as before.

Table 8.10: Impact of particle removal, Cases ZM11(3) and ZM12(2).

Case	Ptc. Rem.	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)			Time (s)
							Add.	Rem.	Final	
ZM11(3)	×	-1.63	-6.53	114.84	66.60	106.20	0.136	-	9.229	132812
	✓	-4.37	-10.49	222.55	**42.85**	193.10	69.517	65.395	3.874	**105779**
ZM12(2)	×	-7.28	-11.43	175.72	250.62	142.64	0.006	-	0.910	17309
	✓	**-6.00**	-18.84	339.38	483.21	272.73	21.176	21.296	0.785	18820

Considering the results that have been presented previously, it is unsurprising that particle removal is generally detrimental to the performance of the E-L method for this configuration. The physical diffusive fluxes are not significant enough to prevent any noise, transferred from the particle phase to the Eulerian field, from contributing to the accumulation of oscillations in that field. Figure 8.22 presents a comparison of the transported Eulerian phases from Case ZM12(2) performed with and without particle removal, which shows that the noise is quite discernible.

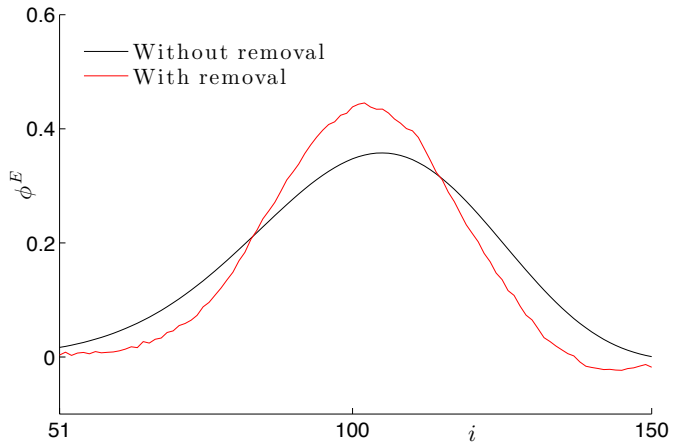


Figure 8.22: Case ZM12(2): Transported Eulerian phase with and without particle removal.

8.4.2 Low-Pass and Inverse Filtering

Having investigated the efficacy of the particle removal mechanism for the Zalesak and modified Zalesak cases, the low-pass filtering and deconvolution algorithms developed in §8.2 and §8.3 are now considered. Each algorithm will be applied separately and then in combination, where appropriate, to provide the best account of their impact on the solution for each case. This impact will be expressed as a relative (percentage) change from the initial results in each of the metrics described in §6.4, where an improvement is shown in bold green and a decline is in red. As the algorithms are only applied to the Lagrangian field once, in a 'post-processing' step at the end of the simulation, their computational cost is not considered.

Deconvolution is expected to play an important role in reducing the total error for Cases Z8 and Z9 from the original Zalesak disc study, as the scalar fields experience zero diffusivity. The deconvolution algorithm is therefore applied prior to the localised low-pass filtering routine. The results are shown in Table 8.11.

Table 8.11: Impact of filtering (LPF) and deconvolution (HPF) for the original Zalesak cases.

Case	LPF	HPF	$\pm\Delta B_L$ (%)	$\pm\Delta B_U$ (%)	$\pm E$ (%)	$\pm\Delta$TV (%)	$\pm E_{sym}$ (%)
Z8	×	✓	15.20	40.94	**-38.71**	13.33	0.57
	✓	×	**-11.68**	**-0.05**	**-0.76**	**-11.91**	**-2.01**
	✓	✓	**-0.74**	29.37	**-39.80**	**-1.31**	**-1.70**
Z9	×	✓	56.63	14.30	**-31.03**	13.51	4.25
	✓	×	**-5.88**	**-7.96**	**-7.16**	**-23.38**	**-11.72**
	✓	✓	56.63	14.30	**-39.89**	**-13.50**	**-9.24**

The deconvolution algorithm is clearly capable of delivering a significant reduction in total error (of at least 31%), as the intrinsic filtering in the Lagrangian reconstruction is corrected around the sharp edges of the disc. However, this comes at the cost of increased unboundedness (up to 56%), increased deviation from the expected change in total variation (around 13%), and a larger symmetry error (up to 4%), as noise in the reconstruction is amplified within the regions to which

deconvolution is locally applied. Low-pass filtering, on the other hand, results in an improvement in every metric, as that noise is locally removed. The effectiveness of filtering is found to be greater in Case Z9, where total error is reduced by 7% (as opposed to only 1% in Case Z8). The lower particle density in Case Z9 results in a noisier field, which is more responsive to the applied filter.

In combination the increased unboundedness caused by the deconvolution algorithm is partially resolved by filtering for Case Z8, but is unaffected in Z9. For total error, change in total variation and symmetry error, however, the overall change is roughly equal to the sum of the changes delivered by each routine individually. This indicates that regions identified for deconvolution will not also be selected for low-pass filtering, and vice versa.

The localised filtering and deconvolution algorithms are now applied to the best E-L solutions from §8.4.1: for the high Schmidt number ($Sc = 10$ and $Sc = 100$) cases results from Cases ZM3, ZM4, ZM11(3) and ZM12(2) without particle removal are used, while Cases ZM7 and ZM8 – for the low Schmidt number ($Sc = 1$) configuration – are with particle removal applied. For all six of these solutions the finite diffusivities have smoothed the transported Lagrangian field sufficiently so that the deconvolution algorithm does not identify any regions for application; therefore the results, presented in Table 8.12, show the impact of localised low-pass filtering only.

Table 8.12: Impact of filtering for the modified Zalesak cases.

Case	Particle Removal	$\pm\Delta B_L$ (%)	$\pm\Delta B_U$ (%)	$\pm E$ (%)	$\pm\Delta$TV (%)	$\pm E_{sym}$ (%)
ZM3	×	**-39.47**	**-37.79**	**-18.45**	**-77.50**	**-18.05**
ZM4	×	**-50.40**	**-23.04**	**-29.53**	**-59.15**	**-30.26**
ZM7	✓	0.00	**-73.78**	**-18.89**	**-78.36**	**-18.78**
ZM8	✓	0.00	**-49.89**	**-29.68**	**-84.61**	**-23.80**
ZM11(3)	×	**-17.27**	**-16.65**	**-6.35**	**-78.88**	**-3.53**
ZM12(2)	×	**-42.09**	**-2.72**	**-25.21**	**-70.51**	**-18.03**

Localised low-pass filtering is found to deliver an improvement in all of the applied metrics, with the exception of the lower boundedness measures for Cases ZM7 and ZM8; however, referring to Table 8.9, these are very nearly zero. Filtering has the greatest impact on the change in total variation, as this measure is most sensitive to noise. The cases with lower particle densities (ZM4, ZM8 and ZM12(2)) again generally show a larger proportional improvement with filtering, as the amplitude of the noise in these cases is typically higher.

8.5 Conclusions

Investigations into the performance of the joint Eulerian-Lagrangian method have revealed that the method suffers from an increased computational cost in comparison to traditional Eulerian schemes, while the Lagrangian reconstruction introduces noise (due to the randomness of the particle distribution) and some intrinsic low-pass filtering (due to the top-hat filtering of the scalar contribution of each particle) into the solution. In this chapter modifications designed to limit or correct these

problems have been developed, tested, and applied.

A particle removal mechanism has been suggested in which particles are removed in regions where the Eulerian phase is considered to be smoother than necessary. The scalar contribution from any removed particle is then added to the Eulerian phase. The 'aggressiveness' of the mechanism in its selection of particles for removal is controlled by the parameters β and λ, appearing in Eqs. 8.1 and 8.6, respectively. These parameters are related to the second and first derivatives of the Eulerian field, and take recommended values of $\beta = 0.2$ and $\lambda = 0.7$; higher values will tend to reduce the overall number of particles. However, the applicability of the particle removal routine is limited to cases with non-zero diffusivity, as noise introduced into the Eulerian phase (via the reconstruction of the scalar contribution of the removed particles) tends to accelerate the accumulation of error in the low-frequency field by making it more susceptible to oscillatory behaviour. This is only lessened where the action of physical diffusion smooths out that noise.

Inverse filtering (deconvolution) corrects the small degree of filtering inherent in the Lagrangian reconstruction routine, and a localising algorithm is applied to maximise its effectiveness in delivering an improved solution. The algorithm selects regions for deconvolution based on the second derivative of the Lagrangian field, and has been tested in two and three dimensions. Deconvolution is capable of reducing total error significantly for cases of zero diffusivity, where the reconstruction of very sharp edges in the scalar field is of the most significance; however, as the smoothness of the field is increased the routine rapidly becomes redundant, as the potential for the correction of errors in the reconstruction becomes outweighed by the amplification of noise. The deconvolution localisation is designed to account for this, though, meaning that only the sharpest changes in gradient are identified for correction.

Low-pass filtering is used to reduce the noise in the Lagrangian reconstruction, and is also localised according to the shape of the underlying Lagrangian field. By their formulation filtering and deconvolution will typically be applied in separate regions. Localised low-pass filtering is shown to deliver an improvement in all of the applied metrics for all of the cases considered here.

A final summary of the performance of the E-L method thus far is presented in Table 8.13, where the total error is compared to the time to solution for each of the 'best' Eulerian and E-L simulations (taking both total error and computational time into consideration for that selection).

Table 8.13: A comparison of total error and time to solution for a selection of the Eulerian (italicised) and E-L simulations presented so far, including particle removal (PR), low-pass filtering (LPF), and deconvolution (HPF).

Configuration	Case	PR	LPF	HPF	Total Error	Time (s)
Zalesak Original	*Z6*	-	-	-	*143.46*	*7.10*
	Z9	×	✓	✓	46.00	15.52
Modified, $Sc = 10$, 1 rev.	*ZM2*	-	-	-	*48.26*	*45.19*
	ZM4	×	✓	×	38.26	99.93
Modified, $Sc = 1$, 1 rev.	*ZM5*	-	-	-	*14.05*	*43.38*
	ZM8	✓	✓	×	11.01	88.02
Modified, $Sc = 100$, 100 revs.	*ZM9*	-	-	-	*335.14*	*4423*
	ZM12(2)	×	✓	×	131.42	17309

The data in Table 8.13 suggests that although the new method is consistently more expensive – by a factor of at least two – it is also capable of delivering more accurate results, where accuracy has been enhanced through a combination of particle removal, low-pass filtering, or deconvolution.

The studies presented in Chapters 6 and 7 provided a useful insight into the performance of the proposed joint Eulerian-Lagrangian method, although they only tested its ability to deal with constant or lessening changes in gradient, and were only concerned with the translation of a pre-defined scalar 'shape'. It will therefore be important to test the method further, by applying it to a case where the scalar field is deformed by shear within the flow. In particular the re-initialisation algorithm will be important for the accurate description of scalar transport in such a case, and this constitutes the focus of Chapter 9.

9 Test Case 3: The Single Vortex

The proposed joint Eulerian-Lagrangian (E-L) method has been rigorously tested using two simple, solid body rotation cases, in two dimensions. The results of these investigations have subsequently suggested some improvements to the method, which have also been developed in two dimensions, and for which additional testing has been carried out in three dimensions for a diffusion-only problem. The next step in the process of analysing and validating the E-L method is to consider a case where the transported scalar field is subjected to shear, causing a deformation of the original field 'shape'. This case will particularly test the re-initialisation algorithm built into the method, which will play a pivotal role if an accurate description of the transported scalar is to be maintained.

One significant disadvantage of the E-L method revealed in Chapters 6 and 7 has been the relatively high computational expense in comparison to traditional Eulerian schemes. Although the particle removal mechanism developed in Chapter 8 is capable of reducing this expense for flows of non-zero diffusivity, the Lagrangian element of an E-L simulation will always make the method more expensive than an Eulerian case at the same resolution. With this in mind, it would be useful to compare the new method to Eulerian simulations at different resolutions; this should reveal whether a more refined Eulerian simulation is capable of matching the improvement in accuracy delivered by the new method, while continuing to have a lower computational cost.

The single vortex test case originally proposed by Rider and Kothe [185] is used, applying the (forced) vortical velocity field of Bell et al. [212] for the deformation of a simple two-dimensional scalar field. As for the original and modified Zalesak configurations of Chapters 6 and 7, no solution of the flow-field itself is required. The problem therefore only tests the numerics of the scalar transport scheme, without any influence from the flow solver.

9.1 Case Objectives

The previous results of Rider and Kothe [185] (as well as Enright et al. [180, 186], and Ubbink and Issa [213], amongst others, who have applied the same case) show that the transported scalar field will develop into a swirling pattern of an increasingly thin 'filament'. This will provide an excellent test of the E-L method overall, but will particularly highlight the ability of the re-initialisation algorithm to identify regions where more particles are required to maintain an accurate description of the scalar field. In addition the idea of 'time inversion' is used – as applied by Rider and Kothe – where the flow-field is reversed half-way through a simulation. This results in the scalar field being transported back to its original position, which permits a direct comparison of the initial and final transported fields, and allows the transport error of the Eulerian and E-L methods to be quantified. However, as will be shown in the first set of single vortex results, care should be taken to consider

any errors that are apparent at the half-way point when the flow-field is reversed.

While the E-L method will be tested at a fixed grid resolution, the corresponding Eulerian simulations will be performed over a variety of computational cell sizes. This will allow an assessment of the improvement in accuracy that can be gained from grid refinement; ultimately, it should then be possible to determine whether the advantages of the E-L method are truly 'cost-effective' compared to traditional schemes. (However, this assessment neglects the cost of solving the flow-field itself: in an actual Large-Eddy Simulation or Direct Numerical Simulation, the additional cost of the E-L method – as a proportion of the overall expense – will be moderated by the costs of transporting momentum and density, and of calculating pressure corrections for each cell. The expense of an Eulerian simulation at a higher grid resolution, relative to a simulation which does not solve the flow, will then become far greater).

9.2 Case Description

The single vortex case consists of an off-centred solid body transported by a vortical velocity field. Because the magnitude of the flow velocity decreases with distance from the centre of the vortex, the body experiences a shear, which results in significant topological changes: after an appreciable length of time has passed the body has been deformed into a thin filament spiralling around the vortex origin. The single vortex case results in structures within the scalar field that are reminiscent of the features found in a realistic simulation of turbulent flow: as in a shear layer, for example, where the resulting Kelvin-Helmholtz instabilities will tend to generate a procession of the vortices that will be simulated here.

9.2.1 Velocity Field

The velocity field imposed in this case, as used by Bell et al. [212], is defined by the stream function:

$$\Psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y) \tag{9.1}$$

A contour plot of this stream function is provided in Fig. 9.1. The streamlines of the single vortex are circular towards the centre, but tend to become more square-shaped as the radius increases: in an orthogonal array of equally-spaced vortices the zero isolines of velocity will form a square grid around each vortex, resulting in the illustrated pattern.

The velocity components u and v (in the Cartesian directions i and j, respectively) are subsequently described by Eqs. 9.2 and 9.3:

$$u = -\frac{\partial \Psi}{\partial y} = -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \tag{9.2}$$

$$v = \frac{\partial \Psi}{\partial x} = 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x) \tag{9.3}$$

Where x and y are the physical co-ordinates in the (Cartesian) i- and j-directions. Note that an alternate definition of u and v as a function of Ψ is commonly used, which would simply change the direction of rotation from clockwise to anti-clockwise.

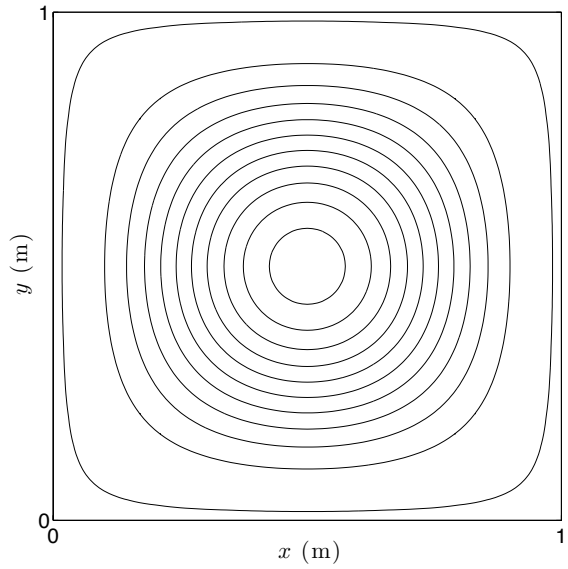


Figure 9.1: Single vortex: Contour plot of the stream function described in Eq. 9.1.

9.2.2 Solid Body

The initial scalar field consists of a disc of radius $r = 0.15$ inside a 2×2 domain, as shown in Fig. 9.2, discretised on to a grid of 200×200 cells. The disc is centred at the position $[1.0, 1.25]$, such that the radius of rotation $R = 0.25$. Note that the original case of Rider and Kothe used a unit-sized domain; this is increased here to prevent any possible influence of the boundary conditions on the scalar field (as in the modified Zalesak case of Chapter 8). The stream function and velocity component variables x and y, appearing in Eqs. 9.1, 9.2 and 9.3, are shifted by 0.5 units in the negative x- and y-directions to reflect the increase in the domain size. This ensures that the solid body is initialised such that it lies across the shear structure of the central vortex. This also means that the generated vortex is one of a repeating pattern with neighbouring vortices along the edges of the domain, although these will not have any influence on the results.

As for the Zalesak cases the disc represents a scalar value $\phi = 1$, while all other points are initialised to $\phi = 0$. Rider and Kothe stipulate that the scalar value within a cell that contains the interface of the disc should be set to the proportion of that cell inside the disc radius. For simplicity, a Monte Carlo approach is used to calculate this proportion: any cell that is identified as containing part of the disc interface is populated with randomly-distributed points, and each of those points is inspected to determine whether it lies within the disc radius. The number of points that lie within the disc, divided by the total number for that cell, provides an approximation for the scalar value that should be set within that cell. The number of random points N_{MC} for each cell is chosen to be 10000, which is expected to provide an accuracy of around 1% within each cell on the circumference (the error in a Monte Carlo approximation being proportional to $1/\sqrt{N_{\mathrm{MC}}}$). As this routine is only used during the initialisation of the scalar field any computational cost associated with it will be negligible compared to the cost of the simulation itself.

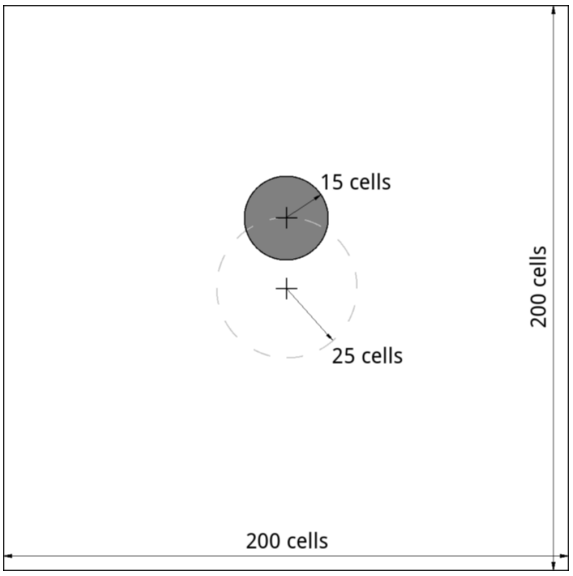


Figure 9.2: Single vortex: Schematic representation. The shaded area represents $\phi = 1$.

9.3 Simulation Numerics

Computations are performed using the same simplified, two-dimensional version of the `PsiPhi` code applied for the Zalesak case. The cell size is initially set to $\Delta = 1.00$ cm, resulting in a domain of 2×2 m. It can be seen from Eqs. 9.2 and 9.3 that the maximum velocity component is then 2 m/s. The time-step width is therefore chosen to be $dt = 0.5\Delta$, resulting in a value CFL = 1.0 for the convective stability criterion. Simulations are performed for 10 seconds, requiring 2000 time-steps at $\Delta = 1.00$ cm, with the velocity field reversed after $t = 5$ s.

The diffusion coefficient for this case is set to zero, such that the Schmidt number Sc is infinite. This maximises the gradients within the scalar field so that the re-initialisation algorithm of the E-L method is tested to the greatest possible extent. This also stipulates that there should be no transport of scalar across the streamlines of the vortex, and, as such, although the initial solid body will be very strongly deformed by the flow-field, it should remain as a single contiguous feature.

An approximation of the analytical solution is generated by performing a very high-resolution simulation with the scalar field represented by Lagrangian particles only. Figure 9.3 shows this solution at time $t = 5$ s, in the region $x = [0.5 - 1.5]$ m, $y = [0.5 - 1.5]$ m.

The single vortex case is primarily intended to provide a rigorous test of the E-L method, but three Eulerian spatial schemes will be applied for comparison. Firstly a high-order Central Differencing Scheme (CDS8) will be used, to demonstrate the 'reversibility' of oscillatory errors. The other Eulerian methods are of the Total Variation Diminishing (TVD) category, where the applied flux limiters are the Cubic-parabolic High Accuracy Resolution Method (CHARM) and SuperBee schemes. These are chosen as CHARM is found to be a good 'all-round' limiter, while SuperBee provides the best possible second-order TVD solution for a zero-diffusion simulation (see §4.5). As the better of the two schemes for this case, SuperBee simulations will also be performed at a range of grid resolutions $\Delta = [0.2 - 1.0]$ cm. The cases presented here are summarised in Table 9.1.

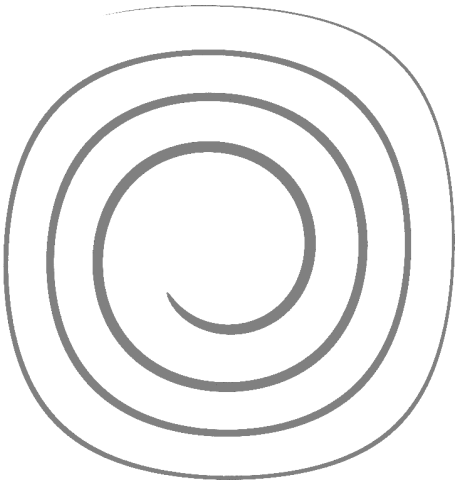


Figure 9.3: Single vortex: Approximate analytical solution at $t = 5$ s.

Table 9.1: Numerical schemes to be tested with the single vortex case.

Case	Method	Eulerian Scheme	Cell Size Δ (cm)	Particle Density L^{ϱ}
SV1	Eulerian	CDS8	1.00	-
SV2	Eulerian	CHARM TVD	1.00	-
SV3(a)	Eulerian	SuperBee TVD	1.00	-
SV3(b)	Eulerian	SuperBee TVD	0.80	-
SV3(c)	Eulerian	SuperBee TVD	0.50	-
SV3(d)	Eulerian	SuperBee TVD	0.40	-
SV3(e)	Eulerian	SuperBee TVD	0.25	-
SV3(f)	Eulerian	SuperBee TVD	0.20	-
SV4	E-L	CDS8	1.00	500
SV5	E-L	CDS8	1.00	100
SV6	E-L	CDS8	1.00	20

The parameters for the E-L method are partly selected based on suitable values identified through the Zalesak studies. The Gaussian filter kernel width is set to the recommended value of $\sigma_f = 5\Delta$, corresponding to 20 filter support points in each direction. Re-initialisation is performed every four computational steps, while the threshold $\phi''_{\max}$ for the second derivative (above which, the local field will be re-filtered to transfer scalar from the Eulerian to the Lagrangian phase) is set with $\gamma = 1$ in Eq. 5.60. This ensures that it is consistent with the expected maximum second derivative of the initial Eulerian field at the chosen value of σ_f. As this case will be simulated with zero diffusivity the particle removal mechanism (introduced in §8.1) is turned off. Localised low-pass filtering and deconvolution are applied to the Lagrangian reconstruction of each E-L case, with the parameters suggested in Chapter 8.

A variety of Lagrangian particle densities L^{ϱ} are tested; Case SV4 is expected to exhibit a lower amplitude of noise in the Lagrangian reconstruction, but at an increased computational cost. This range in L^{ϱ} is intended to provide further insight into the potential of the localised low-pass filtering

algorithm introduced in Chapter 8, where it has been observed that (computationally cheaper) low particle density simulations benefit more from filtering. It should be considered that although the Eulerian-Lagrangian results for the single vortex configuration could be improved by increasing the particle density, it may not be feasible to maintain very high particle numbers (due to computational cost, memory requirement, or both) when the method is applied to a three-dimensional case. Using an increased particle density here could therefore provide a misleading impression of the potential accuracy that might be achieved in three dimensions. It should also be considered that the perimeter of the solid body will become very much elongated over the first half of a simulation – as the vortex stretches the scalar field into a thin filament – such that the total number of particles should be expected to increase rapidly.

9.4 Results and Discussion

Results are presented in greyscale images at a point in time half-way through the simulation ($t = 5$ s), where a scalar value of $\phi = 1$ appears white and $\phi = 0$ appears black. In addition a 'slice' through the scalar field along the vertical mid-plane ($x = 1$ m) is provided, showing the profiles of the initial, intermediate, and final transported fields, as well as of the analytical approximation at $t = 5$ s (shown in Fig. 9.3). Because the final field (after time inversion) should be identical to the initial field, the metrics applied in §6.4, and defined by Eqs. 6.3 to 6.10, may be used. As the 'initialisation error' introduced by the Eulerian-Lagrangian decomposition has already been investigated in Chapter 6, only the initial (pre-decomposition) and final fields are compared here.

To provide further insight into the behaviour of each scheme the probability density function (PDF) of the scalar field before and after transport is determined. This is performed within a region limited to the locality of the initial disc position, to provide a clearer indication of any differences between the schemes at different times. The PDF is constructed by discretising the scalar values in the region of interest into 40 'bins', between $\phi = 0$ and $\phi = 1$, and counting the number of cells within the region containing the value assigned to each bin. Each cell therefore constitutes one sample to the PDF, which is subsequently normalised by the total number of samples. The initial field and its PDF are shown in Fig. 9.4: clearly the distribution is bimodal at initialisation, with peaks at scalar values of $\phi = 0$ and $\phi = 1$. Any non-zero probabilities in the range $0 < \phi < 1$ are due to the cells around the edge of the disc whose scalar contribution is equal to the proportion of the cell inside the disc radius (as determined from the Monte Carlo approach described in §9.2.2). Inspection of this PDF shows that within the assessed region (indicated by the red box in Fig. 9.4) 55% of cells are initialised with zero scalar value, 39% have a value $\phi = 1$, and the remaining 6% are initialised with an intermediate value in the range $0 < \phi < 1$.

Simulations are performed with the same 2.4GHz AMD Opteron^{TM} 280 processor used for the Zalesak studies in Chapters 6 and 7, and the computational expense is estimated from the execution time to completion for each simulation. The flow-field is dramatically different to that of the Zalesak studies; however, as the simulation still consists only of a single scalar in a forced flow, it is assumed that the uncertainties in timing determined in Chapter 6 – around 2% for E-L simulations, and 10% for Eulerian runs – are valid here.

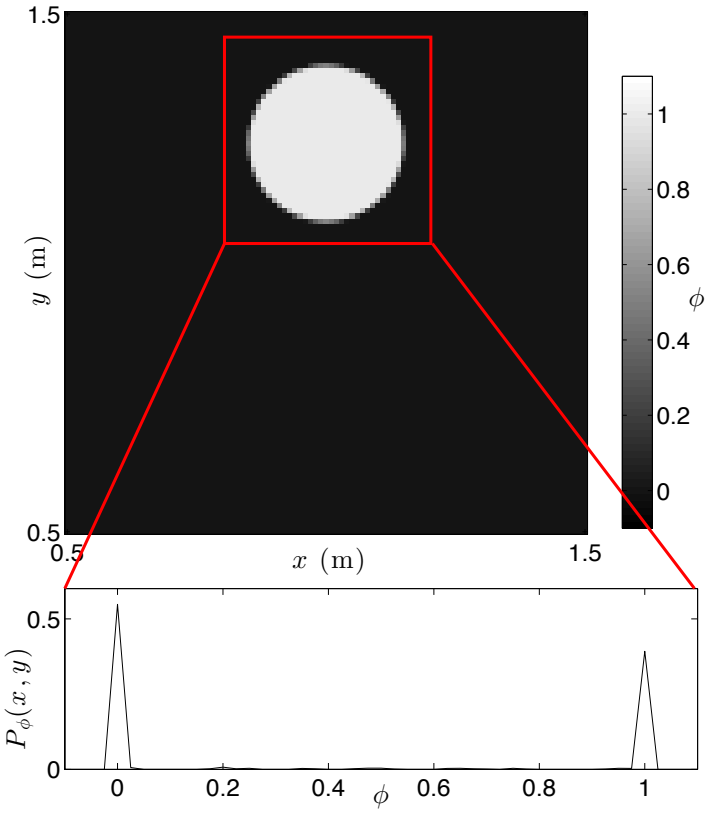


Figure 9.4: Single vortex: PDF of the initial scalar field; assessed region (red box).

9.4.1 Eulerian Simulations

Results from the Eulerian simulations, listed as Cases SV1 to SV3 in Table 9.1, are presented here.

Case SV1 In the first instance the single vortex computation is performed using CDS8 for the convection of the scalar field. As stated previously there is little expectation of a favourable result in this case, as the use of a central approximation for the face-centred scalar values will result in strong oscillations within the transported field. Figure 9.5(a) shows the single vortex at $t = 5$ s (just before the velocity field is reversed), where significant constructive and destructive interference is observed between the 'waves' generated by the convective scheme.

Although this intermediate result is clearly inaccurate (in comparison to the approximate analytical solution presented in Fig. 9.3), the transported field at the end of the simulation, shown in Fig. 9.5(b), seems quite reasonable. In fact, based on the final transported field alone, we will see that CDS8 exhibits the best performance of the Eulerian schemes. The over-predicted fluxes that cause the oscillatory behaviour in the 'positive' direction, before the time inversion, are exactly cancelled out by equal but opposite fluxes in the 'negative' direction during the second half of the simulation. This can clearly provide a misleading assessment of a scheme for such a case as the initial and final fields may look very similar, even though the intermediate field is wrong. The caveat, then, is that the use of time inversion to recover a final field equivalent to the initial one should be treated with some care, while it is clearly of no relevance to real applications.

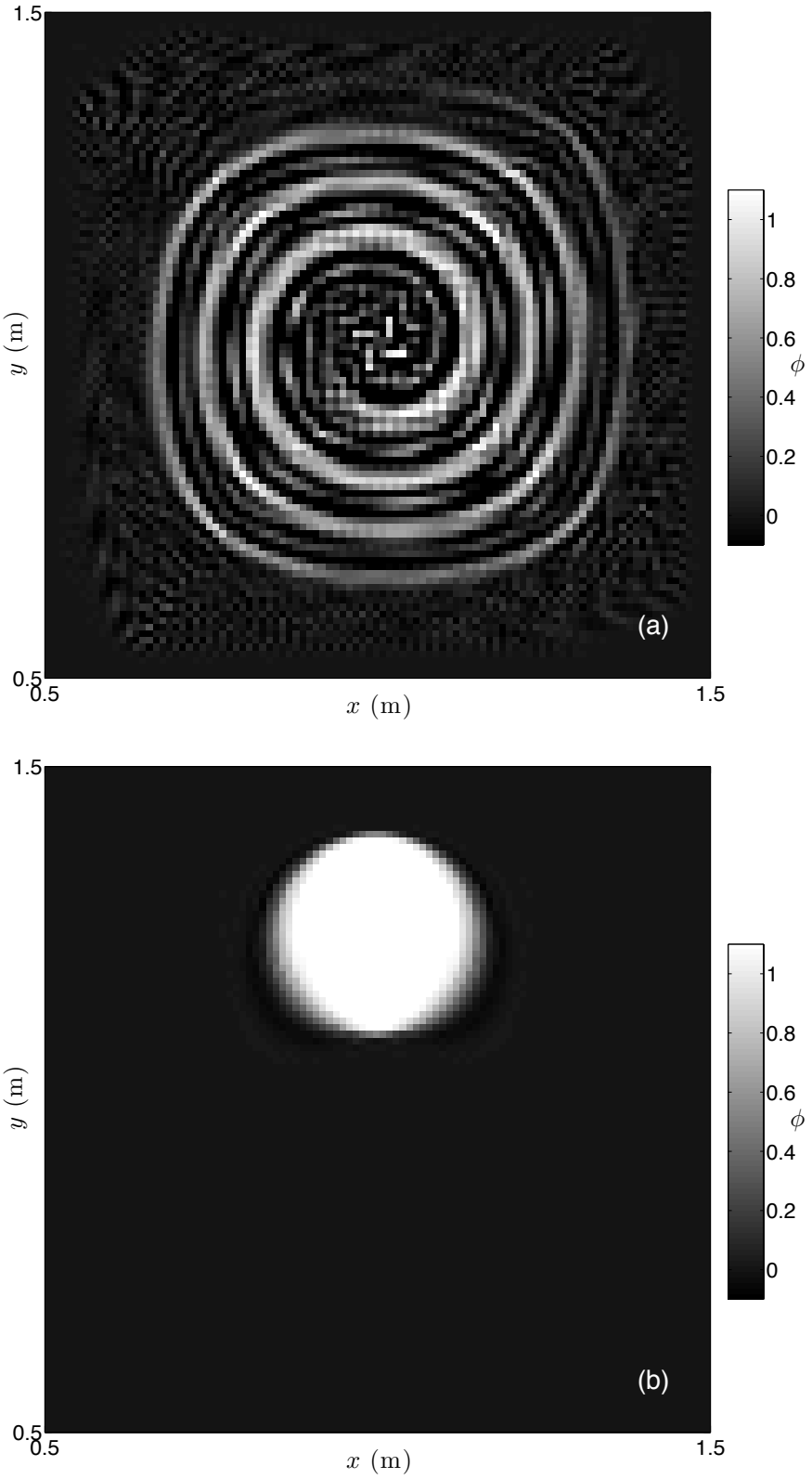


Figure 9.5: Case SV1: CDS8, $\Delta = 1.00$ cm; scalar field at (a) $t = 5$ s; (b) $t = 10$ s.

Case SV2 The next case applies the CHARM TVD scheme. The scalar field at $t = 5$ s is shown in Fig. 9.6(a), while Fig. 9.6(b) shows a slice through the scalar field at times $t = 0$ s, $t = 5$ s and $t = 10$ s and Fig. 9.6(c) shows PDFs of the initial and final ($t = 10$ s) scalar field.

Immediately apparent from Fig. 9.6(a) is the extent by which the scalar has dissipated across the vortex streamlines. While the inner and outer 'tails' of the swirling pattern are discernible, the field has become quite homogenised around the radius of rotation of the initial solid body; in other words, the area of non-zero scalar has increased dramatically. The drop in peak scalar value is made clear by the relatively dull appearance of the field (where the same greyscale is used for each case, to facilitate the comparison of each transported field). This can also be seen in Fig. 9.6(b), where the intermediate solution shows only a vague suggestion of the expected 'ridges' of scalar. Notably from Fig. 9.6(b), the solution at $t = 10$ s shows a significant proportion of the scalar in the range $0 < y < 1$ m (in the wrong half of the domain): so much so, in fact, that the final scalar profile is almost symmetrical. This is indicative of the significant difference between the actual and expected final fields.

Also discernible is the misalignment between (what remains of) the peak closest to the vortex centre – which shall be referred to as the 'inner' peak – in the intermediate field, and the corresponding ridge of the analytical solution. This is due to the spreading of the scalar away from the radius of rotation (which intersects the slice in Fig. 9.6(b) at $y = 0.75$ m and $y = 1.25$ m). The gaps between the peaks on each side of the vortex are filled by numerical diffusion, causing the gradient on the 'outer' side of the inner peak to be reduced at a quicker rate than the gradient of the 'inner' side. As the rate of numerical diffusion (like physical diffusion) is proportional to scalar gradient, this causes a net diffusive flux towards the centre of the vortex at this location. The same effect can be seen in the peaks furthest from the vortex centre – the 'outer' peaks – with a net diffusive flux towards the outside of the domain, but to a lesser extent due to the smaller scalar values. The difference in magnitude of the net radial fluxes can be seen from the right half of Fig. 9.6(b), where the final field has spread further towards the centre than it has towards the outer edge.

The PDF shown in Fig. 9.6(c) quantifies the extent by which the peak value has dropped, as the probability P_ϕ drops to zero above $\phi = 0.25$. The bimodality of the PDF has not been entirely lost, with local peaks at $\phi = 0$ and $\phi = 0.18$, but the non-zero values for P_ϕ between these peaks (as well as, of course, the far-removed position of the right-hand peak) are highly indicative of the acute numerical dissipation that the scalar field has experienced during transport. Further examination of the PDF shows that after transport only 12% of cells in the region of interest have a zero scalar value, while a further 40% lie in the range $0.01 < \phi < 0.16$ and the remaining 48% constitute the peak around $\phi = 0.2$.

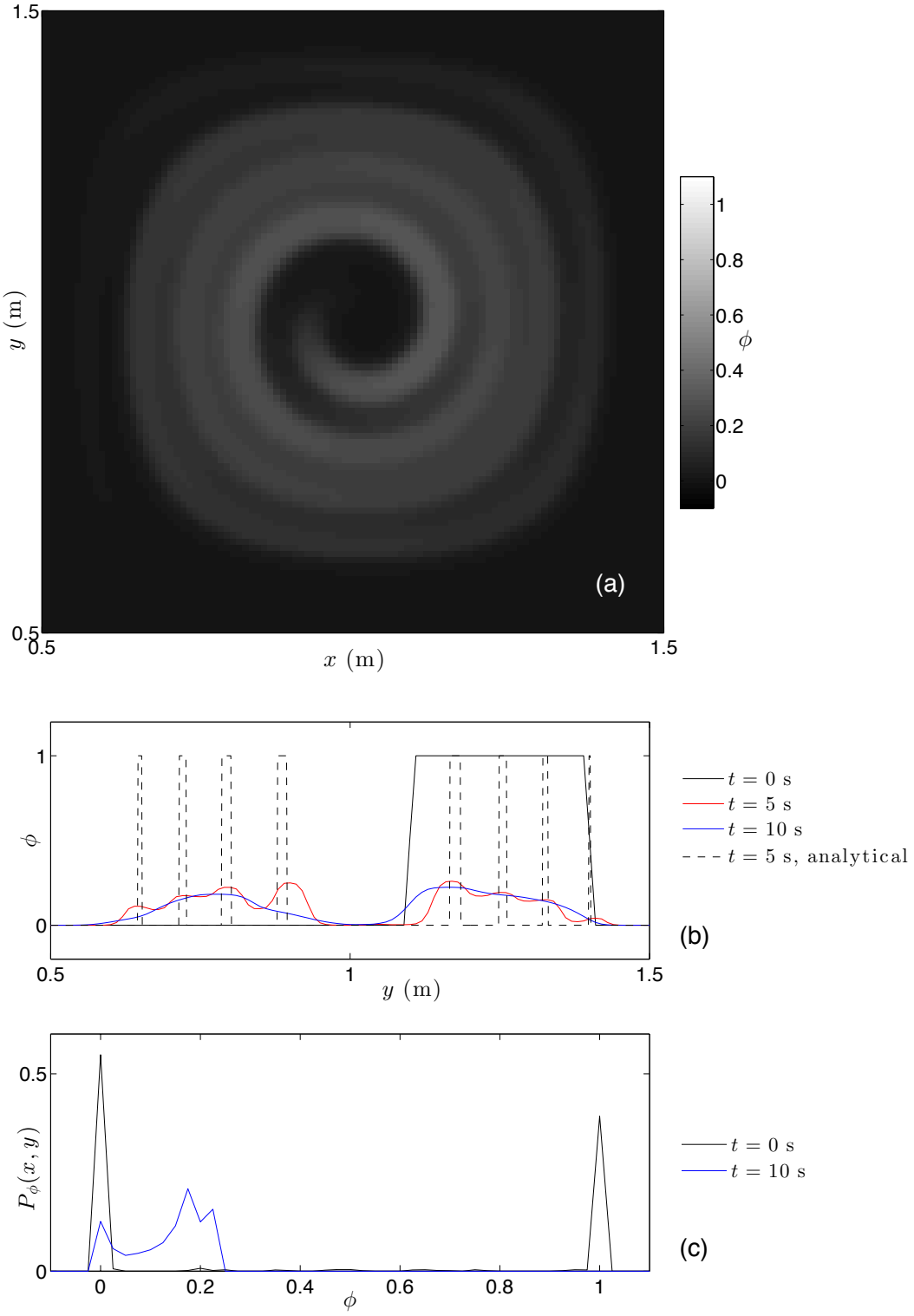


Figure 9.6: Case SV2: CHARM TVD, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

Case SV3 In the third case the TVD scheme is retained; however, the CHARM flux limiter is replaced with the SuperBee limiter, which is expected to be more effective for this configuration (based on the results from the original Zalesak disc study, presented in Chapter 6). Simulations with this scheme are also performed at a range of grid resolutions from $\Delta = 1.0$ cm down to $\Delta = 0.2$ cm. A representative selection of these results are displayed graphically here, although metrics are applied to solutions at all six resolutions (summarised in §9.4.3).

Case SV3(a), $\Delta = 1.0$ cm Results from the coarsest simulation are shown in Fig. 9.7. Compared to the results from Case SV2 the scalar field here bears much more of a resemblance to that of the approximate analytical solution shown in Fig. 9.3, as the swirling pattern generated by the vortex is clearly visible. However, the mid-grey shade of the field indicates that the peak scalar has dropped. The width of the spiral filament is approximately constant along its entire length, rather than becoming thinner towards the outer end, and this coincides with a decreasing scalar value. The outer 'tail' of the spiral thus becomes hard to distinguish.

The profile in Fig. 9.7(b) also shows the reduction in scalar value towards the outside of the spiral, with an appearance as if a constant diffusion coefficient had been applied to the analytical field. The outer peaks are narrow (particularly on the right-hand side of Fig. 9.7(b), where the spiral is at its thinnest), and will therefore diffuse faster than the thicker parts of the spiral towards the centre of the vortex. It can also be seen that the net diffusion away from the radius of the initial disc centre that was observed in Case SV2 is not as prevalent here, as (at $t = 5$ s, at least) the gaps between the peaks have not been filled in. This means that each peak remains symmetrical, since the diffusive flux is equal on each side of the peak. A small amount of scalar is observed in the domain range $0 < y < 1$ m at $t = 10$ s.

The corresponding PDF for this coarse run shows that the transported field has zero P_ϕ above $\phi = 0.43$. The improved description of the transported scalar is indicated by the more pronounced peak (16% of samples) at $\phi = 0$, which demonstrates that the scalar has dissipated into fewer cells. This is supported by the fact that, although there is still a non-zero region between the left- and right-hand bounds of the PDF, the probabilities for $0.01 < \phi < 0.21$ are reduced in comparison to Case SV2; 22% of cell scalar values lie within this range. The remaining 62% of cells form a fairly uniform distribution over $0.21 < \phi < 0.43$.

Case SV3(d), $\Delta = 0.4$ cm Figure 9.8 shows results for the SuperBee simulation at a grid resolution of $\Delta = 0.4$ cm, where the number of cells in the domain is increased to 500×500 and the time-step width is reduced by a factor of 2.5 (to maintain a convective stability criterion of CFL = 1.0).

As apparent from Fig. 9.8(a) the increase in resolution has delivered a dramatic improvement in the description of the transported field at $t = 5$ s, where the edges of the filament appear to be much sharper than at the lower resolution of SV3(a). The lighter shades of grey towards the inner (central) end of the filament also indicate that the scalar value is much higher than previously. However, from Figs. 9.8(b) and (c), it can be seen that the peak scalar value does drop by around 6% over the first half of the simulation, and by a further 2% during the second half. Furthermore

the scalar PDF is not bimodal at $t = 10$ s; only 27% of the cells within the assessed region contain zero scalar, while for the remaining 73% of cells a fairly uniform distribution exists across the range $0.01 < \phi < 0.93$.

Considering the transported field profile shown in Fig. 9.8(b), there is a discernible asymmetry to each of the 'peaks' across the scalar field at $t = 5$ s: the innermost edge of each peak tends to have a sharp top and a smoother base, while on the outer side the base is sharp and the top is smooth. This is consistent with the findings of Case Z6 from the original Zalesak study, where a similar behaviour has been observed (see Fig. 6.9). The final transported field is relatively well captured towards the mid-point of the domain, where the scalar gradient near the centre is accurately recovered; however, the scalar value decreases with increasing radius.

Case SV3(f), $\Delta = 0.2$ cm For the final Eulerian case the SuperBee TVD scheme is applied at a grid resolution of $\Delta = 0.2$ cm, corresponding to a domain size of 1000×1000 cells and requiring a time-step width one-fifth of that of Case SV3(a). The results are presented in Fig. 9.9.

At this resolution the transported scalar field at $t = 5$ s closely resembles the approximate analytical solution shown in Fig. 9.3, where a unit scalar value is maintained for the two innermost revolutions along the filament before the value of ϕ begins to drop towards zero. The most notable difference between this solution and Fig. 9.3 is the relatively constant width of the filament towards the outside of the swirl; as stated previously the thinner parts of the filament will be more susceptible to (numerical) diffusion, meaning that the scalar will spread more rapidly at the outer end.

The profiles of the intermediate and final transported fields presented in Fig. 9.9(b) further reveal the propensity of the SuperBee scheme for unusual behaviour: the 'peaks' across the field at $t = 5$ s display the same asymmetry observed in previous cases, while the final field now has a distinct 'stepped' appearance towards the outer radius. It is suggested that these steps result from the compressive nature of SuperBee, where shallow gradients tend to become steepened. During the first half of the simulation the perimeter of the scalar field is lengthened by the shearing motion of the vortical flow, meaning that the region affected by numerical diffusion increases over time. However, once the flow-field is reversed the perimeter begins to shrink, and this region – that was smoothed before time inversion – now becomes compressed. The step effect can be seen to affect only those parts of the scalar field where the peak scalar value had dropped below $\phi = 1$ at $t = 5$ s.

The scalar PDF shown in Fig. 9.9(c) reveals that the distribution of scalar values is well-preserved compared to the solutions at other resolutions, where the bimodality of the PDF at $t = 10$ s appears to have been maintained. However, spreading of the scalar in the final field is still significant, where 44% of cells in the assessed region now have a value in the range $0.01 < \phi < 0.99$. The proportion of zero-valued cells has dropped from 55% to 36%, while cells with a unit scalar value have dropped from 39% to 20%.

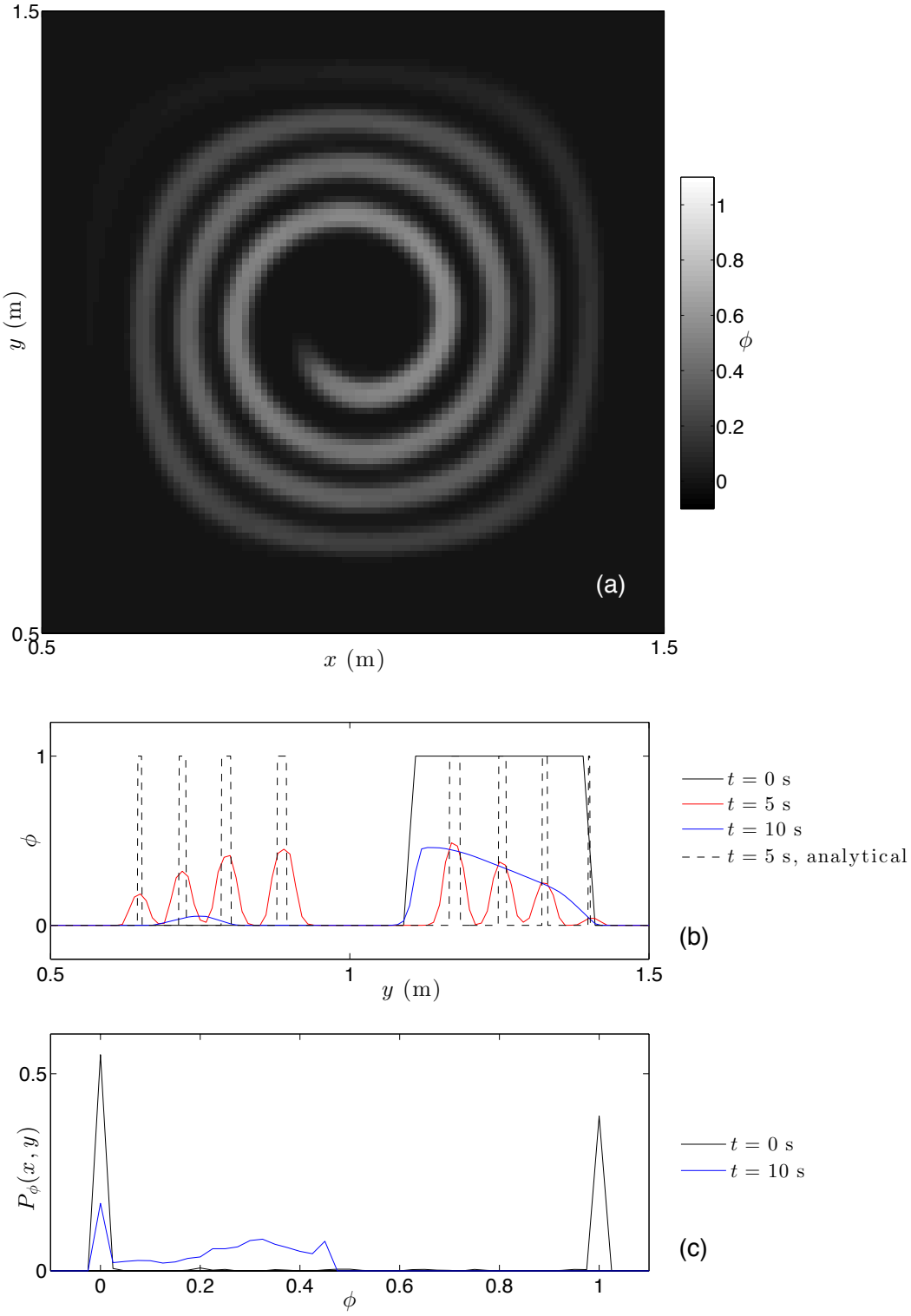


Figure 9.7: Case SV3(a): SuperBee TVD, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

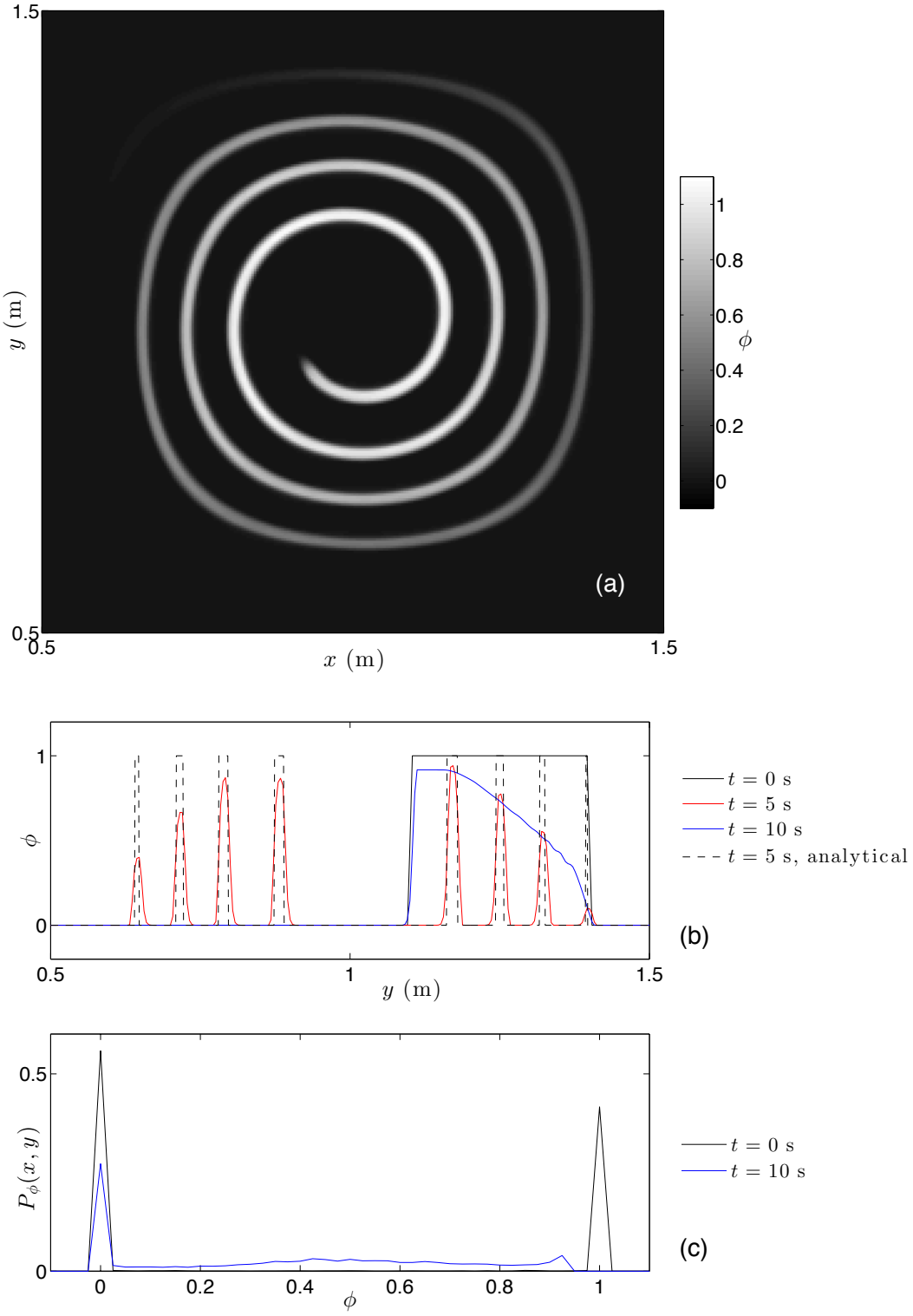


Figure 9.8: Case SV3(d): SuperBee TVD, $\Delta = 0.40$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

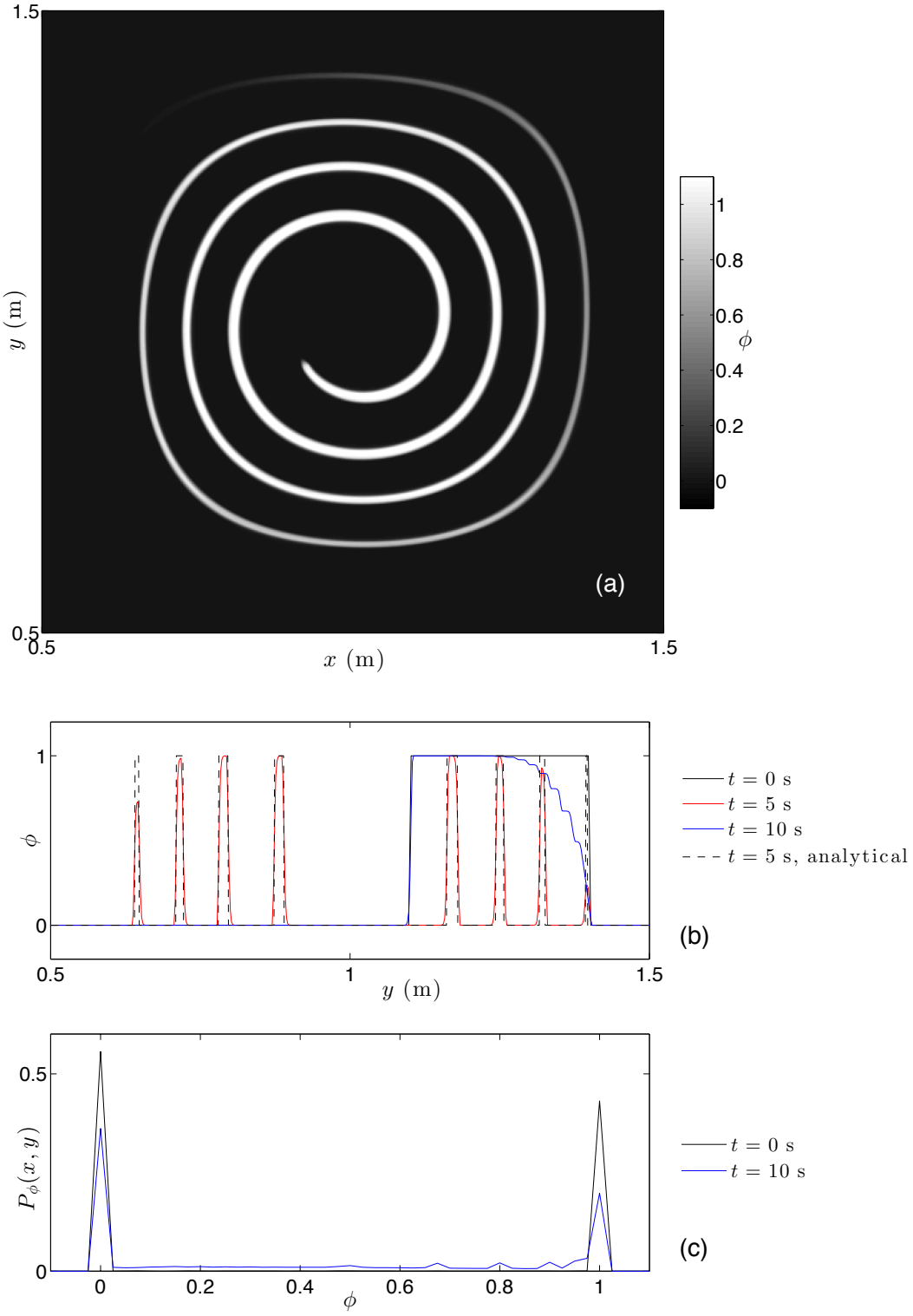


Figure 9.9: Case SV3(f): SuperBee TVD, $\Delta = 0.20$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

9.4.2 Eulerian-Lagrangian Simulations

Results from Cases SV4 and SV6, where the E-L method is applied, are shown in Figs. 9.14 and 9.15. The solution from Case SV5 is intermediate to those of SV4 and SV6, and graphical analysis is therefore omitted here.

Case SV4 Figure 9.14 presents results for the E-L method at a relatively high Lagrangian particle density $L^{\varrho} = 500$, on a computational grid of cell size $\Delta = 1.0$ cm.

First impressions of the intermediate transported field shown in Fig. 9.14(a) are encouraging; the shape of the scalar filament is very well captured, and a unit scalar value appears to be maintained along at least the two innermost rotations of the spiral. The solution presented here looks very similar to the SuperBee TVD simulation at the highest resolution (Fig. 9.9(a)), but with a more pixellated appearance due to the lower grid resolution. Unlike the Eulerian simulations the thin end of the filament continues to narrow until its width occupies a single cell only.

At the very end of the 'tail' it is suggested that the particle field will continue to capture the shape of the scalar on a sub-grid scale, although the reconstruction is incapable of representing that information. This error in the representation – resulting from the finite resolution of the Eulerian grid – is illustrated in Fig. 9.10, where a high scalar value in a single cell is seen to drop to a medium value spread across two cells as the line of the 'tail' crosses the boundary between two columns of cells.

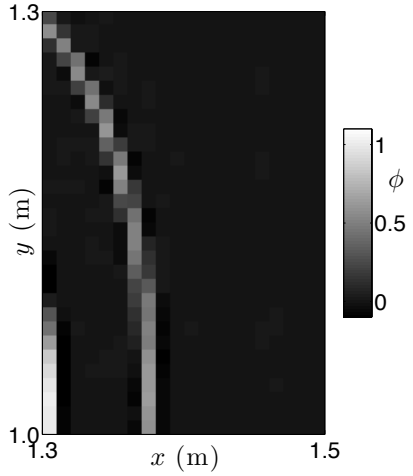


Figure 9.10: Case SV4: Detail of the reconstruction near to the 'tail' of the scalar field.

To explore this sub-grid particle representation further, a small part of the region shown in Fig. 9.10 is inspected in closer detail. Fig. 9.11(a) shows exact particle positions within a 4×6-cell section from the upper left corner of Fig. 9.10, with 'positive' particles shown in red and 'negative' particles in blue, while Fig. 9.11(b) shows a reconstruction of those particles at a resolution of $\Delta = 0.10$ cm (i.e. 10 times finer than the current resolution for Case SV4). The Eulerian contribution in included in the reconstruction. The particle field in this region is dominated by a densely-populated band of positive particles, which are apparent as the white band in the reconstruction.

Considering that the reconstructed scalar values are filtered over the Eulerian cell size (where the red dashed lines in Fig. 9.11(b) denote the Eulerian cell boundaries), it can be seen that the value of ϕ in cell (135, 124) will be around 0.5 (based on the area of the cell appearing white). However, in cells (134, 125) and (135, 125) the value will have dropped to around 0.25.

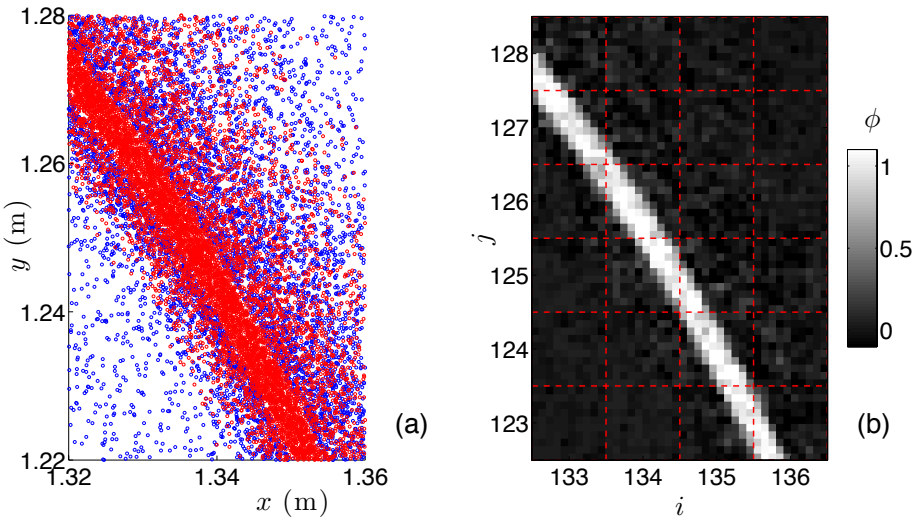


Figure 9.11: Case SV4: (a) Particle positions; (b) high-resolution Lagrangian reconstruction.

Inspection of the scalar field profiles shown in Fig. 9.14(b) indicates that the E-L method accurately predicts the position and amplitude of the scalar peaks at $t = 5$ s. However, the outer peaks show a drop in amplitude from the analytical profile, and it is suggested that this occurs where the width of the filament is less than a single cell. The Eulerian representation of the Lagrangian field can only recover features at a spatial resolution of Δ, while any features smaller than this will be top-hat filtered across the area of the cell (c.f. the 'resolution error' shown in Fig. 9.10). The final transported solution closely matches the analytical field, aside from the usual noisiness in the summation deriving from the Lagrangian reconstruction.

The excellent fidelity of the E-L method in this case is confirmed by considering the scalar PDF presented in Fig. 9.14(c). Unlike the probability distribution for Case SV3(f), where the largest proportion of cells contained intermediate scalar values ($0.01 < \phi < 0.99$), the PDF here has greater bimodality: making some allowance for any noise in the reconstruction we find that 54% of cells in the assessed region are close to zero; 36% are close to a unit scalar value; and the remaining 10% lie in the range $0.04 < \phi < 0.96$. This compares very favourably with the PDF of the initial field, which has proportions of 55%, 39% and 6% in these ranges, respectively.

One other notable feature of the solution at $t = 5$ s is the negative 'spike' appearing either side of each scalar peak in Fig. 9.14(b). The reason for these spikes is not immediately clear; however, the source of the error is established by re-computing Case SV4 without localised deconvolution applied, where Fig. 9.12 shows the corresponding scalar profiles. In the latter case the spikes are noticeably absent from the reconstruction of the final field.

To understand the cause of the spikes the 'stack of boxes' shown in Fig. 8.15 is considered again, where each set of 20 boxes represents the scalar contribution of one of 80 particles randomly

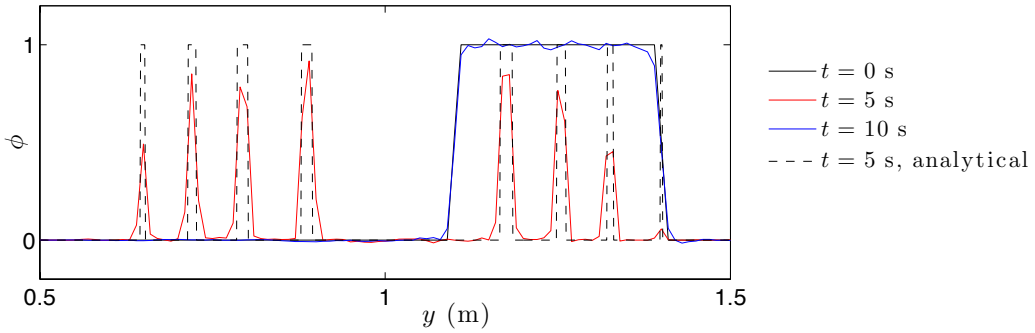


Figure 9.12: Case SV4: Slice through the scalar field at $x = 1$ m, without deconvolution.

positioned in the region $x = [1-2]$. From this we determine that the Lagrangian reconstruction will result in an intrinsic filtering, where the original field $\phi = [0, 1, 0]$ will be recovered as $\phi = \frac{1}{8}[1, 6, 1]$. However, if every particle is transported 0.5 units to the right, the 'stack' would then lie across the range $x = [1-3]$. Boxes would be shared approximately equally between cells i and $(i+1)$, while the reconstruction of the transported field would be $\phi = [0, 0.5, 0.5]$. Furthermore, if this translation is also performed on the *cell-centred* particles in Fig. 8.14, the transported particles would again be shared equally between cells i and $(i+1)$ resulting in a reconstruction of $\phi = [0, 0.5, 0.5]$. Figure 9.13 illustrates both of these cases, for uniform random (red) and cell-centred (blue) particle distributions.

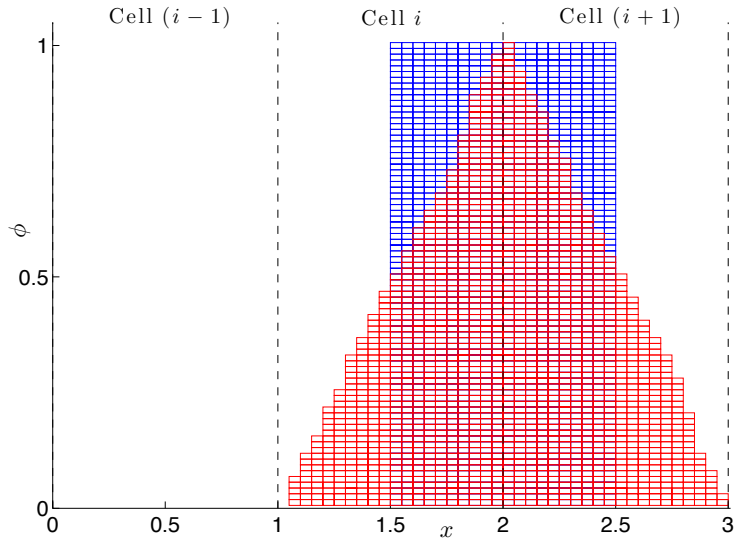


Figure 9.13: Lagrangian reconstructions after particle translation by 0.5 units.

In general, therefore, a convolution error due to the particle distribution (which is zero for initially cell-centred particles) combines with a resolution error due to the finite resolution of the computational grid (which is zero where the particle distribution is aligned with that grid). The convolution error may be corrected with localised deconvolution, while the resolution error may only be reduced by grid refinement. However, it can be seen that the effect of the convolution error tends to zero

as the particle distribution becomes misaligned with the underlying grid, so that deconvolution in this case will attempt to correct an error that is no longer there. In effect, deconvolution will then amplify the resolution error, resulting in the negative spikes observed in Fig. 9.14(b). The initial and (after time inversion) final particle fields should exactly line up with the computational grid, so that these spikes do not appear in the initial and final reconstructions.

Although this is an undesirable side-effect of the localised deconvolution algorithm, it should be stressed that deconvolution may still play an important role in improving the accuracy of an E-L solution. For example, it can be seen that at $t = 5$ s the uncorrected scalar field shown in Fig. 9.12 has scalar values typically around 20% lower than for the deconvoluted field in Fig. 9.14(b), due to the effect of the intrinsic filtering on the very narrow scalar features.

Case SV6 Finally the E-L method is applied with a reduced particle density of $\mathrm{L}^{\varrho} = 20$, and results are presented in Fig. 9.15.

The first impression of the scalar field at $t = 5$ s shown in Fig. 9.15(a) is favourable, where the overall representation of the transported scalar is much improved compared to that of the Eulerian schemes at the same grid resolution. However, as expected, the field clearly contains a higher degree of noise compared to the higher particle density solution of Case SV4.

The intermediate and final transported scalar field profiles shown in Fig. 9.15(b) illustrate the unboundedness of the solution for Case SV6 due to this noise, while the negative 'spikes' identified in Fig. 9.14(b) are still discernible. At this particle density it is suggested that localised deconvolution may have a detrimental effect overall, as any noise within deconvoluted regions will be more susceptible to amplification. Furthermore the final transported solution contains significant noisiness across the diameter of the solid body, suggesting that the localised low-pass filtering is possibly not effective enough with the current parameters. A reduction of the higher amplitude of noise for this low particle density case would require more filtering to be applied, which could be achieved by increasing the top-hat filter width (currently set to $w_{\mathrm{TH}} = 1.05\Delta$) and the maximum effective filter width (currently $b = \Delta$) for the filtering algorithm.

The increased level of noise is reflected in the scalar PDF shown in Fig. 9.15(c): scalar values are more widely spread around $\phi = 0$ and $\phi = 1$, although the PDF of the transported field continues to exhibit a bimodality that an Eulerian method cannot maintain. Dividing the PDF at $t = 10$ s into three regions, and allowing for the increased magnitude of noise from the Lagrangian component, we find that 55% of scalar values lie below $\phi = 0.19$; 37% of values are above $\phi = 0.81$; and the remaining 7% of values are in the range $0.19 < \phi < 0.81$. It should also be noted, however, that the total range of the PDF is indicative of the unboundedness resulting from the particle noise, where the minimum and maximum scalar values are now -0.30 and 1.52 respectively.

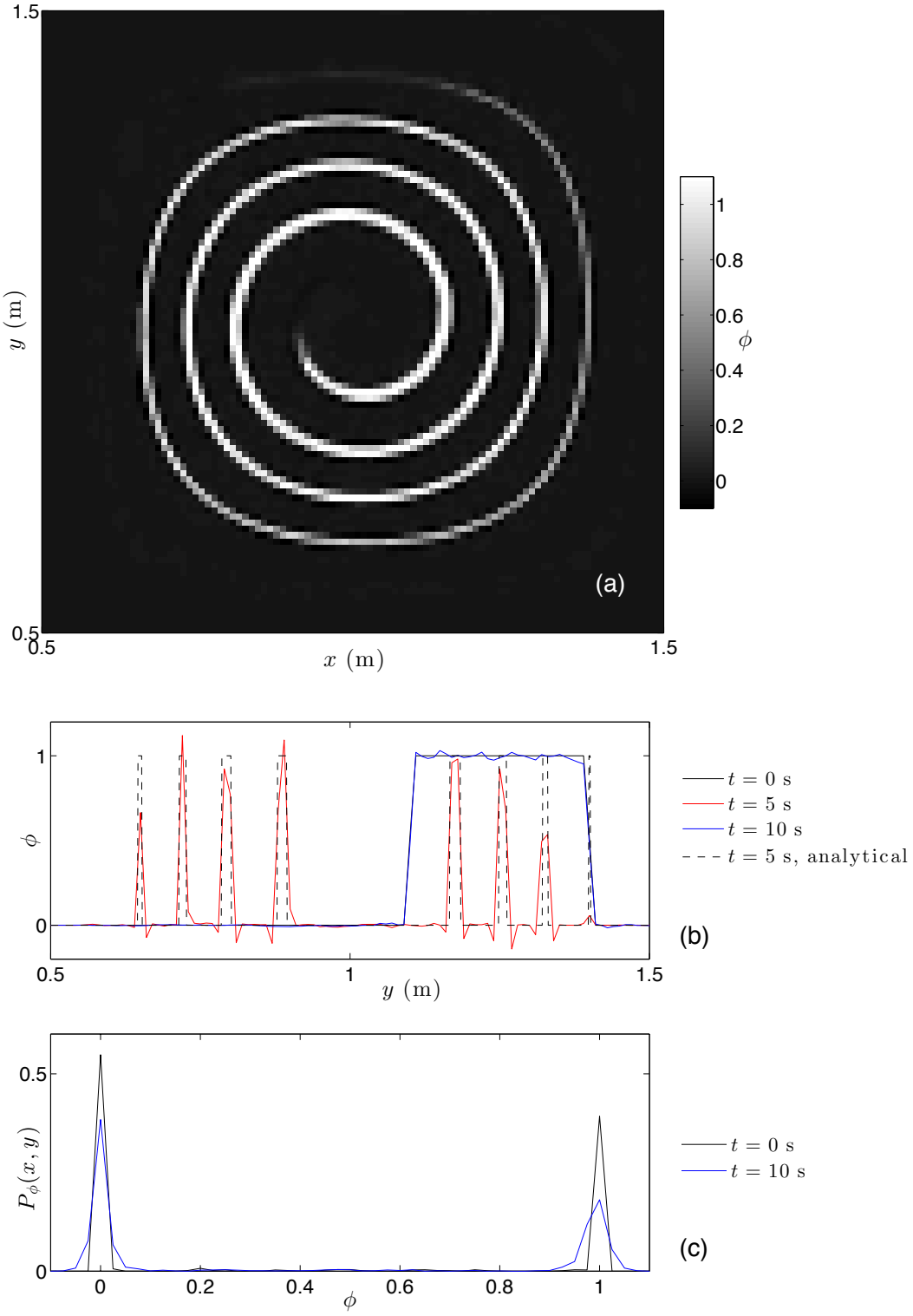


Figure 9.14: Case SV4: E-L method, $L^{\varrho} = 500$, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

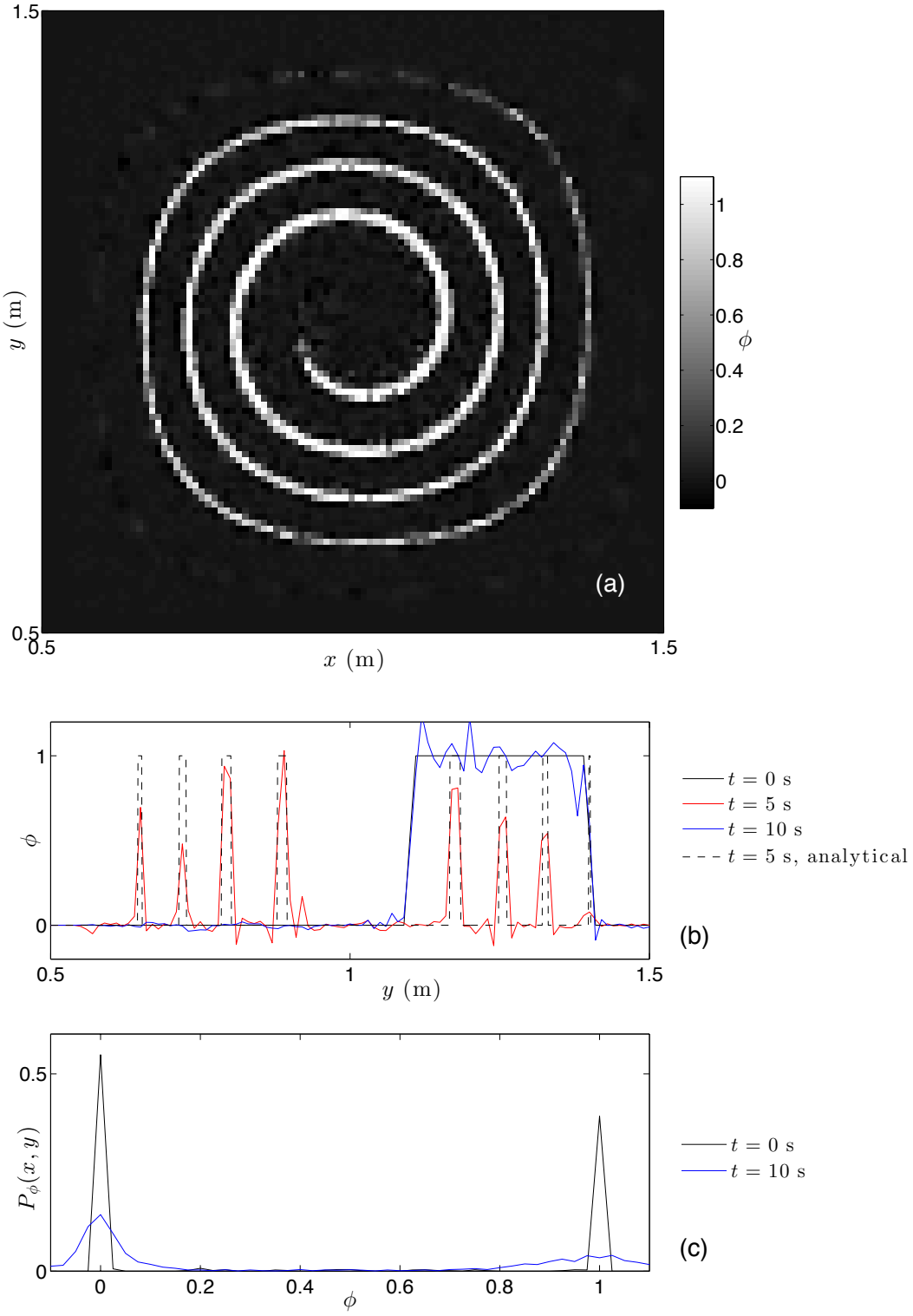


Figure 9.15: Case SV6: E-L method, $L^{\varrho} = 20$, $\Delta = 1.00$ cm; (a) scalar field at $t = 5$ s; (b) slice through the scalar field at $x = 1$ m; (c) scalar PDF.

9.4.3 Tabulated Results

Table 9.2 summarises the metrics gathered for boundedness (ΔB_L and ΔB_U, expressed as percentages), total error (E), change in total variation (ΔTV), asymmetry (E_{sym}) and computational expense (i.e. execution time, in seconds), for Cases SV1 to SV6. Also included, for the E-L simulations, are the numbers of Lagrangian particles at the end of each simulation. For Case SV3 – where a SuperBee TVD scheme is used at a variety of grid resolutions – the measures of total error, change in total variation and asymmetry must be scaled by the area of each cell, so that they are comparable between different resolutions. The definitions of these measures, originally provided in Eqs. 6.7, 6.8 and 6.10 respectively, are therefore modified accordingly; for example, total error becomes:

$$E(\phi, \Delta) = \frac{1}{\Delta^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left| \phi^0(i,\, j) - \phi^t(i,\, j) \right| \tag{9.4}$$

Boundedness measures require no such modification as they are not determined from a summation operation.

Table 9.2: Measurements of accuracy and cost for Cases SV1 to SV6, after $t = 10$ s.

Case	Cell Size Δ (cm)	ΔB_L (%)	ΔB_U (%)	E	ΔTV	E_{sym}	Particles ($\times 10^5$)	Time (s)
SV1	1.00	-8.02	-6.80	89.75	28.34	2.67	-	50.12
SV2	1.00	0.00	77.59	1162.68	-47.84	88.55	-	56.28
SV3(a)	1.00	0.00	53.98	950.89	-24.25	116.33	-	53.58
SV3(b)	0.80	0.00	45.14	857.20	-18.29	88.20	-	105.52
SV3(c)	0.50	0.00	20.70	612.80	-7.35	51.12	-	449.04
SV3(d)	0.40	0.00	8.28	478.34	-2.71	40.51	-	933.78
SV3(e)	0.25	0.00	0.08	245.47	-0.42	21.97	-	3599.23
SV3(f)	0.20	0.00	0.02	174.66	-0.11	15.72	-	7198.49
SV4	1.00	-6.39	-7.25	47.97	59.53	31.89	6.905	500.38
SV5	1.00	-13.95	-17.81	80.21	143.81	54.96	1.376	160.76
SV6	1.00	-29.70	-51.79	218.93	503.44	154.57	0.251	84.48

Results for Case SV1 are included to further illustrate the importance of considering intermediate solutions in a test simulation applying Rider and Kothe's idea of time inversion. The unboundedness observed here is indicative of the oscillatory behaviour of a Central Differencing Scheme (at any order of accuracy); however, the total error and symmetry error for this case are the lowest of all Eulerian schemes (at any resolution), while CDS8 also returns the shortest computational time. Nevertheless, the highly erroneous intermediate solution presented in Fig. 9.5(a) reiterates the unsuitability of CDS8 as a convective scheme for a scalar field containing rapid changes in gradient.

The CHARM TVD flux limiter applied in Case SV2 delivers arguably the worst results presented here, where the total error is greatest and the peak scalar has dropped to less than a quarter of the expected value. The final field is also relatively asymmetric, which is consistent with previous results (such as those from Case ZM9 for the modified Zalesak disc, presented in Chapter 7). The

computational cost is relatively low, but the fidelity of the solution is poor.

From the information for Case SV3(a) it can be seen that the performance of the SuperBee TVD scheme at a cell size of $\Delta = 1.00$ cm is not vastly better than that of the CHARM scheme. Although the measures do show some improvement in comparison the maximum scalar value has still dropped to less than one-half of the expected value, while the total error is relatively large. Symmetry error is also higher for the SuperBee scheme, as observed for previous configurations.

The performance of the SuperBee scheme does show marked improvement as the grid resolution is increased (i.e. Δ is reduced), where total error, change in total variation and symmetry error are all found to be approximately proportional to cell size. Upper boundedness also seems to be fairly linear until it becomes very close to zero for Case SV3(e) at $\Delta = 0.25$ cm. However, plotting ΔB_U and E against Δ – as shown in Figs. 9.16(a) and 9.16(b), respectively – reveals that both exhibit a weak non-linearity.

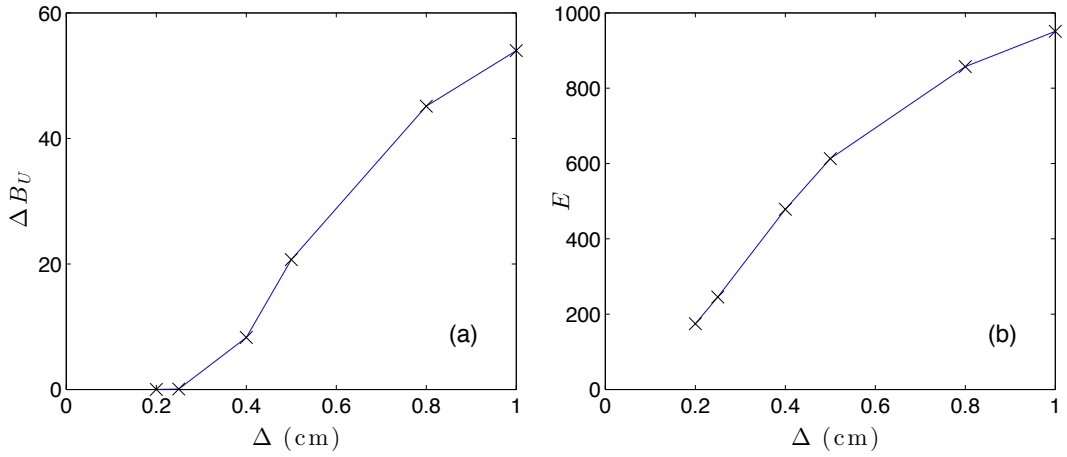


Figure 9.16: Case SV3: Dependence of (a) upper boundedness, (b) total error on Δ.

Before attempting to explain these trends it is useful to consider the behaviour of the boundedness and total error measures if the solid body were instead subjected to a constant physical diffusivity. Figure 9.17 shows how these measures vary in time, where the initial conditions of Fig. 9.2 have been transported by diffusion for a nominal duration.

For this constant diffusion case the observed non-linearity in total error may be explained by considering that the maximum possible error (for a bounded scheme) would be equal to twice the sum of the scalar, and that the total error will tend towards this value as the solution gets worse. The amount by which the shape of the field changes will also decrease in time, as the scalar gradients (and, therefore, the magnitude of the diffusive fluxes) become reduced. Considering upper boundedness the effects of diffusion will take some time to spread from the sharp edges of the solid body towards the centre, but the peak scalar value will then drop steadily as the shape of the body tends towards a Gaussian profile.

The numerical diffusivity of an Eulerian scheme becomes greater as the grid resolution is reduced, and we assume that it is a Gaussian process (like physical diffusion). The profile of the scalar field transported by an Eulerian method on different grids would therefore be the same as the constantly-

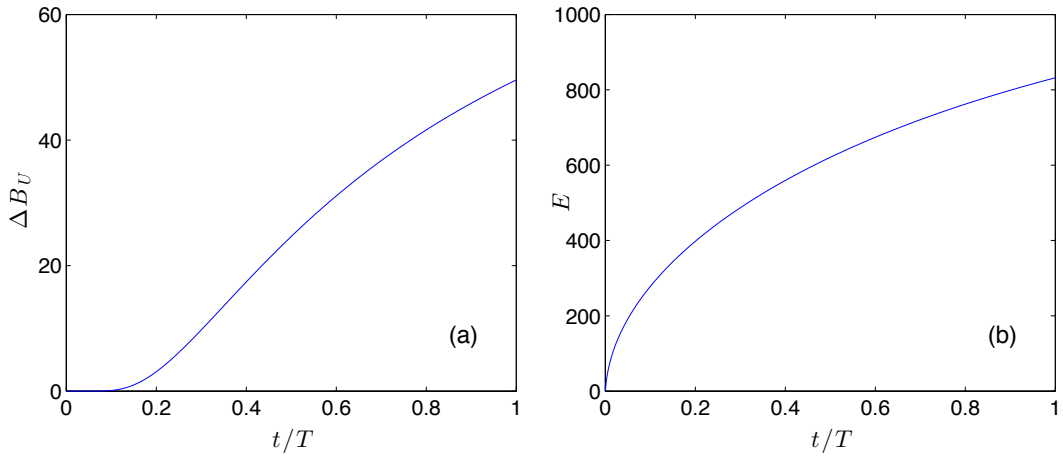


Figure 9.17: Variation of (a) upper boundedness, (b) total error with time, applying a constant diffusivity to the initial SV conditions.

diffused scalar at different times. The fact that the trends shown in Figs. 9.16 and 9.17 are very similar thus tends to support the concept that numerical diffusivity is proportional to cell size. Any differences between them, such as the apparent offset of the trend in Fig. 9.16(b) compared to the curve in Fig. 9.17(b), may be accounted for by the fact that numerical diffusion is not actually a Gaussian process for a SuperBee scheme (or for any other Eulerian method; the transported profiles are typically asymmetric, while numerical diffusion will only act in directions normal to any gradient but parallel to the flow).

Considering computational cost, Fig. 9.18 shows how execution time T varies with the resolution of the grid for Case SV3. From this it can be seen that cost decreases with increasing cell size, where the gradient of the (logarithmic) plot is approximately -3.

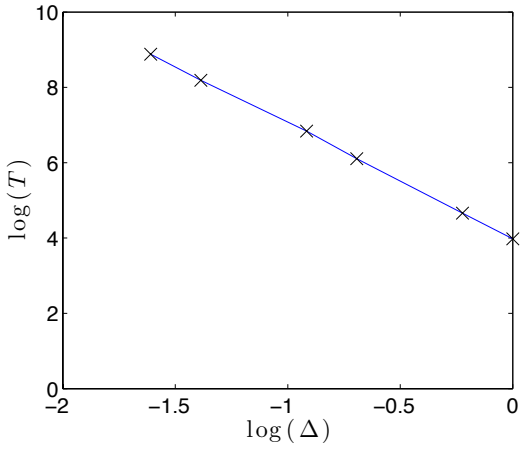


Figure 9.18: Case SV3: Dependence of computational cost on Δ.

This relationship is straightforward to explain: increasing the resolution will correspondingly extend the simulation in three dimensions (the temporal and two spatial). Reducing the cell size by, for example, a factor of two, will quadruple the total number of cells, and double the required

number of time-steps (as the time-step width must be halved to maintain a constant value for the CFL criterion), resulting in an eight-fold overall increase in cost. (In three spatial dimensions, then, it is reasonable to assume that the computational cost would scale with Δ^{-4}).

Results for the three E-L simulations at $t = 10$ s are strongly influenced by the noisiness of the Lagrangian reconstruction, as observed in previous tests. The boundedness and change in total variation are again particularly sensitive to this noise; all three solutions are unbounded, while ΔTV is positive and becomes very large for the lowest particle density simulation (Case SV6, $L^{\varrho} = 20$). Results for the total error are highly encouraging, however: simulations at high (SV4, $L^{\varrho} = 500$) and medium (SV5, $L^{\varrho} = 100$) particle densities are found to have a lower total error than even the finest SuperBee solution, while SV6 is comparable to SuperBee on a $\Delta = 0.25$ cm grid. Symmetry error is likewise relatively favourable in comparison to the CHARM and SuperBee solutions, where SV4 and SV5 significantly out-perform the Eulerian results from the same grid.

The E-L method is found to be considerably more expensive than the Eulerian computations at the same resolution, as expected, although a more standardised measure of performance will be introduced towards the end of the current analysis. Applying a linear fit to the variation of computational cost with total particle number (as shown in Fig. 9.19) suggests that the additional cost of particles is around 0.31 CPU-seconds per time-step per million particles.

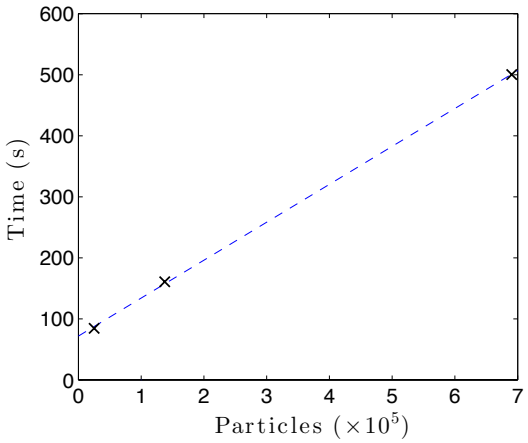


Figure 9.19: E-L simulations: Dependence of computational cost on particle number.

This value is slightly lower than the 0.33 CPU-seconds determined for the original Zalesak case in Chapter 6, and much lower than the 1.1 CPU-seconds from the modified Zalesak case of Chapter 7 (although the latter configuration included diffusion terms in the particle equation of motion). In the original Zalesak configuration the total number of particles was fairly constant, however, whereas for the single vortex cases significant numbers of particles are added as the perimeter of the solid body becomes stretched by the shearing motion of the flow. Figure 9.20 shows the total particle number (blue) and the rate of particle addition (black) for the three E-L simulations, where it can be seen that each case experiences a rapid initial increase in the number of particles. This increase decays over the first half of each simulation, and then drops abruptly once the flow-field is reversed at $t = 5$ s and the perimeter begins to shrink. Re-calculating the particle cost based on the mean

number of particles for each case returns a slightly higher value of 0.37 CPU-seconds per time-step per million particles.

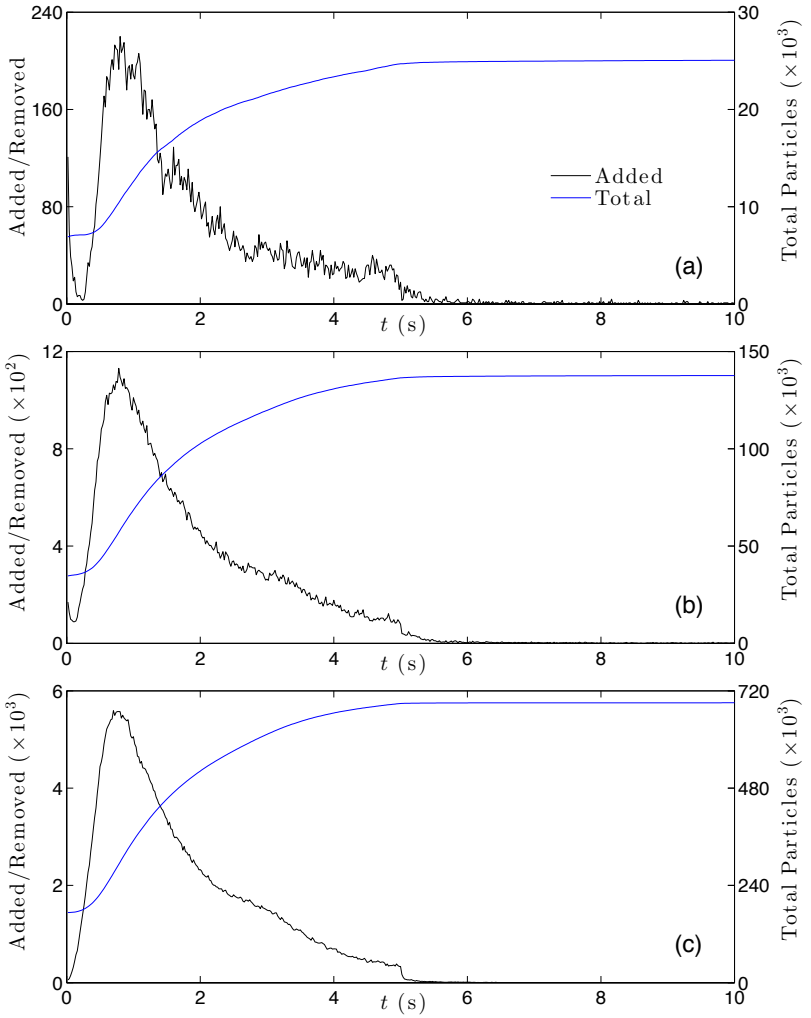


Figure 9.20: E-L simulations: Total and added particle numbers for (a) SV4; (b) SV5; (c) SV6.

9.4.4 Localised Deconvolution and Filtering

The effectiveness of the localised deconvolution algorithm for E-L solutions is investigated. It has been suggested that any improvement gained from the deconvolution of the Lagrangian reconstruction is partially counteracted by the amplification of noise within deconvoluted regions. This is particularly relevant to simulations with low Lagrangian particle densities where the amplitude of the noise will typically be higher (and therefore more affected by amplification), to the extent that deconvolution may degrade the solution in terms of the total error. (Note that this is not related to the amplification of the 'resolution error' discussed in §9.4.2). To assess deconvolution in this context Cases SV4 and SV6 are recomputed both with and without the algorithm.

The effect of modifying the parameters controlling the low-pass filtering algorithm is also examined: the higher amplitude of noise in Case SV6 makes it harder to distinguish between desirable

and unwanted high-frequency information in the Lagrangian reconstruction, and it is suggested that with the present settings the filtering algorithm is having little impact on the solution. As identified previously this is most obvious from Fig. 9.15(b), where the final transported field may be seen to contain large fluctuations across the whole diameter of the solid body. In an attempt to improve this Cases SV4 are SV6 are also recomputed with modified values of $a = 0.025$, $b = 3\Delta$, and $w_{\mathrm{TH}} = 2\Delta$. Increasing a should ensure that more of the noise in the Lagrangian reconstruction is identified for filtering, while increasing b and w_{TH} will cause that noise to be more heavily filtered. The restriction described in §8.2.3, where the number of filter operations can only vary by a maximum of one operation per cell, is designed to prevent a noisy value in one cell from becoming an edge condition for the adjacent filtered cells. However, this restriction also means that the number of filter operations near to a sharp change in gradient will always be low. The suggested changes to b and w_{TH} will essentially allow the algorithm to apply a more effective filter close to a sharp change in gradient. A disadvantage of the modification to a is that some of the 'tail' in the solution at $t = 5$ s may also be identified for filtering; in general, care should be taken in the selection of these parameters to prevent the Lagrangian reconstruction from being too heavily filtered.

Table 9.3 shows results for a variety of deconvolution and filtering configurations for Case SV4. The first row represents the 'base case' without either of the post-processing algorithms applied, and is included for comparison. From this analysis it is found that deconvolution does improve the solution, where total error is reduced by 13%. Low-pass filtering on its own with the current parameters is also effective, while the modified parameters only provide a slight additional improvement for total error (although change in total variation and symmetry error are more substantially reduced). The existing combination of deconvolution and filtering is found to be most useful overall, however, where a 37% reduction is total error is delivered.

Table 9.3: Case SV4: With and without deconvolution (HPF); with modified filtering (LPF).

HPF	LPF			ΔB_L	ΔB_U	E	ΔTV	E_{sym}
	a	b (Δ)	w_{TH} (Δ)	(%)	(%)			
×	-	-	-	-6.60	-7.62	75.55	123.05	44.00
✓	-	-	-	-6.60	-7.62	65.74	126.07	44.93
✓	0.010	1.0	1.05	-6.39	-7.25	47.97	59.53	31.89
×	0.010	1.0	1.05	-6.47	-7.25	57.96	56.01	31.06
×	0.025	3.0	2.00	-4.76	-6.70	55.45	26.70	24.14

Table 9.4 shows the same variety of configurations for Case SV6, where it can be seen that deconvolution is found to have a negative effect on the solution. Each of the five metrics becomes worse to some degree; boundedness and change in total variation in particular are relatively adversely affected, which is consistent with their greater sensitivity to noise. Correspondingly, although the existing combination of deconvolution and filtering does provide some improvement compared to the 'base case', the existing filtering on its own is rather more effective. The most significant improvement, however, is delivered by the filtering algorithm with modified parameters, where noise is removed to a far greater extent than previously.

From these results it is apparent that deconvolution may not be appropriate for simulations

Table 9.4: Case SV6: With and without deconvolution (HPF); with modified filtering (LPF).

HPF	LPF			ΔB_L	ΔB_U	E	ΔTV	E_{sym}
	a	b (Δ)	w_{TH} (Δ)	(%)	(%)			
×	-	-	-	-24.54	-31.39	242.07	552.77	165.41
✓	-	-	-	-29.70	-51.79	271.63	669.56	192.13
✓	0.010	1.0	1.05	-29.70	-51.79	218.93	503.44	154.57
×	0.010	1.0	1.05	-24.54	-31.39	189.11	386.09	128.02
×	0.025	3.0	2.00	-24.54	-31.39	120.21	193.55	78.27

employing low Lagrangian particle densities. This will be of significance for the studies presented in Chapters 10 and 11, where the maximum permissible number of particles in a three-dimensional flow at a high resolution will be limited by the computational cost of those particles (in terms of processing time, and the amount of memory required). The information in Tables 9.3 and 9.4 also suggests that the low-pass filtering parameters may be modified to improve the effectiveness of filtering for these cases, where noise in the particle reconstruction will have a greater impact. Such modifications should be treated with care, however, to avoid making the filtering algorithm too aggressive; too much filtering will result in a reduced accuracy as desirable high-frequency information is inadvertently removed.

9.4.5 Standardised Performance

The most promising characteristic observed in Table 9.2 is the performance of the E-L cases in comparison to Eulerian solutions of similar computational cost. In light of this an additional measure is introduced to provide an indication of overall performance, by considering the balance between computational cost and accuracy for each method.

The measure should be formulated so that it is approximately constant for the six different solutions to Case SV3, as this would permit a comparison of each convective scheme that is independent of resolution. From the dependence of computational cost on grid size for Case SV3, shown in Fig. 9.18, it may be inferred that:

$$T \approx c_1 \, \Delta^{-3} \tag{9.5}$$

Where T is execution time and c_1 is a constant. Total error E, on the other hand, has been found to exhibit a weak non-linearity in its dependence on cell size (as shown in Fig. 9.16(b)), but is taken to be approximately linear over the present range of grid sizes:

$$E \approx c_2 \, \Delta \tag{9.6}$$

Where c_2 is another constant. Re-arranging Eqs. 9.5 and 9.6 to express Δ in terms of T and E, respectively, yields:

$$\Delta \approx 1/\sqrt[3]{T/c_1} \tag{9.7}$$

$$\Delta \approx E/c_2 \tag{9.8}$$

Equating Eqs. 9.7 and 9.8 subsequently results in:

$$E\sqrt[3]{T} \approx c_2\sqrt[3]{c_1} = \text{constant} \tag{9.9}$$

This suggests that the product on the left-hand side of Eq. 9.9 may be used to provide a general indication of the performance of each scheme. The quantity should ideally be as low as possible; furthermore a 'good' solution may have low total error or a short execution time, although the linear dependence on error means that the measure 'prefers' an accurate solution to a quick execution. Although this quantity is not non-dimensional, its dimensions do not have any particular physical significance and are subsequently neglected. Figure 9.21 compares the standardised performance of the numerical methods used in Cases SV2 to SV6, where Eulerian simulations are shown in blue and E-L solutions are in red. The solution to SV6 without deconvolution and with modified filtering parameters is also included.

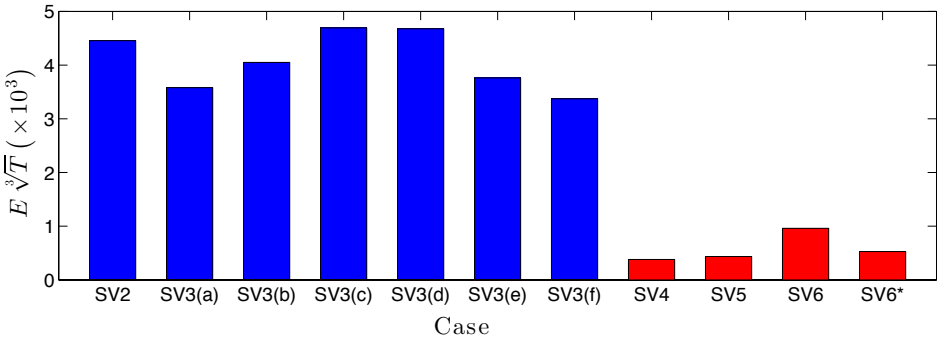


Figure 9.21: Standardised performance. (*With no deconvolution, and modified filtering).

This analysis provides an indication of the improvement in performance that the E-L method is capable of delivering for the single vortex configuration; the additional cost of the E-L solutions is justified by the significant reduction in error in the final transported field. It should be noted that the Eulerian schemes perform particularly poorly for this configuration, however, as the distortion of the scalar field creates a large perimeter over which numerical diffusion will have an impact.

The standardised performance measure introduced here is also applied to solutions from the previous configurations presented in Chapters 6 and 7. The standardised performance is useful for these previous cases as the E-L method has been shown to perform variably under the variety of physical conditions that these cases have covered. Figure 9.22 compares the performance of the four 'best' E-L solutions presented in Table 8.13, where the standardised measure for each case is normalised by (i.e. expressed as a percentage of) the standardised measure for the corresponding Eulerian solution in Table 8.13.

From Fig. 9.22 it can been seen that the E-L method performs particularly well for the original Zalesak case (Case Z9), and for the long-running modified Zalesak configuration (Case ZM12(2), with modified decomposition parameters). However, for the single-revolution modified Zalesak case at a high Schmidt number (Case ZM4), the standardised performance of the E-L method is worse than that of an Eulerian method. This suggests that even though the E-L method may deliver a lower

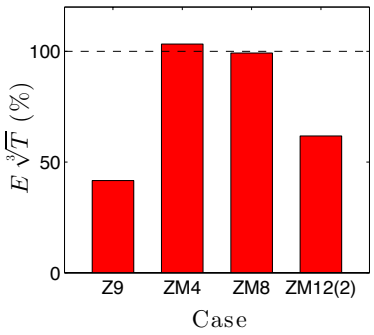


Figure 9.22: Standardised performance, Zalesak and modified Zalesak simulations.

total error, it would be possible to improve on the E-L solution by increasing the resolution of the corresponding Eulerian simulation while maintaining a lower computational cost. The performance of the E-L method for the low Schmidt number modified Zalesak case (ZM8) is found to be almost equivalent to an Eulerian method.

9.5 Conclusions

In the present chapter the joint Eulerian-Lagrangian method has been used to compute the transport of a scalar quantity within a vortical flow-field, using the test case originally suggested by Rider and Kothe [185]. The shearing motion of the flow, which is reminiscent of that which may be found within real turbulence, causes a significant distortion of the scalar topology: an initial disc shape is stretched into a thin, spiralling filament. By reversing the direction of the flow-field half-way through the simulation, the final scalar field should ideally match the initial field.

Eulerian solutions to the single vortex problem have been generated using a high-order CDS, the CHARM TVD scheme, and the SuperBee TVD scheme. The former proves to be very accurate if only the initial and final fields are considered; however, the intermediate field is strongly affected by oscillations consistent with the use of a central approximation for the scalar value at a cell face. Solutions from the two TVD schemes suffer from the effects of significant numerical diffusion, where the sharp perimeter of the filament is almost entirely lost.

As the better of the two TVD schemes the SuperBee limiter is also applied for simulations over a variety of grid resolutions, where the highest resolution simulation employs a cell size that is one-fifth of the original simulation. Mesh refinement is shown to provide a substantial improvement in the description of the transported field, but at the expense of an increased computational cost, while numerical diffusion is still found to have a significant effect. The total error measure applied here varies approximately linearly with cell size for the SuperBee cases, while execution time scales with the cube of the resolution.

The E-L method is capable of delivering a very accurate representation of the scalar field, even at a relatively low Lagrangian particle density. By formulating a standardised measure of performance based on total error and execution time, it is suggested that the improvement in accuracy of the E-L method out-weighs its additional computational cost for this case. In other words, an Eulerian

simulation with a grid refinement sufficient to match the accuracy of an E-L solution would be far more computationally expensive.

This standardised performance measure is also applied to the previous 'best' solutions from the original and modified Zalesak cases, and indicates that the E-L method also out-performs the Eulerian approach for the original Zalesak configuration and the long-running modified Zalesak test. However, it should be noted that the application of this measure to these cases continues to assume a linear dependence of total error on cell size.

This part of the investigation has revealed some additional weaknesses in the E-L method. The deconvolution algorithm is found to create 'spikes' within the Lagrangian reconstruction when the particle field is not aligned with the underlying Eulerian mesh, as it will tend to amplify a 'resolution error' that is quite separate from the deconvolution error it is designed to eliminate. Deconvolution is also found to be inappropriate for cases with low Lagrangian particle densities as the amplification of noise within deconvoluted regions may negate any improvement gained by the deconvolution itself. The low-pass filtering algorithm exhibits some shortcomings for low particle density simulations, where it is suggested that the recommended parameters are modified to make the algorithm more aggressive in identifying regions for filtering. This becomes necessary as attenuation of the larger amplitudes of noise in the Lagrangian reconstruction requires a wider effective filter width.

Sufficient testing has now been performed on the joint Eulerian-Lagrangian method in two dimensions. A number of improvements and modifications have been suggested and developed throughout the investigation, and the method is now considered to be in a suitable state for application to a three-dimensional flow of physical significance. Such a case will be presented in Chapter 10.

10 Test Case 4: The Mixing Layer

Chapters 6, 7 and 9 have presented the results of a comprehensive and rigorous test of the joint Eulerian-Lagrangian (E-L) method, and have provided insight into its behaviour and performance. A number of modifications to the basic method proposed in Chapter 5 have been suggested and developed in Chapter 8, and evaluated in Chapters 8 and 9, such that the method is considered to be optimal at this stage. It would now be instructive to consider a three-dimensional, physically meaningful flow, to see how the method behaves for a 'real world' problem.

The E-L method is implemented in a full version of the incompressible `PsiPhi` flow solver, and is applied to the simulation of a planar shearing flow. The resolution of the computational domain is fine enough that the Kolmogorov length-scale is sufficiently resolved; accordingly no turbulence modelling is required, as the code is performing a Direct Numerical Simulation (DNS) of the flow.

To keep the present case as simple as possible, the transported scalar quantity continues to represent a passive scalar that moves with the flow-field but does not have any influence upon it. This avoids any complications associated with coupling between the scalar and the flow.

10.1 Case Objectives

The main objective of the mixing layer test case is to gain insight into the applicability of the E-L method for simulations of physically representative situations. A realistic, turbulent flow will contain highly complex, chaotic structures, and these will cause even greater distortions of the scalar field than in previous cases. Such a problem will therefore interrogate every aspect of the E-L method and should immediately highlight any shortcomings in its formulation or in the chosen method configuration.

The E-L method has been developed in such a way that the extension from two spatial dimensions to three should be as simple as possible, while some initial three-dimensional testing of the code has already been performed in Chapter 8. The mixing layer case now provides an opportunity to validate the method for a fully three-dimensional configuration.

Considering the results that have been presented so far – and in particular, the standardised performance comparisons made in Chapter 9 (Figs. 9.21 and 9.22) – it is apparent that the E-L method may not always be the most suitable choice for a given configuration, particularly where a scalar quantity experiences high diffusive fluxes (although the method should be viable and accurate for such cases). The mixing layer study is therefore designed to provide further guidance as to which cases are good candidates for its application, and this will be achieved by performing simulations under a variety of physical conditions (Schmidt numbers).

The computational cost of the present case will be considerably greater than that experienced in

previous chapters, and this will allow a more detailed and representative analysis of that cost to be made. The significant number of cells required, resulting from the very high resolution demanded by a DNS, means that simulations will need to be performed on several processes operating in parallel. This will test the various Message Passing Interface (MPI) subroutines that have been implemented for the E-L method to be applied on massively-parallel computing architectures. (Parallelisation must be implemented for almost every aspect of the E-L method; it becomes particularly significant within the filtering operations used for decomposition and localised low-pass filtering, as well as during Lagrangian particle transport and reconstruction).

Unlike the previous test cases an analytical solution for the mixing layer scalar field will not be available. Although the simplicity of the flow geometry makes a time-averaged solution relatively straightforward to predict, a DNS is often too expensive to run to statistical convergence. Instead, the instantaneous structures of the scalar topology are considered, and these will be partly dependent on the randomly-generated turbulence that is super-imposed upon the initial flow-field. By storing and re-using that initial turbulence it is possible to ensure that the simultaneous flow-fields of solutions from different simulations will be identical. In terms of accuracy, then, a qualitative assessment will be made by comparing E-L solutions to corresponding computations performed with a traditional Eulerian scheme.

Although a time-dependent analytical solution is infeasible for a mixing process driven by both inertial and viscous forces, such a solution can be found easily for a diffusion-only process where the scalar profile will resemble a Gaussian cumulative distribution. This will be used as the basis for a quantitative analysis of the transported scalar field in each case; comparisons will be made between the 'widths' of the mixing layer, the scalar gradients, and the derived scalar dissipation rates, to further investigate any differences between the Eulerian and E-L solutions.

10.2 Case Description

A large number of plane mixing layer flows have been studied experimentally and computationally, as they typically consist of a very simple and accessible geometry while still involving a complex flow. Planar mixing also has considerable practical importance as it is often the main mode of convective mixing in, for example, combustion devices [214]. The usefulness of the plane mixing layer as a test case here lies in its inherent (physical) instability, where the shearing motion between two streams generates a procession of Kelvin-Helmholtz vortices. These structures in themselves have appreciable physical significance, being the driving force behind the development of, for example, ocean waves.

In experiments it is common to generate a plane shear layer after a thin splitter plate between two streams of differing velocity, density, or both. Early experimental work was performed by Brown and Roshko, who investigated density and compressibility effects in nitrogen-helium plane mixing layers [215, 216], while Bernal and Roshko measured streamwise vortex structures in both (gaseous) nitrogen-helium planar mixing and with (liquid) streams of dyed water [217]. Amongst other notable developments Breidenthal devised an improved visualisation technique by the plane mixing of two liquid dilute chemical reactants to form a visible product [218, 219], while Papamoschou and Roshko

studied mixing in supersonic flows up to a convective Mach number $M_c = 4.0$ [220].

In simulations it is sometimes more convenient to consider the relative velocities of the two mixing streams, as lower flow speeds permit longer computational time-steps (and hence less computational expense). In such cases the plane mixing layer is then defined in terms of a velocity u_0, where two streams of opposing speed are moving at $\pm\frac{1}{2}u_0$. This approach is used in the early DNS work of Comte et al. [221], who used 128^3 grid points to simulate turbulence vorticity and passive scalar mixing, as well as by Vreman et al. [222], who tested a selection of turbulent stress tensor closure models by Large-Eddy Simulation (including the Smagorinsky and dynamic Germano techniques described in §3.5). More recently Patano and Sarkar [223] studied density and compressibility effects using DNS with up to $512 \times 256 \times 128$ grid points at convective Mach numbers up to $M_c = 1.1$. Patano et al. [224] used the same simulation configuration, but at slower flow speeds ($M_c = 0.3$, i.e. quasi-incompressible), to study scalar mixing in a turbulent reacting shear layer under the assumption of infinitely fast chemistry. Mahle et al. [225] explored the effects of compressibility by performing similar reactive simulations at $M_c = 0.15$ and $M_c = 0.7$.

The present case consists of a cuboid domain measuring $18 \times 10 \times 6$ mm. The upper and lower domain boundaries are defined as walls, while boundaries in the horizontal (x) and lateral (z) directions are periodic. In this manner the configuration may be considered to represent the flow between two infinite parallel plates. Between these plates are initialised two horizontal streams of equal but opposite momentum, where the interface is parallel to the enclosing walls and is located at $y = 5$ mm. The velocities of the lower and upper streams are set uniformly to $u_0^{++} = 2$ m/s and $u_0^{-} = -2$ m/s, respectively[1], with the exception of a thin near-wall layer where flow speeds are attenuated to zero at the wall. (Such a layer proves to be useful during the earliest stages of each simulation, where the pressure correction algorithm attempts to correct any continuity errors within the initial flow-field). Figure 10.1 shows a schematic of the domain and initial bulk flow conditions.

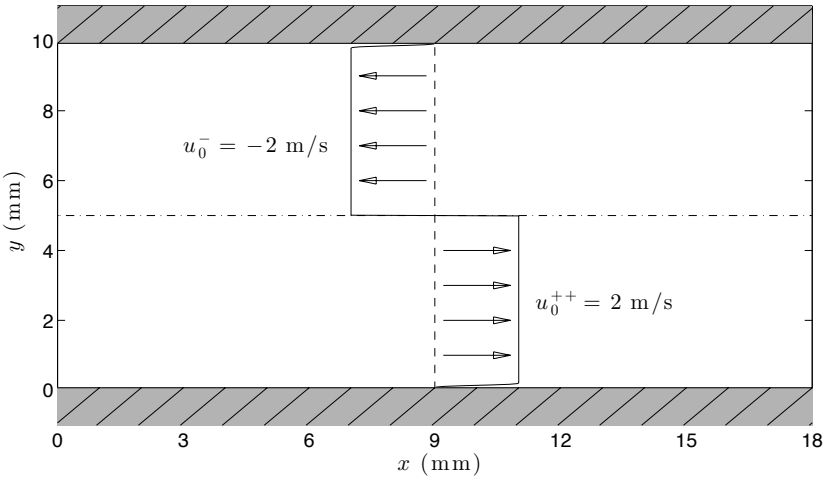


Figure 10.1: Initial mixing layer bulk flow profile.

The initial scalar profile is also very simple: the u^{++} (lower) stream is initialised with $\phi = 1$,

[1]The notation u^{++} is applied here to avoid confusion with the commonly-used non-dimensional wall velocity u^+

while the opposing stream is set to $\phi = 0$. The initial interface between the two regions is 'sharp', varying across the height of a single computational cell. The physical properties of the two streams are assumed to be identical to those of air at standard ambient temperature ($T = 298.15$ K) and atmospheric pressure ($p = 1.01325 \times 10^5$ Pa). The quantity ϕ then represents the concentration of an arbitrary, inert, neutrally-bouyant gas, whose addition otherwise has no influence on the physical properties of the fluid.

The intention for the present case is to simulate the mixing layer without any modelled contribution to the flow, which demands a cell size on the order of the smallest scales within that flow. Accordingly it is necessary to estimate the Kolmogorov scale, η_K, based on prior knowledge of the flow properties. An expression for η_K has already been provided in Eq. 2.19; however, this requires the dissipation rate ε, which is an unknown quantity. Following the approach of Taylor, an approximation for ε is formulated by considering that the rate of energy dissipation within a turbulent flow is equal to the rate at which kinetic energy is being produced by that flow [226]. In general, the turbulent kinetic energy k per unit mass may be expressed as:

$$k = \frac{1}{2}u'^2 \tag{10.1}$$

The production of k is assumed to be inversely proportional to the integral time-scale of the vortices within the flow, which is equal to L/u'. The kinetic energy production rate, and therefore the dissipation rate, may thus be approximated as:

$$\varepsilon \approx \frac{u'^2}{L/u'} = \frac{u'^3}{L} \tag{10.2}$$

Substitution of Eq. 10.2 into Eq. 2.19 subsequently yields:

$$\eta_K \approx \left(\frac{\nu^3 L}{u'^3}\right)^{\frac{1}{4}} \tag{10.3}$$

Applying Sutherland's law [201] (Eq. 5.53) for air at standard ambient temperature yields a dynamic viscosity $\mu = 1.846 \times 10^{-5}$ kg/m/s. The fluid density, meanwhile, may be obtained from the equation of state for an ideal gas at the pressure and temperature prescribed above, yielding $\rho = 1.177$ kg/m^3. According to the definition in §2.1.2, the fluid kinematic viscosity is therefore $\nu = 1.569 \times 10^{-5}$ m^2/s.

All that is required for an approximation of the Kolmogorov length-scale are estimates of the integral length- and velocity-scales for Eq. 10.3. If we assume that the length-scale of the largest eddies will be of the order of half the plate separation ($L \approx 0.005$ m), while the velocity fluctuation will be roughly equal to the speed of each stream ($u' \approx 2$ m/s), application of Eq. 10.3 returns a Kolmogorov scale estimate of $\eta_K \approx 3.94 \times 10^{-5}$ m $= 39.4$ μm.

A grid resolution may now be chosen to ensure that turbulent scales are fully-resolved. However, some consideration should also be given to the length-scales of the scalar field itself: the intention is to perform simulations at a variety of Schmidt numbers, which means that the turbulent characteristics of the scalar field will be quite different to those of the flow-field. For the highest Schmidt

number configuration – details of which will be provided in §10.3 – diffusive scalar fluxes will approach zero, meaning that scalar features will become very small. To quantify the limit of scalar feature size it is useful to consider a local scalar dissipation length-scale, which is analogous to the Kolmogorov scale at which turbulent kinetic energy dissipates. An extension to Kolmogorov's description of the turbulent energy cascade (Eq. 2.15) for passive scalars was proposed independently by Obukhov [227] and Corrsin [228], who formulated an expression relating the spectral density k_ϕ of the scalar to the dissipation rates:

$$k_\phi(\omega) = C_\phi \, \chi \, \varepsilon^{-1/3} \, \omega^{-5/3} \tag{10.4}$$

Where χ is the *scalar* dissipation rate, which will be discussed in greater depth in §10.5.2, and C_ϕ is the Obukhov-Corrsin constant, which is found to have a value of around 0.4 from experiments [229]. From this Obuhkov and Corrsin proposed what is now known as the Obukhov-Corrsin scale, η_ϕ, at which the smallest scalar features will occur:

$$\eta_\phi = \frac{\eta_K}{Sc^{3/4}} \approx \left(\frac{\nu^3 L}{u'^3} \right)^{\frac{1}{4}} Sc^{-3/4} \tag{10.5}$$

However, as discussed by Vaishnavi et al. [230], Eq. 10.5 is valid only for Schmidt numbers less than unity; for configurations with $Sc \geq 1$, an alternative definition has been proposed by Batchelor [231]:

$$\eta_B = \frac{\eta_K}{Sc^{1/2}} \approx \left(\frac{\nu^3 L}{u'^3} \right)^{\frac{1}{4}} Sc^{-1/2} \tag{10.6}$$

Where η_B is the Batchelor scale. (Clearly the expressions in Eqs. 10.5 and 10.6 will both tend to η_K as $Sc \to 1$). Based on the approximation for η_K it is thus possible to estimate how well-resolved the scalar field itself will be for each of the simulations described in §10.3.

10.3 Simulation Numerics

In order to perform a DNS of the mixing layer case the estimated Kolmogorov length-scale must be sufficiently resolved, although Moin and Mahesh stipulate that the required degree of resolution additionally depends on the accuracy of the spatial schemes applied in the simulation [232]. However, in their detailed discussion of spatial resolution requirements, Moin and Mahesh also suggest that:

> 'The Kolmogorov length-scale, η, is commonly quoted as the smallest scale that needs to be resolved. However, this requirement is probably too stringent. The smallest resolved length-scale is required to be of $O(\eta)$, not equal to η.'

Moin and Mahesh subsequently identify the good agreement of early spectral DNS results with experiments (as presented by, for example, Moser and Moin [233] and Spalart [234]), where η_K was not resolved; they therefore propose that reliable velocity statistics (means and variances) will be obtained as long as the grid captures 'most' of the turbulence dissipation, which will occur at scales above η_K. Other authors (for example Yakhot and Sreenivasan [235], and Donzis et al. [236]) have

suggested that the accurate simulation of localised anomalous scaling behaviour and intermittency in three-dimensional turbulence may demand resolutions finer than the Kolmogorov scale, while Wan et al. [237] also point out the need for finer resolutions to recover statistical moments higher than the first (mean) and second (variance) orders.

Based on these considerations the present computational domain is discretised on to a grid of cubic cells of size $\Delta = 20$ μm, meaning that the estimated Kolmogorov scale is 'over-resolved' by around a factor of two. Discretisation of the simulated volume described in §10.2 at this resolution equates to a computational domain of $900 \times 500 \times 300$ cells in the i-, j- and k-directions, respectively, for a total of 135 million cells. However, one additional Eulerian simulation will be performed at an even finer resolution of $\Delta = 10$ μm, where dimensions will be $1800 \times 1000 \times 600$ totalling 1.08 billion cells. Simulations are performed on the 26.2 TeraFLOP [238] Cray XT6m architecture of the University of Duisburg-Essen, which consists of 344 12-core, 1.9 GHz AMD Magny Cours processors and offers a total of 4.5 Terabytes of Random Access Memory (RAM). Taking into account the resources available and the expected execution time the present domain is decomposed in three dimensions across a total of 144 sub-domains (parallel processes), or 480 sub-domains for the finer case. This also allows for the fact that the parallel efficiency of a code will typically decrease as the number of processes is increased, due to the shift in balance between computation and communication costs: doubling the number of processes in each direction of a three-dimensional domain decomposition will reduce the computation (number of cells in a sub-domain) on each process by a factor of eight, but will only reduce communication costs ('surface area' of a sub-domain) by a factor of four. Assigning just under one million cells (or just over two million for the finer case) to each process should maintain a reasonable parallel efficiency, while also avoiding restrictions due to the amount of RAM available to each process.

To ensure the stability of the flow solution the CFL criterion is set to a relatively low value of 0.15. This value results from initial investigations that demonstrate the susceptibility of the present configuration to a phenomenon known as 'checker-boarding', where an instability within the density field can create a significant additional overhead within the pressure correction algorithm. Based on this CFL number, the cell size given above, and the magnitude of the prescribed flow-field velocities, each time-step will have an expected duration of $dt \approx 1.5 \times 10^{-6}$ s (or $dt \approx 7.5 \times 10^{-7}$ s for the finer case).

The stability of the simulation is augmented by linearly 'ramping up' the CFL criterion over the first 100 time-steps. This reduces the workload of the pressure correction algorithm early on in the simulation – considering that the initial turbulence field may not satisfy continuity – and further mitigates the development of unphysical artefacts within the transported density field. Correction of the initial turbulence field will also tend to create a number of pressure correction waves throughout the domain, and initial results demonstrate that constructive interference between these waves may de-stabilise a simulation. In particular the waves contribute to the formation of the checker-boarding effect mentioned previously. This occurs because the simulated vessel is effectively closed; its periodicity means that the pressure correction waves cannot leave the domain. As a further stability measure, then, the mass error (and, therefore, pressure correction) is set to zero within the 'halo' (or 'ghost') regions of the periodic boundaries during each time-step, resulting in the

cancelling of the pressure correction waves as they pass from one side of the domain to the other.

The total simulated time in this case only needs to be long enough for the satisfactory generation of Kelvin-Helmholtz vortices. Based on the estimated integral length-scale and the relative flow velocities, a duration of 20 ms would be equivalent to eight 'eddy turnover' times. Based on the approximation for dt above this is expected to result in a total of around 13000 time-steps for each simulation at $\Delta = 20$ μm. The advancement in time is performed using the same low-storage, third-order accurate Runge-Kutta scheme applied in previous chapters and described in §3.2.

The convective transport of active scalar quantities such as density and the components of momentum is performed using an eighth-order Central Differencing Scheme (CDS8). (These quantities will always vary smoothly in space at the chosen grid resolution, so that CDS may be used without risk of oscillatory behaviour). Gradient approximations for the calculation of diffusive momentum fluxes are also made with eighth-order accuracy.

The initial turbulent field consists of three components of velocity, and is generated using the digital filtering approach described in §3.7.1. The width of the applied Gaussian filter is chosen such that the resulting fields have a length-scale $L = 0.1$ mm, and these fields are subsequently scaled to have a turbulent kinetic energy $k = 0.24$ m^2/s^2 per unit mass. This corresponds to initial velocity fluctuations of $u_i' = 0.4$ m/s. It should be noted that the purpose of this turbulence is only to disrupt the initial bulk flow: without some sort of disturbance the transition from laminar to turbulent flow would take a considerable (physical, and computational) time, being precipitated only by asymmetries from precision or rounding errors in the flow-field. The turbulence parameters are selected such that the initial small-scale velocity fluctuations will be rapidly dissipated by the molecular viscosity of the simulated fluid. Once they have served their purpose in disrupting the initial flow they should not, therefore, have a significant overall effect on the flow-field. The three turbulent velocity fields are written to file when the first simulation is performed, and are subsequently re-used for all other simulations so that the flow-fields are identical throughout the study. Furthermore, by interpolating between points within those stored fields, it is possible to recreate the same initial conditions for the simulation at a finer grid resolution.

To further explore the applicability and behaviour of the E-L method, simulations are performed at a variety of Schmidt numbers. Corresponding Eulerian simulations employing a Total Variation Diminishing (TVD) scheme (with the CHARM flux limiter) are carried out for comparison. Table 10.1 summarises these cases. The lowest Schmidt number is chosen to be representative of a typical gas mixture (for example, methane in air); the highest value would be unusual in the gas phase, but would be characteristic of a liquid mixture such as ethanol in water. Included for each value of Sc are the corresponding Obukhov-Corrsin or Batchelor scales, with respect to Δ, as estimated from Eqs. 10.5 and 10.6 respectively. Note that these scales are *not* resolved for Cases ML2 to ML5, and further discussion of this point will be provided in §10.5. The effects of increasing the grid resolution will be most apparent at higher Schmidt numbers, while the significant computational expense expected at $\Delta = 10$ μm discourages the realisation of more than one such simulation. In addition it would seem unnecessary to apply a higher resolution for the E-L method, as it has already been shown to out-perform more highly-resolved Eulerian simulations in Chapter 9. A resolution of $\Delta = 10$ μm is thus used for Case ML5-EU only.

Table 10.1: Simulation configurations for the mixing layer study.

Case	Δ (μm)	Dimensions (Cells)	Method	Particle Removal	Sc	Scalar Length-scale	
						η_ϕ/Δ	η_B/Δ
ML1-EU	20	$900 \times 500 \times 300$	CHARM TVD	-	0.7	2.574	-
ML2-EU					5	-	0.881
ML3-EU					20	-	0.441
ML4-EU					100	-	0.197
ML5-EU					1000	-	0.062
ML5-EUf	10	$1800 \times 1000 \times 600$	CHARM TVD	-	1000	-	0.125
ML1-EL	20	$900 \times 500 \times 300$	Eulerian-Lagrangian	✓	0.7	2.574	-
ML2-EL					5	-	0.881
ML3-EL					20	-	0.441
ML4-EL					100	-	0.197
ML5-EL					1000	-	0.062
ML4-ELb	20	$900 \times 500 \times 300$	Eulerian-Lagrangian	×	100	-	0.197
ML5-ELb					1000	-	0.062

For E-L simulations the configuration used in previous chapters is assumed. The width of the Gaussian filter kernel used for the initial scalar decomposition is set to $\sigma_f = 5\Delta$, requiring $n_f = 20$ support points in each direction. For this highly-resolved three-dimensional configuration a Lagrangian particle density of $\mathrm{L}^\varrho = 20$ is selected; although higher particle densities would lead to improved accuracy (i.e. less noise in the particle reconstruction), the computational cost of a large number of particles must also be considered. For the highest Schmidt number case in particular the surface area of the scalar interface is expected to expand drastically over the course of the simulation, which will lead to a significant increase in the total particle number. Considering the amount of RAM available to each processing core, and the number of computational cells per process, a hard limit of around 15 million particles per sub-domain is imposed.

For re-initialisation the particle addition parameter (defining the upper limit of the second derivative of the low-frequency scalar field) is set to $\gamma = 1$. Scalar information will therefore be transferred from the Eulerian phase to the Lagrangian phase in regions where the smoothness of the low-frequency field becomes less than that at the start of the simulation. For those cases with particle removal parameter values of $\lambda = 0.7$ and $\beta = 0.2$ are applied (as suggested in §8.1), where an associated reduction in particle number is expected to reduce the overall cost of the simulation. However, the studies presented in §8.4.1 suggest that particle removal may also reduce the accuracy of the E-L method under high Schmidt number conditions; accordingly Cases ML-EL4 and ML-EL5 are also performed without particle removal, to gain further insight to how the mechanism behaves for these higher Schmidt number flows.

Results from the single vortex study presented in Chapter 9 indicate that localised deconvolution is not appropriate for E-L simulations at low Lagrangian particle densities due to the amplification of noise in a deconvoluted region exceeding the improvement gained from the deconvolution. This algorithm is not, therefore, applied here. Localised low-pass filtering, on the other hand, is expected to play an important role in reducing the noisiness of the Lagrangian reconstruction; following the

extended analysis of Case SV6 presented in §9.4.4 parameters of $a = 0.025$, $b = 3\Delta$ and $w_{\mathrm{TH}} = 2\Delta$ are selected.

10.4 Flow-Field Development

Before analysing the performance of the E-L method for the present mixing layer case, the development of the flow-field by which the scalar is transported is considered. Inspection of the flow-fields from the different simulations in Table 10.1 confirms that re-use of the same initial (and for Case ML5-EUf, interpolated) turbulence results in an identical flow for each simulation.

Figure 10.2 shows the initial horizontal and vertical turbulent velocities super-imposed on to the bulk flow shown schematically in Fig. 10.1. (Initial lateral velocities are comparable to vertical velocities, and are omitted). The mid-grey bands at the top and bottom of Figs. 10.2(a) and 10.2(b) represent the zero velocities imposed within the bounding walls; the prescribed boundary layer at each of these surfaces is also just discernible. More clearly visible in each of these images are the isotropic, small-scale turbulent structures, which are intended to dissipate quickly while promoting a transition to turbulence between the two streams.

The flow-field at $t = 2.5$ ms (Fig. 10.3) shows some evidence of the initial turbulence in both the horizontal and vertical velocity components, although those initial fluctuations have largely dissipated due to the molecular viscosity of the fluid. Most notably a developing procession of Kelvin-Helmholtz vortices is observed between the two streams, where five small vortex cores can be seen to have formed. At this early stage those vortices appear to be stretched, measuring typically 2 mm long by around 1 mm high; in addition the vertical velocities at the interface ($y = 5$ mm) within a vortex are lower than the corresponding horizontal velocities within that vortex. Both of these observations indicate that the vortices are not yet fully-developed. Boundary layers at the upper and lower wall surfaces are now much more clearly visible in Fig. 10.3(a), although their influence is far-removed from the interface.

Finally the flow-field is inspected at $t = 12$ ms (Fig. 10.4): no evidence of the initial turbulence remains, while the small vortices observed in Fig. 10.3 have agglomerated into two larger structures each of around 5 mm in diameter. Maximum and minimum vertical velocities within those structures are approximately equal to the bulk horizontal velocities, suggesting that the vortices have been further accelerated by the surrounding flow. Although the boundary layers at the walls have continued to thicken, their influence is still separated from the mixing layer.

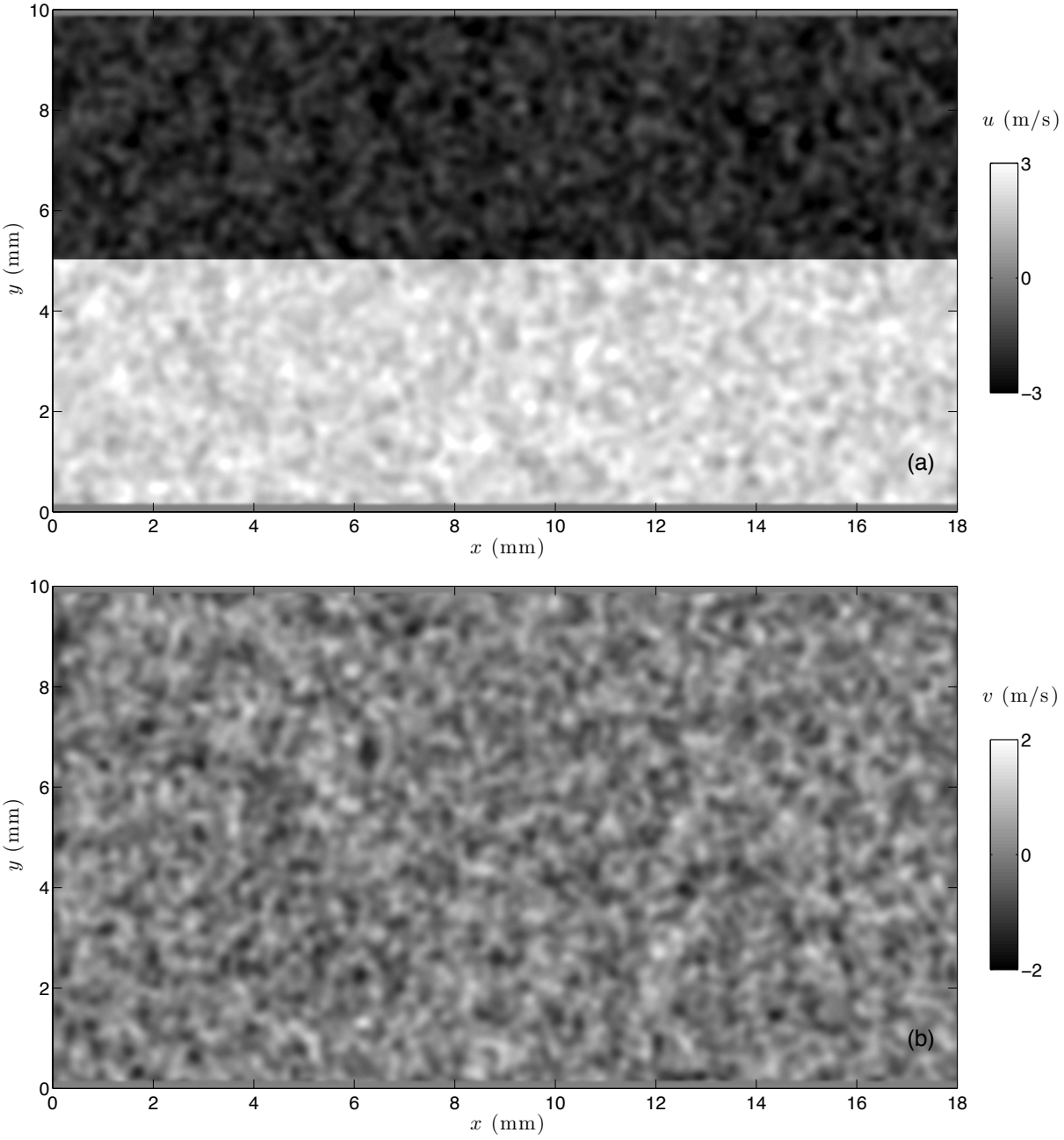


Figure 10.2: Initial mixing layer flow-field; (a) horizontal and (b) vertical velocities.

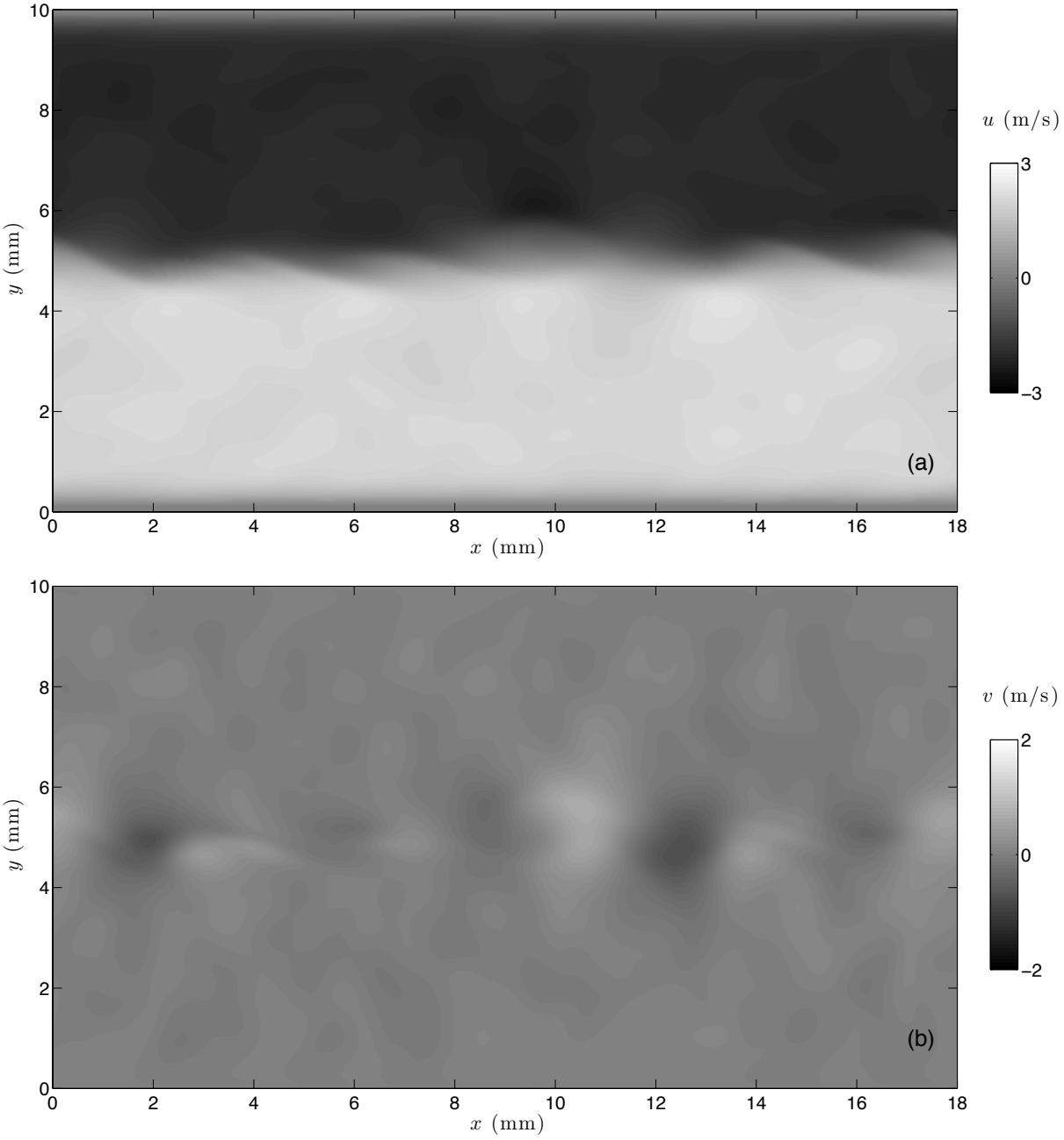


Figure 10.3: Mixing layer flow-field at $t = 2.5$ ms; (a) horizontal and (b) vertical velocities.

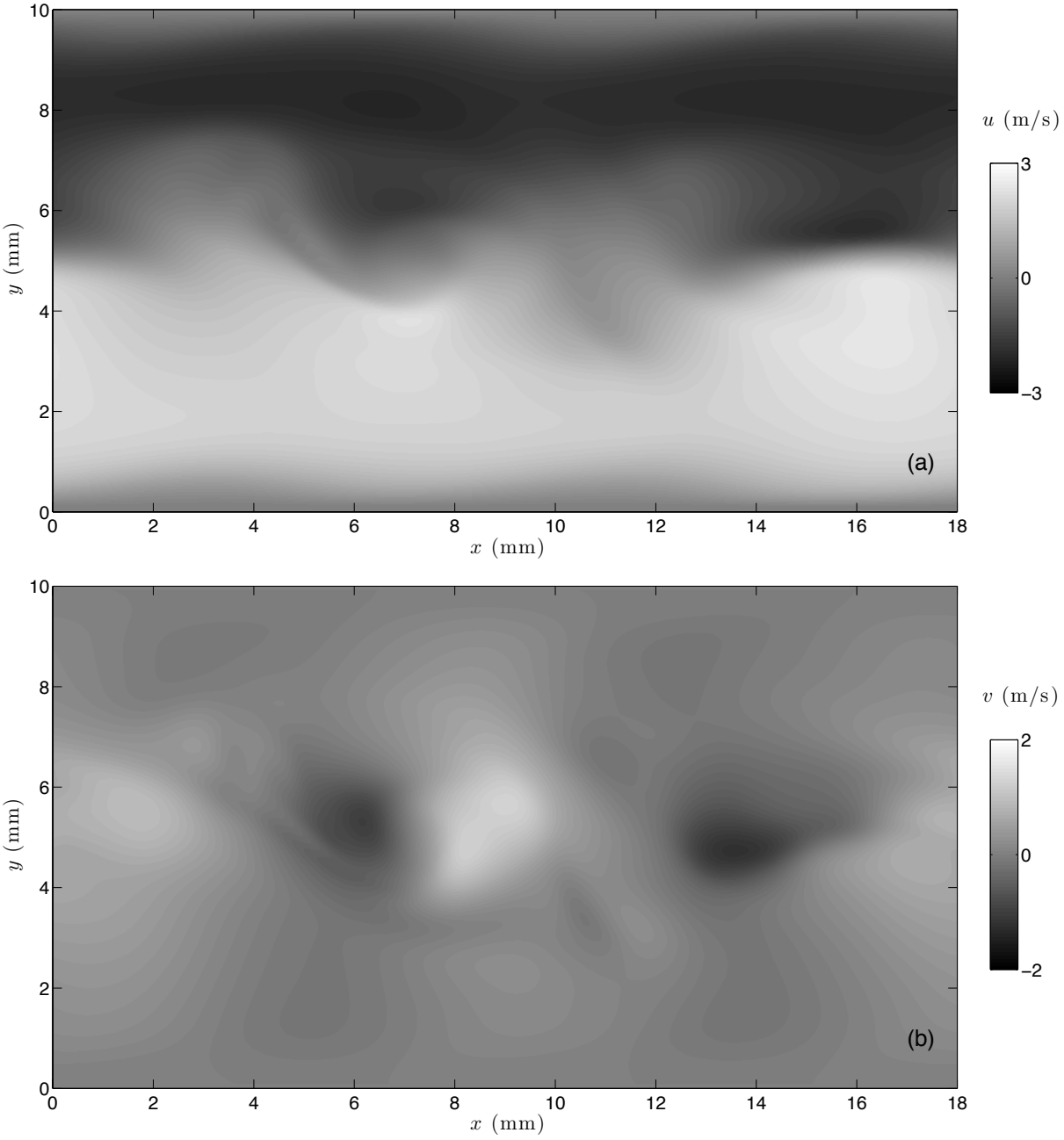


Figure 10.4: Mixing layer flow-field at $t = 12$ ms; (a) horizontal and (b) vertical velocities.

10.5 Results and Discussion

The following analysis of the mixing layer simulations is divided into three sections. Firstly the transported scalar fields are inspected directly, to allow a qualitative assessment of the performance of the Eulerian and E-L methods for each case. Next the topology of each solution is quantified in terms of the scalar gradient and mixing layer thickness, and is compared between methods and against the analytical solution for a one-dimensional diffusion-only configuration. The scalar gradient is also used to approximate the instantaneous scalar dissipation rate, to see how predictions of this quantity differ for each solution. Finally the computational cost of each simulation is considered, with reference to the total number of particles required in each E-L case.

10.5.1 Scalar Fields

Figures 10.7 to 10.11 show 'slices' of the scalar field for the configurations listed in Table 10.1, through the vertical mid-plane of the domain (i.e. at $z = 3$ mm), at various times. Results are shown without particle removal for the E-L solutions to Cases ML4 and ML5; further discussion of the effects of particle removal will be provided subsequently. The effect of grid refinement on the Eulerian solution to Case ML5 is also presented separately.

Case ML1 The first visual comparison is drawn between the Eulerian and E-L solutions for Case ML1, where the transported fields at $t = 20$ ms are shown in Fig. 10.7. The two images show that the solutions are almost completely identical, and further analysis reveals that there is a total absolute difference of just 0.002% between the Eulerian and E-L results. (This qualitative analysis provides no indication of the absolute accuracy of each method; for example, any temporal errors are likely to be consistent between the two solutions).

Within the transported scalar field itself a variety of structures that have been created by interactions between the two streams are observed. Within the left-hand half of the domain the scalar field is generally smooth; on the right-hand side, however, there exists a relatively sharp 'ridge' around $(x, y) = [12, 4]$ mm, which has formed in the highly-strained region between two adjacent vortices. In general feature sizes within the scalar field are similar to those observed within the velocity fields presented in Fig. 10.4, as the momentum diffusivity is comparable to the diffusivity of the scalar. Any small scalar feature forming within the flow is rapidly dispersed by viscous forces, as its perimeter (across which diffusion acts) will be large in relation to its volume. This corresponds to the relatively large Obukhov-Corrsin scale estimated for this configuration, where scalar features (fluctuations) will dissipate more rapidly than turbulent features (i.e. $\eta_\phi > \eta_K$).

Noticeably absent from Fig. 10.7(b) is any evidence of Lagrangian particles (i.e. noise from the particle reconstruction), which suggests that the particle removal algorithm has been effective in removing particles from the smoother regions of the flow. Although the scalar field does contain some relatively sharp changes in gradient, the convective transport processes generating these sharper fronts will be countered by the significant physical diffusive fluxes of this configuration. This balance of transport processes is quantified by considering the 'grid' Reynolds number Re_Δ, based on the bulk flow velocity and the cell size Δ, from which the mass Péclet number (introduced in §2.1.2) at

the surface of a cell is evaluated as:

$$Pe = Re_\Delta \cdot Sc = \frac{u_0 \, \Delta}{\nu} \, Sc \approx 1.8 \tag{10.7}$$

This suggests that in the higher-velocity regions of the flow the convective fluxes (within a single cell) will be around twice as significant as the diffusive fluxes; in slower regions, where Re_Δ will be lower, diffusion will therefore dominate. (An exact balance between convective and diffusive fluxes (i.e. $Pe = 1$) will occur at a flow velocity of 1.12 m/s). The result of this is that even though the particle addition routine may transfer information from the Eulerian to the Lagrangian phase where gradients are becoming steeper, the particle removal routine will probably remove those new particles shortly after: particles will be rapidly dispersed by relatively high diffusive fluxes, and will quickly become redundant.

Case ML2 Similar results are observed for cases applying a Schmidt number of $Sc = 5$, although the transported scalar fields shown in Fig. 10.8 are clearly much less diffuse than for the previous case: scalar features are more prominent in this configuration, while feature sizes are smaller. At this Schmidt number the finest scalar features may not be resolved, as the estimated Batchelor scale is (slightly) less than Δ. A direct comparison of the Eulerian and E-L solutions shows little variation between the two, while the absolute difference is still low (0.065%). The apparent lack of noise in the E-L solution at $t = 20$ ms again suggests that the particle removal mechanism has transferred all of the scalar information to the Eulerian phase. The mass Péclet number in the bulk flow is larger for this configuration – $Pe \approx 12.8$ – but scalar transport within the slower regions inside and between vortices may still be dominated by diffusion. A unit Péclet number occurs for flow speeds of around 0.16 m/s at this Schmidt number.

Case ML3 Figure 10.9 shows the transported scalar fields after $t = 18$ ms with $Sc = 20$, where we now observe a distinct noisiness in some regions of the E-L solution. Furthermore some minor differences between the two fields are now discernible, where some of the scalar features in the E-L summation are very slightly more pronounced than those in the Eulerian field; for example, structures within the region $x = [14-16]$ mm, $y = [2-4]$ mm are noticeably sharper in Fig. 10.9(b). The two fields are very similar overall, however, while regions from which particles have been removed in the E-L solution are largely indistinguishable from the corresponding regions of the Eulerian solution. The Batchelor scale is even more under-resolved for this configuration – by around a factor of two – although there are perhaps no obviously under-resolved scalar features visible in these slices.

The mass Péclet number, based on the free-stream mesh Reynolds number and the present Schmidt number, is now $Pe = 51.0$. This suggests that convection is dominant throughout almost the entirety of the flow, with the exception of only the near-stationary ($u < 0.04$ m/s) regions in the vortex cores.

Case ML4 For the next case the Schmidt number is increased to $Sc = 100$. Transported fields after $t = 13$ ms are shown in Figs. 10.10(a) and 10.10(b) from the Eulerian and E-L approaches, respectively. As stated previously the E-L solution is shown without particle removal (i.e. Case ML4-ELb); briefly, particle removal is found to be unsuitable for a case with very low diffusion, but its effect will be explored in greater depth towards the end of this part of the discussion.

At this Schmidt number the differences between the two solutions become more prominent, as the smaller scalar features within the flow are more defined with the E-L method. The low diffusivity for this configuration means that any deformation of the scalar field that occurs over the course of the simulation persists, rather than being smeared out, for all but the smallest structures. Regions where the numerical diffusion of the Eulerian method seems to have caused an under- or over-prediction of scalar value are more apparent. An example of this is the lenticular shape centred on $(x, y) = [6, 6]$ mm, where some thin layers of zero scalar, appearing black in Fig. 10.10(b), are non-zero (grey) in Fig. 10.10(a). It should be stressed, however, that no absolute assessment regarding the accuracy of either method may be made without comparing with an (unobtainable) analytical solution.

The field has been subjected to significant straining motions around the location $(x, y) = [4, 6]$ mm, where a swirling structure is seen to be stretched into thin filaments as it is drawn into a nearby vortex. Although these very fine features will tend to diffuse away rapidly, at this Schmidt number it is expected that they become finer than the grid itself. In this case the sub-grid features – which may continue to be well-captured by the Lagrangian particle distribution, as previously suggested by Fig. 9.11 – will effectively be spread across the extent of a single cell when the particle field is reconstructed. This has previously been identified as a 'resolution error' in §9.4.2, and is illustrated by Fig. 9.10. More specifically it may be inferred that the failure to capture these features is due to the grid spacing being around five times larger than the estimated Batchelor scale, found from Eq. 10.6, for this Schmidt number.

Case ML5 For the last case we consider a configuration at the very high Schmidt number of $Sc = 1000$, where diffusive fluxes approach zero; the mass Péclet number for computational cells in the free stream is now $Pe = 2550.4$, indicating that convection is entirely dominant. The Eulerian solution in Fig. 10.11(a) and the E-L field of Fig. 10.11(b) now contain significant differences throughout the interfacial region, where features in the Eulerian field are very noticeably more diffuse. A visual comparison of Fig. 10.11(a) and the Eulerian solution from Case ML4 (Fig. 10.10(a)) suggests that the diffusivity (overall, i.e. physical and numerical) in each case is approximately the same. This leads to the suggestion of an 'effective' diffusivity D_{eff} experienced by the scalar, which may be expressed as:

$$D_{\text{eff}} = \frac{\nu}{Sc} + D_{\text{num}} \tag{10.8}$$

Where the numerical diffusivity D_{num} depends on the chosen Eulerian scheme only, and will therefore be constant for each test (i.e. independent of Sc). This is borne out in the results presented so far: the effect of numerical diffusion is obfuscated by the dominant physical diffusion at low Schmidt numbers, but becomes far more apparent as Sc is increased. Figure 10.5 shows this variation

of D_{eff}, where the numerical diffusivity has been given a selection of arbitrary values, for Schmidt numbers in the range $0.7 \leq Sc \leq 1000$. The kinematic viscosity is that of the present configuration. Although D_{num} is undetermined, the observation that the smoothnesses of Cases ML4-EU and ML5-EU are similar (at the Schmidt numbers indicated by the dashed lines in Fig. 10.5) might suggest that its value is of the order $D_{\text{num}} \approx 10^{-6}$ m^2/s.

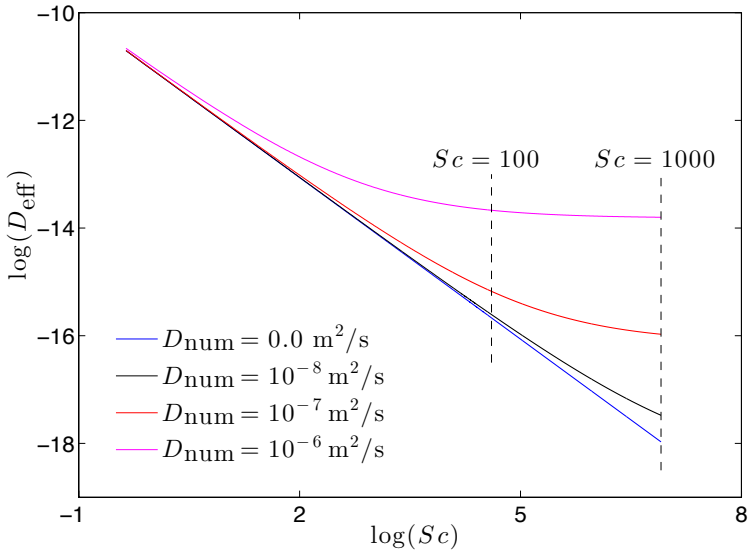


Figure 10.5: Effective diffusivity with varying Sc, for a selection of constant D_{num}.

Inspection of Fig. 10.11(b) again indicates that resolution error becomes significant at high Schmidt numbers, where the Batchelor scale is under-resolved by an order of magnitude. Regions are observed (e.g. at $(x, y) = [16, 5]$ mm) where the scalar has been stretched into a thin swirling filament around a vortical structure before being convected into a straining part of the flow, so that the filament has been compressed into an even thinner form. While it is likely that the Lagrangian particles in such regions will continue to represent these thin features accurately, the grid resolution is not sufficient to recover them.

At this stage it is instructive to analyse the Lagrangian field; such an analysis is performed for this case in particular, as an accurate description of the transported scalar field relies most heavily on the particle contribution. Figure 10.6 shows the Lagrangian component (positive, red; negative, blue) at $t = 12$ ms. Also included are arrows indicating the magnitude and direction of the flow-field, with one arrow for every 20 cells (i.e. 0.4 mm spacing), and contours of the scalar summation at $\phi = 0.1$ (dotted line), $\phi = 0.5$ (solid) and $\phi = 0.9$ (dashed). The particle distribution map shows that particle contributions are restricted to interfacial regions of the scalar field, as expected, while higher contributions are typically found where deformation of that field is greatest. This demonstrates that the particle addition algorithm continues to perform adequately, as the small, sharp features within these regions demand a larger number of particles to be correctly described.

Visualisation of the flow-field in this manner also shows how the scalar interface is stretched and compressed by the flow. Compression is most apparent along the plane between $(x, y) = [10, 6]$ mm and $(x, y) = [12, 4]$ mm, where fluid moving in a 'north-east' direction, due to the vortex centred

around $(x, y) = [7, 5]$ mm, impinges on fluid travelling in a 'south-west' direction. This creates a local stagnation plane, causing a sharpening of the scalar gradient as the two sides of the interface are pushed together, and therefore constitutes one of the key mechanisms within the flow for which particle addition becomes essential. The other important mechanism is the shearing motion inside the flow vortices (studied in two dimensions in Chapter 9), which will tend to stretch the scalar interface and increase the area over which particles are required.

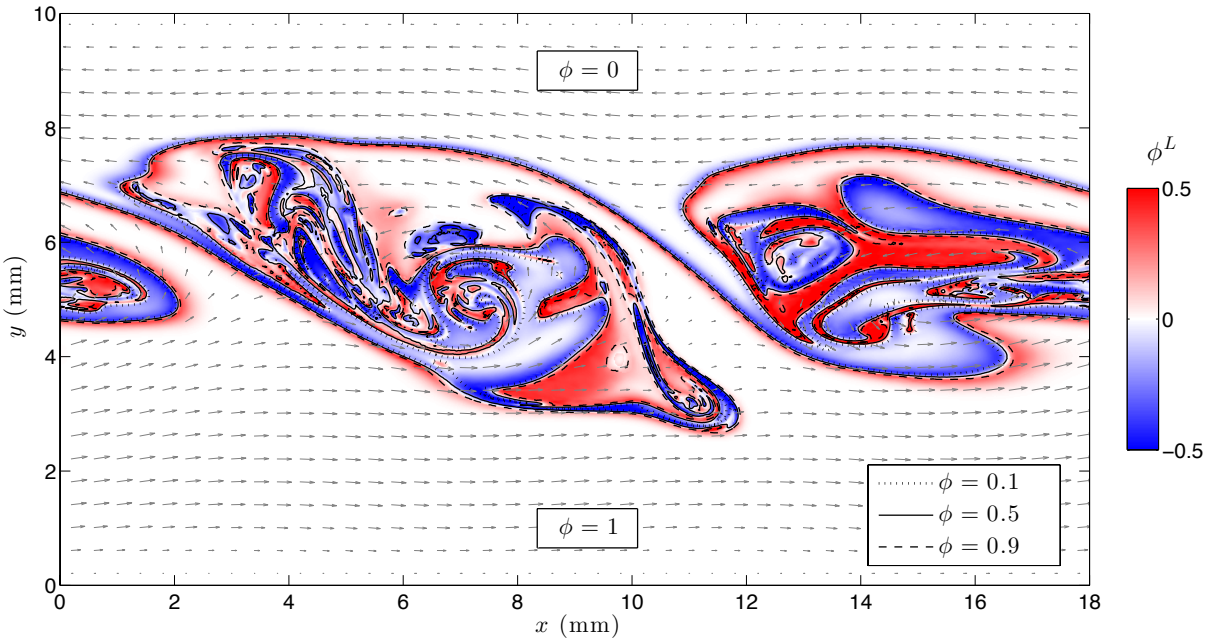


Figure 10.6: Case ML5-ELb: Lagrangian field at $t = 12$ ms, with flow-field and scalar contours.

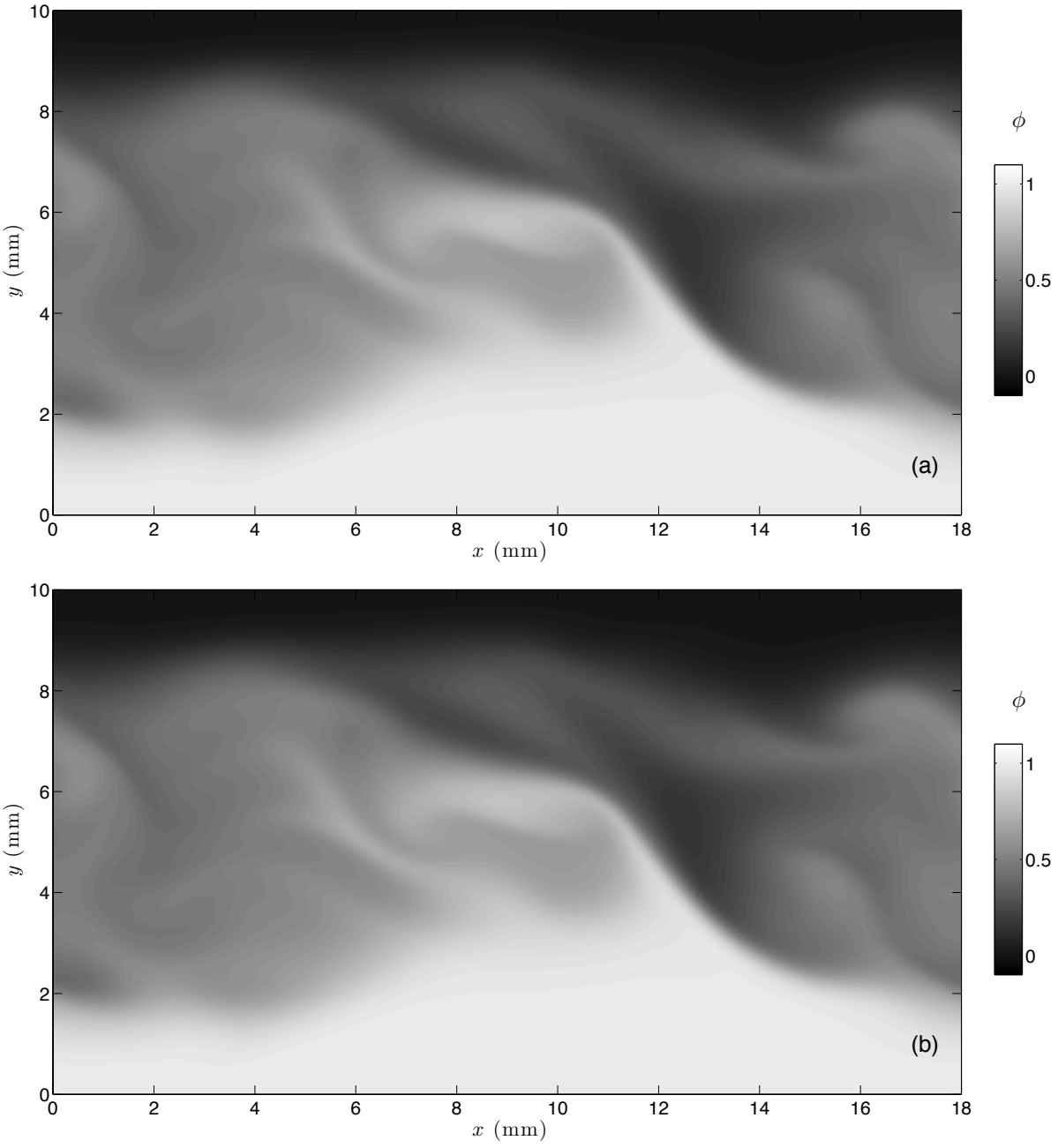


Figure 10.7: Case ML1: Scalar field at $t = 20$ ms, $Sc = 0.7$; (a) Eulerian, (b) E-L method.

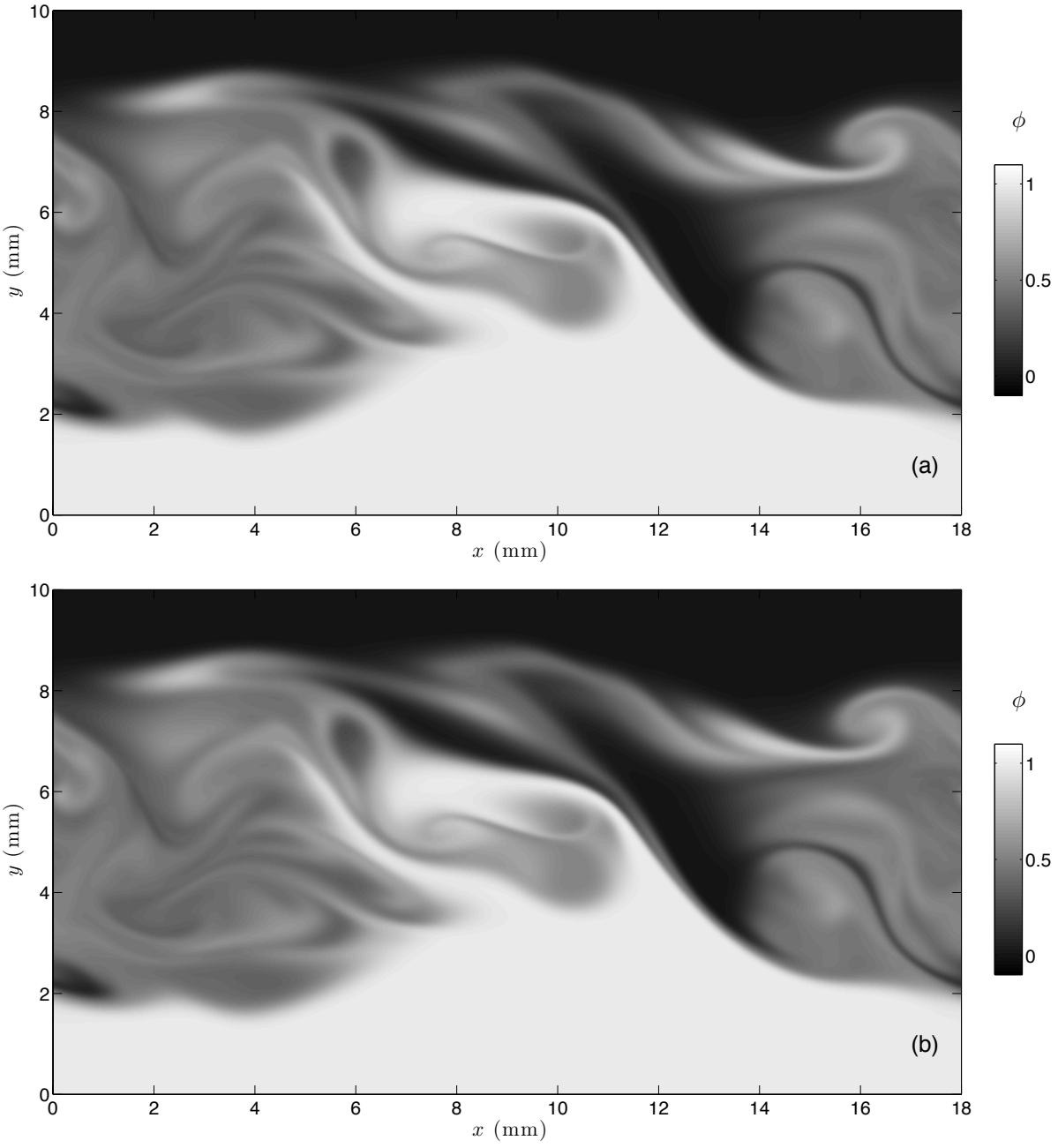


Figure 10.8: Case ML2: Scalar field at $t = 20$ ms, $Sc = 5$; (a) Eulerian, (b) E-L method.

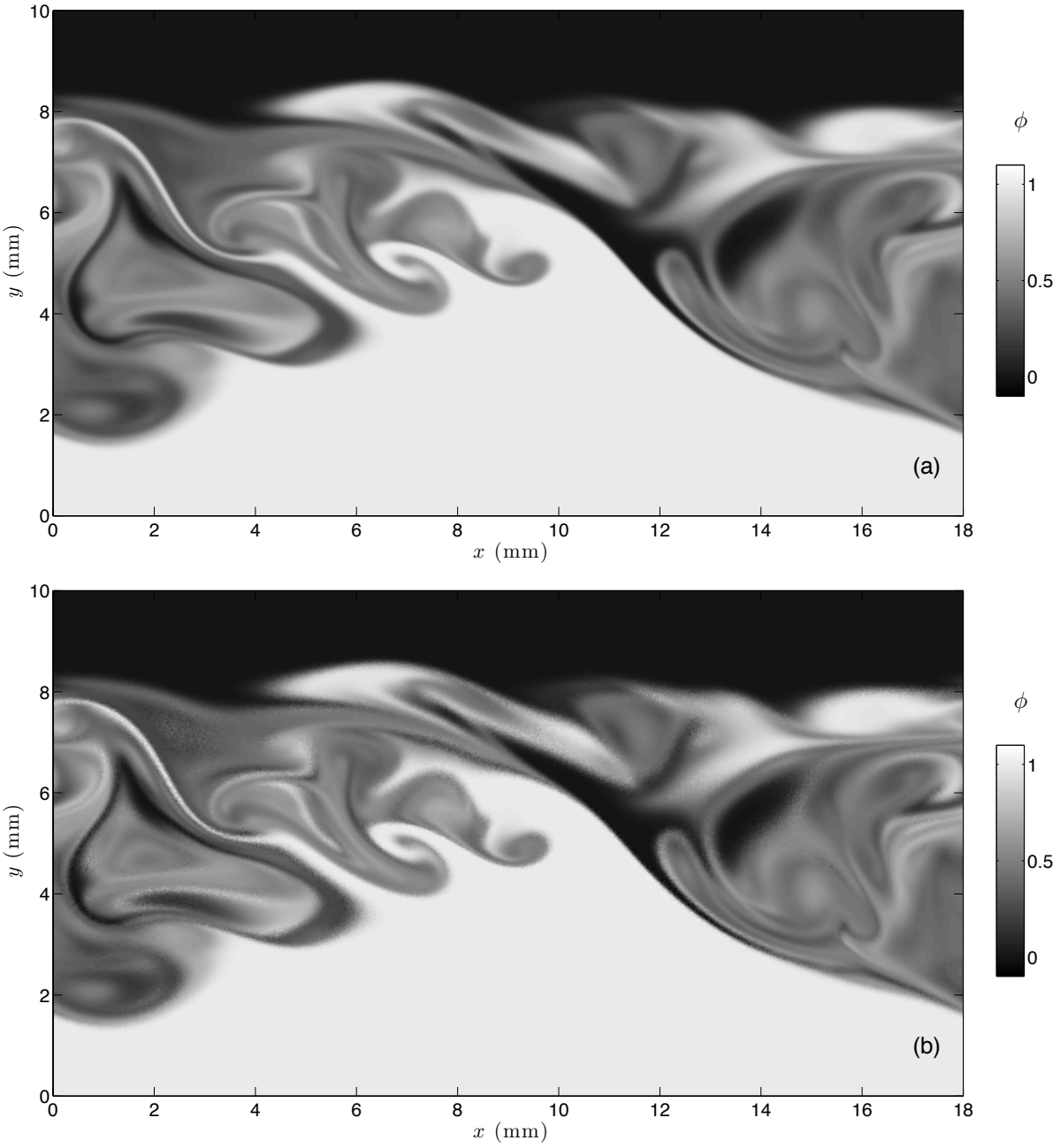


Figure 10.9: Case ML3: Scalar field at $t = 18$ ms, $Sc = 20$; (a) Eulerian, (b) E-L method.

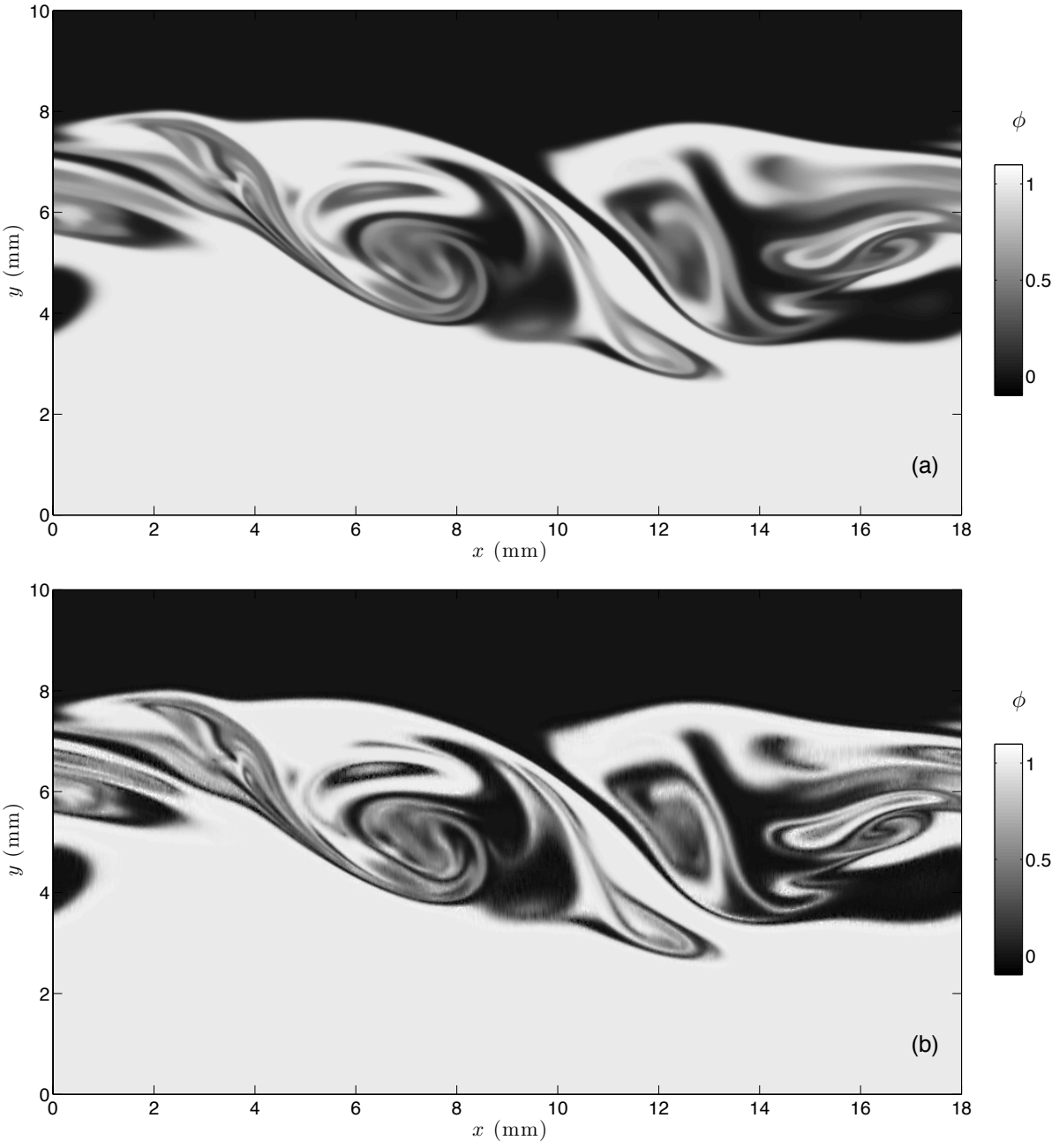


Figure 10.10: Case ML4: Scalar field at $t = 13$ ms, $Sc = 100$; (a) Eulerian, (b) E-L method.

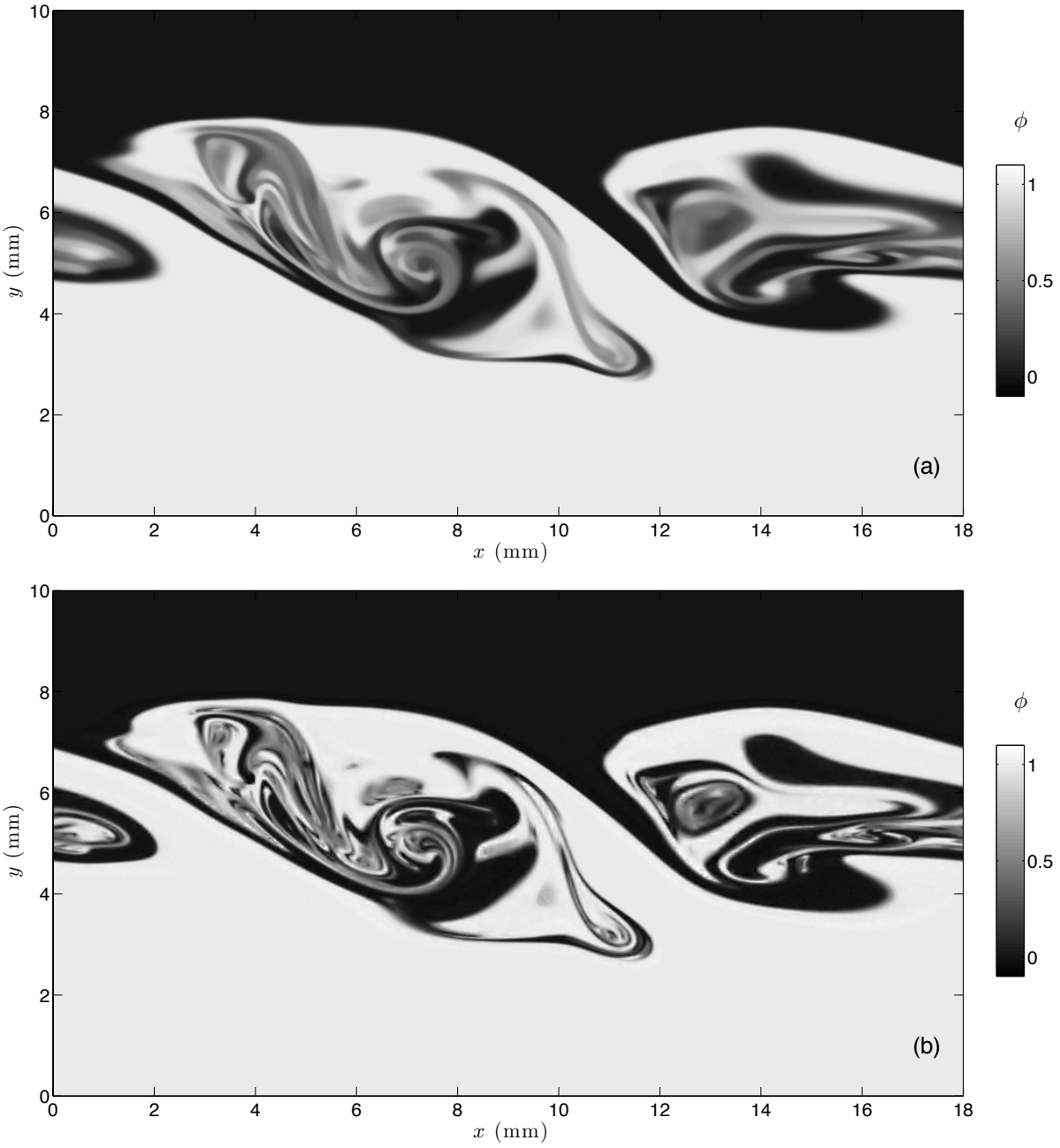


Figure 10.11: Case ML5: Scalar field at $t = 12$ ms, $Sc = 1000$; (a) Eulerian, (b) E-L method.

Effects of Particle Removal At this stage the effect of Lagrangian particle removal on the scalar topology is discussed, to demonstrate why results are presented without the mechanism for Cases ML4-EL and ML5-EL. In §8.4.1 it was found that particle removal may have an adverse effect on configurations at higher Schmidt numbers: the mechanism introduces noise from the Lagrangian reconstruction into the Eulerian field, and a degree of physical diffusion is required to prevent this noise from causing a distortion of the low-frequency phase. In this instance the opposite effect is observed, however, as the E-L method with particle removal implemented results in a smoother field than the corresponding simulation without the mechanism. This is illustrated in Figs. 10.12 and 10.13, which compare small regions of the vertical mid-plane slices through the solutions for Cases ML4 and ML5, respectively. Although the smoothness of the solution is not as pronounced as for an Eulerian simulation, the representation of the smallest scalar features within the regions shown is quite noticeably worse compared to the corresponding E-L solution with particle removal disabled.

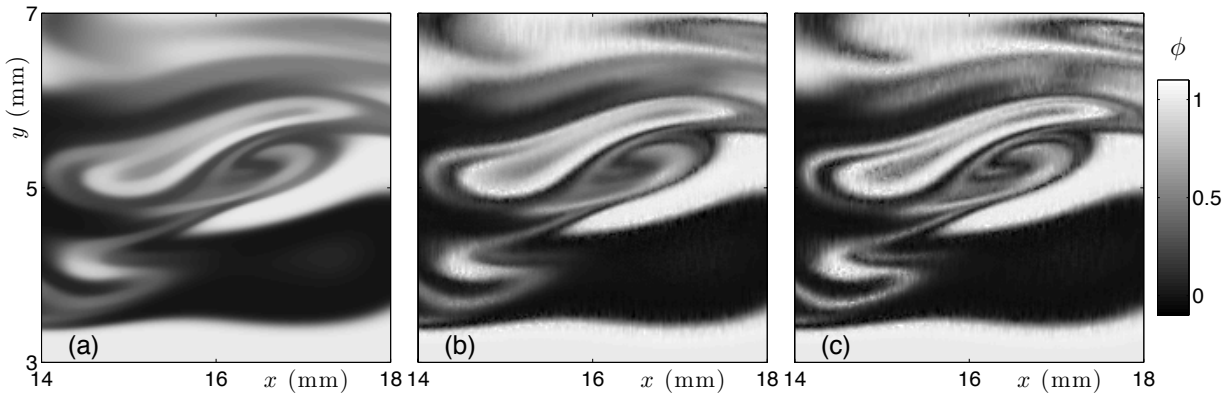


Figure 10.12: Case ML4: (a) Eulerian; (b) E-L method; (c) E-L without particle removal.

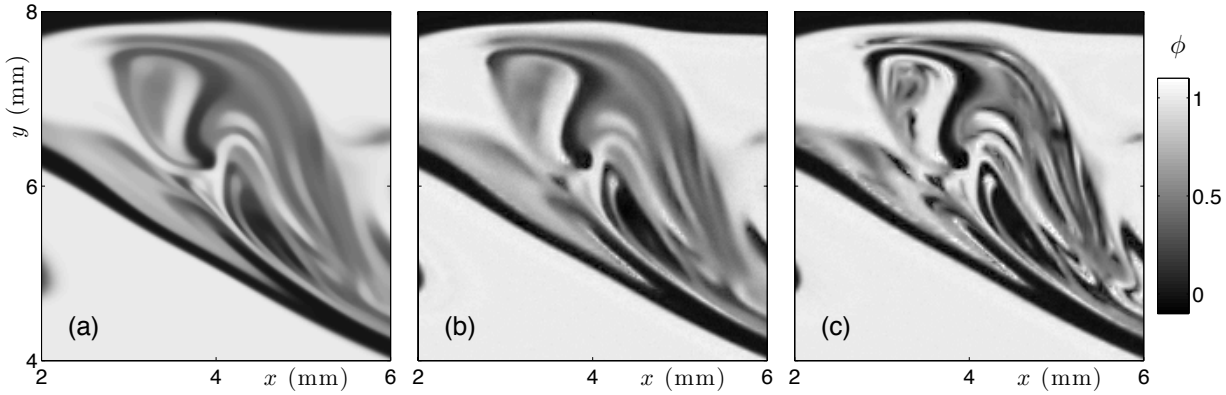


Figure 10.13: Case ML5: (a) Eulerian; (b) E-L method; (c) E-L without particle removal.

This smoothing may be due to the intrinsic filtering of the Lagrangian reconstruction previously described in §8.3, which is introduced into the solution when the scalar contribution to be transferred from the particles to the low-frequency phase is reconstructed on the Eulerian mesh. Although both

the quantity of transferred scalar and the amount of intrinsic filtering in each re-initialisation event will be small, the cumulative effect of this filtering may lead to the observed smoothing within the E-L summation. A solution to this problem may be to apply the deconvolution algorithm of §8.3.2 to the reconstructed scalar field to be transferred; however, this deconvolution will: a) amplify the noise in that field, and b) amplify the 'resolution error' in the field, as identified in §9.4.2. Both of these side-effects will increase the noisiness of the transferred information, and therefore also of the Eulerian phase, which will make that phase more susceptible to oscillatory behaviour.

A reduction in the smoothing introduced by the mechanism may also be achieved by altering the balance between particle addition and removal, which is controlled by the parameters γ, λ and β. Reducing λ and β will tend to make the removal mechanism less 'aggressive' in selecting particles for removal, although this will come at the cost of an increased total particle number, and therefore a more expensive simulation. Further analysis of the balance between computational cost and the number of particles will be presented in §10.5.3.

Effects of Grid Refinement To conclude these visual assessments the transported scalar field from the more highly-resolved Eulerian simulation of Case ML5, performed at $\Delta = 10$ μm, is considered. Figures 10.14(a–c) compare slices of the scalar at $t = 12$ ms through the lateral mid-plane of the domain (i.e. at $y = 5$ mm), for Cases ML5-EU, ML5-EUf and ML5-ELb, respectively. It should be noted that some minor topological differences are observed in the scalar field from the solution at the higher resolution; this will be due to, for example, the finite accuracy of the interpolation method used to produce the initial conditions for this simulation. However, the fact that the overall shape of each field is almost identical provides a strong indication that the Kolmogorov scale is well-resolved on both grids.

A comparison of these three fields illustrates how an increase in resolution reduces the effect of numerical diffusivity in an Eulerian solution, as the smallest scalar structures are captured in much more detail. This is apparent throughout almost the entirety of the field displayed here, but is most obvious within the highly convoluted region across the width of the domain between $x = 5$ mm and $x = 8$ mm. At the same time, however, it would seem that the E-L method continues to provide an improved description of the scalar field: although the features in the range $x = [5 - 8]$ mm are less diffuse in Fig. 10.14(b) than they are in Fig. 10.14(a), the edges of the scalar field still appear to be more clearly defined in Fig. 10.14(c).

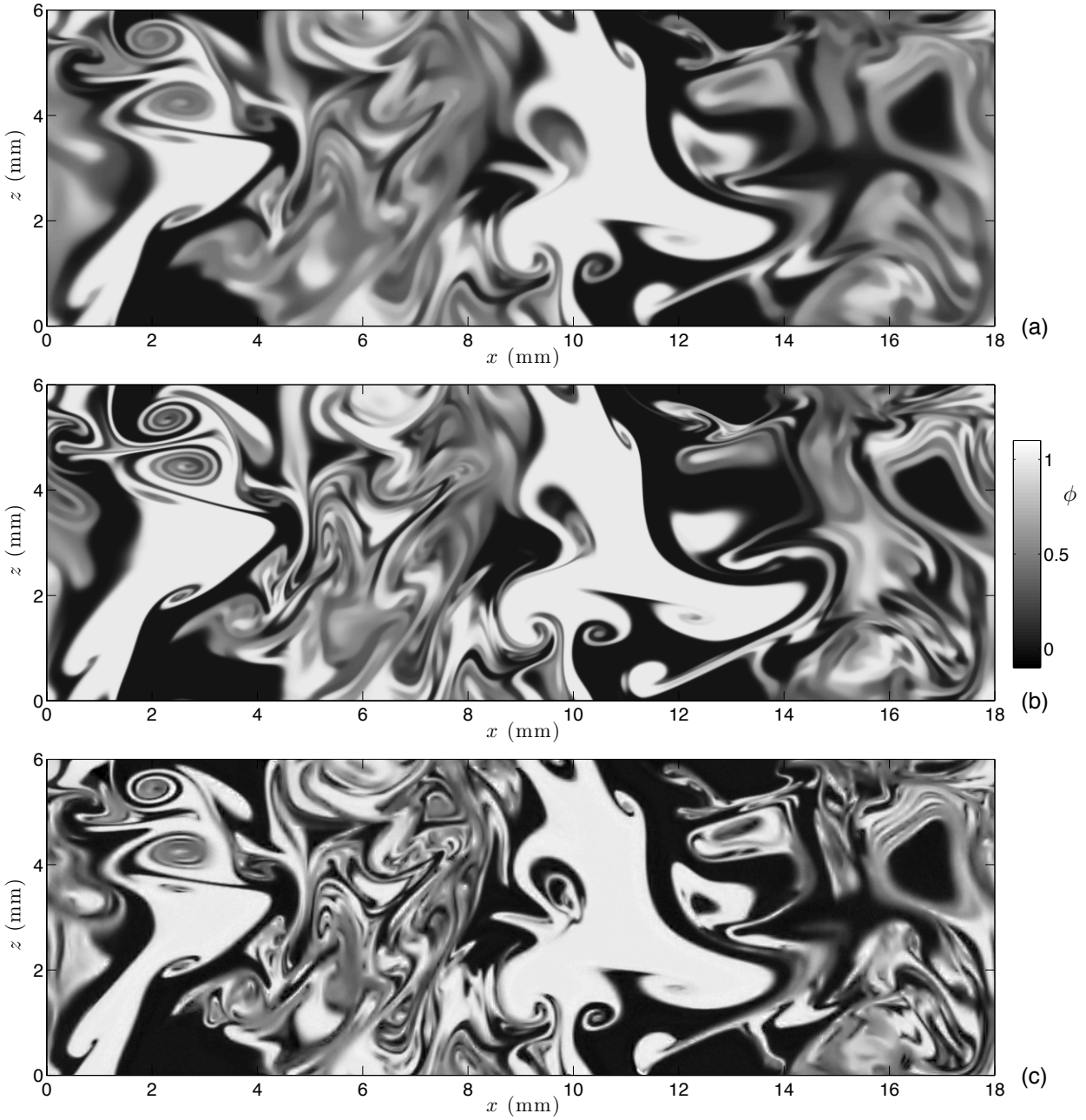


Figure 10.14: Case ML5: Scalar field at $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm.

10.5.2 Scalar Topology

The visual comparisons presented in §10.5.1 indicate the potential of the E-L method to improve the accuracy of scalar transport in high Schmidt number flows. However, they do not provide any quantitative analysis, and this forms the basis for the next stage of the investigation. A measure of the 'true' accuracy of the Eulerian and E-L methods tested here is difficult to formulate, as there is no analytical solution for comparison. By making some assumptions about how the scalar topology might vary in time, it is at least possible to make a fairly coarse comparison between the transported fields in each case and some expectation.

Analysis of the scalar topology is divided into three parts. Firstly, scalar gradients is considered; next, the thickness of the mixing layer is defined and measured; and finally, the instantaneous scalar dissipation rate is calculated.

Scalar Gradient The behaviour of the initial scalar field described in §10.2 is considered if it were to be transported by diffusion only. It is assumed that the physical diffusion process is Gaussian, as indicated by Einstein's solution (Eq. 5.26) to the diffusion equation (Eq. 5.25), and the three-dimensional field is described with a one-dimensional profile (as the initial scalar field varies only in the y-direction). This profile may be expressed analytically as a Gaussian cumulative distribution function varying in time:

$$\phi(x,\, t) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{4Dt}}\right)\right] \tag{10.9}$$

Differentiation of Eq. 10.9 recovers a Gaussian distribution describing the (absolute) gradient of the scalar field:

$$\left|\frac{d\phi}{dx}\right| = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-x^2}{4Dt}\right) \tag{10.10}$$

From Eq. 10.10, the maximum scalar gradient may be approximated at $x = 0$:

$$\left|\frac{d\phi}{dx}\right|_{\max} = \frac{1}{\sqrt{4\pi Dt}} \tag{10.11}$$

For a given diffusivity D, it is therefore possible to determine the maximum gradient of the one-dimensional profile at time t. Furthermore the evaluation of Eqs. 10.9 and 10.10 over some region $-r \leq x \leq r$, which adequately captures the whole scalar profile, allows a Probability Distribution Function (PDF) of scalar gradient conditioned on scalar value to be produced. The analytical PDF may then be compared to the distribution obtained from the solution at time t of a simulation with that diffusivity D.

Before this analytical distribution is generated, however, it is important to consider how the scalar gradient of a transported scalar field will be determined. In three dimensions the magnitude of the scalar gradient at the centre of a cell $(i,\, j,\, k)$ may be estimated (with second-order accuracy)

according to:

$$|\nabla\phi|_{(i,j,k)} \approx \left[\left(\frac{\phi(i+1,j,k) - \phi(i-1,j,k)}{2\Delta} \right)^2 + \left(\frac{\phi(i,j+1,k) - \phi(i,j-1,k)}{2\Delta} \right)^2 + \left(\frac{\phi(i,j,k+1) - \phi(i,j,k-1)}{2\Delta} \right)^2 \right]^{1/2} \quad (10.12)$$

The accuracy of this approximation may be improved by extending the computational stencil in each direction, using stencil weights determined from the algorithm of Fornberg [114]. However, any such approximation is based on the assumption that the scalar field is relatively smooth over the extent of the stencil. Applying Eq. 10.12 to the initial mixing layer scalar field will return a maximum gradient of $0.5/\Delta$, even though the gradient at the interface is known to be $1/\Delta$. (Higher-order expressions do improve the approximation, but an eighth-order scheme will still return a significant under-estimation of $\left|\nabla\phi\right| \approx 0.63/\Delta$ at the interface). The result of this is that the maximum scalar gradient of the one-dimensional profile determined from Eq. 10.11 is not comparable to any scalar gradient calculated from the actual transported field using Eq. 10.12. This may be accounted for, however, by evaluating Eq. 10.9 over a very high-resolution mesh – chosen to be three orders of magnitude finer than Δ – and then applying Eq. 10.12 to that discretised analytical solution (where the gradient at a given point x is determined from the analytical values at $x - \Delta$ and $x + \Delta$). This means that any inaccuracy in the gradient approximation due to Eq. 10.12 will at least be consistent between the analytical and actual scalar gradient PDFs. Evaluating Eq. 10.9 at a very high resolution will ensure that any discretisation error in the gradient approximation is negligible. Furthermore this approach has the added benefit of taking into account the ability of the finer grid, applied to Case ML5-EUf, to capture steeper scalar gradients.

Figure 10.15 shows the expected scalar gradient conditioned on scalar value for the five Schmidt numbers tested here, after $t = 20$ ms, at a grid resolution of $\Delta = 20$ μm. (Corresponding plots generated for $\Delta = 10$ μm would simply be twice as tall in the y-axis). The steepest gradient in each case occurs at $\phi = 0.5$, while gradients reach zero at $\phi = 0$ and $\phi = 1$. The maximum gradient increases with the Schmidt number, as indicated by Eq. 10.11, while the least diffusive case is almost two orders of magnitude steeper than the most diffusive.

It should be noted that any comparison between these analytical gradient profiles and the actual transported fields will not account for deformation due to the straining and shearing motions of the flow. For example, and as identified previously, Fig. 10.6 shows regions where scalar gradients will be steepened by impinging streams. This analysis of scalar gradients may therefore be suitable only for comparisons between the Eulerian and E-L results, while the information provided in Fig. 10.15 is indicative of the gradients that may be expected in each case. However, as the diffusivity of the scalar is reduced (i.e. as Sc approaches infinity), the scalar field will maintain its initial 'step-like' interface – even as the interfacial plane is deformed by the flow-field – such that the measured gradients should become more closely matched to the analytical PDF.

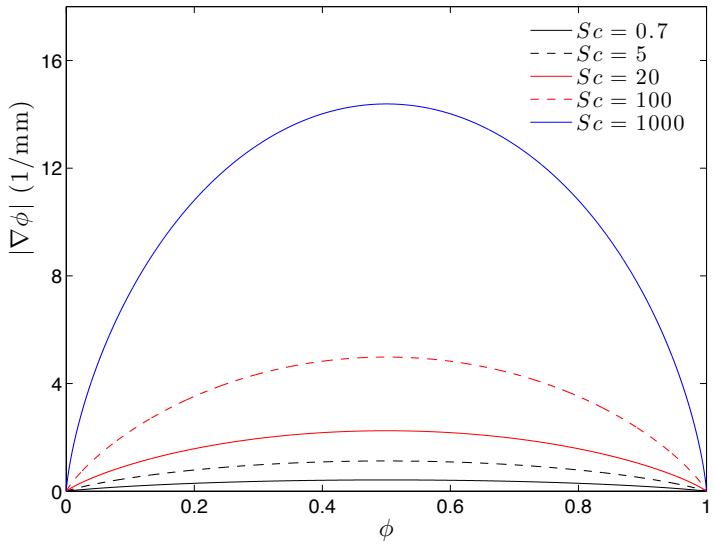


Figure 10.15: $|\nabla\phi|$ conditional on ϕ, for an analytical 1D profile with varying Sc, after $t = 20$ ms.

Figures 10.16 to 10.19 show PDFs of scalar gradient conditioned on scalar value for Cases ML2 to ML5, respectively, at $t = 12$ ms. (The analyses for Cases ML1 and ML2 prove to be very similar; the former is therefore omitted). These plots are generated by calculating the scalar gradient for each computational cell within the domain using Eq. 10.12, and retrieving the scalar value within that cell; these two quantities are then 'binned', with bin widths of $\Delta\phi = 0.001$ for scalar value and $\Delta\left|\nabla\phi\right| = 0.01$/mm for scalar gradient, and the summation over all cells is normalised by the total number of cells. The value at each point on the plot then represents the probability of a cell containing the scalar value and gradient corresponding to that point. The colour scales of the plots are selected to ensure that information in the range $0 < \phi < 1$ is visible, as there will be high probabilities of a cell having zero gradient at $\phi = 0$ or $\phi = 1$. Also plotted are the expected profiles based on Eq. 10.9, at $t = 12$ ms, at the corresponding diffusivity of each case.

Figure 10.16 compares profiles from the Eulerian and E-L solutions for Case ML2. Considering that the slices through the vertical mid-plane for these simulations (shown in Fig. 10.8) are very similar, it is perhaps unsurprising that the gradient profiles are also almost identical. Each plot is characterised by relatively high ($P > 5 \times 10^{-6}$) probability densities around $\phi = 0$ and $\phi = 1$, and relatively high densities in the region of $\phi = 0.5$ at zero or shallow ($|\nabla\phi| < 0.7$/mm) gradients. (Further analysis of the data reveals that 18% of cells have zero gradient and zero scalar value, while 15% have zero gradient and a unit value). It can be seen that much of the data in the plot falls inside the analytical PDF (denoted by the white dashed line). This results from the deformation of the scalar field: as the interface is 'folded' by a vortex, a region of low scalar value becomes surrounded by high scalar; diffusive fluxes then 'fill in' the gaps, resulting in these regions of zero or low gradient at scalar values of other than $\phi = 0$ and $\phi = 1$. However, there are also gradients outside of (above) the analytical trend: the maximum gradient is found to be $|\nabla\phi| = 8.8$/mm, compared to $|\nabla\phi| = 1.5$/mm from the analytical plot. This is indicative of the steepening of the scalar field in compressive regions of the flow.

Although these distributions exhibit a slight skewness towards scalar values below $\phi = 0.5$, this would seem to be purely coincidental; profiles generated for Case ML2 at times shortly before or after that shown are either symmetrical or skewed slightly in the opposite direction, and this is likely to be a result of the turbulent nature of the flow.

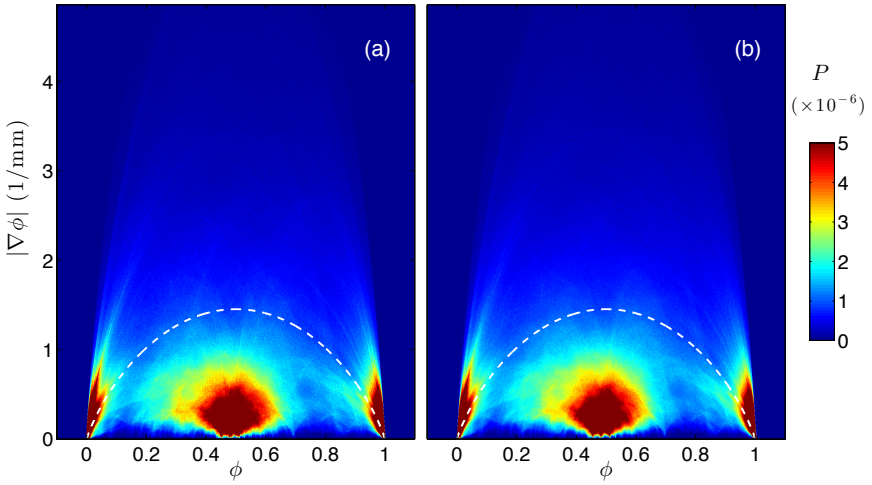


Figure 10.16: Case ML2: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method.

Figure 10.17 compares scalar gradient profiles for the Eulerian and E-L solutions to Case ML3 at $Sc = 20$. Once again the profiles are largely similar, reflecting the similarity of the images presented in Fig. 10.9, although the probability densities for shallow gradients ($|\nabla\phi| < 2.0$/mm) around $\phi = 0.5$ are noticeably lower for the E-L method. This suggests that scalar gradients tend to be steeper around $\phi = 0.5$, or that more of the cells have scalar values of $\phi = 0$ or $\phi = 1$.

A feature of the Eulerian solution PDF is the faintly 'banded' appearance of some parts of the distribution. These bands exhibit a parabolic shape similar to that of the analytical PDF, and it is suggested that they represent regions of the scalar field that vary smoothly from $\phi = 0$ to $\phi = 1$ (for example across the top edge of the mixing layer shown in Fig. 10.9(a), along the line of $y = 8$ mm). The one-dimensional scalar profile across such a region may therefore closely represent a Gaussian cumulative distribution. The bands become noticeable once regions of the flow possess very similar topologies, such that the PDF 'bins' into which those topologies fall become relatively more filled; this also explains why the bands become more obvious as they tend towards $\phi = 0$ and $\phi = 1$. In comparison these bands are absent from the E-L distribution, as Lagrangian noise in the scalar summation will disrupt the smoothness of the scalar field. (Note, however, that this banding is apparent in both the Eulerian and E-L profiles previously shown in Fig. 10.16, which provides a further indication of the lack of particle noise in the E-L solution to Case ML2).

For Case ML4 the disparity between the gradient distributions from the Eulerian and E-L solutions (without particle removal) becomes more pronounced, as shown in Fig. 10.18. (Note that the range of the colour scale for these images has been reduced, to maintain the clarity of the PDFs in both parts). The distribution for the E-L solution around $\phi = 0.5$ is generally over higher gradients: while the Eulerian PDF exhibits relatively large probabilities at zero or very shallow gradients

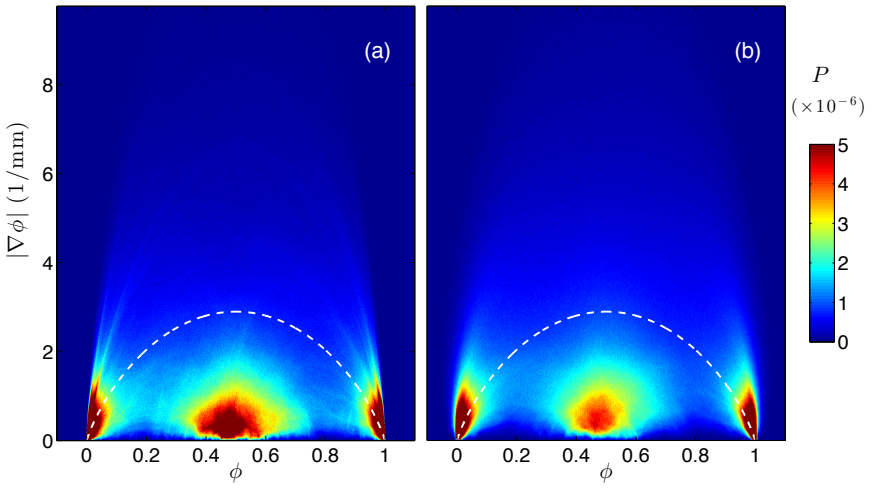


Figure 10.17: Case ML3: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method.

($|\nabla\phi| < 0.5$/mm) in this region, the E-L solution does not. The PDF from the E-L method also contains higher probability densities at higher gradients (above $|\nabla\phi| = 3.0$/mm) and more of the total probability density falls outside of the analytical PDF, as indicated by the lighter shade of blue for $|\nabla\phi| > 6.0$/mm. All of these factors suggest that the E-L solution generally contains steeper gradients, although it should be noted that the absence of zero gradients around $\phi = 0.5$ will partly be due to noise from the Lagrangian reconstruction at this scalar value.

Another feature of the Eulerian-Lagrangian PDF is the spread of the distribution around $\phi = 0$ and $\phi = 1$, where relatively high probabilities ($P > 2 \times 10^{-6}$) are found as low as $\phi = -0.05$ and as high as $\phi = 1.05$. This is indicative of the unboundedness due to noise in the Lagrangian reconstruction, even with low-pass filtering applied. Further evidence of the effect of noise on the E-L solution is provided by analysis of the data plotted in Fig. 10.18: 63% of cells in the Eulerian solution have zero gradient at $\phi = 0$ or $\phi = 1$, while in the E-L summation only 51% of cells do.

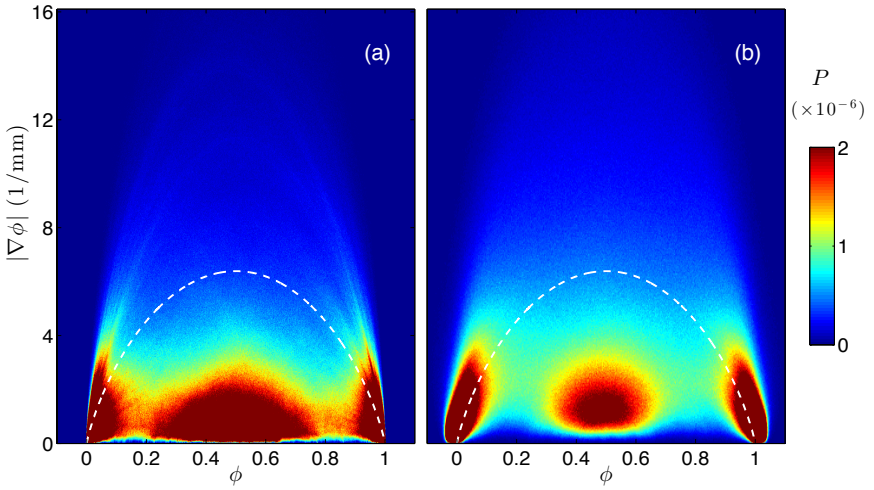


Figure 10.18: Case ML4: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, (b) E-L method.

Finally scalar gradient PDFs for the three solutions to Case ML5 are considered. The profile from the E-L simulation without particle removal is presented in Fig. 10.19(c), while the distribution from the more highly-resolved simulation (Case ML5-EUf, at $\Delta = 10$ μm) is included in Fig. 10.19(b).

The first observation to be made is that the PDF for Case ML5-EU, shown in Fig. 10.19(a), is remarkably similar to that from Case ML4-EU in Fig. 10.18(a). This supports the previous suggestion that the effective diffusivity D_{eff} approaches a constant value between $Sc = 100$ and $Sc = 1000$ (as illustrated in Fig. 10.5), indicating that the numerical diffusivity of the CHARM TVD scheme applied for these simulations is in the region of 10^{-6} m^2/s at $\Delta = 20$ μm.

The analysis of Fig. 10.19 demonstrates the capability of the E-L method to improve the description of scalar transport. Probabilities of relatively shallow gradients ($|\nabla\phi| < 3.0$/mm) around $\phi = 0.5$ are considerably lower than for the Eulerian solutions at both resolutions. (Some of the shallow gradients in the E-L gradient PDF will be as a result of errors in the description of the smallest scalar structures due to the under-resolution of the Batchelor scale). The contrast between the Eulerian and E-L distributions provides an indication of the ability of the E-L method to transport steep scalar gradients more accurately. The Eulerian simulation at $\Delta = 10$ μm suffers from less resolution error than its lower-resolution counterpart as the finer grid is more capable of capturing small structures, and this is indicated by the reduced probabilities around $\phi = 0.5$ in Fig. 10.19(b).

Figure 10.19(c) shows the correlation between the PDFs recovered from the E-L simulation and derived from the analytical profile. The 'ridges' of relatively high probabilities originating at $\phi = 0$ and $\phi = 1$ appear to follow the analytical PDF closely, suggesting that the small diffusive fluxes for the present configuration are accurately predicted by the E-L method. Apart from the higher probability densities at shallow gradients around $\phi = 0.5$, the data for Case ML5-EUf is similarly distributed; however, as indicated by the analytical PDF for this case (the whole extent of which is not shown, for clarity), the higher resolution should be able to capture gradients twice as steep. This suggests that the fine Eulerian simulation captures the steeper regions of the scalar topology about as well as the E-L method on the coarser grid.

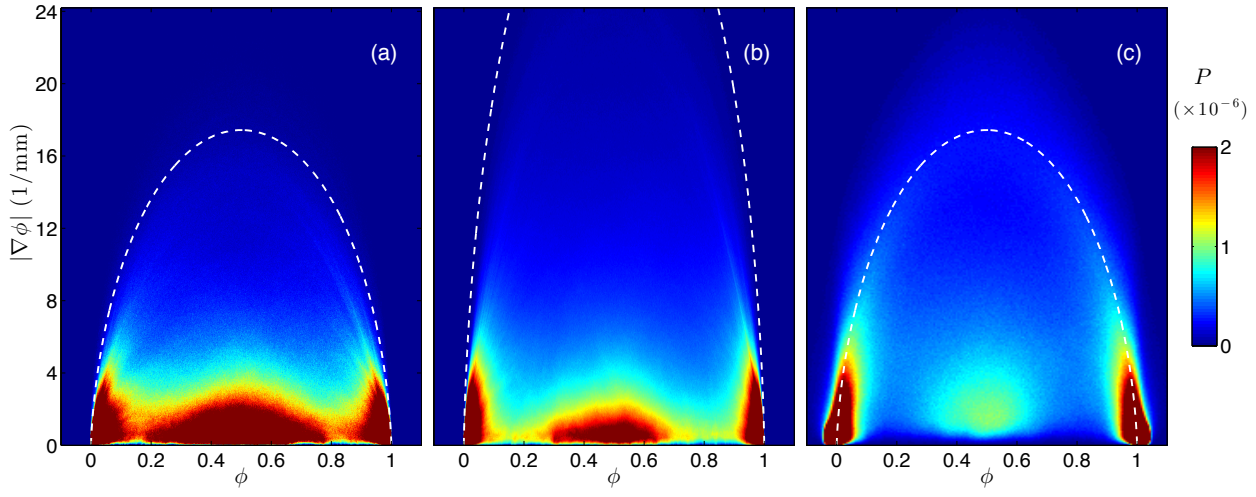


Figure 10.19: Case ML5: $|\nabla\phi|$ conditional on ϕ, after $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm.

Mixing Layer Thickness In the previous discussion the analysis has been informed by the application of Eq. 10.12 to an analytical solution for a diffused one-dimensional profile (Eq. 10.9). Equation 10.9 is now used directly to estimate the thickness of the mixing (or mass transfer) layer for each configuration at a given time. This is then compared to the corresponding mixing layer thicknesses measured from the Eulerian and E-L solutions for Cases ML1 to ML5.

The first requirement is to define that mixing layer thickness. A comparable quantity would be the momentum boundary layer thickness δ, which is often (but not always) defined in terms of the velocity thickness: for a wall-bounded flow with a free stream velocity u_∞, δ is equal to the distance y from the wall to $u(y) = 0.99\,u_\infty$. Sizaret et al. [239] modify the velocity thickness to define their own 'concentration boundary layer thickness' as $\delta_c = y|_{c\,=\,0.99\,c_\infty}$ (where c is the local concentration and c_∞ is the concentration in the free stream). However, a more convenient definition is suggested by Brown and Roshko [216], and used by Bernal and Roshko [217], who define the concentration thickness as the distance between points in the flow where the concentration deviates by 1% from the value in each free stream. In the context of the one-dimensional analytical profile, the mixing layer thickness δ_ϕ is therefore defined as:

$$\delta_\phi = x|_{\phi\,=\,0.01} \;-\; x|_{\phi\,=\,0.99} \tag{10.13}$$

For analysis of the simulation results it is necessary to calculate a comparable mean width for the mixing layer from an instantaneous scalar field. Ideally this would be determined as the average distance between the two iso-surfaces at $\phi = 0.01$ and $\phi = 0.99$, for a given time, over the entire three-dimensional domain. However, it is simpler to consider a two-dimensional approximation based on the slices presented in §10.5.1, where δ_ϕ is defined as the area between the contours at $\phi = 0.01$ and $\phi = 0.99$ divided by the length of the contour at $\phi = 0.5$. This area may be approximated as the number of cells in the slice with cell-centred scalar values in the range $0.01 \leq \phi \leq 0.99$, multiplied by Δ^2. (While it would also be straightforward to calculate the volume of the domain in the range $0.01 \leq \phi \leq 0.99$ in this manner, the difficulty in three dimensions comes from the need to determine, and calculate the area of, the $\phi = 0.5$ iso-surface; the analysis in two dimensions is much easier, and is expected to give similar results). Figure 10.20 illustrates this definition: the shaded area represents the region $0.01 \leq \phi \leq 0.99$, and the dotted line is the contour at $\phi = 0.5$. The example shown is for Case ML1-EU at $t = 12$ ms.

Based on the definition above mixing layer thicknesses are now determined from the analytical profile of Eq. 10.9, and from the solutions for Cases ML1 to ML5. The analysis is performed at $t = 12$ ms. Figure 10.21 shows the variation of δ_ϕ with Schmidt number, where both quantities are plotted on logarithmic scales.

From Fig. 10.21 it can be seen that the mixing layer thickness is proportional to the inverse root of the Schmidt number for the analytical case (i.e. the gradient of the analytical trend is $-1/2$). The linearity of this (logarithmic) trend is consistent with that shown in Fig. 10.5 for a configuration where the numerical diffusivity D_{num} is zero. At the same time the Eulerian trend shown here may be compared to any of the configurations with $D_{\text{num}} > 0$ in Fig. 10.5, and perhaps most closely matches the curve for $D_{\text{num}} = 10^{-6}$ m^2/s.

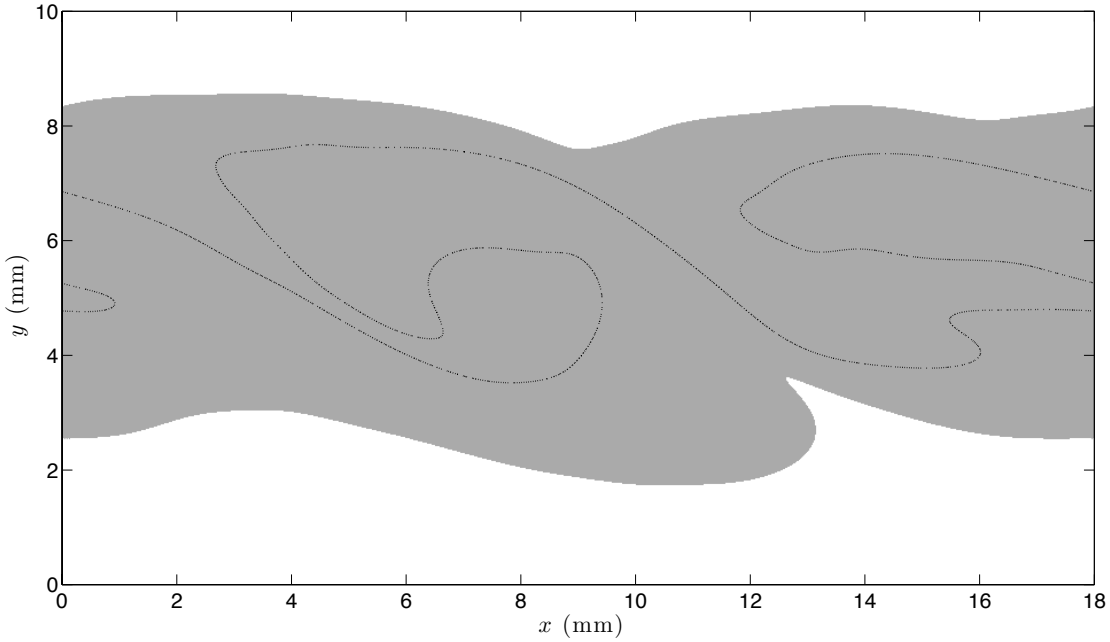


Figure 10.20: Mixing layer thickness: area between $\phi = 0.01$ and $\phi = 0.99$ (shaded) divided by length of $\phi = 0.5$ contour (dotted line).

For the Eulerian and E-L simulations we observe the same relationship between Sc and δ_ϕ over this first two points (representing Cases ML1 and ML2), at a narrower mixing layer thickness than in the analytical case (by around 40%) due to the compressive strain in some regions of the flow. Significantly, however, this linearity is lost from the Eulerian trend for $Sc \geq 20$, while the Eulerian-Lagrangian trend also becomes non-linear at $Sc = 100$ and $Sc = 1000$. (The dashed line in Fig. 10.21 indicates a linear fit through the first two points of the E-L trend, to illustrate this deviation). It is suggested that the non-linearity derives from a combination of two factors: the numerical diffusivity of the CHARM TVD scheme, which will only influence the Eulerian simulations; and the resolution error affecting Batchelor-scale scalar features, which will impact upon both Eulerian and E-L solutions.

Although only one data point is available for the Eulerian method at $\Delta = 10$ μm, its location in Fig. 10.21 is instructive: the fine grid reduces both the numerical diffusion and the resolution error of the solution, resulting in a narrower mixing layer compared to the Eulerian solution at $\Delta = 20$ μm, but the mixing layer thickness is still greater than that of the E-L simulation. This assessment correlates with that presented in two dimensions in Chapter 9, where the E-L method was found to out-perform the Eulerian approach over a range of grid sizes.

Scalar Dissipation Rate The final part of this analysis relates to the scalar dissipation rate, an important quantity that is often derived from the scalar field. The scalar dissipation rate finds application in, for example, turbulent combustion modelling, where Bilger [240] demonstrates that the instantaneous chemical reaction rate is directly proportional to the scalar dissipation rate (under the assumption of 'fast' chemistry conditions). As the name suggests, (turbulent) scalar dissipation

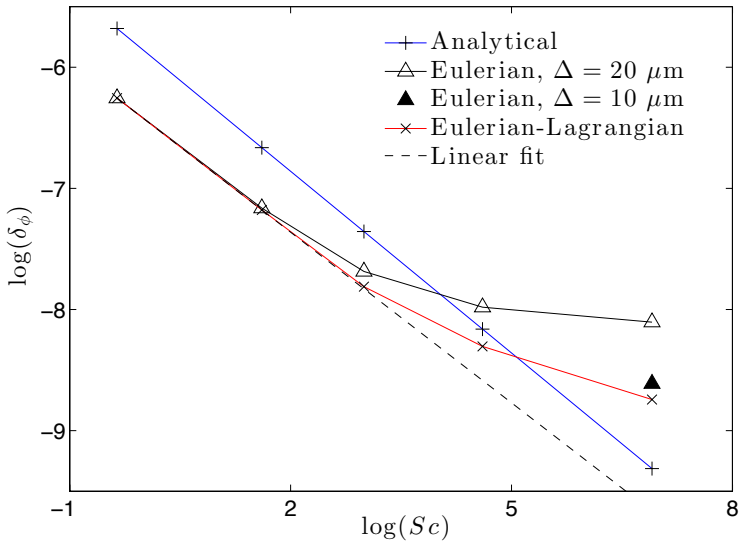


Figure 10.21: Mixing layer thickness with varying Sc at $t = 12$ ms.

is analogous to the (turbulent) dissipation rate of kinetic energy; this is discussed in detail in the review presented by Bilger [241]. Pitsch and Steiner [242] use Large-Eddy Simulation (LES) to investigate turbulent scalar dissipation in a diffusion flame, and discuss its use in many of the common (finite rate) chemistry modelling approaches (such as the flamelet model of Peters [243, 244], and the transported PDF method of Pope [22]).

The instantaneous scalar dissipation rate χ is calculated and compared for a selection of the Eulerian and E-L simulations that have been performed here, from the definition[2]:

$$\chi = 2D \left(\frac{\partial \phi}{\partial x_i} \right)^2 \tag{10.14}$$

The three components of the scalar gradient are approximated with second-order accuracy using an expression of the form in Eq. 8.4. It is also straightforward to apply Eq. 10.14 for the analytical profile described by Eq. 10.9, where the maximum scalar dissipation rate χ_{an} may be determined directly by substitution of Eq. 10.11 for the scalar gradient term. This results in the expression:

$$\chi_{\text{an}} = \frac{1}{2\pi t} \tag{10.15}$$

For the one-dimensional step case the maximum scalar dissipation rate is a function of time only; at $t = 0$ χ_{an} will be infinite, as the interface will be infinitely steep. However, values of χ_{an} found from Eq. 10.15 should not be compared too closely to the values calculated from simulations, as this simple analysis does not take into account the steepening of scalar gradients due to straining motions on the flow.

Figures 10.22(a–b) compare the instantaneous scalar dissipation rate for the Eulerian and

[2]Note that some authors provide a definition that does not include the multiplicative factor of two, such as that in [241]

E-L solutions to Case ML3, respectively, at $t = 20$ ms, through the lateral mid-plane slice (i.e. at $y = 5$ mm). The scale used for χ is logarithmic to ensure that a sufficient range of values are represented. For reference the scalar dissipation rate from Eq. 10.15 at $t = 20$ ms is 7.96/s, which is included in Fig. 10.22 as the red contour in each plot. Comparison of the two fields indicates that the magnitudes and distributions of χ are similar for each solution; however, the noise in the Lagrangian reconstruction has a significant effect on some regions of the flow, where scalar dissipation suffers from localised under- or over-prediction compared to its Eulerian counterpart.

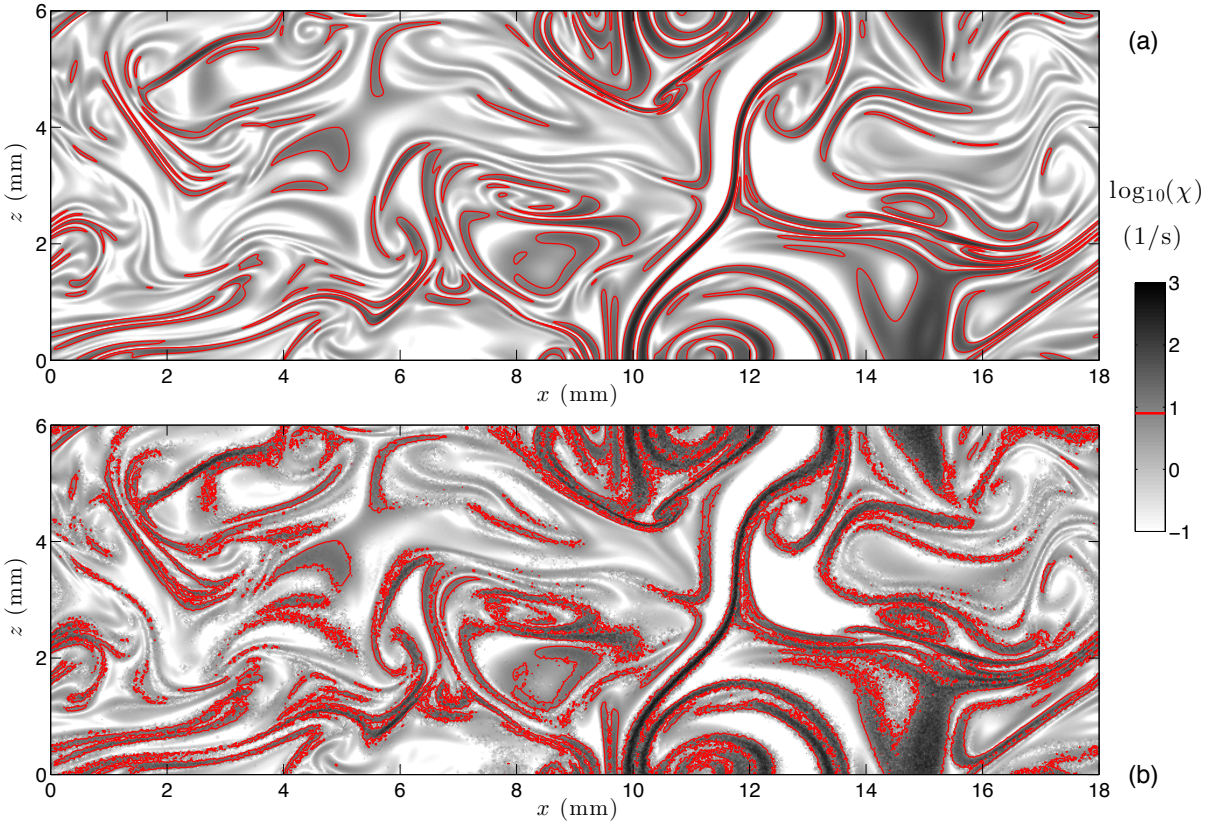


Figure 10.22: Case ML3: Scalar dissipation rate at $t = 20$ ms; (a) Eulerian, (b) E-L method.

The instantaneous values of χ from Case ML5 are presented for the Eulerian ($\Delta = 20$ μm), high resolution Eulerian ($\Delta = 10$ μm) and E-L solutions at $t = 12$ ms, through the lateral mid-plane ($y = 5$ mm), in Figs. 10.23(a–c) respectively. In general the values of χ are considerably lower for this configuration than for Case ML3 – by an order of magnitude, at least – which results from the linear dependence of χ on the diffusivity of the scalar. (Comparing Figs. 10.17 and 10.19, it can be seen that scalar gradients are typically less than an order of magnitude greater in Case ML5; considering Eq. 10.14, the squared gradient term is therefore less than two orders greater, while the diffusivity is reduced by two orders). However, the analytical value χ_{an} – again denoted by the red contour in each plot – is higher here ($\chi_{\text{an}} = 13.26$/s), as a shorter physical time has elapsed. Peak scalar dissipation rates from the Eulerian and E-L solutions along sharper regions of the scalar interface are approximately equal. The more highly-resolved Eulerian simulation shows a more widespread distribution of χ at and above χ_{an} than the Eulerian simulation at $\Delta = 20$ μm, due to

the ability of the finer grid to capture steeper gradients. The small degree of filtering intrinsic to the particle reconstruction in the E-L summation tends to reduce the peak scalar dissipation rate, so that it approximately matches that of the Eulerian simulation at $\Delta = 10$ μm.

A significant feature of these results is that values of χ inside the smaller scalar structures are lower in the Eulerian simulations, at both grid resolutions, than for the E-L simulation. Scalar dissipation rates differ by up to two orders of magnitude within many of the small features in the regions between $x = 5$ mm and $x = 8$ mm. This suggests that the E-L method provides a definite improvement at higher Schmidt numbers, which would be significant for any simulation in which the modelling of scalar mixing by its dissipation rate is important.

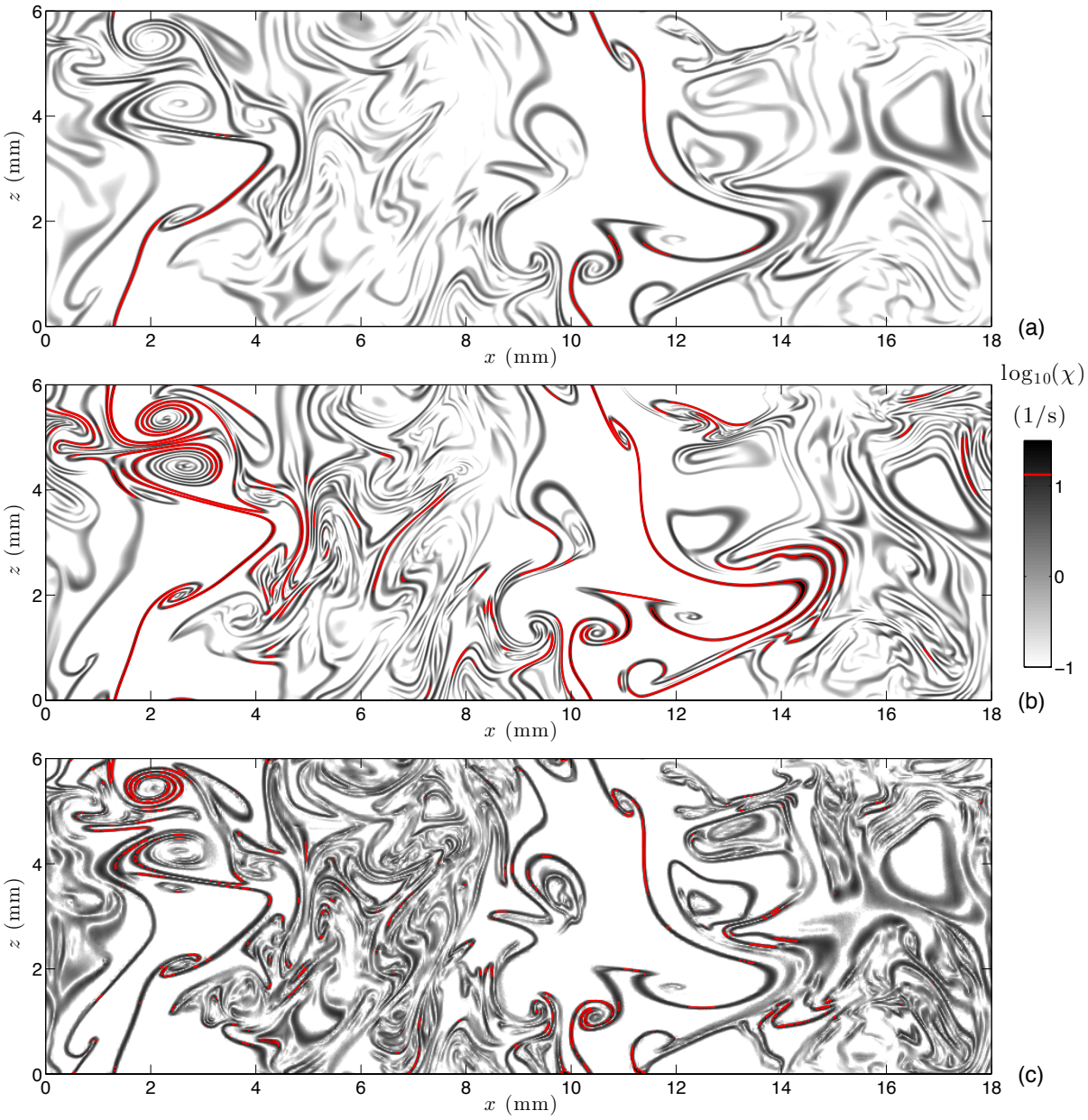


Figure 10.23: Case ML5: Scalar dissipation rate at $t = 12$ ms; (a) Eulerian, $\Delta = 20$ μm, (b) Eulerian, $\Delta = 10$ μm, (c) E-L, $\Delta = 20$ μm.

10.5.3 Computational Cost

The final stage of the mixing layer study comprises an investigation of the computational cost for each simulation, and, for the E-L method, an analysis of how the total particle number affects that cost. Table 10.2 summarises the cost of the cases that make up this study, in Central Processing Unit hours[3] (CPUh). Also shown is the simulated physical time for each case, where the high-resolution Eulerian simulation and the three most expensive E-L simulations were terminated before 20 ms had been simulated due to their increased expense. Costs are also, therefore, quoted in CPUh required per simulated millisecond, to permit a direct comparison between each case.

Table 10.2: Computational cost of the mixing layer simulations.

Case	Simulated Time (ms)	Cost (CPUh)	Cost/ms (CPUh)
ML1-EU	20.00	4414.60	220.73
ML2-EU	20.00	4425.47	221.27
ML3-EU	20.00	4422.03	221.10
ML4-EU	20.00	4421.42	221.07
ML5-EU	20.00	4448.21	222.41
ML5-EUf	12.11	103670.22	8557.32
ML1-EL	20.00	5220.22	261.01
ML2-EL	20.00	5450.39	272.52
ML3-EL	20.00	8306.37	415.32
ML4-EL	20.00	12163.09	608.15
ML5-EL	18.28	13824.00	756.19
ML4-ELb	13.50	13824.00	1023.68
ML5-ELb	12.18	13824.00	1134.98

A general point to note from the data in Table 10.2 is the significant cost of each simulation: Case ML1-EU required the equivalent of around 6 CPU-months, while ML5-EUf demanded 11.8 CPU-*years*. This highlights the importance of massively-parallel computing systems for the approaches of DNS and (highly-resolved) LES, as such simulations would be impossible on standard consumer hardware. Furthermore the Eulerian simulations used a total of around 40 Gigabytes (at $\Delta = 20$ μm) of RAM, while particle storage in the E-L simulations further increased this figure. For the Eulerian simulation at $\Delta = 10$ μm, no less than 320 Gigabytes of RAM were required.

Eulerian Simulations From Table 10.2 it can be seen that Eulerian simulations at $\Delta = 20$ μm are all of approximately equal cost. Even though the physical conditions in each case are quite different, the number of operations for the calculations of pressure correction, convection and diffusion will be the same. At the higher resolution the cost to simulate each millisecond of physical time is much greater, which is largely to be expected: doubling the resolution increases the number of cells by a factor of eight, and the number of time-steps required by a factor of two. However, rather than a 16× increase in cost, we observe more than twice that – around 39× – above the expense of a

[3]Simulations are performed on a parallel computing architecture, as described in §10.3; it is therefore more meaningful to consider the combined cost for all processing cores than just the execution 'wall time'

simulation at $\Delta = 20$ μm. Three main factors are identified as the cause of this additional increase, the first of which relates to the parallel efficiency E_m of the code. Some comments regarding the balance of communication and computation in a parallel code have already been made in §10.3, and these are expanded upon here. In the ideal case a simulation performed sequentially (i.e. on one process) in a time of t_s seconds will require $t_m = t_s/m$ seconds when performed on m processes, from which E_m is defined as:

$$E_m = \frac{t_s}{m\, t_m} \tag{10.16}$$

A code with perfect parallel efficiency therefore exhibits a performance of $E_m = 1$. However, real computational codes will typically operate at $E_m < 1$, due to, for example, the additional cost of communication between processes. Previous (unpublished) work suggests that the `PsiPhi` code has a parallel efficiency of $E_m > 0.6$ [245], although this lower bound was determined from a simulation with a particularly poor ratio of communication to computation. Based on the 'worst case' the computational expense of the larger simulation may therefore be expected to increase by up to 67%, although it should be noted that E_m will vary from architecture to architecture, and should be *considerably* higher than 0.6 under the present conditions[4].

The second factor concerns the choice of `FORTRAN` compiler used in each case. The 144-core simulations (at $\Delta = 20$ μm) were compiled with `pgf95 v10.4`, developed by the Portland Group; for the larger 480-core run, however, compiling with `pgf95` resulted in a fatal error on execution related to the initialisation of the MPI environment. Further investigation of this problem proved inconclusive, but the issue was solved by switching to the Intel Corporation `ifort v11.1` compiler. Compilation options in both cases were specified in the same way – using the third ('O3') level of optimisation – but the actual implementation of this option almost certainly differs between the two compilers. An in-depth analysis of compiler performance is beyond the scope of the present work, but for comparison, Case ML3-EU compiled with `ifort v11.1` ran at a cost of 227.27 CPUh/ms. Although the executable compiled with `ifort` is therefore slightly slower than that compiled with `pgf95`, the cost from $\Delta = 20$ μm to $\Delta = 10$ μm is still increased by a factor of 37.6 based on this value.

Thirdly, due to a run-time limit of 96 hours on the Cray XT6m used for these calculations, it was necessary to perform Case ML5-EUf over three separate executions of the code; the values presented in Table 10.2 represent a summation over all three runs. Binary 'restart' files were generated at the end of each run, from which the next was initialised. The (parallelised) writing and reading of such a quantity of data clearly takes some extra time, although only a few minutes were required for each operation. More significantly, however, it was found that the second and third code executions were considerably less efficient than the first, where the individual computational costs per millisecond for each run were 6139, 12909 and 11530 CPUh/ms, respectively. Considering that exactly the same executable was submitted in each instance, the disparity between these costs is hard to account for. (Inspection of the transported density and pressure correction fields shows no

[4] This analysis is rather simplified: parallel efficiency will depend on both the 'strong' and 'weak' scaling performance of the code, as well as the chosen domain decomposition, and a number of other factors

indication of the 'checker-boarding' effect mentioned previously, which might cause such an increase in cost, while the pressure correction algorithm continues to perform only the minimum prescribed number of iterations during each time-step). A possible source may be the computer architecture itself: although the executable file is the same for each run, the physical locations of the cores on which it is executed are probably not. Inter-process communication will typically be quicker where it is performed between cores on the same node (processor), as messages will be passed in memory; between processes on different nodes, however, communication will take place over interconnects, and will be slower. The speed of communication may also vary between processes in the same physical 'rack', and processes in different racks, and finally, data transfer rates may depend on the amount of communication required by other calculations being performed at the time, as interconnects between nodes and racks will have a finite bandwidth. The way in which processes were distributed, as well as the overall use of the machine by other users, may therefore have had a negative impact upon the latter code executions.

Eulerian-Lagrangian Simulations Joint Eulerian-Lagrangian runs show considerable variations in computational expense. To examine this Fig. 10.25 presents the change in total particle number with time, including particle addition and removal data.

From the information in Fig. 10.25 it can be seen that all five simulations are initialised with 25 million Lagrangian particles (as expected, given that the initial conditions are identical for each case). However, for Case ML1-EL – shown in Fig. 10.25(a) – the particle number drops rapidly, with a peak removal rate of 2.1×10^5 particles per re-initialisation event (i.e. every four time-steps) at $t = 1$ ms, and has reached a total of zero particles after around 2 ms. This corresponds with the observations made in §10.5.1, where the transported scalar at $t = 20$ ms appears to contain none of the noise associated with a reconstructed Lagrangian field. A small number of particle additions are also made within the first 2 ms due to the sharpening of gradients within the small scalar structures created by the initial turbulent field. The net result of this behaviour is that the computational cost of Case ML1-EL shows an increase of around 18% per simulated millisecond when compared to an Eulerian simulation. Given that the Eulerian and E-L solutions to Case ML1 are very nearly identical, this additional expense is perhaps unjustifiable for the present configuration.

The total number of particles also drops immediately from its initial value for Case ML2-EL, as shown in Fig. 10.25(b). A peak in particle addition due to the early formation of small-scale features is closely followed by a peak in removal rate at $t = 2$ ms, as the kinetic energy of the 'artificial' initial turbulence, which drives the creation of those features, rapidly dissipates. By $t = 5$ ms particle addition and removal are mostly balanced; beyond $t = 10$ ms a net removal rate results in a gradual decrease in the total number of particles, with only 2% of the initial particle number remaining at the end of the simulation. This matches the observations made in §10.5.1, where the transported scalar field was found to exhibit little or no noise from the particle reconstruction. The relatively low particle numbers result in a computational cost close to that of Case ML1-EL, where we now observe a 23% increase compared to the corresponding Eulerian simulation. The minimal difference between the two solutions to Case ML2 again suggests that the E-L method may not be beneficial for this configuration.

Particle addition and removal trends change quite dramatically for Case ML3-EL, as illustrated in Fig. 10.25(c), where the total particle number experiences a net increase at the start of the simulation. The scalar features generated by the initial turbulence are more persistent due to the lower diffusivity of this configuration, and the particles required for their description are therefore more likely to be retained. The number of particles continues to increase almost monotonically for the first 15 ms, as vortices within the flow-field generate sharpening scalar gradients. Towards the end of the simulation the particle removal mechanism begins to dominate, as the total particle number drops from 71 million at $t = 14.3$ ms to 45 million at $t = 20$ ms. It is likely that at this stage much of the scalar field that is still undergoing deformation is already populated with particles (such that no further particle addition is required, resulting in the observed drop in particle addition rate), while particles continue to be removed from smoother regions. The higher total particle number for this configuration precipitates a cost around 87% greater than that of an Eulerian simulation; however, at this Schmidt number the impact of numerical diffusion begins to become apparent (as observed from the mixing layer thicknesses of Fig. 10.21), suggesting that the extra computational resources required for the application of the E-L method may now be worthwhile.

Cases ML4-EL and ML5-EL exhibit similar behaviour when compared to one another, as shown in Figs. 10.25(d) and 10.25(e) respectively, and are therefore considered together. Looking firstly at simulations with particle removal included, the number of active particles is found to grow steadily over the course of each run until a plateau is reached at around $t = 15$ ms. During these first 15 ms the particle removal rate lags behind the particle addition rate by around 1 ms, and it is proposed that this is approximately equal to the time it would take for a fluid element to pass across a vortex (based on the velocities and length-scales within the flow). Consider an undisturbed region of the scalar interface that interacts with such a vortex: as that region is drawn in the change of gradient in the low-frequency component will become sharper, and particles will be added. The region will then pass through the vortex and out the other side, where it may become smooth due to divergence of the flow-field or (for these configurations, small) diffusive fluxes, whereupon particles will be removed again. While this may provide some explanation of the clear correlation between particle addition and removal rates, the effect would be hard to observe directly.

A fairly constant total particle number is maintained for the period beyond $t = 15$ ms in each of these simulations (with particle removal applied). For Case ML4-EL this total is around 190 million particles, while for ML5-EL the total rises to 285 million. The difference in the final particle number is a result of the difference in diffusivity between the two configurations, as small scalar features (requiring larger numbers of particles) will tend to be diffused away more rapidly in Case ML4. This also accounts for the larger particle addition (and, subsequently, removal) rates in ML5-EL, where in excess of 1.1×10^6 particles are added or removed at each re-initialisation event. In comparison, maximum particle addition or removal rates of around 6×10^5 per event are observed for Case ML4-EL. A cause of the plateau in total particle number for each case has already been suggested in the analysis of ML3-EL: after around 15 ms much of the scalar field that is continuing to undergo deformation will have been populated with Lagrangian particles, such that further addition in these regions becomes unnecessary. Overall, these simulations cost around three times more than their Eulerian counterparts (2.75× for ML4, and 3.42× for ML5). However, Figs. 10.12 and 10.13 have

already provided some indication that the particle removal mechanism is unsuitable for these high Schmidt number configurations.

A more significant cost impact is observed for Cases ML4-ELb and ML5-ELb in Table 10.2, where E-L simulations are performed without particle removal. Computational expense is increased by factors of 4.63 and 5.14 for these two configurations, respectively, which may be explained by the (dashed line) trends shown in Figs. 10.25(d) and 10.25(e). Rates of particle addition for the two simulations are relatively constant and are similar in magnitude, at between 1.5×10^5 and 2.5×10^5 new particles each time the low-frequency field is re-initialised. This is much lower than the addition rates observed for simulations including particle removal, where in the latter case it is suggested that a particle that is removed in one instance will often have to be replaced later. The total particle number climbs steadily over time without particle removal to balance it, reaching 457 million at $t = 13.5$ ms for Case ML4-ELb and 517 million at $t = 12.2$ ms for ML5-ELb. The additional cost of performing these simulations with the E-L method is considerable when compared to Eulerian simulations at the same resolution, but is justified by the perceived improvement in the description of the transported scalar field. Furthermore the results from the E-L method are still favourable in comparison with the Eulerian simulation at a higher resolution, as demonstrated in Figs. 10.14, 10.19 and 10.21, while the latter is an order of magnitude more computationally expensive.

Finally the additional computational expense of the Lagrangian particles is determined for this part of the study. This is achieved by determining the average total particle number in time for each simulation from the data presented in Fig. 10.25, and performing a linear fit between these averages and the corresponding computational costs. This fit is shown in Fig. 10.24, from which the additional cost is found to be 14.98 CPU-seconds per time-step per million particles. (The y-intercept of the fit is 229.1 CPUh/ms, which agrees well with the costs of the Eulerian simulations).

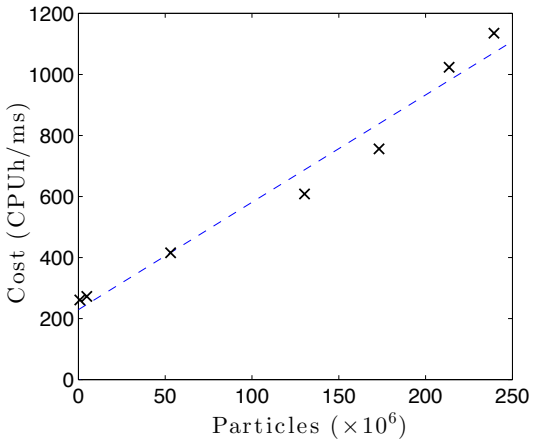


Figure 10.24: E-L simulations: Dependence of computational cost on particle number.

The particle cost in this configuration is therefore an order of magnitude greater than that of the modified Zalesak case in Chapter 7, which is mostly due to the increased dimensionality of the mixing layer test case. (The particle cost is greater still than the Zalesak and single vortex studies of Chapters 6 and 9, respectively, although these did not include particle diffusion terms).

The additional expense of the Lagrangian reconstruction routine in three dimensions has already been illustrated in Fig. 5.11; the explanation for this in §5.3.2 also applies to the interpolation of the velocity field for particle convection, where the particle velocity is determined from eight cell-centred values (in 3D) instead of four (in 2D). Note that no interpolation is required for particle diffusion terms in this case, because the density and viscosity fields are uniform; however, three Gaussian-distributed random numbers (generated by a Box-Muller transformation) are required for the components of the Wiener term used to describe particle diffusion, instead of two.

Performing a simulation in three dimensions also increases the expense of routines specific to the E-L method, which is most significant for the frequently-applied re-initialisation mechanisms of particle addition and removal. For the most part the E-L routines are expected to scale linearly with the spatial dimensionality d of the simulated domain (for a fixed number of cells and particles, although the Eulerian data set will scale with 2^d), as they operate independently in each direction. An exception to this – in addition to the reconstruction algorithm mentioned previously – is the particle removal routine, which also requires a reconstruction to create an Eulerian description of the transferred particle scalar information.

The other major factor affecting particle cost in this case is the fact that simulations are now being performed across parallel processes. This means that at every time-step it is necessary to: identify any particles that are within the 'ghost' region of a neighbouring process; copy those particles into a buffer array; communicate that buffer to the neighbour; and copy the information from the buffer into the main particle array. This requires the inclusion of several conditional statements in the code to determine whether a particle is a 'ghost' somewhere else, and to establish whether that particle has left the ('real') domain of the process it currently resides on.

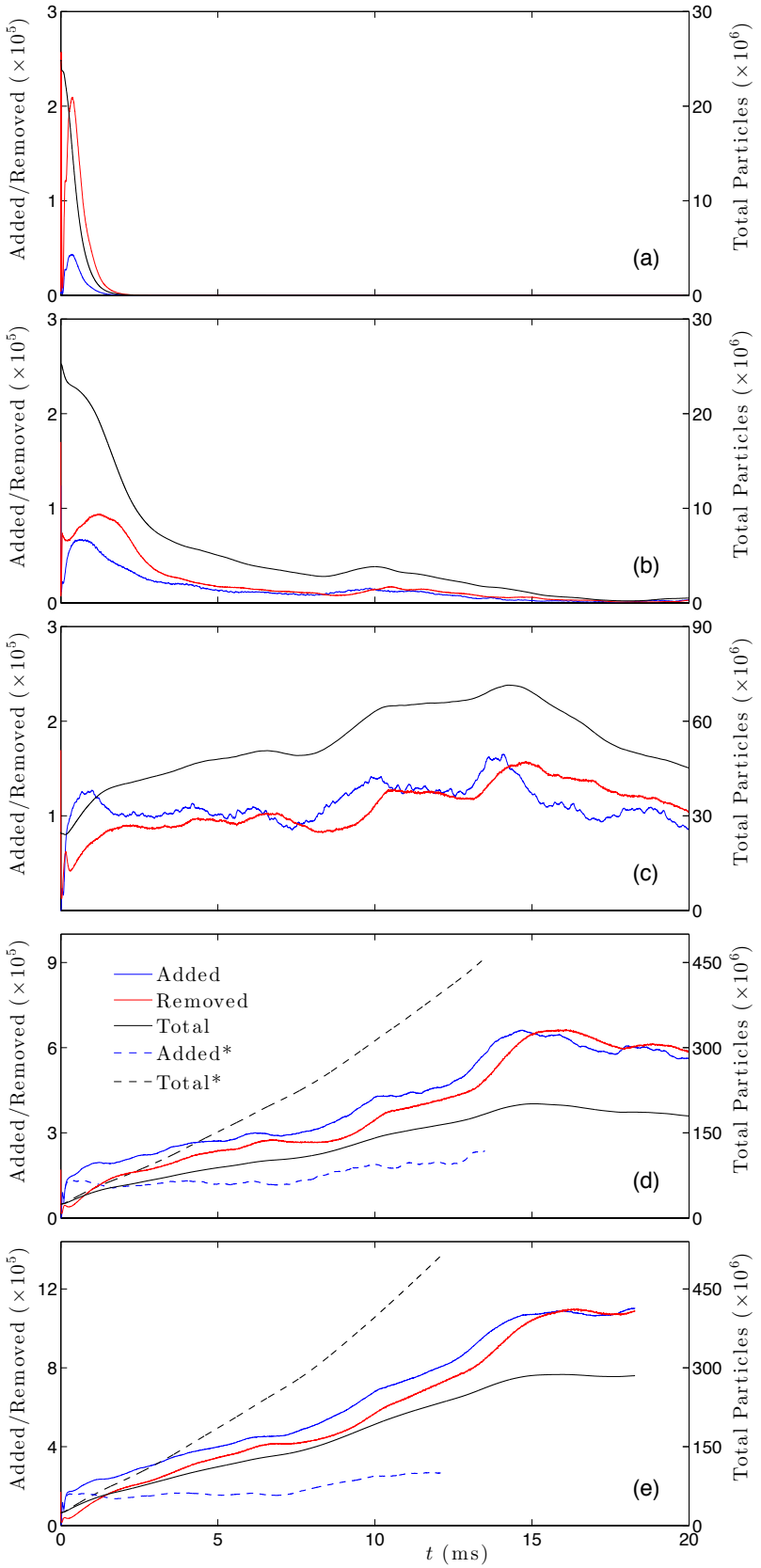


Figure 10.25: Total active; added; removed particle numbers, for Cases ML1-EL to ML5-EL, (a) to (e) respectively. *Without particle removal.

10.6 Conclusions

Chapter 10 has provided the first physically-representative test of the joint Eulerian-Lagrangian method, where a three-dimensional turbulent shear layer between two counter-flowing streams has been simulated. By restricting the physical size of the simulated region it has been possible to perform a Direct Numerical Simulation (DNS) of the flow, where it is estimated that the Kolmogorov scale has been over-resolved by a factor of two. The incompressible `PsiPhi` flow solver has been applied for the solution of the flow-field. The computational cost of this part of the study has been significant: the standard configuration ($\Delta = 20$ μm) has required 135 million computational cells, while a further, more refined simulation ($\Delta = 10$ μm) has used in excess of one billion cells. The most expensive simulation has utilised 11.8 CPU-years of computing time, while the overall cost is around 22.6 CPU-years.

A transported scalar has been used to represent the mixture fraction of an inert, passive fluid that is otherwise identical to the air with which it mixes. The mixing process has been studied at a variety of Schmidt numbers. Eulerian simulations applying a Total Variation Diminishing scheme with the CHARM flux limiter have been compared to E-L simulations. The E-L method has used eighth-order Central Differencing for the low-frequency phase and a particle density of $\mathrm{L}^{\varrho} = 20$ for the Lagrangian particles. Re-initialisation parameters have been prescribed with the values recommended in §5.4.4 for particle addition and §8.1 for the removal algorithm. The localised low-pass filtering algorithm has been applied with the parameters suggested in §9.4.4, which are modified from the original recommended values to account for the increased noisiness of the Lagrangian reconstruction at this low value of L^{ϱ}. Localised deconvolution, on the other hand, has not been used, due to the adverse performance of the algorithm at low particle densities demonstrated in §9.4.4.

Slices through the scalar field have been compared for five configurations of differing diffusivity, which has indicated that results from the Eulerian and E-L methods are largely similar for $Sc \leq 20$; however, significant differences become apparent at $Sc = 100$ and $Sc = 1000$. The numerical diffusivity of the Eulerian method causes small details in the scalar field to be lost. Increasing the resolution of the Eulerian simulation for the $Sc = 1000$ case reduces this numerical diffusion, but the E-L method appears to show an improved ability to capture smaller scalar features over the Eulerian simulations at both resolutions. On the other hand, noise from the Lagrangian reconstruction continues to be detrimental. The effect of particle removal on the E-L solutions has also been investigated and is found to be unsuitable for simulations at $Sc \geq 100$. Simulations with particle removal appear to deliver smoother scalar fields than simulations without the mechanism, and it is proposed that this is due to intrinsic filtering in the particle reconstruction that is required by the algorithm.

The topology of the scalar field has be studied closely, by: comparing probability density functions of scalar gradient conditional on scalar value; defining and measuring the mixing layer thickness; and calculating the scalar dissipation rate. An analytical solution for the case of a one-dimensional 'step' scalar profile undergoing diffusion only has been used for comparison. At any time $t > 0$, the scalar gradient profile exhibits a peak at $\phi = 0.5$ and zero gradients at $\phi = 0$ or $\phi = 1$. The magnitude of the peak gradient is a function of the diffusivity of the configuration and the length of

time that has passed. At low Schmidt numbers the gradient PDF is quite dissimilar to the analytical PDF, as the turbulent mixing of the scalar field by the underlying flow-field is not accounted for in the analytical solution; the PDFs from the Eulerian and E-L simulations are also practically identical. As the Schmidt number is increased, however, the PDFs become more distinct, while the E-L PDF tends towards the analytical profile at $Sc = 1000$. For this case the Eulerian solution at $\Delta = 10$ μm still contains traits consistent with numerical diffusion, such as low gradients around $\phi = 0.5$, and appears to be worse than the E-L solution.

The mixing layer thickness δ_ϕ has been defined as the mean distance between the contours at $\phi = 0.01$ and $\phi = 0.99$, and is found to be proportional to the inverse root of the Schmidt number for the analytical case. This relationship between δ_ϕ and Sc is also found for solutions to the low Schmidt number ($Sc \leq 5$) cases, but breaks down at $Sc > 5$ for the Eulerian method and $Sc > 20$ for the E-L method. It is proposed that the deviation from the inverse root trend is a result of numerical diffusion, which only affects the Eulerian simulations, and the under-resolution of the Batchelor scale by the computational grid, which impacts both Eulerian and E-L solutions. This has been explored further by calculating the mixing layer thickness for the more highly-resolved Eulerian case: δ_ϕ is found to be closer to the analytical trend as numerical diffusion and resolution error are reduced, but is still larger than the value of δ_ϕ determined from the E-L simulation at $Sc = 1000$.

Scalar dissipation rate χ is an important derived quantity in the context of, for example, combustion modelling, as it is proportional to the chemical reaction rate of two mixing species. The scalar dissipation rate is a function of the square of the scalar gradient, and particularly shows marked differences between Eulerian and E-L solutions at high Schmidt numbers. Noise in the Lagrangian component of an E-L summation is significant as it can cause localised over- and under-predictions of instantaneous scalar dissipation. However, for the configuration at $Sc = 1000$ the E-L solution has been found to deliver a more widespread distribution of χ within regions containing small scalar structures, where scalar dissipation rates may be under-predicted by two orders of magnitude in the Eulerian simulations at both grid resolutions.

Finally the computational costs of these simulations have been determined. The Eulerian simulations at $\Delta = 20$ μm consistently required around 220 CPUh per simulated millisecond, but this cost increased by almost 40 times when the cell size was reduced. Three reasons for this additional cost (above the expected factor of 16, due to the eight-fold increase in the number of cells and two-fold increase in the number of time-steps) have been suggested, which are the sub-optimal parallel efficiency of the code; a change in the compiler used to produce the executable in the more highly-resolved case; and variations in the distribution of processes and instantaneous loads on the machine used to perform the calculations. The computational cost of an E-L simulation is approximately proportional to the number of Lagrangian particles in that simulation, where it has been found that around 15 additional CPU-seconds are required per time-step per million particles. Higher Schmidt number simulations demand more Lagrangian particles, and the E-L calculations are therefore increasingly more expensive compared to the Eulerian simulations on the same 20 μm grid. However, this increased cost is justifiable for configurations at $Sc \geq 20$ – particularly when compared to the much larger cost of the more highly-resolved Eulerian simulation – as the E-L

method has been shown to improve the description of the scalar topology.

In this part of the investigation the joint Eulerian-Lagrangian method has been comprehensively studied using DNS, and has shown potential for the improved simulation of scalar mixing in high Schmidt number flows. However, the cost of these DNS calculations has been formidable, which has resulted from the necessity to apply a computational grid that resolves the Kolmogorov scale. Of more practical use is the Large-Eddy Simulation (LES) methodology, where this restriction on resolution is lifted; instead, LES uses a model to represent the smallest scales. The focus of Chapter 11, therefore, will be to demonstrate that the joint Eulerian-Lagrangian method continues to be compatible where the LES approach is applied.

11 Test Case 5: The Turbulent Opposed Jet

The final configuration to which the joint Eulerian-Lagrangian (E-L) method is applied is the laboratory-scale Turbulent Opposed Jet (TOJ) geometry developed by Coppola, Coriton and Gomez at the Yale Center for Combustion Studies [246, 247]. This apparatus has formed the basis of a collaboration between the author and the research group of Gomez, where a combined experimental (at Yale) and computational (with the `PsiPhi` flow solver) study of reactive flow has been performed [100]. While some elements of the present work will refer to that study, the simulations performed here will only consider a passive scalar in a non-reactive flow, as in previous chapters.

In Chapter 10 a grid resolution was selected that fully resolved the flow-field, down to the (estimated) Kolmogorov scale, such that no turbulence modelling was required. This allowed the E-L method to be analysed in the context of Direct Numerical Simulation. The duration and size of the domain in this study require that the Kolmogorov scale will *not* be resolved, necessitating the closure of the sub-grid velocity fluctuations by a modelled term (i.e. the code will perform a Large-Eddy Simulation, or LES).

11.1 Case Objectives

The performance of the E-L method has been thoroughly investigated throughout previous chapters, so that this final phase of the work mainly constitutes a validation of the method for use in an LES.

The flow-field from the mixing layer test case was characterised almost entirely by shear; the region of interest in the opposed jet configuration is dominated by strain, where two impinging jets create a stagnation plane upon which a scalar interface resides. The attraction of the TOJ configuration for a reactive case is that the reaction zones will also reside on (for a diffusion flame) or near to (for premixed combustion) this plane, which should be (approximately, i.e. statistically) stationary. The impinging jets will steepen scalar gradients on the stagnation plane, meaning that the particle addition routine in E-L simulations will play a vital role. A key feature of the TOJ case is that it will therefore provide additional insight into the balance of particle addition and removal.

Experimental data is readily available for velocity statistics (means and fluctuations) within some areas of the flow-field, allowing the accuracy of the physics in the `PsiPhi` code to be verified (although this is not the main aim of the exercise). Eulerian and E-L solutions will be compared to highlight any improvement in the representation of scalar transport and mixing processes. To facilitate the comparison, and to reduce the computational cost of the study, a precursor simulation of the entire TOJ geometry will be performed to generate transient inflow data at the boundaries of the region of interest. Repeated re-use of this inflow data will ensure that the same flow-field is experienced in each simulation, while the simulated volume will also be significantly reduced.

11.2 Case Description

The experimental TOJ geometry is intended to provide highly turbulent flow conditions representative of those found in practical applications, while providing good optical access for laser-based measurements such as Laser Doppler Anemometry (LDA) or Particle Image Velocimetry (PIV). The flow is well-defined and easy to reproduce, while the apparatus is designed to remove any dependence on boundary conditions far upstream of the region of interest. In addition the design is relatively compact, so that measurements and simulations may be performed at a reduced expense in comparison to more traditional axisymmetric jets.

A typical TOJ configuration consists of two vertical, opposed, co-axial nozzles, with a separation of the order of the nozzle exit diameter. Turbulence is generated inside each nozzle as the flow passes through a perforated turbulence-generating plate (TGP); near to this plate the flow will be characterised by intermittency between the jets and wakes issuing from the TGP, where velocity fluctuations will be high. The turbulence will be anisotropic and inhomogeneous, and the range of length-scales it contains will be far from representative of a fully-developed turbulent flow. As the fluid progresses downstream the turbulence will become more isotropic and developed, but the velocity fluctuations will also dissipate. Part of the challenge for the experimentalist, therefore, is to optimise the configuration so that it delivers a flow at the stagnation plane that is homogenous and isotropic, while still containing the maximum possible turbulent kinetic energy.

The TOJ configuration has been applied for many years for the examination of, for example, non-premixed (diffusion), partially-premixed, and premixed flames. Early experimental cases include the premixed studies of Kostiuk et al. [248, 249] and the partially-premixed flames of Mastorakos et al. [250]. More recently the research group of Lindstedt has been employing a method of fractal turbulence generation – inspired by the work of Vassilicos and co-workers [251, 252, 253] – to study 'flameless' oxidation, as reported by Geipel et al. [254] and Goh et al. [255]. Improvements to the design of the apparatus itself have been explored by Geyer et al. [256, 257], who have performed Raman-Rayleigh line measurements along the burner axis, and by Böhm et al. [258], who have made high-speed PIV flow-field measurements close to the TGP through a glass nozzle.

The focus for this case is the TOJ apparatus developed by Coppola and Gomez at the Yale Center for Combustion Studies [246], and subsequently measured for a reactive case by Coppola et al. [247]. Their design employs a very high (∼95%) blockage TGP, from which a jet issues into a relatively large plenum chamber within each nozzle. This creates a recirculation zone within the chamber, which contributes to jet destabilisation and break-up. The flow then enters a contraction and exits on to the stagnation plane. For reactive experiments an annular nitrogen shroud shields the stagnation plane and flame from the environment, dilutes any unburnt reactants, and prevents the recirculation of combustion products. A single nozzle of the Yale TOJ is shown in Fig. 11.1, where the exits of the opposing nozzles are separated by 12.7 mm (i.e. one nozzle diameter). Experiments are performed with the nozzles aligned vertically to mitigate the effect of buoyancy on measurements of variable density flow; the schematic shown shown here is rotated by 90°.

The TGP consists of 12 thin slits in a radial pattern, as shown in Fig. 11.1. The plate blockage results in a jet velocity from the TGP around 1.8 times that of the bulk flow velocity at the nozzle

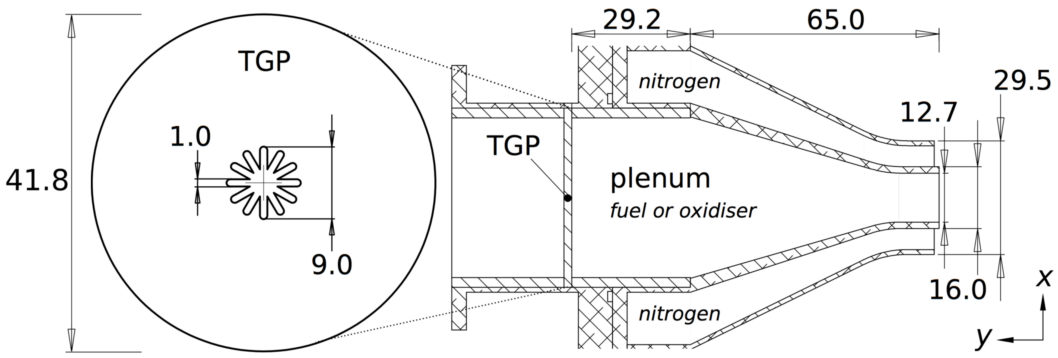


Figure 11.1: Schematic of a single nozzle. Dimensions in mm, not to scale.

exit. The TGP opening is designed to have a much larger perimeter than a traditional perforated TGP orifice, and very strong shear across this perimeter serves to rapidly destabilise the jet. Based on the nozzle exit diameter, the prescribed volumetric flow rate of 50 l/min through each nozzle (corresponding to a bulk flow velocity $U_b \approx 7$ m/s at the nozzle exit), and fluid properties of air at standard temperature and pressure, the Reynolds number for the present configuration is 5410.

Experimental measurements for this case are described in detail by Coppola et al. [247] and are only summarised here. Velocity means and fluctuations were determined by PIV, where the flow was seeded with $1-2$ μm olive oil droplets. A laser sheet of 0.6 mm thickness illuminated the region between the nozzle exits, and the droplet positions were captured by a charge-coupled device digital camera operating in double exposure mode. Image pairs were processed by an adaptive multi-pass fast Fourier transform cross-correlation algorithm using the TSI Inc. 'Insight 3G' software, yielding a velocity field with a resolution of 138×138 μm in the illuminated plane. Hot Wire Anemometry (HWA) was also used to record a time-series of velocity data 0.5 mm downstream of one nozzle exit, allowing turbulent (integral) time- and length-scales to be determined.

The simulations performed here recreate the isothermal experiments reported in [100] (Case OJ50), where both streams are cold air. To ensure that the simulated stagnation plane is statistically stationary, the velocity of the surrounding co-flow is set to 4 m/s (as for reactive experiments).

11.3 Simulation Numerics

A non-reactive simulation of the Yale TOJ has already been performed using the `PsiPhi` LES code, as reported in [100], and experience from this is drawn on for the re-computations performed here.

To simulate the region between the TGPs of the two opposing nozzles described in §11.2 a domain measuring 201 mm in length is required, where the nozzle axis is aligned with the y-direction. Based on the nozzle exit diameter, and to ensure that outflow boundary conditions have a negligible impact on the flow dynamics within the region of interest (i.e. on the stagnation plane close to the nozzle axis), the dimensions of the domain in the horizontal (x) and lateral (z) directions are chosen to be 50 mm; a slightly smaller domain size was used for the study presented in [100] to no adverse effect. The length of simulated time is chosen to be $t = 500$ ms, allowing time-averaged statistics to be gathered for quantities of interest. This duration has proved to deliver sufficient convergence

in the first and second statistical moments for the previous study [100]. Considering the size and duration of the simulation it would therefore be infeasible to perform calculations at resolutions near to those used for the mixing layer study (Chapter 10), as the computational expense would be three orders of magnitude greater for the present case.

The region of interest for the TOJ configuration constitutes a relatively small proportion of the overall volume of the domain. It is therefore more economical to divide the simulation campaign into two stages: the 'precursor' stage is concerned with the accurate prediction of the flow throughout the entire geometry, while the 'mixing layer' stage focuses on scalar transport processes within the region between the nozzle exits. The precursor simulation provides inflow boundary conditions for subsequent mixing layer simulations, where repeated re-use of this information mitigates the need to simulate the same flow-field (upstream of each nozzle exit) many times. Velocity and length-scale statistics from the precursor simulation may be compared to experimental results to ensure that key features of the flow, such as the location of jet break-up within each nozzle, are accurately captured.

11.3.1 Precursor Simulation

Following [100] a grid resolution of $\Delta = 0.5$ mm is used, requiring a total of 4.08 million cells over a domain measuring $50 \times 204 \times 50$ mm. (The domain length is extended slightly to ensure an even decomposition of the domain across parallel processes; this extra length is accounted for by including the thickness of each TGP in the domain). The simulation is performed on the same Cray XT6m architecture used in Chapter 10, where the domain is decomposed in three dimensions over 24 sub-domains. The processes are composed in a $2 \times 6 \times 2$ arrangement to yield the greatest efficiency in communication (i.e. such that each sub-domain is as close to a cube as possible, thus having the smallest surface across which communication must be performed).

The transport of a passive scalar quantity is not of interest for the precursor simulation, as a solution for the velocity field only is required. The three momentum components and the fluid density are therefore the only transported scalars. Convective fluxes are approximated using an eighth-order Central Differencing Scheme (CDS8) for momentum, and a CHARM Total Variation Diminishing (TVD) scheme for density. (In Chapter 10, CDS8 was also used for density; in this case, however, this leads to instabilities around sharp corners in the geometry). Diffusion of momentum is estimated with an eighth-order central approximation for the gradient at a cell face.

The relatively complex geometry of this case is described using the immersed boundary method, where cells inside a wall are set to have zero (cell-centred) momentum, density, pressure and (cell-faced) velocity normal to the wall. The use of CDS8 for momentum requires the inclusion of a 'flux blending' approach, where approximations of fluxes are made with reduced accuracy (i.e. smaller computational stencils) in near-wall regions. Directly adjacent to the wall CDS is replaced with a first-order Upwind Differencing Scheme. The description of the physical geometry itself is not ideal, as angled or curved surfaces cannot be accurately represented by cubic cells; however, the walls are sufficiently removed from the region of interest (i.e. the stagnation plane) that this does not impact significantly on the solution [100]. The TGP opening suffers from a similarly poor description, where each slit is only two cells across. Accordingly it is necessary to fine-tune the inlet velocity to ensure that the volumetric flow rate prescribed in the experiments is recovered, as the cross-sectional area

of the simulated opening is inaccurate.

The CFL criterion for the present case is set to 0.25, which has been found to provide a reliable numerical stability for this configuration; higher CFL numbers tend to result in instabilities in the transported density field within the nozzles. Based on this CFL number, a cell size of $\Delta = 0.5$ mm, and a jet velocity of around 22 m/s (to recover a flow rate of 50 l/min at the nozzle exit), a time-step width of $dt \approx 5.5 \times 10^{-6}$ s is expected. This corresponds to a total of around 88000 time-steps for a simulation of 500 ms. Advancement in time is performed using the same low-storage, third-order accurate Runge-Kutta scheme applied in previous chapters and described in §3.2. The CFL number is also 'ramped up' over the first 100 time-steps, as in Chapter 10.

The simulation of the Yale TOJ does not employ periodicity in any direction. This therefore requires the prescription of inflow and outflow boundary conditions at each face of the domain, in addition to the initial turbulent momentum field within. Artificial turbulence (both for the initial flow-field and at the inflow boundaries) is generated using the digital filtering procedure described in §3.7.1. The integral length-scale is set to be equal to the TGP slit width ($L = 1$ mm) while the turbulence is scaled to have fluctuations of $u_i' = 1.5$ m/s. This latter value is chosen from a parameter study, and is found to give a satisfactory agreement between the velocity statistics from the simulation and experiments. Outflow conditions are prescribed according to §3.7.2, where the entrainment of fluid into the domain is avoided by enforcing a positive momentum component in the direction normal to the outflow plane. A von Neumann condition is prescribed for the pressure field at both inflows and outflows.

The Kolmogorov scale for the Yale TOJ configuration is estimated to be $\eta_K \approx 0.032$ mm from experiments [100], although this value is based on measurements at the nozzle exit; it is likely that even smaller scales exist within the nozzles themselves. The current grid resolution therefore under-resolves the Kolmogorov scale by at least an order of magnitude, requiring the inclusion of a model to account for the effect of the smallest eddies. In this case the Germano model [79] is applied, using the extended version of Piomelli and Liu [80] as described in §3.5.2. The test filter width is set to $\widehat{\Delta} = 2\Delta$, while the local computed values of C are spatially filtered over $\widehat{\Delta}$ and are limited to the range $0 < C < 0.173^2$. The lower bound is set to prevent the backscatter of turbulent energy. The upper limit prevents instabilities due to spurious high values of C, and is based on the value of C_s for the (static) Smagorinsky model recommended by Lilly [259]; it should be noted, however, that the calculated values of C are typically below this limit.

A simulation duration of $t = 500$ ms is chosen as this yields convergence in the first (mean) and second (variance) moments of velocity statistics. This duration includes a period of 100 ms at the start of the simulation, equivalent to around six nozzle 'flow-through' times, to allow the flow-field to develop before samples for statistics are collected. Samples are subsequently gathered every 0.4 ms, corresponding to the integral time-scale at the nozzle exit estimated from experiments, to ensure that each sample is statistically significant. Statistics are therefore based on a total of 1000 samples by the end of the simulation. For comparison with experimental measurements simulation results are considered along the vertical axis between the nozzle exits, as well as across the nozzle exit diameter (0.5 mm downstream). For the latter the number of (simulated) samples is doubled by combining statistics along the horizontal and lateral directions, based on the observed axis-symmetry of the

measured and simulated flow in [100].

Inflow information for subsequent E-L simulations is gathered at the end of every computational time-step beyond 100 ms. To limit the storage requirement of this large quantity of data – consisting of three velocity components across two nozzle exit planes, for at least 70000 steps – values are only stored over a 13×13 mm region encompassing the area of a nozzle exit. Consistent with the prescribed co-flow, velocities outside of this region are assumed to be equal to 4 m/s in later simulations. Given a grid resolution of $\Delta = 0.5$ mm each inflow data file will therefore contain an array of 26×26 values, plus one additional value to record the physical time to which that data file corresponds. It is then straightforward to interpolate between these data points, in space or in time, as required.

11.3.2 Mixing Layer Simulations

For the latter stage of the simulation campaign the transport of an inert scalar quantity is added, and the domain is reduced to a $50 \times 13 \times 51$ mm volume between the nozzle exits. The resolution of the computational grid is increased, with a cell size of $\Delta = 0.25$ mm resulting in a total of 2.12 million cells. The domain is decomposed in three dimensions over 24 sub-domains, as before. In this case the sub-domains are configured in a $4 \times 1 \times 6$ arrangement, necessitating the extension of the domain in the z-direction to ensure an even division of cells over these processes.

Transport of the three components of momentum is performed using CDS8 for convection and an eighth-order central approximation for diffusion, as before, while density is again convected by CHARM TVD. The domain prescribed for mixing layer simulations does not contain any immersed boundary cells, meaning that no special wall treatment is required. Mixing layer simulations apply the same pressure and outflow boundary conditions, time integration and statistics parameters, and turbulence modelling as in the precursor stage.

The CHARM TVD scheme will also be used in Eulerian simulations for the convection of the transported scalar, while the low-frequency phase in E-L simulations will be transported by CDS8. An eighth-order central gradient approximation will be used for Eulerian diffusion in both methods.

For velocity inflow conditions the time history data generated by the precursor simulation is linearly interpolated from the previous grid ($\Delta = 0.5$ mm) to the present grid ($\Delta = 0.25$ mm). A linear interpolation is also performed in time between inflow slices, to ensure that the time of the inflow data matches the time of the current time-step in the mixing layer simulation. The re-use of the inflow data generated by the precursor simulation causes independent mixing layer simulations to have identical flow-fields, much as a repeated use of the same initial turbulence yielded identical flows for the mixing layer case in Chapter 10. Because time-averaged statistics are of interest here, this equivalence of flow-fields is of less relevance; however, re-using the same inflow data also mitigates the need to perform a separate precursor simulation for each mixing layer simulation, while still allowing instantaneous results to be compared directly.

The mixing layer simulations are concerned with scalar transport by the Eulerian and E-L methods, so that boundary conditions for mixture fraction are required. To ensure that the correct net scalar flux is maintained inflow conditions of $\phi = 1$ at the lower nozzle exit and $\phi = 0$ at the upper nozzle exit are applied. A unit scalar value is prescribed across the whole of the lower inflow plane,

so that the inert co-flow, which would typically be nitrogen in experiments, is replaced with the species or quantity that the transported scalar is defined to represent. In addition to simplifying the scalar boundary conditions, the reasons for this are two-fold. Firstly, in the simulation of a reactive non-premixed flame the combustion process is often modelled using, for example, a steady flamelet model [20, 243, 244], where there is (by definition) a non-zero rate of reaction wherever a mixture of fuel and oxidiser is present. However, if a single transported scalar quantity is used to describe the chemical state within the domain (i.e. a fuel/oxidiser mixture fraction), it is not possible to include the inert co-flow gas as a separate component. It is therefore more practical to set the co-flow adjacent to the fuel nozzle to be fuel, and the co-flow at the oxidiser nozzle to oxidiser. Secondly, in this manner Lagrangian particles are only added at the interface between the two opposing streams (i.e. on the stagnation plane), and are not required between the co-flow and the lower ($\phi = 1$) stream.

This test case is chosen to validate the E-L method in the context of LES, and Schmidt numbers of $Sc = 0.7$ and $Sc = 1000$ are selected to be representative of the molecular mixing behaviour of gaseous and liquid flows, respectively. Table 11.1 summarises the cases to be studied here, and includes the Obukhov-Corrsin (Eq. 10.5) and Batchelor (Eq. 10.6) scalar length-scales, with respect to the cell size, for these Schmidt numbers. These scales are based on the estimate of the Kolmogorov scale η_K at the nozzle exit quoted above. (An estimate of η_K from Eq. 10.3 – using the integral length-scale and vertical velocity fluctuation at the nozzle exit reported in experiments, as well as the kinematic viscosity of air – yields $\eta_K = 0.027$ mm, which is in good agreement with the experimental value obtained by an alternative means[1]). The chosen cell size is incapable of resolving these scalar length-scales even for the more diffusive configuration, and under-resolves the Batchelor scale by three orders of magnitude for the higher Schmidt number case.

Table 11.1: Mixing layer simulation configurations for the TOJ study.

Case	Method	Sc	Scalar Length-scale	
			η_ϕ/Δ	η_B/Δ
TOJ1-EU	CHARM TVD	0.7	0.1673	-
TOJ2-EU		1000	-	0.0040
TOJ1-EL	Eulerian-	0.7	0.1673	-
TOJ2-EL	Lagrangian	1000	-	0.0040

The initial scalar decomposition for E-L simulations is performed by applying a Gaussian filter kernel of width $\sigma_f = 5\Delta$, requiring $n_f = 20$ support points in each direction, as suggested in §5.3.1. The resolution of the simulated domain is reduced compared to that of the mixing layer configuration of Chapter 10; an increased Lagrangian particle density of $\mathrm{L}^\varrho = 100$ is therefore possible (within the bounds of reasonable computational cost) and is used to improve the accuracy of the solution by a reduction of the noise in the Lagrangian reconstruction. The resolution, domain size, parallel decomposition and particle density of this configuration suggests a maximum of 5 million particles

[1] In experiments, the Taylor microscale is estimated to be $\lambda = 0.8$ mm. This value is used to calculate a Taylor Reynolds number $Re_\lambda = u'\lambda/\nu = 164$. Following Geyer et al. [257], the relationship $\lambda/\eta_K = \sqrt[4]{15}\sqrt{Re_\lambda}$ is used to determine the above value of $\eta_K \approx 0.032$ mm

per sub-domain.

The re-initialisation parameters follow those recommended in §5.4.4 (for particle addition, i.e. $\gamma = 1$) and §8.1 (for removal, i.e. $\lambda = 0.7$ and $\beta = 0.2$). Localised deconvolution is not applied, as the Lagrangian particle density for this case is considered to be too low for deconvolution to yield any improvement in accuracy. Localised low-pass filtering is used, however, since it is expected to improve accuracy through noise reduction in the Lagrangian reconstruction. Filtering parameters of $a = 0.01$, $b = 1.5\Delta$ and $w_{\mathrm{TH}} = 1.2\Delta$ are selected to take into account the reduction in noise associated with the increased Lagrangian particle density, while at the same time preventing the localisation routine from being overly 'aggressive' in its selection of regions to filter.

In this part of the investigation the computational cost of the Eulerian simulations is reduced significantly, by exploiting a useful feature of the Eulerian approach. The flow-field in each mixing layer simulation will be identical; the Eulerian calculations of scalar transport at different Schmidt numbers may therefore be performed concurrently, by defining a separate transported scalar quantity for each Schmidt number. The flow must only be solved once, while solutions for scalar transport at different diffusivities are obtained simultaneously. Profiling of the code has shown that the convective transport of a scalar with a TVD scheme accounts for around 14% of the total computational cost in `PsiPhi` [245], such that the overall reduction in cost gained by transporting the two scalars concurrently may be as much as 40%.

This highlights a potential disadvantage of the E-L method that should be discussed here. By definition the representation of a scalar quantity by Lagrangian particles requires an explicit position and velocity for each particle, while the advection of the scalar field is described by the motion of the particle ensemble. In a simulation in which more than one scalar is transported using the E-L method it is possible to represent those scalars using one 'set' of particles, as long as the distributions of those scalars are identical throughout the entire simulation. However, as soon as the distributions differ – due to, for example, different initial or boundary conditions, or differing diffusivities as in this case – it is necessary to prescribe an entirely separate set of particles for the description of each scalar. Given that the cost of the Lagrangian solution is large compared to that of the Eulerian transport or flow-solving aspects of the code, the same cost reduction benefits in transporting two scalars 'at once' will not be gained (although there will still be some slight advantage derived from the fact that only one flow is being solved). Furthermore the limit on the maximum number of particles within each sub-domain, which is dictated by the amount of Random Access Memory (RAM) available to each process, may make such an approach infeasible; in this case, however, each process has access to sufficient RAM to allow concurrent E-L simulations to be performed.

11.4 Results and Discussion

Results from each stage of the Yale TOJ investigation are presented below. Firstly the veracity of the precursor simulation is verified by inspecting the flow-field and comparing velocity statistics to measured results, before the behaviour of the E-L method in the context of an LES is discussed.

11.4.1 Precursor Simulation

To ensure that the precursor simulation has accurately recreated the flow-field within the TOJ configuration of Coppola et al., velocity statistics from this simulation are compared to the measurements of Coriton and Gomez (previously presented in [100]). In addition the suitability of the computational grid used in the precursor LES is analysed by considering a 'quality indicator', as implemented previously by the author [100].

Pope recommends that for an LES including near-wall regions, the resolution of the grid should capture at least 80% of the total turbulent kinetic energy within the flow [260]. While it should be stressed that a quality indicator is not sufficient to ensure an accurate simulation, it can at least suggest regions within the simulated domain where the grid resolution is lacking in this respect. Based on Pope's suggestion Celik et al. [59] have formulated a model for the viscosity-based quality index LES_IQ_ν, which has the form:

$$\text{LES_IQ}_\nu = \left[1 + \alpha_\nu \left(\frac{\nu_t + \nu}{\nu}\right)^n\right]^{-1} \tag{11.1}$$

In which the constants α_ν and n take values of 0.05 and 0.53, respectively. Considering the limiting case of a DNS (in which the modelled viscosity ν_t tends to zero) the maximum value of LES_IQ_ν reaches 95.2%, while Celik et al. refer to $\text{LES_IQ}_\nu > 80\%$ as being indicative of a 'good LES'. In [100], however, we consider the more straightforward viscosity ratio:

$$r_\nu = \frac{\nu_t}{\nu} \tag{11.2}$$

Where for a DNS $r_\nu \to 0$, and for a 'good LES' (based on the arguments of Celik et al., i.e. $\text{LES_IQ}_\nu > 80\%$), our measure requires $r_\nu < 20$. In a simulation in which the (static) Smagorinsky model is applied the numerator of Eq. 11.2 is a linear function of the chosen Smagorinsky constant C_s, such that the 'quality' of a simulation may be misrepresented by a poor selection of this constant; the indicator of Celik et al. would also be susceptible to such a problem. In the present case, however, this issue is avoided by the use of the dynamic Germano procedure.

A selection of instantaneous, mean and fluctuation plots of horizontal, vertical and lateral velocities across a vertical plane through the entire simulated domain are presented, to illustrate the nature of the flow inside and between the opposing nozzles. Mean and fluctuation values of vertical and horizontal velocities along the axis between, and across, the nozzle exits are also recorded and compared to experimental values. In addition normalised auto-correlations for each velocity component are generated from the simulation at the nozzle exits (according to Eq. 2.16), and are used to estimate the integral length-scales within the flow at this point (applying Taylor's hypothesis, as in Eq. 2.17). The integral length-scale determined in the bulk flow direction (i.e. along the axis) may then be compared to that measured in experiments.

Velocity Statistics: Full Flow-Field Data Figures 11.2(a–b) shows instantaneous and fluctuation values of horizontal velocities throughout the simulated TOJ domain, while Figs. 11.2(c–d) shows instantaneous and mean vertical velocities, Figs. 11.2(e–f) shows instantaneous and fluctua-

tion lateral velocities, and Figs. 11.2(g–h) shows the instantaneous and mean viscosity ratio as an indicator of LES quality. The axes in these plots are normalised by the radius R of the nozzle exit (i.e. $R = 6.35$ mm), and the blue outlines denote the nozzle walls. The first six plots illustrate the highly turbulent nature of the flow within each nozzle plenum, where the high-velocity (~ 22 m/s) jet issuing from the TGP within each nozzle rapidly breaks up due to very high shear across the shortest axis of the inlet slits (i.e. normal to the projections in Fig. 11.2). This shear leads to high turbulence production on the surface of the jet, which rapidly becomes corrugated (as evident from the relatively high horizontal and lateral velocity fluctuations, of up to 3.5 m/s, near to the TGP inlet). Recirculation within each nozzle is apparent from the counter-flowing fluid inside each plenum, and further contributes to jet break-up. Maximum recirculation velocities between the TGP and the end of the contraction in each nozzle of 6.78 m/s (instantaneous) and 4.35 m/s (mean) are recorded, corresponding to around 20% of the inlet jet velocity in the mean. Based on the magnitudes of the horizontal and lateral velocity fluctuations and the mean vertical velocity (as a function of axial position) turbulence intensities from 20% to 40% are observed within each plenum, reducing to around 30% at each nozzle exit.

A maximum instantaneous value in the viscosity ratio of $r_\nu = 7.18$ is found throughout the domain, while the maximum in the mean is $\bar{r}_\nu = 5.43$. These maxima are found to lie within the very high shear regions close to the TGPs; beyond the regions 10mm from each TGP the instantaneous viscosity ratio is typically less than 4. These values suggest that the turbulent kinetic energy within the flow is quite well-resolved, and allows a comparison between experimental and numerical velocity statistics to be drawn with some confidence.

Velocity Statistics: Comparisons with Experiments Figure 11.3 compares velocity statistics from the precursor simulation to those measured in experiments. Profiles of (a) mean vertical velocity, (b) vertical velocity fluctuation and (c) horizontal velocity fluctuation are presented along the axis between the nozzle exits; across the nozzle exit (at a point 0.5 mm downstream, for the convenience of the experimental measurements) are shown (d) mean vertical velocity, (e) mean horizontal velocity, and (f) horizontal turbulence intensity. In each plot the red line corresponds to predicted values, while the '+' symbols represent values measured in experiments. For consistency with the experimental data the latter three quantities are averaged across both nozzles.

The profiles in Figs. 11.3(a), 11.3(d) and 11.3(e) show that the simulation captures the mean velocity field very accurately. Mean vertical velocities along the axis match closely, although the mean position of the stagnation plane is found to be shifted by around 0.5 mm towards the lower nozzle. (During preliminary studies to determine a suitable inflow turbulence level, the stagnation plane was typically observed to stabilise around a position $y = \pm 0.5$ mm. This variation in the mean position may be due to some low-frequency instability in the flow-field, as two identical simulations are often found to yield slightly different statistics). Horizontal velocities across the nozzle exit are accurate up to within 1 mm of the wall. The small peak around the centreline in the mean vertical velocity profile across the nozzle exit may indicate some legacy of the TGP inlet jet persisting beyond the exit of the nozzles. This peak is more defined in the experimental data, but is captured to some extent by the simulation.

Figure 11.3(b) displays a distinct under-prediction of up to 13% in the vertical velocity fluctuation along the axis, with the greatest deviation occurring around the mean position of the stagnation plane (i.e. near to $y/R = 0$). At this position a maximum fluctuation of 3.7 m/s is recorded in the simulation, compared to a measured maximum of 4.2 m/s. The reasons for this discrepancy are not immediately clear, given that the horizontal velocity fluctuations shown in Fig. 11.3(c) are quite well-matched. A possible cause may be due to the inertia of the (physical) particles used for the PIV: the mass of these particles may cause them to penetrate beyond the stagnation plane as they decelerate, resulting in an artificially-elevated measurement of velocity fluctuation. Vertical velocities tend to show greater fluctuations around the stagnation plane due to the steep velocity gradients in this region, as well as any oscillations of the stagnation plane itself. Low-frequency oscillations of the stagnation plane may not be captured over the duration of the simulation, but could be included in experimental results; this may further account for the under-prediction of vertical velocity fluctuations. Simulated horizontal turbulence intensities at the nozzle exits are well-predicted on the centreline with values of around 25% from both simulations and experiments, but show a marked attenuation towards the nozzle walls. This is most likely due to the dissipative nature of the immersed boundary treatment applied in the `PsiPhi` code, where first-order Upwind Differencing is applied adjacent to walls. The measured and predicted levels of turbulence are higher than the intensities of 9% [261] and 14% [254] achieved in previous TOJ experiments.

It should be noted that a good simulated representation of velocity profiles throughout this TOJ configuration is dependent on an accurate prediction of the position of jet break-up within each nozzle plenum, which itself is highly sensitive to the level of turbulence prescribed at the domain inlets. A higher level of inflow turbulence will tend to accelerate this break-up, resulting in a lower turbulence level at the nozzle exit. Unfortunately, without any experimental data for guidance, it is difficult to determine whether the simulated break-up occurs in the correct location (although the good agreement shown in Fig. 11.3 suggests that it is captured quite accurately).

Velocity Auto-Correlations and Integral Length-Scales The temporal auto-correlation functions of horizontal (blue), vertical (black) and lateral (red) velocities are shown in Fig. 11.4. Based on the integration (Eq. 2.17) of these auto-correlation functions up to $R = 0.1$, integral time-scales of $\mathcal{T}_{uu} = 0.265$ ms, $\mathcal{T}_{vv} = 0.517$ ms and $\mathcal{T}_{ww} = 0.259$ ms are recorded in the horizontal, vertical and lateral directions, respectively. From these values the horizontal, vertical and lateral integral length-scales of the simulated turbulence at the nozzle exit are found to be $L_{uu} = 1.96$ mm, $L_{vv} = 3.83$ mm, and $L_{ww} = 1.91$ mm, respectively. This longitudinal (vertical) integral length-scale compares well with the value measured in experiments of 3.50 mm, and is closer than the simulated value of 3.01 mm reported in [100].

For an incompressible, homogenous, isotropic fluid flow, it is possible to demonstrate [262] that:

$$L_{uu} = L_{ww} = \frac{1}{2} L_{vv} \tag{11.3}$$

As an indication of the isotropy of the flow at the nozzle exit, then, it is worth noting that from the precursor simulation results we find the relationships $L_{uu} = 0.512 L_{vv}$ and $L_{ww} = 0.498 L_{vv}$.

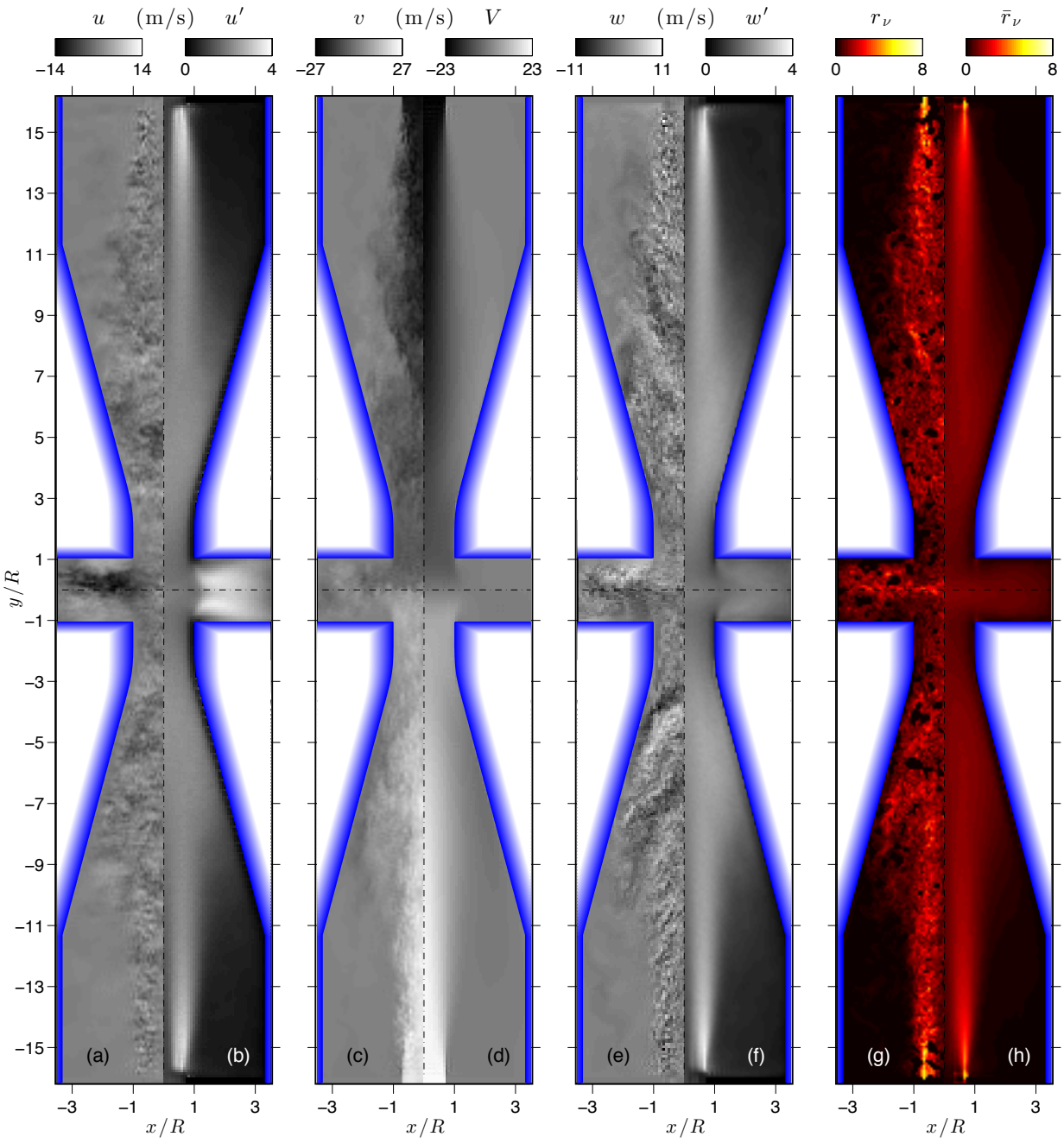


Figure 11.2: Precursor simulation: Horizontal velocities, (a) instantaneous, (b) fluctuation; vertical velocities, (c) instantaneous, (d) mean; lateral velocities, (e) instantaneous, (f) fluctuation; viscosity ratio r_ν, (g) instantaneous, (h) mean.

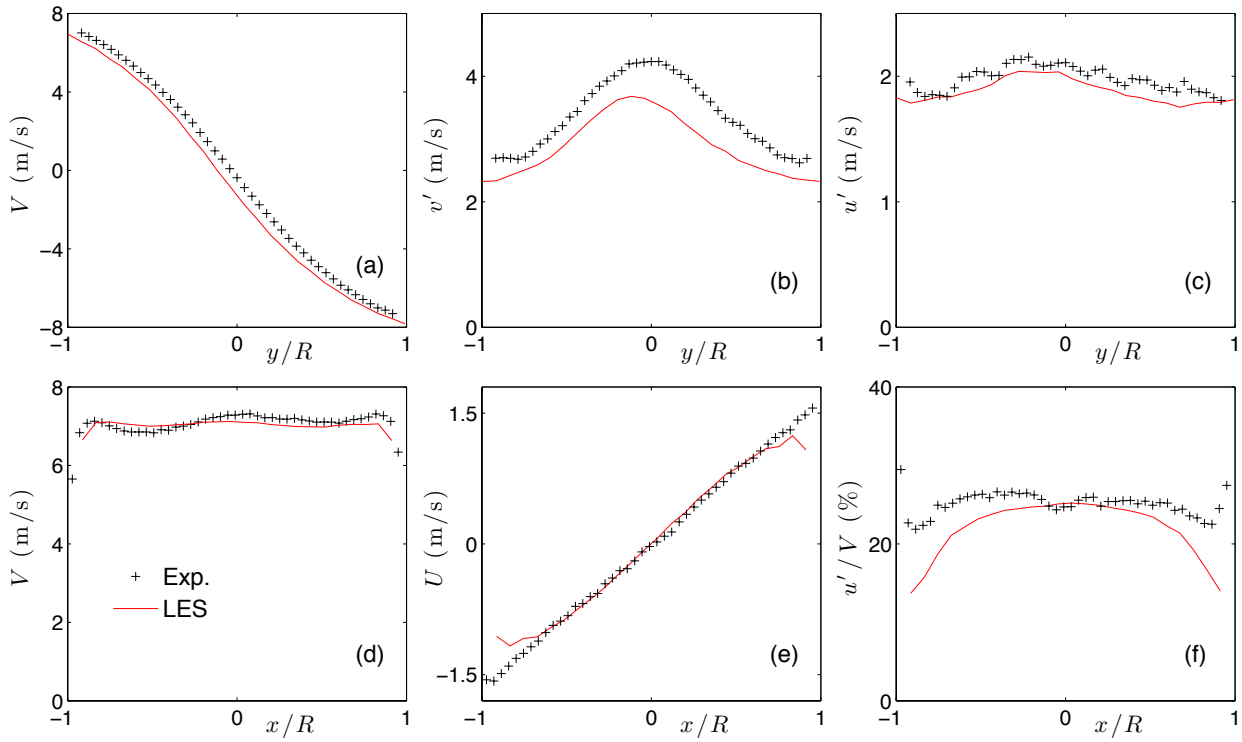


Figure 11.3: Precursor simulation: Velocities from PIV and LES; along the axis: vertical (a) mean, (b) fluctuation; (c) horizontal fluctuation; across the nozzle exit: (d) vertical mean; horizontal (e) mean, (f) turbulence intensity.

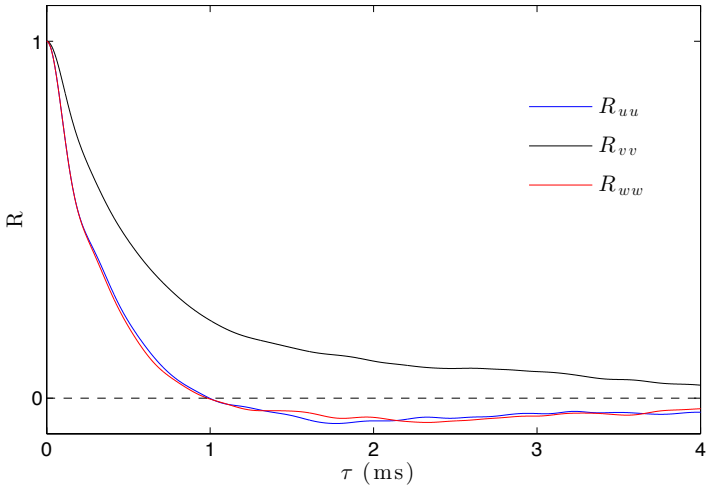


Figure 11.4: Precursor simulation: Normalised auto-correlations for u, v, w velocities.

11.4.2 Mixing Layer Simulations

It is prudent to verify that the simulated flow continues to provide an accurate representation of that measured in experiments. Figure 11.5 shows (a) mean and (b) fluctuation values of the vertical velocity, as well as (c) horizontal velocity fluctuations, along the axis between the nozzle exits. Although the mean values continue to show very good agreement there has been a noticeable reduction in fluctuations: v' is reduced by around 14% at the inlets, while u' has dropped by as much as 16% compared to the precursor simulation data. These quantities improve towards the stagnation plane, however, where they have both recovered to within around 5% of the precursor LES values. The reduction in fluctuations at the inlet may be due to aliasing in the inflow data, where the temporal resolution of the inflow samples is unable to capture some of the highest frequencies in the velocity field. The interpolation between samples (in both time and space) will similarly cause a reduction in velocity fluctuations.

The instantaneous and mean viscosity ratio values for the mixing layer simulations are shown in Fig. 11.6. Maximum values of $r_\nu = 3.17$ (instantaneous) and $\bar{r}_\nu = 1.33$ (mean) indicate that the grid resolution used in these simulations is capable of resolving more of the turbulent kinetic energy than the (coarser) precursor simulation, as expected, and that the quality of the LES continues to be sufficient in this respect. It should be noted that some of the turbulent energy at the inlet will have been lost due to the reduction in velocity fluctuations identified above.

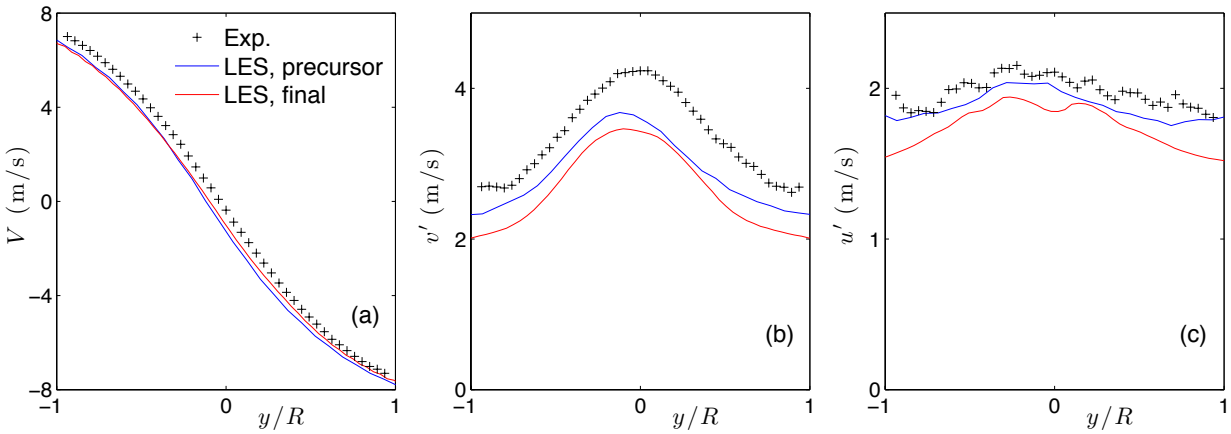


Figure 11.5: Mixing layer simulations: Velocities from PIV and LES; along the axis: vertical (a) mean and (b) fluctuation, (c) horizontal fluctuation.

Before analysing the behaviour of the Eulerian and E-L methods in the TOJ configuration, the computational cost of the mixing layer simulations is briefly discussed. To simulate the transport of two scalar fields for a total of 500 ms the Eulerian method requires around 1700 Central Processing Unit hours (CPUh), or 3.37 CPUh per millisecond of simulated physical time. The E-L method, on the other hand, requires around 7200 CPUh, or 14.39 CPUh/ms. The E-L method is therefore around four times more expensive, which is comparable to the increase in cost observed for the mixing layer study of Chapter 10.

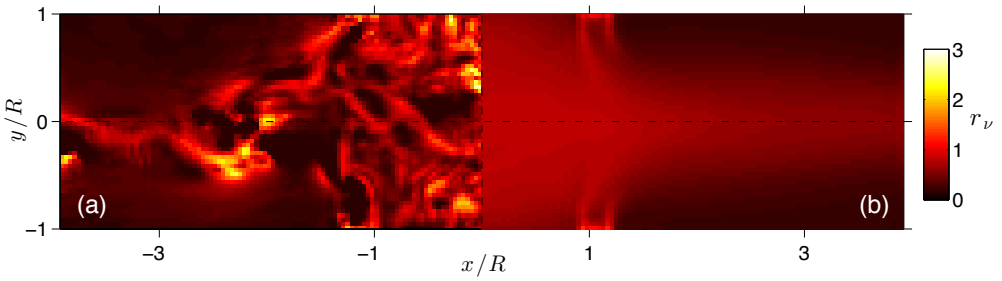


Figure 11.6: Mixing layer simulations: Viscosity ratio r_ν; (a) instantaneous, (b) mean.

Volume Renderings Figure 11.7 presents an instantaneous visual representation of the flow at the stagnation plane, with, from left to right: (a) the transported scalar (e.g. mixture fraction); (b) horizontal, (c) vertical, and (d) lateral velocities. In Fig. 11.7(a) the scalar is marked in white, with zero scalar regions in black; in Figs. 11.7(b), 11.7(c) and 11.7(d) positive velocities are shown in white, while negative velocities are black. The blue lines in each volume rendering signify the extent of the computational domain. Specifically these images are generated from Case TOJ2-EL (although the flow-fields are expected to be identical for all cases). The instantaneous scalar field illustrates the highly turbulent nature of the flow, where a strongly corrugated interface exists between the regions of $\phi = 0$ and $\phi = 1$. The turbulence at the stagnation plane is also clearly apparent in the vertical velocity field, where the two streams from the opposing nozzles impinge and interact. Horizontal and lateral velocities around the stagnation plane are dominated by the divergence of the flow from the stagnation point at the centre of the domain, although some evidence of turbulent structures in these directions is also quite visible.

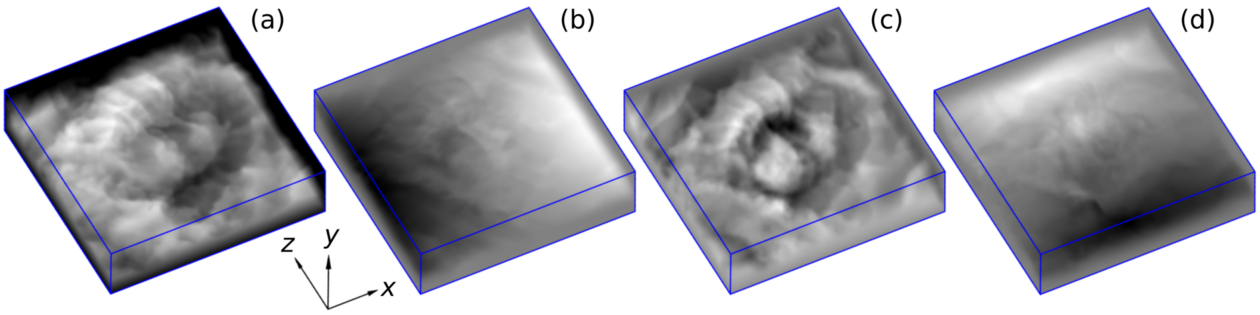


Figure 11.7: Case TOJ2-EL: Volume-rendered instantaneous fields; (a) mixture fraction; (b) horizontal, (c) vertical, and (d) lateral velocities.

Instantaneous Scalar Fields A first comparison of the performance of the Eulerian and E-L methods for this configuration is provided in Fig. 11.8, which shows the instantaneous transported scalar fields across the xy-plane, at a nominal time of $t = 0.240$ s, for Cases TOJ2-EU (Fig. 11.8(a)) and TOJ2-EL (Fig. 11.8(b)). A significant observation is that even at this high Schmidt number the Eulerian method appears to capture the interface relatively well, although the effect of numerical

diffusion is still apparent when compared to the E-L solution. The interface will tend to lie on the stagnation plane, and, since the velocity of the flow in the locality of this plane will be zero, the scalar convective fluxes along this location will also be zero. The Eulerian method will therefore be less susceptible to numerical diffusion in this configuration, as it is only while the stagnation plane itself moves that the scalar field will be subject to any numerical diffusivity. The E-L solution contains some evidence of noise in the Lagrangian reconstruction near to the interface, although the impact and extent of that noise has been lessened by the application of the localised low-pass filtering algorithm.

Inspecting the scalar distributions shown in Fig. 11.8 regions of the interface that are thinly stretched, flat, locally curved and strongly corrugated are observed, as well as regions that exhibit a locally-thickened mixing layer.

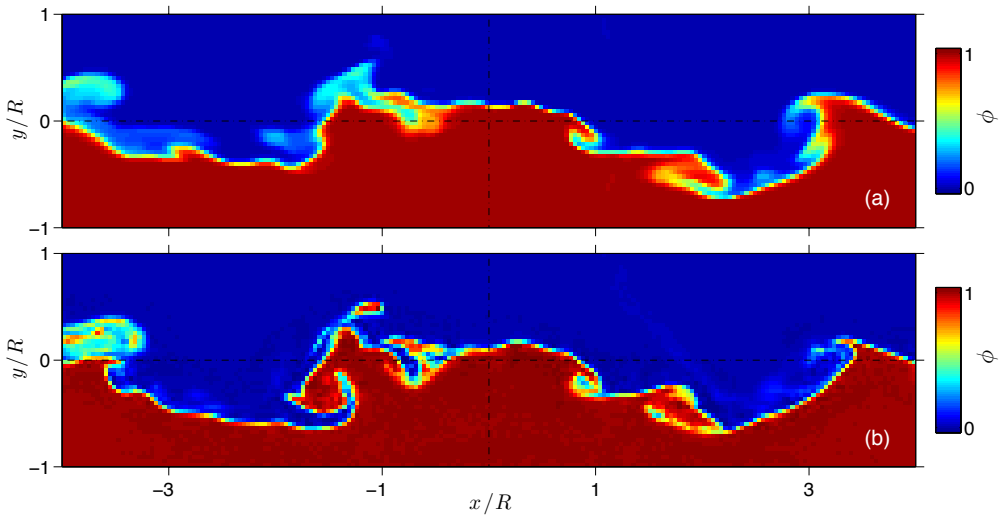


Figure 11.8: Case TOJ2: Instantaneous mixture fraction from (a) Eulerian, (b) E-L simulations.

Scalar Profiles Figures 11.9(a) and 11.9(b) compare the mean and fluctuation scalar profiles along the centreline of the domain from the Eulerian and E-L solutions, for Case TOJ1 ($Sc = 0.7$), and Case TOJ2 ($Sc = 1000$), respectively. The slight shift of the stagnation plane towards the lower nozzle previously identified in Fig. 11.3 continues to be apparent here, and is consistent throughout these and subsequent results.

Although the two configurations experience significantly different physical diffusivities, the profiles from each are remarkably similar. The mean scalar profiles are almost identical, both between the Eulerian and E-L simulations and between the two Schmidt number configurations. It is suggested that the similarity between the profiles at different Schmidt numbers is due to a very short residence time at the stagnation plane, where fluid carrying the scalar quantity is transported away from the stagnation point and out of the computational domain before any significant diffusive fluxes have accumulated. The only slight difference between TOJ1 and TOJ2 lies in the peak scalar fluctuations, where the maximum value of ϕ' is around 2% greater in Case TOJ2 (for both Eulerian and E-L results). A guideline at $\phi = 0.5$ is included in each plot to highlight this small variation. Between the

Eulerian and E-L profiles, however, a difference of around 11% in the peak fluctuation is observed, which is consistent for both Schmidt numbers.

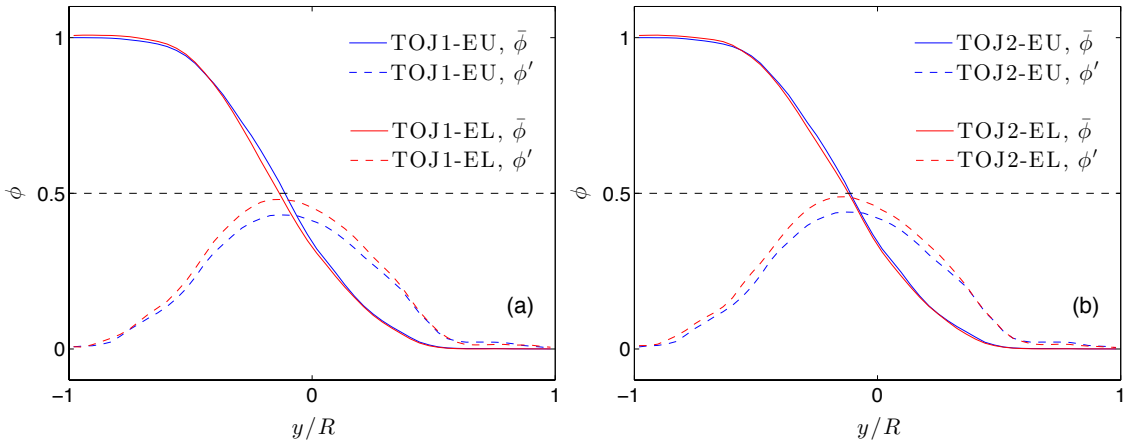


Figure 11.9: Eulerian vs E-L scalar statistics at $t = 0.5$ s: (a) $Sc = 0.7$, (b) $Sc = 1000$.

Given that a CHARM TVD scheme has been shown to exhibit numerical diffusion, and that the effect of this extra diffusivity would be to attenuate any fluctuations in the scalar field, it is notable that the E-L method yields larger scalar fluctuation values. Conversely, it should also be considered that the effect of the noise in the Lagrangian reconstruction may cause an artificially-elevated scalar fluctuation to be recorded. It is therefore difficult to suggest which of the profiles in Fig. 11.9 is closest to the 'correct' (analytical) answer.

One-Dimensional Analysis Although an analytical solution is not feasible for the present case it is possible to construct a heavily-simplified 'opposed jet' in one dimension, from which such a solution may be compared to corresponding one-dimensional Eulerian and E-L results. To replicate the motion of the three-dimensional stagnation plane due to the turbulent flow-field, a one-dimensional scalar interface is oscillated with a sinusoidal displacement about an initial position $y = 0$. The amplitude of the displacement is set to one-quarter of the nozzle separation so that the resulting mean and fluctuation profiles occupy the middle half of the domain, and the calculation is performed over 10 cycles. The probability P of finding the interface at a position $y \in [-0.5R, 0.5R]$ will follow an arcsine distribution of the form:

$$P(y) = \frac{1}{\pi\sqrt{(y + 0.5R)(0.5R - y)}} \tag{11.4}$$

While the mean scalar profile within the region $-0.5R \leq y \leq 0.5R$ will have the form of the corresponding arcsine cumulative distribution function:

$$\bar{\phi}(y) = \frac{2}{\pi} \arcsin\left(\sqrt{y + 0.5R}\right) \tag{11.5}$$

The diffusivity of the scalar for this test is set to zero. The results from this calculation should

not be directly compared to those shown in Fig. 11.9: in the mixing layer simulations the position of the stagnation plane (and scalar interface) would be expected to follow a Gaussian probability distribution centred on the mid-point of the domain. This yields a Gauss error function in the mean scalar profile and a Gaussian curve for the fluctuation profile. The position of the one-dimensional interface, on the other hand, is limited to the region $y/R = \pm 0.5$, and, according to Eq. 11.4, is statistically more likely to be found near to these limits.

The corresponding one-dimensional Eulerian and E-L simulations consist of an interface initially located at $y = 0$, with a co-sinusoidal velocity field imposed such that the amplitude of the interface displacement is equal to that of the analytical calculation. The decomposition parameters, re-initialisation (particle addition and removal) parameters and localised low-pass filtering parameters for the E-L simulation are taken from the full three-dimensional TOJ configuration. The same third-order accurate Runge-Kutta time integration scheme is used as in the three-dimensional simulations.

The resulting scalar mean and fluctuation profiles are shown in Fig. 11.10, where it can be seen that the maximum scalar fluctuations in the analytical case are 0.5. The analytical mean scalar profile and peak fluctuation values are well-matched by the E-L simulation, with ϕ' being over-predicted by only 0.004 (1% of analytical) at the mid-point of the domain. In the Eulerian simulation, however, the peak scalar fluctuation is under-predicted by around 0.122 (24% of analytical) at the mid-point, while the mean scalar profile is also quite clearly smoothed near the limits of the interface position (i.e. around $y/R = \pm 0.5$). The E-L method does deviate significantly from the analytical result outside of the range of motion of the interface, where scalar fluctuations are over-predicted by up to 0.04.

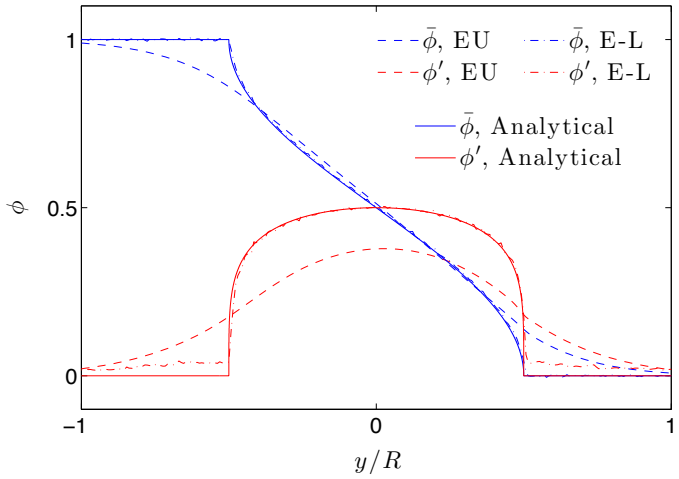


Figure 11.10: Analytical, Eulerian and E-L scalar profiles for a 1D oscillating interface.

From these results it is apparent that numerical diffusion does have a significant impact on the prediction of scalar fluctuations, and that the magnitude of such under-predictions is more significant than any over-prediction in an E-L result due to noise in the Lagrangian reconstruction (for the present Lagrangian particle density, at least). Furthermore the results of this one-dimensional test suggest that any such over-prediction in the E-L solution will typically occur away from the region

of interest (i.e. the stagnation plane) in an opposed jet configuration.

It is suggested that the error in the prediction of scalar fluctuations through the use of a CHARM TVD scheme may have some bearing on the observed under-prediction of vertical velocity fluctuations apparent in Figs. 11.3 and 11.5. Although the momentum terms themselves are transported by CDS8 the density field is transported by CHARM TVD, and the two are closely coupled.

Eulerian-Lagrangian Components For subsequent analysis of the behaviour of the E-L method it is sufficient that only one set of results is considered, as the variation of the Schmidt number for this configuration makes little difference to the scalar field statistics; the higher Schmidt number (Case TOJ2-EL) solution is chosen.

An impression of the balance that exists between the Eulerian and Lagrangian phases is gained by considering their first and second statistical moments, as shown in Fig .11.11(a) for the low-frequency phase ϕ^E and Fig. 11.11(b) for the high frequencies in ϕ^L. Here, and in subsequent statistical plots, the rotational symmetry of the flow-field is exploited to quadruple the number of statistical samples by averaging over the xy- and yz-planes and by subsequently averaging each quantity in a reflection about the centreline.

Considering the Eulerian phase we observe the characteristic stagnation plane scalar distribution, albeit for a smoothed scalar profile. The turbulence of the flow-field and any oscillatory motion of the scalar interface is evident in the relatively high fluctuations (approaching 0.5, or 50% of the maximum low-frequency scalar value) around the mid-plane of the domain. The Lagrangian phase, on the other hand, is found to have a small contribution to the mean overall, with peak mean values of only ± 0.02. The high-frequency phase exhibits a general distribution of particles with a negative scalar contribution towards the upper nozzle (where $\phi = 0$) and positive particles towards the lower nozzle ($\phi = 1$). The low mean values, coupled with relatively high fluctuations (around 0.25), are consistent with a narrow band of positive and negative particles in close proximity to one another moving erratically about the mid-plane of the domain under the effect of a highly-turbulent flow.

Re-Initialisation: Particle Addition As stated above the stability of the stagnation plane in the TOJ configuration allows some insight into the balance that exists between particle addition and removal in the E-L method. The high strain rates around the stagnation plane serve to steepen the scalar gradient in this region, necessitating the continuous addition of particles as high frequencies (sharp changes in gradient) are generated within the low-frequency phase. Figure 11.12 shows the mean and fluctuation particle addition statistics for Case TOJ2-EL. The magnitudes of these statistics are small, as they represent the first and second moments of particle addition for each individual re-initialisation event. As a source term in the Eulerian-Lagrangian transport equations for LES (Eqs. 5.10 and 5.11) these values could equally be expressed as rates of scalar transfer within each cell, where the range of values in the mean field becomes approximately $-80/\mathrm{s} < \bar{S}_{\phi^L} < 90/\mathrm{s}$. Alternatively the rate of particle addition in each cell can be determined by multiplying these scalar transfer rates by the Lagrangian particle density $\mathrm{L}^{\varrho} = 100$, suggesting that up to 9000 particles are added per cell per second.

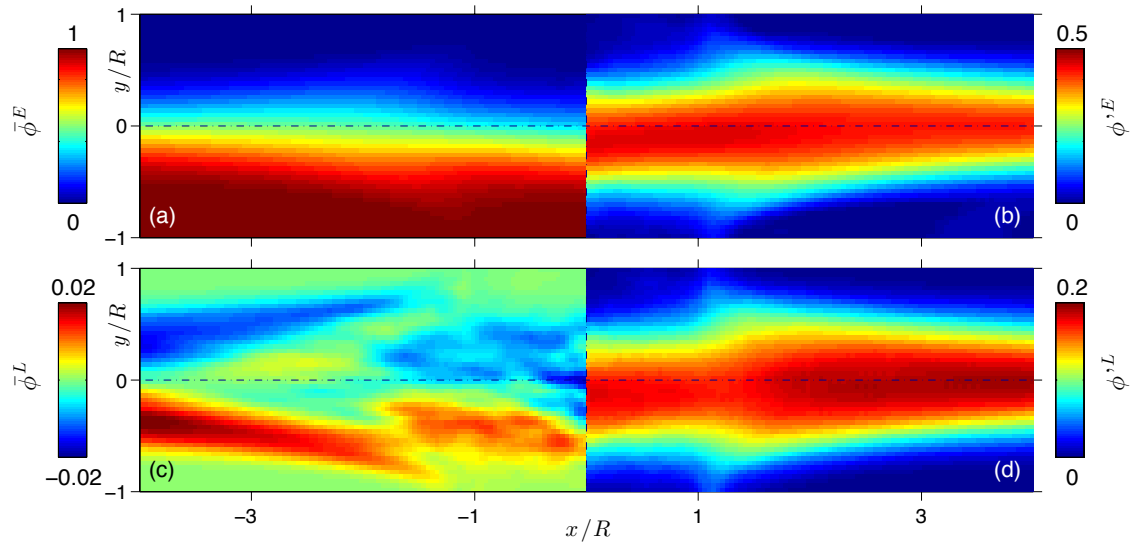


Figure 11.11: E-L components, Case TOJ2-EL: Eulerian (a) mean, (b) fluctuation; Lagrangian (c) mean, (d) fluctuation.

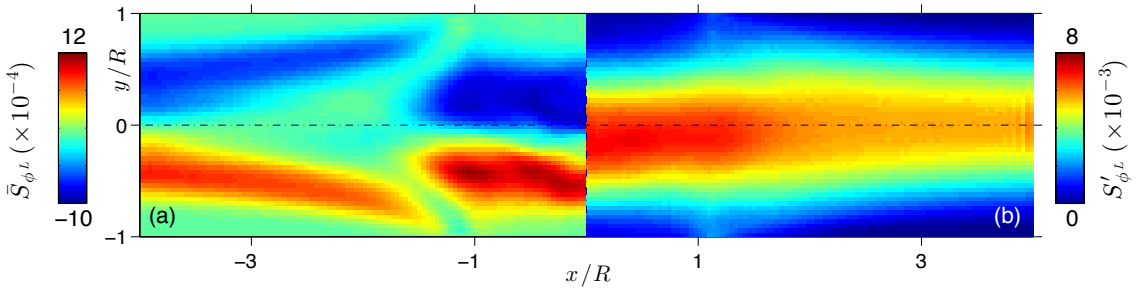


Figure 11.12: Particle addition, Case TOJ2-EL: Lagrangian source (a) mean, (b) fluctuation.

It can be seen that the magnitudes of positive and negative particle addition are approximately equal, and that the distributions are generally mirrored about the mean stagnation plane position. The most significant sources to the Lagrangian phase occur close to the axis of the TOJ apparatus, where the jets issuing from the nozzles directly impinge upon each other. In this region the strain rates in the flow-field will be largest – as the jets are pushed away from the stagnation point by the relatively high pressures at the centre of the stagnation plane – and the requirement of additional particles to remove high-frequency information from the low-frequency phase will be greatest.

Beyond one nozzle radius in the horizontal direction the distribution of particle addition is less well-defined, but is mostly dominated by new particles being created where the slower co-flow streams approach the scalar interface. (Recall that each co-flow stream is set with a scalar boundary condition equal to that of its corresponding nozzle). Between these regions, at a distance of around $2R$ from the axis, there exists a region with a net source contribution of approximately zero. From the non-zero fluctuations at this point, however, it seems that the addition of positive and negative particles happens to cancel out here (unlike within the regions immediately downstream of the nozzle exits, where it is reasonable to say that there are definitely no particles added at all). It is

suggested that the scalar interface is most highly perturbed around $x = \pm 2R$, such that there are approximately equal probabilities of finding the interface above, on, or below the mid-plane of the domain. This is consistent with the mean Lagrangian distribution shown in Fig. 11.11(c), where it can be seen that the total particle contributions also approximately cancel out within this region.

Re-Initialisation: Particle Removal Particle removal statistics are considered in terms of the particle removal probability P_{MC} and the Eulerian source term $S_{\phi E}$, as shown in Figs. 11.13(a) and 11.13(b) respectively. The probability of particle removal near to the stagnation plane is very low (both in the mean and the fluctuation values), as there will usually be a gradient (or change of gradient) in the low-frequency phase in this region. Further away from the mean position of the scalar interface the probability of a particle being selected for removal increases, as particles are less likely to be required here (i.e. the frequencies in the scalar field are low enough to be accurately transported by an Eulerian approach only).

The magnitudes of the Eulerian source term are found to be small, as these values represent the first and second moments of particle removal at each individual re-initialisation event. Expressed as a scalar transfer rate these values translate into a range of around $-10/\mathrm{s} < \bar{S}_{\phi E} < 10/\mathrm{s}$, which is equivalent to about 1000 particles removed per cell per second. It should be noted that this removal rate does not account for those particles that are transported out of the domain.

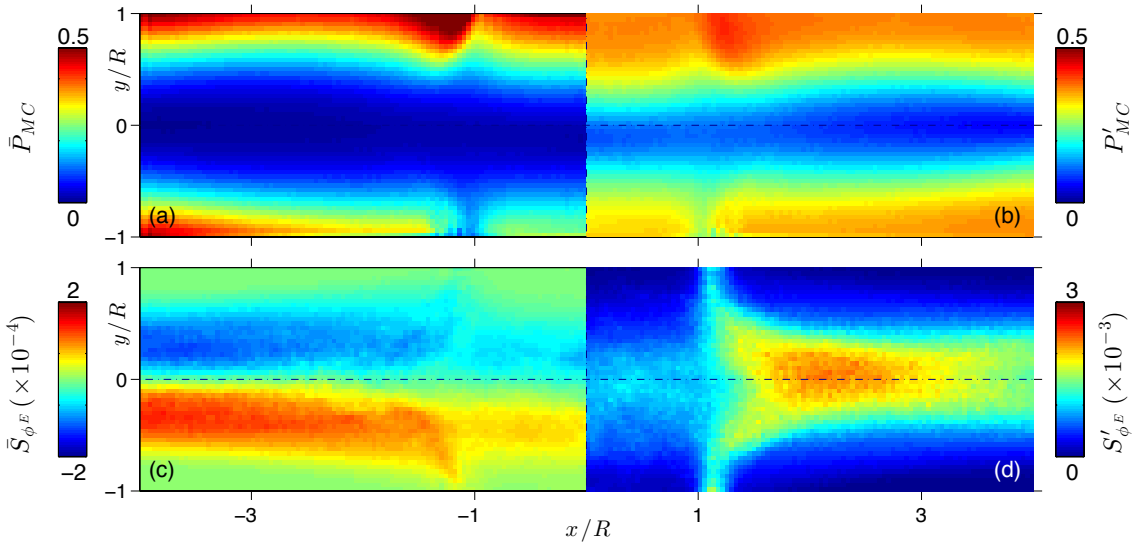


Figure 11.13: Particle removal, Case TOJ2-EL: Probability (a) mean, (b) fluctuation; Eulerian source (c) mean, (d) fluctuation.

In terms of the distribution of the Eulerian source term, it is apparent that fewer particles are removed towards the axis where the jets from the opposing nozzles meet. The distribution of particle addition shown in Fig. 11.12 indicates that a balance between particle addition and removal has been achieved. (High values in the Eulerian source term here would otherwise suggest that this balance was not ideal, as it would indicate that particles were being removed immediately after their introduction).

Particle Numbers The final part of this analysis considers the numbers of particles required in an E-L simulation. Figure 11.14 shows the variation in total particle number with time, as well as the numbers of particles added and removed, for Case TOJ2-EL.

From this set of data there are 25.36 million particles on average within the computational domain at any given time, or around 12 particles in each computational cell, with a standard deviation of around 8% of this value. Per re-initialisation event (i.e. every four steps) there are on average 368000 particles added (with a fluctuation of 7%) and 56000 removed (with a fluctuation of 23%). Given that the mean total shows no systematic increase over time, it is apparent that the balance of particles are deactivated as they leave the simulated domain (i.e. around 78000 particles per step).

The data for Case TOJ1-EL is not included here; the trends, means and standard deviations are almost identical to those shown in Fig. 11.14, while there exists a correlation between individual peaks and troughs within the two data sets. This is expected, as the flow-fields and scalar profiles for the two simulations are approximately the same.

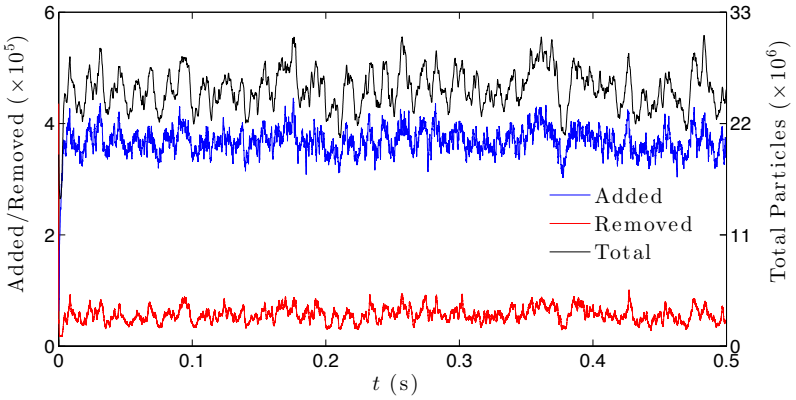


Figure 11.14: Particle numbers, Case TOJ2-EL: Total active, added, and removed.

11.5 Conclusions

In this chapter the joint Eulerian-Lagrangian method has been tested in the context of a Large-Eddy Simulation, where a Turbulent Opposed Jet geometry based on that developed at the Yale Center for Combustion Studies has been simulated. The size of the domain required for such a simulation, coupled with the desire to generate fully-converged time-averaged statistical data, has made the use of Direct Numerical Simulation impractical as a predictive tool.

Simulation of the flow-field and scalar transport has been performed in two stages. A precursor calculation of the entire geometry has been carried out to generate a time history of inflow data at the nozzle exits, and this data has then been repeatedly re-used in a series of Eulerian and E-L simulations of the region of interest (between the nozzle exits). Such an approach has delivered a significant saving in computational cost, as the majority of the physical domain (i.e. upstream of the nozzle exits) has only been simulated once. The accuracy of this precursor simulation has been verified by comparing predicted mean and fluctuation velocity statistics, as well as the longitudinal

integral length-scale at the nozzle exit, with values measured in experiments. An LES 'quality indicator' based on an index proposed by Celik et al. has also been used to ensure that the selected computational grid sufficiently resolves the turbulent kinetic energy of the flow. Simulation numerics are the same as those applied in the mixing layer case of Chapter 10, except that it has been necessary to use a TVD scheme for the convective transport of density (to prevent oscillatory behaviour in this field near to immersed boundaries).

Mixing layer simulations applying the inflow data from the precursor simulation have been performed with Eulerian and E-L methods for the convective transport of an inert scalar quantity at Schmidt numbers of $Sc = 0.7$ and $Sc = 1000$. In practice it is observed that the differences in scalar means and fluctuations at differing diffusivities are small. It is suggested that this is due to the short residence time of the scalar on the stagnation plane, as well as the relatively large motion of the stagnation plane in comparison to the narrow mixing layer thickness in both configurations. Furthermore it is revealed that the mean scalar profiles from Eulerian and E-L computations are almost identical. Numerical diffusion plays a less significant role in Eulerian simulations of this case as the scalar interface generally resides on the stagnation plane, where local velocities (and therefore convective fluxes) are small or zero. The impingement of the opposed jets also serves to steepen scalar gradients, which may obfuscate any numerical diffusion.

The most notable difference between the two methods is found in the scalar fluctuations, where the numerical diffusion of the CHARM TVD scheme causes an under-prediction of 24% in this quantity in a one-dimensional analysis. This same analysis also demonstrates the ability of the E-L method to accurately capture these fluctuations, and that any over-prediction of scalar fluctuation values by the E-L method due to noise in the Lagrangian particle reconstruction is small in comparison (for the Lagrangian particle density used in the present case, at least). It is therefore suggested that the scalar fluctuation profile from the E-L simulation of the Yale TOJ is more accurate than that from the Eulerian simulation.

The high rates of strain across the stagnation plane and scalar interface make an effective application of the E-L particle addition routine essential, as high frequencies are introduced into the low-frequency phase. At the same time a balance between the particle addition and removal routines is important, as the removal of newly-created Lagrangian particles is typically undesirable. The effectiveness of the particle addition routine and the balance between addition and removal have been investigated and verified by analysing the first and second statistical moments of the Lagrangian and Eulerian source terms.

12 Conclusions and Further Work

This chapter presents conclusions on the overall performance of the joint Eulerian-Lagrangian method, and suggests some avenues for further application and development.

12.1 Conclusions

Traditional numerical schemes for the approximation of convective fluxes in a time-dependent Computational Fluid Dynamics simulation are known to suffer from numerical diffusion, where sharp changes in a scalar gradient are artificially smoothed during transport. The purpose of this work has been to develop a novel method for the numerical integration of the convection term in the general transport equation (Eq. 2.9) to eliminate the numerical diffusion of existing schemes. A transported scalar quantity is decomposed into smooth (low-frequency) and rapidly-varying (high-frequency) components, by the application of a Gaussian low-pass filter. Separate transport equations for the two components have been formulated for constant and variable density cases in the context of Direct Numerical Simulation and Large-Eddy Simulation, and may be solved independently.

The low-frequency scalar field is convected with good accuracy in the Eulerian framework using an eighth-order Central Differencing Scheme (CDS), which suffers from negligible numerical diffusion but exhibits unphysical oscillatory behaviour when transporting sharp changes in gradient. For consistency diffusive fluxes in the Eulerian framework are also approximated with eighth-order accuracy. The high-frequency information in the scalar field is described and transported using massless Lagrangian particles. These deliver excellent accuracy in transport regardless of the local scalar topology, but are more computationally expensive than an Eulerian method. Particle convection applies an equation of motion where the particle velocity is interpolated from the underlying (Eulerian) flow-field, while a stochastic Wiener process is used to represent particle diffusion.

After transport an Eulerian description of the Lagrangian field is reconstructed from the individual particle scalar contributions, to allow the total transported scalar field to be recovered from the summation of the Eulerian and Lagrangian components. This reconstruction introduces some noise into the solution, due to the random distribution of the point-like particles. The accuracy and expense of the joint Eulerian-Lagrangian method may be controlled by the selection of an appropriate Lagrangian particle density, where a higher density yields a reduction in the amplitude of the Lagrangian noise but increases the computational cost. Re-initialisation (particle addition) is used to maintain a suitable smoothness in the Eulerian field to prevent the development of oscillations due to the use of a CDS, while particle removal reduces the computational cost of the method and

may also deliver an improvement in accuracy. Localised low-pass filtering is used to reduce the impact of the noise in the Lagrangian particle reconstruction, while a localised deconvolution algorithm removes a small degree of filtering inherent in the reconstruction routine. These algorithms may be applied separately or in combination, depending on the nature of the problem to be solved. The novelty of the joint Eulerian-Lagrangian method lies in its combination of the Eulerian and Lagrangian approaches, drawing upon the advantages of each to deliver a more accurate solution at a reasonable additional cost.

Three cases were selected to test the joint Eulerian-Lagrangian (E-L) method in two dimensions, before two more complex and physically-representative configurations were studied in three dimensions. For the first of the two-dimensional cases the rotating slotted disc of Zalesak was used to provide a clear analysis and comparison of the behaviour of the Eulerian and E-L methods. For the Zalesak study the SuperBee Total Variation Diminishing (TVD) scheme was found to deliver the best results of the applied Eulerian schemes, while the E-L method was at least twice as accurate (in terms of the total error between the predicted and analytical fields) but also at least twice as computationally expensive (in terms of the time to solution).

In the second configuration the Zalesak case was modified to provide some variation in the scalar topology, to include diffusion transport terms, and to test the performance of the E-L method over long simulation durations (i.e. for more computational time-steps). The results from this part of the study suggested that the E-L method in its initial form was only as accurate as an Eulerian scheme for cases of relatively high diffusivity (i.e. low Schmidt number), while continuing to be between two and ten times more expensive (depending on the prescribed Lagrangian particle density). Over longer durations at low diffusivity, however, the E-L method was capable of accurately transporting a scalar field where the CHARM and SuperBee TVD schemes delivered highly inaccurate results. For the long-running simulations the significant distortion of the scalar field by the Eulerian schemes advocated the use of the E-L method, even though the latter was at least four times more expensive for this case.

Based on the results of the Zalesak and modified Zalesak tests, a number of improvements to the method were identified and added. These were: i) a particle removal algorithm, designed to reduce computational cost and potentially improve accuracy; ii) a localised low-pass filtering algorithm, to remove noise in the Lagrangian reconstruction; and iii) a localised deconvolution algorithm, to correct some intrinsic filtering in the weighted averaging procedure used to produce the Lagrangian reconstruction. The particle removal algorithm proved to be beneficial only for low Schmidt number cases, while the deconvolution algorithm was only applicable to zero diffusion (i.e. infinite Schmidt number) configurations at relatively high Lagrangian particle densities. The low-pass filtering algorithm, however, has been shown to improve accuracy for all of the cases tested.

The third two-dimensional configuration considered a scalar field deformed by a single vortex, where the shear in the flow elongated the shape of that scalar field. This case tested the particle

re-initialisation algorithm, where particles are added to regions of the flow in which the rate of change of scalar gradient is increasing. Investigations into the computational 'value' (in terms of the accuracy and cost) of the E-L method were carried out by performing Eulerian simulations over a range of grid resolutions and comparing the E-L results against them. In this case the E-L method was found to be up to one order of magnitude more accurate (in terms of total error) than an Eulerian simulation of equivalent computational cost. The good performance of the E-L method was further highlighted by the formulation of a standardised performance measure, which, while preferring a low error solution to a quick execution, showed the E-L method to be considerably more effective for the single vortex case.

Having defined the methodology of the E-L method, and with a clear understanding of its behaviour, a more realistic three-dimensional shear layer was studied for Schmidt numbers ranging from 0.7 to 1000. The size of the domain, as well as the simulated duration, were selected to allow a Direct Numerical Simulation (DNS) to be performed; however, although the Kolmogorov length-scale was resolved by the computational grid, the Batchelor scale (representing the size of the smallest scalar structures) was not. With no analytical solution for direct comparison the accuracy of the Eulerian and E-L methods was instead quantified by considering probability distributions of scalar gradient conditioned on scalar value, mixing layer thicknesses, and instantaneous scalar dissipation rates. For each of these properties of the scalar field the E-L method out-performed the Eulerian method; most notably, the mixing layer thickness at a high Schmidt number was captured more accurately by the E-L method (with a cell size $\Delta = 20$ μm) than by an Eulerian simulation at twice the spatial resolution ($\Delta = 10$ μm). Although the E-L method was up to five times more expensive than Eulerian simulations at the same resolution, it was capable of capturing more scalar detail and described the scalar gradients more accurately at higher Schmidt numbers ($Sc \geq 100$). The E-L method at $Sc = 1000$ with a cell size $\Delta = 20$ μm exhibited better overall performance compared to the more highly-resolved Eulerian simulation ($\Delta = 10$ μm), at one-eighth of the computational cost, and with at least an equivalent (and arguably better) accuracy.

The final configuration was based on the laboratory-scale Turbulent Opposed Jet (TOJ) apparatus developed by the research group of Gomez at the Yale Center for Combustion Studies. The desire to generate time-averaged statistical data, coupled with a much larger domain size than that of the mixing layer case, made a DNS impractical. Large-Eddy Simulations (LES) were therefore performed, and the simulation of the TOJ geometry was divided into two phases. A precursor calculation was made to predict the flow-field inside the opposing nozzles, and to generate a time history of velocity data for use as the inflow conditions in a smaller, more highly-resolved series of Eulerian and E-L simulations of the stagnation plane and mixing layer between the nozzle exits. By comparing time-averaged statistics for the first and second moments of velocity from the precursor simulation to measurements made at Yale, the precursor simulation was shown to have captured the details of the flow with good accuracy. In the latter simulations the mean scalar profiles from the Eulerian and E-L simulations were almost identical, because the thickness of the mixing layer was typically small compared to the motion of the stagnation plane; the mean profiles were also

weakly dependent on the applied Schmidt number. A notable difference between the two approaches was observed in the prediction of scalar fluctuations across the mid-plane of the domain, however. It was subsequently shown that an Eulerian simulation should be expected to under-predict such fluctuations, while an E-L simulation captures them accurately.

The original objective of this study was to develop and test a novel method for the approximation of convective fluxes within the Eulerian framework of a DNS or LES, to eliminate the numerical diffusion of traditional convection schemes. The results presented throughout this study demonstrate that this objective has been met. The significant improvement in accuracy delivered by the joint Eulerian-Lagrangian method has an additional benefit in that it may reduce the cost of numerical convective mixing studies, as it is able to capture small scalar features on less highly-resolved computational grids. It is hoped that the joint Eulerian-Lagrangian method will become a powerful and useful predictive tool for many branches of Computational Fluid Dynamics in the future.

12.2 Further Applications

In addition to the shear mixing layer and Turbulent Opposed Jet configurations presented in Chapters 10 and 11, respectively, the joint Eulerian-Lagrangian method has a potential for practical application to a wide variety of other realistic problems. Two such applications are suggested here.

12.2.1 Downstream Mixing in a Bluff Body Jet

Bluff body jets are widely used in research as they consist of a simple, well-defined geometry with a predictable and repeatable flow-field, and offer good access within all regions of the flow domain for measurements by a range of experimental techniques. Some of the earliest work with bluff body jets has been presented by Longwell et al. [263] and Spalding [264], amongst others, while a variety of bluff body configurations have since developed over the years.

A typical bluff body geometry features a high-speed central jet emitting from a thick-walled nozzle, which is surrounded by a relatively low velocity co-flow. In experiments involving reactions the central jet would consist of a fuel while the co-flow would consist of an oxidiser. Recirculation zones are created in the downstream region of the nozzle exit as the jet and co-flow pass the end of the nozzle, and these zones are particularly useful in reactive experiments as they stabilise and anchor the flame.

To illustrate the potential benefits of the E-L method for a bluff body jet simulation, the geometry of Dally et al. [265, 266] is considered. This geometry consists of a nozzle with an inner (jet) diameter of 3.6 mm and an outer diameter of 50 mm. The 'HM1' flow configuration studied by Dally et al. includes a jet velocity of 118 m/s and a co-flow velocity of 40 m/s, where the jet consists of a methane/hydrogen mixture and the co-flow is air. The flow-field of this configuration is characterised by strong shear across the surface of the jet, which causes the jet to break up rapidly. Within the region of jet break-up the scalar (e.g. mixture fraction) gradients are still relatively steep, while the Péclet number is high; this results in convective mixing by the dominant

turbulent transport processes. Further downstream the turbulent kinetic energy of the flow will dissipate, and convective mixing processes will be augmented by diffusion.

The potential of the E-L method for this case lies in the accurate description of convective mixing near to the nozzle exit, where scalar gradients are still steep. The convective transport of those gradients may not be accurately captured by a traditional Eulerian scheme, but should be well-represented when Lagrangian particles are included for the scalar field. Further downstream, where the particles may no longer be required, the particle removal algorithm proposed in §8.1 would be effective in transferring their scalar contribution to the Eulerian phase.

12.2.2 Mixing in a Stirred Tank

In the science of Chemical Engineering there is frequently a need to mix two or more chemical components. The mixing process may include a reaction between those components, or may simply be required to produce a complete mixture for subsequent use. A good example of the latter would be the mixing of colours in the production of paint, where specific quantities of base pigmentations must be thoroughly mixed to produce derived colours to an exact specification.

To continue with this example the mixture of paint may constitute an attractive case for the application of the E-L method. The mixing process is typically driven almost entirely by convection, as the resins used (e.g. acrylic emulsion, epoxy or urethane) have high Schmidt numbers and are therefore slow to mix by molecular diffusion. The convective mixing process is often driven by a stirred tank, consisting of a rotating paddle inside a cylindrical drum that is lined with baffles to disrupt the vortical flow pattern. The interactions between the paddle and baffles create complex flow structures at a range of length-scales, and these ensure that the paint constituents achieve the homogeneity required for a consistent coloration. The flow inside stirred tanks (beyond the mixing of paints) has previously been studied by a number of authors both experimentally [267, 268, 269] and numerically [270, 271, 272].

The traditional numerical schemes tested previously in this study would be unable to accurately predict the mixing process inside a stirred tank containing high-Schmidt number fluids, due to the over-prediction of diffusive fluxes between the mixing components. Interfacial methods (such as the level set method mentioned in §5.1) would also be inappropriate, as they would not account for the contribution of the (small, but non-negligible) physical diffusive fluxes. The ability of the E-L method to accurately describe the transport of small scalar features within a high Schmidt number flow may prove beneficial not only for the example of a paint mixture, but also within a wider range of disciplines from pharmacology to minerals processing.

12.3 Further Development

Although the present work has endeavoured to explore the performance of the E-L method as fully as possible, many avenues for further development exist. Some developments that have become apparent during this study, and that may increase the potential of the E-L method as a simulation tool, are discussed here.

12.3.1 Re-Initialisation

The proposed particle addition and removal algorithms have been shown to improve stability, increase accuracy, and reduce the computational cost of the E-L method. However, they rely upon the selection of values for the parameters γ, β and λ, which represent, respectively: the second derivative of the Eulerian field above which Eulerian information will be transferred to the Lagrangian phase; the second derivative of the Eulerian field below which information will be transferred from the Lagrangian phase to the Eulerian phase; and the gradient of the Eulerian field above which information will not be transferred from the Lagrangian phase to the Eulerian phase. The particle addition and removal algorithms are closely linked in their functionality, and care must be taken to ensure that a suitable balance exists in the 'aggressiveness' of each to add or remove Lagrangian particles. Ideally, their formulation would be modified so that this balance (and subsequently, the behaviour of the two algorithms) is controlled by one parameter only.

Throughout the present work re-initialisation has been performed at a user-defined frequency. This could be improved upon, however, by automating the particle addition routine so that it is only applied when necessary: for example, when the second derivative of the Eulerian field has exceeded some triggering value. This value could be taken as the second derivative of the initial Eulerian field, which is presently the suggested limit for particle addition.

While particle removal has been shown to work well for a variety of cases it has also exhibited some deficiencies that limit its application, for example under the high Schmidt number conditions of Case ML5-EL (see Chapter 10). Confidence in the implementation of the E-L method would be gained by modifying the removal algorithm to work with any Schmidt number, so that no choice regarding its use would need to be made. Some experimentation into the formulation of the algorithm may also be useful: in particular, it may be more effective to take the absolute values of the second derivative of the Eulerian field in each direction, such that Eqs. 8.7 and 8.8 would read:

$$\nabla^2 \phi^E = \sum_{i=1}^{d} \left| \frac{\partial^2 \phi^E}{\partial x_i^2} \right| \tag{12.1}$$

$$\begin{aligned} \nabla^2 \phi^E(i,\, j,\, k) \approx \left|\phi^E(i-1,\, j,\, k) - \phi^E(i,\, j,\, k)\right| + \left|\phi^E(i+1,\, j,\, k) - \phi^E(i,\, j,\, k)\right| + \\ \left|\phi^E(i,\, j-1,\, k) - \phi^E(i,\, j,\, k)\right| + \left|\phi^E(i,\, j+1,\, k) - \phi^E(i,\, j,\, k)\right| + \\ \left|\phi^E(i,\, j,\, k-1) - \phi^E(i,\, j,\, k)\right| + \left|\phi^E(i,\, j,\, k+1) - \phi^E(i,\, j,\, k)\right| \end{aligned} \tag{12.2}$$

Using the absolute value of the second derivative in this context may prevent some particles from being removed erroneously, in regions where the (signed) rates of change of gradient in perpendicular directions happen to cancel out.

12.3.2 Localised Low-Pass Filtering and Deconvolution

The localised low-pass filtering algorithm has been demonstrated to improve the accuracy of the E-L method by reducing noise in the Lagrangian reconstruction. However, this requires the selection of values for the parameters a, b and w_{TH}, which represent, respectively: the rate of change of

gradient below which the reconstruction will be filtered; the maximum effective Gaussian filter width that will be used; and the width of the top-hat filter that is iteratively applied to actually remove the particle noise. Varying these values allows the effectiveness of the low-pass filtering algorithm to be modified; ideally, however, the routine would be formulated so that it could be applied in any situation with a standard configuration. This would require further investigation, but may demand the use of an alternative filtering methodology (such as the wavelet restoration method of Donoho [211]).

The localised deconvolution algorithm, as it stands, does not have any parameters; however, it would also benefit from some modification to allow it to be applied to any case with confidence. It has been shown that a key disadvantage of the deconvolution algorithm is its amplification of the 'resolution error' that occurs around a sharp scalar interface. This problem could possibly be solved by including, outside of the existing Lagrangian implementation, a number of 'tracker' particles that are initialised at the centre of each computational cell. These particles would then trace the translation and deformation of the scalar interface, and would allow the significance of resolution error in a given cell to be assessed (where a tracker particle close to the centre of a cell would suggest a small resolution error in that cell, while a particle near to the edge of the cell would indicate a larger resolution error, as discussed in §9.4.2). The application of localised deconvolution could thus be limited to cells only in which the resolution error is relatively small.

12.3.3 Resolving the Batchelor Scale

In Chapter 9 the potential of the E-L method to deliver very high resolution scalar information has been illustrated in Fig. 9.11. The concept of performing a sub-grid Lagrangian reconstruction may have particular significance for Direct Numerical Simulations (DNS) at relatively high Schmidt numbers, where the Kolmogorov length-scale (representing the size of the smallest turbulent structures) is resolved by the computational grid, but the Batchelor scale (the size of the smallest scalar structures) is not.

The low-frequency scalar component will always vary smoothly over the prescribed grid, so that these smooth scalar values can be interpolated on to a sub-grid with a negligible loss of scalar information. Performing a Lagrangian reconstruction at a sub-grid resolution also comes at no additional computational cost. In this way the flow-field can be solved at a resolution that captures the Kolmogorov scale, while the E-L summation can be determined at a resolution that captures the Batchelor scale, with the only additional computational cost coming from the interpolation of the Eulerian information on to the sub-grid. For cases where the Batchelor scale must be resolved and is smaller than the Kolmogorov scale, this is potentially more efficient than solving both the flow and the scalar field at a Batchelor-scale resolution (as would be required in an Eulerian-only simulation). The memory requirement of the sub-grid reconstruction approach is significantly less than for solving all quantities at the Batchelor scale, as only the scalar field would need to be stored at the higher resolution; the time-step width would also be restricted only by the resolution of the flow-field grid.

A possible disadvantage of this approach is that an increased Lagrangian particle density would be required to maintain the same level of accuracy in the sub-grid reconstruction as in a Lagrangian

reconstruction on the regular grid. For example, if the sub-grid was more finely-resolved than the computational grid by a factor of 2, the particle density would need to be increased by a factor of 8 (in three dimensions). However, this does not take into account the potential of the localised low-pass filtering algorithm (which would be applied on the sub-grid) for improving the accuracy of the particle reconstruction, such that it may be acceptable to increase the Lagrangian density by a factor of only 4, perhaps. The extra computational cost of using more particles should still be less than the cost of performing the entire calculation at the sub-grid resolution, which would be expected to suffer a 16× increase in computational cost (for a doubling of the spatial resolution in three dimensions, and a corresponding reduction in time-step width to maintain a constant CFL number).

12.3.4 The Eulerian-Lagrangian Method for Density Transport

In the studies presented in this work the transported scalar quantity has been assumed to be passive, i.e. its distribution has no influence upon the development of the flow-field. Future development should aim towards the application of the E-L method for active scalar quantities, for example in the transport of density (and subsequently, for scalar quantities involved in a chemical reaction). Applying the Eulerian-Lagrangian decomposition to the mass conservation (continuity) equation (Eq. 2.3) yields:

$$\frac{\partial \rho^E}{\partial t} + \frac{\partial}{\partial x_i}(\rho^E u_i) = 0 \tag{12.3}$$

$$\frac{\partial \rho^L}{\partial t} + \frac{\partial}{\partial x_i}(\rho^L u_i) = 0 \tag{12.4}$$

Where the transported density field is subsequently recovered from:

$$\rho = \rho^E + \rho^L \tag{12.5}$$

Equations 12.3 and 12.4 may be applied directly in the context of DNS, or may be manipulated for application in LES following a similar procedure to that of §5.2.1. A specific point of interest is the relatively poor prediction of vertical velocity fluctuations in the Turbulent Opposed Jet (TOJ) simulation of Chapter 11, as observed in Figs. 11.3 and 11.5, where the density field is transported using a traditional TVD scheme. It may be possible that the under-prediction of these velocity fluctuations is related to the under-prediction of scalar fluctuations shown in Fig. 11.10, and that the application of the E-L method for density transport in the TOJ case would improve the prediction of the flow-field in the region of the stagnation plane. It should be noted, however, that in cases where density is transported by the E-L method the noise in the Lagrangian reconstruction may affect the solution, as localised changes in gradient due to that noise cause significant pressure corrections and fluid accelerations.

The application of the E-L method may be extended further to the simulation of, for example, non-premixed combustion, where density and mixture fraction fields are both transported in low- and high-frequency parts. The complexity of such a simulation would be increased by the coupling

between the two quantities, and the noise in the Lagrangian reconstruction may have a significant impact upon this. However, no additional equations would need to be solved: it should be sufficient to update the density field using the component continuity equations in Eqs. 12.3 and 12.4, and subsequently the summation in Eq. 12.5, before transporting the mixture fraction field using a relevant set of equations from §5.2.1 (depending on whether a DNS or an LES is being performed). In other words, further transport equations for the combinations of $\rho^E\phi^E$, $\rho^E\phi^L$, $\rho^L\phi^E$ and $\rho^L\phi^L$ should not be required in the context of either DNS or LES.

Bibliography

[1] L. F. Richardson, 'Weather Prediction by Numerical Process', Cambridge University Press, Cambridge (1922)

[2] A. Thom, 'The Flow Past Circular Cylinders at Low Speeds', Proceedings of the Royal Society of London Series A 141 (1933) 651–666

[3] M. Kawaguti, 'Numerical Solution of the NS Equations for the Flow Around a Circular Cylinder at Reynolds Number 40', Journal of the Physical Society of Japan 8 (1953) 747–757

[4] 'Top500 List – November 2012 – TOP500 Supercomputing Sites', accessed 14/11/2012, `http://www.top500.org/list/2012/11/`

[5] J. Dongarra, 'Performance of Various Computers Using Standard Linear Equations Software', Computer Science Technical Report Number CS–89–85, accessed 14/11/2012, `http://www.netlib.org/benchmark/performance.ps`

[6] F. Black, M. Scholes, 'The Pricing of Options and Corporate Liabilities', Journal of Political Economy 81 (1973) 637–654

[7] M. Knudsen, 'The Kinetic Theory of Gases: Some Modern Aspects', Methuen & Co., London (1934)

[8] S. Schaff, P. Chambre, 'Fundamentals of Gas Dynamics', Princeton University Press, Princeton (1958)

[9] O. Aktas, N. R. Aluru, U. Ravaioli, 'Application of a Parallel DSMC Technique to Predict Flow Characteristics in Microfluidic Filters', IEEE Journal of Microelectromechanical Systems 10 (2001) 538–549

[10] Q. Fan, H. Xue, 'Compressible Effects in Microchannel Flows', IEEE/CPMT Electronics Packaging Technology Conference (1998) 224–228

[11] D. Keefer, 'Investigating Landslides Caused by Earthquakes – A Historical Review', Surveys in Geophysics 23 (2002) 473–510

[12] J. Aguirre, K. Irikura, 'Nonlinearity, Liquefaction, and Velocity Variation of Soft Soil Layers in Port Island, Kobe, During the Hyogo-Ken Nanbu Earthquake', Bulletin of the Seismological Society of America 87 (1997) 1244–1258

[13] O. Reynolds, 'An Experimental Investigation of the Circumstances Which Determine Whether the Motion of Water Shall Be Direct or Sinuous, and the Law of Resistance in Parallel Channels', Philosophical Transactions of the Royal Society of London Series A 174 (1883) 935–982

[14] O. Reynolds, 'On the Dynamical Theory of Incompressible Viscous Flows and the Determination of the Criterion', Philosophical Transactions of the Royal Society of London Series A 186 (1894) 123–161

[15] A. N. Kolmogorov, 'The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers', Doklady Akademii Nauk SSSR 30 (1941) 299–303

[16] K. R. Sreenivasan, 'On the Universality of the Kolmogorov Constant', Physics of Fluids 7 (1995) 2778–2784

[17] A. N. Kolmogorov, 'Dissipation of Energy in Local Isotropic Turbulence', Doklady Akademii Nauk SSSR 32 (1941) 19–21

[18] J. Warnatz, U. Maas, R. W. Dibble, 'Combustion: Physical and Chemical Fundamentals, Modelling and Simulation, Experiments, Pollutant Formation (4th Edition)', Springer-Verlag, Berlin (2006)

[19] W. P. Jones, B. E. Launder, 'The Prediction of Laminarization with a Two-Equation Model of Turbulence', International Journal of Heat and Mass Transfer 15 (1972) 301–314

[20] F. Williams, 'Turbulent Mixing in Nonreactive and Reactive Flows', in: S. Murthy (Editor), 'Recent Advances in Theoretical Descriptions of Turbulent Diffusion Flames', Plenum Press, New York (1975) 189–208

[21] C. Dopazo, E. E. O'Brien, 'Functional Formulation of Non-Isothermal Turbulent Reactive Flows', Physics of Fluids 17 (1974) 1968–1975

[22] S. B. Pope, 'PDF Methods for Turbulent Reactive Flows', Progress in Energy and Combustion Science 11 (1985) 119–192

[23] W. Kollmann, 'The PDF Approach to Turbulent Flow', Theoretical and Computational Fluid Dynamics 1 (1990) 249–285

[24] S. B. Pope, 'Lagrangian PDF Methods for Turbulent Flows', Annual Review of Fluid Mechanics 26 (1994) 23–63

[25] A. Y. Klimenko, R. W. Bilger, 'Conditional Moment Closure for Turbulent Combustion', Progress in Energy and Combustion Science 25 (1999) 595–687

[26] S. H. Kim, K. Y. Huh, L. Tao, 'Application of the Elliptic Conditional Moment Closure Model to a Two-Dimensional Non-Premixed Methanol Bluff-Body Flame', Combustion and Flame 120 (2000) 75–90

[27] A. Kronenburg, R. W. Bilger, 'Modelling Differential Diffusion in Non-Premixed Turbulent Reacting Flow: Model Development', Combustion Science and Technology 166 (2001) 195–227

[28] A. Kronenburg, R. W. Bilger, J. H. Kent, 'Second Order Conditional Moment Closure for Turbulent Jet Diffusion Flames', Proceedings of the Combustion Institute 27 (1998) 1097–1104

[29] E. Mastorakos, R. W. Bilger, 'Second-Order Conditional Moment Closure for the Autoignition of Turbulent Flows', Physics of Fluids 10 (1998) 1246–1248

[30] A. Y. Klimenko, S. B. Pope, 'The Modelling of Turbulent Reactive Flows Based on Multiple Mapping Conditioning', Physics of Fluids 15 (2003) 1907–1925

[31] A. Y. Klimenko, 'Matching Conditional Moments in PDF Modelling of Non-Premixed Combustion', Combustion and Flame 143 (2005) 369–385

[32] A. P. Wandel, A. Y. Klimenko, 'Testing Multiple Mapping Conditioning Mixing for Monte-Carlo Probability Distribution Function Methods', Physics of Fluids 17 (2005) 128105

[33] M. J. Cleary, A. Kronenburg, 'Multiple Mapping Conditioning for Extinction and Re-Ignition in Turbulent Diffusion Flames', Proceedings of the Combustion Institute 31 (2007) 1497–1505

[34] M. J. Cleary, A. Y. Klimenko, 'A Generalised Multiple Mapping Conditioning Approach for Turbulent Combustion', Flow, Turbulence and Combustion 82 (2009) 477–491

[35] P. Vaishnavi, A. Kronenburg, 'Multiple Mapping Conditioning of Velocity in Turbulent Jet Flames', Combustion and Flame 157 (2010) 1863–1865

[36] K. Vogiatzaki, A. Kronenburg, M. J. Cleary, J. H. Kent, 'Multiple Mapping Conditioning of Turbulent Jet Diffusion Flames', Proceedings of the Combustion Institute 32 (2009) 1679–1685

[37] E. R. Hawkes, R. Sankaran, J. H. Chen, S. A. Kaiser, J. H. Frank, 'An Analysis of Lower-Dimensional Approximations to the Scalar Dissipation Rate Using Direct Numerical Simulations of Plane Jet Flames', Proceedings of the Combustion Institute 32 (2009) 1455–1463

[38] P. Domingo, L. Vervisch, 'DNS of Partially Premixed Flame Propagating in a Turbulent Rotating Flow', Proceedings of the Combustion Institute 31 (2007) 1657–1664

[39] 'Top500 List – November 2010 – TOP500 Supercomputing Sites', accessed 02/12/2010, `http://www.top500.org/list/2010/11/`

[40] J. W. Deardorff, 'A Numerical Study of Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers', Journal of Fluid Mechanics 41 (1970) 453–480

[41] U. Schumann, 'Subgrid Scale Model for Finite Difference Simulations of Turbulent Flow in Plane Channels and Annuli', Journal of Computational Physics 18 (1975) 376–404

[42] A. Leonard, 'Energy Cascade in Large-Eddy Simulations of Turbulent Fluid Flows', Advances in Geophysics A 18 (1974) 237–248

[43] B. Vreman, B. Geurts, H. Kuerten, 'Subgrid-Modelling in LES of Compressible Flow', Applied Scientific Research 54 (1995) 191–203

[44] C. Mengler, 'Grobstruktursimulation der Strömungs- und Mischungsfelder komplexer, anwendungsnaher Konfigurationen', PhD Thesis, Technischen Universität Darmstadt (2001)

[45] O. Colin, F. Ducros, D. Veynante, T. Poinsot, 'A Thickened Flame Model for Large Eddy Simulation of Turbulent Premixed Combustion', Physics of Fluids 12 (2000) 1843–1863

[46] C. Fureby, 'A Fractal Flame-Wrinkling Large Eddy Simulation Model for Premixed Turbulent Combustion', Proceedings of the Combustion Institute 30 (2005) 593–601

[47] F. Charlette, C. Meneveau, D. Veynante, 'A Power-Law Flame Wrinkling Model for LES of Premixed Turbulent Combustion. Part I: Non-Dynamic Formulation and Initial Tests', Combustion and Flame 131 (2002) 159–180

[48] Ö. L. Gülder, 'Turbulent Premixed Flame Propagation Models for Different Combustion Regimes', Proceedings of the Combustion Institute 23 (1990) 835–842

[49] R. Knikker, D. Veynante, C. Meneveau, 'A Dynamic Flame Surface Density Model for Large Eddy Simulation of Turbulent Premixed Combustion', Physics of Fluids 16 (2004) 91–94

[50] H. G. Weller, G. Tabor, A. D. Gosman, C. Fureby, 'Application of a Flame-Wrinkling LES Combustion Model to a Turbulent Mixing Layer', Proceedings of the Combustion Institute 27 (1998) 899–907

[51] A. R. Kerstein, 'Linear-Eddy Model of Turbulent Scalar Transport and Mixing', Combustion Science and Technology 60 (1988) 391–421

[52] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Transport. Part 2: Application to Shear Layer Mixing', Combustion and Flame 75 (1989) 397–413

[53] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Transport. Part 3: Mixing and Differential Molecular Diffusion in Round Jets', Journal of Fluid Mechanics 216 (1990) 411–435

[54] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Scalar Transport. Part 4: Structure of Diffusion Flames', Combustion Science and Technology 81 (1992) 75–96

[55] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Transport. Part 5: Geometry of Scalar Interfaces', Physics of Fluids 3 (1991) 1110–1114

[56] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Transport. Part 6: Microstructure of Diffusive Scalar and Mixing Fields', Journal of Fluid Mechanics 231 (1991) 361–394

[57] A. R. Kerstein, 'Linear-Eddy Modelling of Turbulent Transport. Part 7: Finite-Rate Chemistry and Multi-Stream Mixing', Journal of Fluid Mechanics 240 (1992) 289–313

[58] O. Stein, 'Large Eddy Simulation of Combustion in Swirling and Opposed Jet Flows', PhD Thesis, Imperial College London (2009)

[59] I. Celik, Z. Cehreli, I. Yavuz, 'Index of Resolution Quality for Large Eddy Simulations', Journal of Fluids Engineering 127 (2005) 949–958

[60] M. Klein, 'An Attempt to Assess the Quality of LES in the Context of Implicit Filtering', Flow, Turbulence and Combustion 75 (2005) 131–147

[61] A. M. Kempf, B. J. Geurts, T. Ma, M. W. A. Pettit, O. T. Stein, 'Quality Issues in Combustion LES', Journal of Scientific Computing 49 (2011) 51–64

[62] B. J. Geurts, J. Fröhlich, 'A Framework for Predicting Accuracy Limitations in Large-Eddy Simulation', Physics of Fluids 14 (2002) L41

[63] J. Meyers, B. J. Geurts, M. Baelmans, 'Database-Analysis of Errors in Large-Eddy Simulations', Physics of Fluids 15 (2003) 2740–2755

[64] J. Floyd, A. M. Kempf, A. Kronenburg, R. H. Ram, 'A Simple Model for the Filtered Density Function for Passive Scalar Combustion LES', Combustion Theory Modelling 13 (2009) 559–588

[65] J. Williamson, 'Low-Storage Runge-Kutta Schemes', Journal of Computational Physics 35 (1980) 48–56

[66] J. C. Butcher, 'Numerical Methods for Ordinary Differential Equations (2nd Edition)', Wiley, Chichester (2003)

[67] R. Courant, K. Friedrichs, H. Lewy, 'Über die Partiellen Differenzengleichungen der Mathematischen Physik', Mathematische Annalen 100 (1928) 32–74

[68] C.-W. Shu, S. Osher, 'Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes', Journal of Computational Physics 77 (1988) 439–471

[69] M. H. Carpenter, C. A. Kennedy, 'Fourth-Order 2N-Storage Runge-Kutta Schemes', NASA (1994) TM–109112

[70] E. Isaacson, H. B. Keller, 'Analysis of Numerical Methods', Dover, New York (1966)

[71] J. H. Ferziger, M. Perić, 'Computational Methods for Fluid Dynamics (Vol. 2)', Springer-Verlag, Berlin (1999)

[72] S. V. Patankar, D. B. Spalding, 'A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows', International Journal of Heat and Mass Transfer 15 (1972) 1787–1806

[73] S. V. Patankar, 'Numerical Heat Transfer and Fluid Flow', Taylor & Francis, New York (1980)

[74] A. M. Kempf, 'Large-Eddy Simulation of Non-Premixed Turbulent Flames', PhD Thesis, Technischen Universität Darmstadt (2003)

[75] J. Boussinesq, 'Essai sur la Theorie des Eaux Courantes', Memoires de l'Academie Scientifique Institut National de France 23 (1877) 680

[76] J. S. Smagorinsky, 'General Circulation Experiments with the Primitive Equations, 1. The Basic Experiment', Monthly Weather Review 91 (1963) 99–164

[77] D. K. Lilly, 'On the Application of the Eddy Viscosity Concept in the Inertial Subrange of Turbulence', National Center for Atmospheric Research (1966) Manuscript 123

[78] J. W. Deardorff, 'On the Magnitude of the Subgrid Scale Eddy Coefficient', Journal of Computational Physics 7 (1971) 120–133

[79] M. Germano, U. Piomelli, P. Moin, W. H. Cabot, 'A Dynamic Subgrid-Scale Eddy Viscosity Model', Physics of Fluids 3 (1991) 1760–1765

[80] U. Piomelli, J. Liu, 'Large-Eddy Simulation of Rotating Channel Flows Using a Localized Dynamic Model', Physics of Fluids 7 (1995) 839–848

[81] M. Germano, 'Turbulence: The Filtering Approach', Journal of Fluid Mechanics 238 (1992) 325–336

[82] U. Piomelli, W. H. Cabot, P. Moin, S. Lee, 'Subgrid-Scale Backscatter in Turbulent and Transitional Flows', Physics of Fluids 3 (1991) 1766–1771

[83] F. Nicoud, F. Ducros, 'Subgrid-Scale Stress Modelling Based on the Square of the Velocity Gradient Tensor', Flow, Turbulence and Combustion 62 (1999) 183–200

[84] L. Selle, G. Lartigue, T. Poinsot, R. Koch, K. U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, D. Veynante, 'Compressible Large Eddy Simulation of Turbulent Combustion in Complex Geometry on Unstructured Meshes', Combustion and Flame 137 (2004) 489–505

[85] S. Roux, G. Lartigue, T. Poinsot, U. Meier, C. Bérat, 'Studies of Mean and Unsteady Flow in a Swirled Combustor Using Experiments, Acoustic Analysis, and Large Eddy Simulations', Combustion and Flame 141 (2005) 40–54

[86] J. Fröhlich, C. P. Mellen, W. Rodi, L. Temmerman, M. A. Leschziner, 'Highly Resolved Large-Eddy Simulation of Separated Flow in a Channel with Streamwise Periodic Constrictions', Journal of Fluid Mechanics 526 (2005) 19–66

[87] H. Baya Toda, K. Truffin, F. Nicoud, 'Is the Dynamic Procedure Appropriate for All SGS Models?', Fifth European Conference on Computational Fluid Dynamics (2010)

[88] D. K. Lilly, 'A Proposed Modification of the Germano Subgrid-Scale Closure Method', Physics of Fluids 4 (1992) 633–635

[89] J. Bardina, J. H. Ferziger, W. C. Reynolds, 'Improved Turbulence Models Based on Large Eddy Simulation of Homogeneous, Incompressible, Turbulent Flows', PhD Thesis, Stanford University (1983)

[90] B. Zhou, G. Cui, N. Chen, 'A Dynamic Procedure Based on the Scale-Similarity Hypotheses for Large-Eddy Simulation', Frontiers of Energy and Power Engineering in China 1 (2007) 468–472

[91] F. Nicoud, H. Baya Toda, O. Cabrit, S. Bose, J. Lee, 'Using Singular Values to Build a Subgrid-Scale Model for Large Eddy Simulations', Physics of Fluids 23 (2011) 085106

[92] T. M. Eidson, 'Numerical Simulation of the Turbulent Rayleigh-Benard Problem Using Subgrid Modelling', Journal of Fluid Mechanics 158 (1985) 245–268

[93] P. J. Mason, 'Large-Eddy Simulation of the Convective Atmospheric Boundary Layer', Journal of the Atmospheric Sciences 46 (1989) 1492–1516

[94] S. Liu, C. Meneveau, J. Katz, 'On the Properties of Similarity Subgrid-Scale Models as Deduced from Measurements in a Turbulent Jet', Journal of Fluid Mechanics 275 (1994) 83–119

[95] F. Sarghini, U. Piomelli, E. Balaras, 'Scale-Similar Models for Large-Eddy Simulations', Physics of Fluids 11 (1999) 1596–1607

[96] S. Liu, C. Meneveau, J. Katz, 'Experimental Study of Similarity Subgrid-Scale Models of Turbulence in the Far-Field of a Jet', Journal of Applied Sciences Research 54 (1995) 177–190

[97] F. Porté-Agel, M. B. Parlange, C. Meneveau, W. E. Eichinger, 'A Priori Field Study of the Subgrid-Scale Heat Fluxes and Dissipation in the Atmospheric Surface Layer', Journal of the Atmospheric Sciences 58 (2001) 2673–2698

[98] M. Klein, A. Sadiki, J. Janicka, 'A Digital Filter Based Generation of Inflow Data for Spatially Developing Direct Numerical or Large Eddy Simulations', Journal of Computational Physics 186 (2003) 652–665

[99] O. Stein, A. M. Kempf, J. Janicka, 'LES of the Sydney Swirl Flame Series: An Initial Investigation of the Fluid Dynamics', Combustion Science and Technology 179 (2007) 173–189

[100] M. W. A. Pettit, B. Coriton, A. Gomez, A. M. Kempf, 'Large-Eddy Simulation and Experiments on Non-Premixed Highly Turbulent 'Opposed Jet' Flows', Proceedings of the Combustion Institute 33 (2011) 1391–1399

[101] J. Janicka, A. Sadiki, 'Large Eddy Simulation of Turbulent Combustion Systems', Proceedings of the Combustion Institute 30 (2005) 537–547

[102] W. Sander, B. Weigand, 'Direct Numerical Simulation and Analysis of Instability Enhancing Parameters in Liquid Sheets at Moderate Reynolds Numbers', Physics of Fluids 20 (2008) 053301

[103] D. Ahlman, G. Velter, G. Brethouwer, A. V. Johansson, 'DNS of Non-Isothermal Turbulent Wall Jets', Physics of Fluids 21 (2009) 035101

[104] M. W. Renfro, A. Chaturvedy, G. B. King, N. M. Laurendeau, A. M. Kempf, A. Dreizler, J. Janicka, 'Comparison of OH Time-Series Measurements and Large-Eddy Simulations in Hydrogen Jet Flames', Combustion and Flame 139 (2004) 142–151

[105] A. M. Kempf, M. Klein, J. Janicka, 'Efficient Generation of Initial and Inflow Conditions for Transient Turbulent Flows in Arbitrary Geometries', Flow, Turbulence and Combustion 74 (2005) 67–84

[106] A. M. Kempf, S. Wysocki, M. W. A. Pettit, 'An Efficient, Parallel Low-Storage Implementation of Kleins Turbulence Generator for LES and DNS', Computers & Fluids 60 (2012) 58–60

[107] G. Batchelor, 'The Theory of Homogeneous Turbulence', Cambridge University Press, Cambridge (1953)

[108] F. Cavallo Marincola, T. Ma, A. M. Kempf, 'Large Eddy Simulations of the Darmstadt Turbulent Stratified Flame Series', Proceedings of the Combustion Institute 34 (2013) 1307–1315

[109] B. M. Franchetti, F. Cavallo Marincola, S. Navarro-Martinez, A. M. Kempf, 'Large Eddy Simulation of a Pulverised Coal Jet Flame', Proceedings of the Combustion Institute 34 (2013) 2419–2426

[110] H. K. Versteeg, W. Malalasekera, 'An Introduction to Computational Fluid Dynamics: The Finite Volume Method (2nd Edition)', Prentice-Hall, Harlow (2007)

[111] T. K. Sengupta, G. Ganerwal, A. Dipankar, 'High Accuracy Compact Schemes and Gibbs Phenomenon', Journal of Scientific Computing 21 (2004) 253–268

[112] B. P. Leonard, 'A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation', Computer Methods in Applied Mechanics and Engineering 19 (1979) 59–98

[113] C. Hirsch, 'Numerical Computation of Internal and External Flows: Computational Methods for Inviscid and Viscous Flows', Wiley, Chichester (1990)

[114] B. Fornberg, 'Generation of Finite Difference Formulas of Arbitrarily Spaced Grids', Mathematics of Computation 51 (1988) 699–706

[115] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, 'Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III', Journal of Computational Physics 71 (1987) 231–303

[116] X.-D. Liu, S. Osher, T. Chan, 'Weighted Essentially Non-Oscillatory Schemes', Journal of Computational Physics 115 (1994) 200–212

[117] C.-W. Shu, 'High Order Weighted Essentially Non-Oscillatory Schemes for Convection Dominated Problems', Society for Industrial and Applied Mathematics Review 51 (2009) 82–126

[118] A. Harten, 'High Resolution Schemes for Hyperbolic Conservation Laws', Journal of Computational Physics 135 (1997) 260–278

[119] B. Van Leer, 'Towards the Ultimate Conservative Difference Scheme V. A Second-Order Sequel to Godunov's Method', Journal of Computational Physics 32 (1979) 101–136

[120] N. P. Waterson, H. Deconinck, 'Design Principles for Bounded Higher-Order Convection Schemes – a Unified Approach', Journal of Computational Physics 224 (2007) 182–207

[121] P. K. Sweby, 'High Resolution Schemes Using Flux-Limiters for Hyperbolic Conservation Laws', Society for Industrial and Applied Mathematics Journal on Numerical Analysis 21 (1984) 995–1011

[122] B. Koren, 'A Robust Upwind Discretisation Method for Advection, Diffusion and Source Terms', in: C. B. Vreugdenhil, B. Koren (Editors), 'Numerical Methods for Advection–Diffusion Problems', Vieweg, Wiesbaden (1993) 117–138

[123] P. L. Roe, 'Characteristic-Based Schemes for the Euler Equations', Annual Review of Fluid Mechanics 18 (1986) 337–365

[124] B. Van Leer, 'Towards the Ultimate Conservative Difference Scheme III. Upstream-Centred Finite-Difference Schemes for Ideal Compressible Flow', Journal of Computational Physics 23 (1977) 263–275

[125] S. R. Chakravarthy, S. Osher, 'High Resolution Applications of the Osher Upwind Scheme for the Euler Equations', Proceedings of the American Institute of Aeronautics and Astronautics 6th Computational Fluid Dynamics Conference (1983) 363–373

[126] N. P. Waterson, H. Deconinck, 'A Unified Approach to the Design and Application of Bounded Higher-Order Convection Schemes', in: C. Taylor, P. Durbetaki (Editors), 'Proceedings of the Ninth International Conference on Numerical Methods in Laminar and Turbulent Flow', Pineridge Press, Swansea (1995) 203–214

[127] F. S. Lien, M. A. Leschziner, 'Upstream Monotonic Interpolation for Scalar Transport with Application to Complex Turbulent Flows', International Journal for Numerical Methods in Fluids 19 (1994) 527–548

[128] G. D. Van Albada, B. Van Leer, W. W. Roberts, 'A Comparative Study of Computational Methods in Cosmic Gas Dynamics', Astronomy & Astrophysics 108 (1982) 76–84

[129] B. Van Leer, 'Towards the Ultimate Conservative Difference Scheme II. Monotonicity and Conservation Combined in a Second Order Scheme', Journal of Computational Physics 14 (1974) 361–370

[130] G. Zhou, 'Numerical Simulations of Physical Discontinuities in Single and Multi-Fluid Flows for Arbitrary Mach Mumbers', PhD Thesis, Chalmers University of Technology, Goteborg, Sweden (1995)

[131] P. H. Gaskell, A. K. C. Lau, 'Curvature-Compensated Convective Transport: SMART, a New Boundedness-Preserving Transport Algorithm', International Journal for Numerical Methods in Fluids 8 (1988) 617–641

[132] F. H. Harlow, 'PIC Method for Fluid Dynamics Calculations', Los Alamos Scientific Laboratory Report (1961) 5170

[133] F. H. Harlow, 'The Particle-in-Cell Method for Numerical Solution of Problems in Fluid Dynamics', Los Alamos Scientific Laboratory Report (1962) 5288

[134] F. H. Harlow, 'The Particle-in-Cell Computing Method for Fluid Dynamics', Methods in Computational Physics 3 (1964) 319–343

[135] F. H. Harlow, 'PIC and its Progeny', Computer Physics Communications 48 (1988) 1–10

[136] A. A. Amsden, F. H. Harlow, 'Numerical Calculation of Supersonic Wake Flow', Journal of the American Institute of Aeronautics and Astronautics 3 (1965) 2081–2086

[137] D. M. Snider, 'An Incompressible Three-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows', Journal of Computational Physics 170 (2001) 523–549

[138] T. L. Cook, R. B. Demuth, F. H. Harlow, 'PlC Calculations of Multiphase Flow', Journal of Computational Physics 41 (1981) 51–67

[139] F. H. Harlow, A. A. Amsden, J. R. Nix, 'Relativistic Fluid Dynamics Calculations with the Particle-in-Cell Technique', Journal of Computational Physics 20 (1976) 119–129

[140] C. K. Birdsall, 'Particle-in-Cell Charged-Particle Simulations, plus Monte Carlo Collisions with Neutral Atoms, PIC-MCC', IEEE Transactions on Plasma Science 19 (1991) 65–85

[141] C. K. Birdsall, A. B. Langdon, 'Plasma Physics via Computer Simulation', Taylor & Francis, New York (2004)

[142] D. Sulsky, S.-J. Zhou, H. L. Schreyer, 'Application of a Particle-in-Cell Method to Solid Mechanics', Computer Physics Communications 87 (1995) 236–252

[143] J. U. Brackbill, H. M. Ruppel, 'FLIP: A Method for Adaptively Zoned, Particle-in-Cell Calculations of Fluid Flows in Two Dimensions', Journal of Computational Physics 65 (1986) 314–343

[144] J. U. Brackbill, D. B. Kothe, H. M. Ruppel, 'FLIP: A Low-Dissipation, Particle-in-Cell Method for Fluid Flow', Computer Physics Communications 48 (1988) 25–38

[145] F. H. Harlow, J. E. Welch, 'Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface', Physics of Fluids 8 (1965) 2182–2189

[146] J. E. Welch, F. H. Harlow, J. P. Shannon, B. J. Daly, 'The MAC Method. A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-flow Problems Involving Free Surfaces', Los Alamos Scientific Laboratory Report (1966) 3425

[147] J. C. Martin, W. J. Moyce, 'Part IV. An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane', Philosophical Transactions of the Royal Society of London Series A 224 (1952) 312–324

[148] F. H. Harlow, J. P. Shannon, 'The Splash of a Liquid Drop', Journal of Applied Physics 38 (1967) 3855–3866

[149] T. V. Gerya, D. A. Yuen, 'Characteristics-Based Marker-in-Cell Method with Conservative Finite-Differences Schemes for Modelling Geological Flows with Strongly Variable Transport Properties', Physics of the Earth and Planetary Interiors 140 (2003) 293–318

[150] S. McKee, M. F. Tomé, J. A. Cuminato, A. Castelo, V. G. Ferreira, 'Recent Advances in the Marker and Cell Method', Archives of Computational Methods in Engineering, 11 (2004) 107–142

[151] S. Vincent, G. Balmigère, J. P. Caltagirone, E. Meillot, 'Eulerian-Lagrangian Multiscale Methods for Solving Scalar Equations – Application to Incompressible Two-Phase Flows', Journal of Computational Physics 229 (2010) 73–106

[152] G. E. Moore, 'Cramming More Components onto Integrated Circuits', Electronics 38 (1965) 114–117

[153] G. A. Bird, 'Approach to Translational Equilibrium in a Rigid Sphere Gas', Physics of Fluids 6 (1963) 1518–1519

[154] G. A. Bird, 'Monte Carlo Simulation of Gas Flows', Annual Review of Fluid Mechanics 10 (1978) 11–31

[155] G. A. Bird, 'Molecular Gas Dynamics and the Direct Simulation of Gas Flows', Oxford University Press, Oxford (1994)

[156] J. Hardy, O. de Pazzis, Y. Pomeau, 'Molecular Dynamics of a Classical Lattice Gas: Transport Properties and Time Correlation Functions', Physical Review A 13 (1976) 1949–1961

[157] U. Frisch, B. Hasslacher, Y. Pomeau, 'Lattice-Gas Automata for the Navier-Stokes Equations', Physical Review Letters 56 (1986) 1505–1508

[158] S. Chen, G. D. Doolen, 'Lattice Boltzmann Method for Fluid Flows', Annual Review of Fluid Mechanics 30 (1998) 329–364

[159] R. P. Lindstedt, S. A. Louloudi, E. M. Váos, 'Joint Scalar Probability Density Function Modelling of Pollutant Formation in Piloted Turbulent Jet Diffusion Flames with Comprehensive Chemistry', Proceedings of the Combustion Institute 28 (2000) 149–156

[160] R. P. Lindstedt, S. A. Louloudi, 'Joint-Scalar Transported PDF Modelling of Soot Formation and Oxidation', Proceedings of the Combustion Institute 30 (2005) 775–783

[161] W. P. Jones, S. Navarro-Martinez, 'Numerical Study of n-Heptane Auto-Ignition Using LES-PDF Methods', Flow Turbulence and Combustion 83 (2009) 407–423

[162] S. Viswanathan, H. Wang, S. B. Pope, 'Numerical Implementation of Mixing and Molecular Transport in LES/PDF Studies of Turbulent Reacting Flows', Journal of Computational Physics 230 (2011) 6916–6957

[163] M. J. Andrews, P. J. O'Rourke, 'The Multiphase Particle-in-Cell (MP-PIC) Method for Dense Particulate Flows', International Journal of Multiphase Flow 22 (1996) 379–402

[164] D. M. Snider, P. J. O'Rourke, M. J. Andrews, 'Sediment Flow in Inclined Vessels Calculated Using a Multiphase Particle-in-Cell Model for Dense Particle Flows', International Journal of Multiphase Flow 24 (1998) 1359–1382

[165] N. A. Patankar, D. D. Joseph, 'Modelling and Numerical Simulation of Particulate Flows by the Eulerian-Lagrangian Approach', International Journal of Multiphase Flow 27 (2001) 1659–1684

[166] N. A. Patankar, D. D. Joseph, 'Lagrangian Numerical Simulation of Particulate Flows', International Journal of Multiphase Flow 27 (2001) 1685–1706

[167] F. C. Lockwood, A. P. Salooja, S. A. Syed, 'A Prediction Method for Coal-Fired Furnaces', Combustion and Flame 38 (1980) 1–15

[168] D. M. Snider, S. M. Clark, P. J. O'Rourke, 'Eulerian-Lagrangian Method for Three-Dimensional Thermal Reacting Flow with Application to Coal Gasifiers', Chemical Engineering Science 66 (2011) 1285–1295

[169] W. F. Noh, 'CEL: A Time-Dependent Two-Space Dimensional Coupled Eulerian-Lagrangian Code', in: B. Alder, S. Fernbach, M. Rotenberg (Editors), 'Methods in Computational Physics, Volume 3: Fundamental Methods in Hydrodynamics', Academic Press, New York (1964) 117–179

[170] R. M. Franck, R. B. Lazarus, 'Mixed Eulerian-Lagrangian Method', in: B. Alder, S. Fernbach, M. Rotenberg (Editors), 'Methods in Computational Physics, Volume 3: Fundamental Methods in Hydrodynamics', Academic Press, New York (1964) 47–67

[171] J. G. Trulio, 'Theory and Structure of the AFTON Codes', Air Force Weapons Laboratory Technical Report, Kirtland Air Force Base (1966) AFWL-TR-66-19

[172] C. W. Hirt, A. A. Amsden, J. L. Cook, 'An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds', Journal of Computational Physics 14 (1974) 227–253

[173] T. J. R. Hughes, W. K. Liu, T. K. Zimmerman, 'Lagrangian-Eulerian Finite Element Formulation for Viscous Flows', Computer Methods in Applied Mechanics and Engineering 29 (1981) 329–349

[174] T. Belytschko, D. F. Flanagan, J. M. Kennedy, 'Finite Element Method with User-Controlled Meshes for Fluid-Structure Interactions', Computer Methods in Applied Mechanics and Engineering 33 (1982) 689–723

[175] A. Huerta, W. K. Liu, 'Viscous Flows for Large Free Surface Motion', Computer Methods in Applied Mechanics and Engineering 69 (1988) 277–324

[176] M. Souli, J. P. Zolesio, 'Arbitrary Lagrangian-Eulerian and Free Surface Methods in Fluid Mechanics', Computer Methods in Applied Mechanics and Engineering 191 (2001) 451–466

[177] A. Sarthou, S. Vincent, J. P. Caltagirone, P. Angot, 'Eulerian-Lagrangian Grid Coupling and Penalty Methods for the Simulation of Multiphase Flows Interacting with Complex Objects', International Journal for Numerical Methods in Fluids 56 (2008) 1093–1099

[178] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, 'Arbitrary Lagrangian-Eulerian Methods', in: E. Stein, R. de Borst, T. J. R. Hughes (Editors), 'Encyclopedia of Computational Mechanics', Wiley, Chichester (2004) 413–437

[179] R. P. Fedkiw, 'Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method', Journal of Computational Physics 175 (2002) 200–224

[180] D. Enright, R. Fedkiw, J. Ferzinger, I. Mitchell, 'A Hybrid Particle Level Set Method for Improved Interface Capturing', Journal of Computational Physics 183 (2002) 83–116

[181] S. Osher, J. Sethian, 'Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations', Journal of Computational Physics 79 (1988) 12–49

[182] S. Osher, R. Fedkiw, 'Level Set Methods and Dynamic Implicit Surfaces', Springer, New York (2003)

[183] J. Sethian, 'Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts', Journal of Computational Physics 169 (2001) 503–555

[184] S. T. Zalesak, 'Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids', Journal of Computational Physics 31 (1979) 335–362

[185] W. J. Rider, D. B. Kothe, 'Reconstructing Volume Tracking', Journal of Computational Physics 141 (1998) 112–152

[186] D. Enright, F. Losasso, R. Fedkiw, 'A Fast and Accurate Semi-Lagrangian Particle Level Set Method', Computers and Structures 83 (2005) 479–490

[187] S. E. Hieber, P. Koumoutsakos, 'A Lagrangian Particle Level Set Method', Journal of Computational Physics 210 (2005) 342–367

[188] F. Losasso, J. O. Talton, N. Kwatra, R. Fedkiw, 'Two-Way Coupled SPH and Particle Level Set Fluid Simulation', Institute of Electrical and Electronics Engineers Transactions on Visualisation and Computer Graphics 14 (2008) 797–804

[189] D. Gaudlitz, N. A. Adams, 'On Improving Mass-Conservation Properties of the Hybrid Particle-Level-Set Method', Computers & Fluids 37 (2008) 1320–1331

[190] Z. Li, F. A. Jaberi, T. I.-P. Shih, 'A Hybrid Lagrangian-Eulerian Particle-Level Set Method for Numerical Simulations of Two-Fluid Turbulent Flows', International Journal for Numerical Methods in Fluids 56 (2008) 2271–2300

[191] W. H. Press, G. B. Rybicki, 'Enhancement of Passive Diffusion and Suppression of Heat Flux in a Fluid with Time Varying Shear', The Astrophysical Journal 248 (1981) 751–766

[192] A. Einstein, 'Investigations on the Theory of the Brownian Movement', Dover, New York (1956)

[193] A. Kolmogorov, 'Über die Analytischen Methoden in der Wahrscheinlichkeitsrechnung', Mathematische Annalen 104 (1931) 415–458

[194] T. W. Parks, C. S. Burrus, 'Digital Filter Design', Wiley, New York (1987)

[195] N. J. Loy, 'An Engineer's Guide to FIR Digital Filters', Prentice-Hall, New Jersey (1988)

[196] B. Vreman, B. Geurts, H. Kuerten, 'Realizability Conditions for the Turbulent Stress Tensor in Large-Eddy Simulation', Journal of Fluid Mechanics 278 (1994) 351–362

[197] M. Frigo, S. G. Johnson, 'The Design and Implementation of FFTW3', Proceedings of the Institute of Electrical and Electronics Engineers 93 (2005) 216–231

[198] E. Catmull, R. Rom, 'A Class of Local Interpolating Splines', in: R. E. Barnhill, R. F. Riesenfeld (Editors), 'Computer Aided Geometric Design', Academic Press, New York (1974) 317–326

[199] G. E. P. Box, M. E. Muller, 'A Note on the Generation of Random Normal Deviates', The Annals of Mathematical Statistics 29 (1958) 610–611

[200] R. W. Bilger, 'Turbulent Flows with Nonpremixed Reactants', in: P. Libby, F. Williams (Editors), 'Topics in Applied Physics: Turbulent Reacting Flows', Springer, Berlin (1980) 65–113

[201] W. Sutherland, 'The Viscosity of Gases and Molecular Force', Philosophical Magazine (Series 5) 36 (1893) 507–531

[202] M. M. Fyrillas, K. K. Nomura, 'Diffusion and Brownian Motion in Lagrangian Coordinates', The Journal of Chemical Physics 126 (2007) 164510

[203] W. Kinzelbach, G. Uffink, 'The Random Walk Method and Extensions in Groundwater Modelling', in: J. Bear, M. Y. Corapcioglu (Editors), 'Transport Processes in Porous Media', Kluwer, Dordrecht (1991) 761–787

[204] A. Kucukelbir, F. J. Sigworth, H. D. Tagare, 'A Bayesian Adaptive Basis Algorithm for Single Particle Reconstruction', Journal of Structural Biology 179 (2012) 56–67

[205] R. Snieder, 'A Guided Tour of Mathematical Methods for the Physical Sciences', Cambridge University Press, Cambridge (2004)

[206] R. Scardovelli, S. Zaleski, 'Interface Reconstruction with Least-Square Fit and Split Eulerian-Lagrangian Advection', International Journal for Numerical Methods in Fluids 41 (2003) 251–274

[207] D. Hartmann, M. Meinke, W. Schröder, 'Differential Equation Based Constrained Reinitialisation for Level Set Methods', Journal of Computational Physics 227 (2008) 6821–6845

[208] C.-W. Shu, S. Osher, 'Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes II', Journal of Computational Physics 83 (1989) 32–78

[209] P. A. Jansson, R. H. Hunt, E. K. Plyler, 'Resolution Enhancement of Spectra', Journal of the Optical Society of America 60 (1970) 596–599

[210] N. Wiener, 'Extrapolation, Interpolation, and Smoothing of Stationary Time Series', Wiley, New York (1949)

[211] D. L. Donoho, 'Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data', in: I. Daubechies (Editor), 'Different Perspectives on Wavelets (Proceedings of Symposia in Applied Mathematics)', American Mathematical Society, Providence (1994) 173–205

[212] J. B. Bell, P. Colella, H. M. Glaz, 'A Second-Order Projection Method for the Incompressible Navier-Stokes Equations', Journal of Computational Physics 85 (1989) 257–283

[213] O. Ubbink, R. I. Issa, 'A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes', Journal of Computational Physics 153 (1999) 26–50

[214] J. Vanormelingen, E. Van den Bulck, 'Scalar Transport in Plane Mixing Layers', Heat and Mass Transfer 35 (1999) 383–390

[215] G. L. Brown, A. Roshko, 'The Effect of Density Difference on the Turbulent Mixing Layer', Advisory Group for Aerospace Research and Development Conference Proceedings 93 (1971) 23.1–23.11

[216] G. L. Brown, A. Roshko, 'On Density Effects and Large Structure in Turbulent Mixing Layers', Journal of Fluid Mechanics 64 (1971) 775–816

[217] L. P. Bernal, A. Roshko, 'Streamwise Vortex Structure in Plane Mixing Layers', Journal of Fluid Mechanics 170 (1986) 499–525

[218] R. E. Breidenthal, 'Chemically Reacting, Turbulent Shear Layer', Journal of the American Institute of Aeronautics and Astronautics 17 (1979) 310–311

[219] R. E. Breidenthal, 'Structure in Turbulent Mixing Layers and Wakes Using a Chemical Reaction', Journal of Fluid Mechanics 109 (1981) 1–24

[220] D. Papamoschou, A. Roshko, 'The Compressible Turbulent Shear Layer: An Experimental Study', Journal of Fluid Mechanics 197 (1988) 453–477

[221] P. Comte, M. Lesieur, E. Lamballais, 'Large and Smallscale Stirring of Vorticity and a Passive Scalar in a 3D Temporal Mixing Layer', Physics of Fluids 4 (1992) 2761–2778

[222] B. Vreman, B. Geurts, H. Kuerten, 'Large-Eddy Simulation of the Turbulent Mixing Layer', Journal of Fluid Mechanics 339 (1997) 357–390

[223] C. Pantano, S. Sarkar, 'A Study of Compressibility Effects in the High-Speed Turbulent Shear Layer Using Direct Simulation', Journal of Fluid Mechanics 451 (2002) 329–371

[224] C. Pantano, S. Sarkar, F. A. Williams, 'Mixing of a Conserved Scalar in a Turbulent Reacting Shear Layer', Journal of Fluid Mechanics 481 (2003) 291–328

[225] I. Mahle, J. Sesterhenn, R. Friedrich, 'DNS of Compressible Inert and Infinitely Fast Reacting Mixing Layers', in: C. Tropea, S. Jakirlic, H.-J. Heinemann, R. Henke, H. Hönlinger (Editors), 'New Results in Numerical and Experimental Fluid Mechanics VI, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 96', Springer Berlin (2007) 372–380

[226] G. I. Taylor, 'Statistical Theory of Turbulence', Proceedings of the Royal Society of London Series A 151 (1935) 421–444

[227] A. M. Obukhov, 'Structure of the Temperature Field in a Turbulent Flow', Izvestiya Akademii Nauk SSSR 13 (1949) 55–69

[228] S. Corrsin, 'On the Spectrum of Isotropic Temperature Fluctuations in Isotropic Turbulence', Journal of Applied Physics 22 (1951) 469–473

[229] K. R. Sreenivasan, 'The Passive Scalar Spectrum and the Obukhov-Corrsin Constant', Physics of Fluids 8 (1996) 189–196

[230] P. Vaishnavi, A. Kronenburg, C. Pantano, 'On the Spatial Length Scales of Scalar Dissipation in Turbulent Jet Flames', Journal of Fluid Mechanics 596 (2008) 103–132

[231] G. K. Batchelor, 'Small Scale Variation of Convected Quantities Like Temperature in a Turbulent Fluid', Journal of Fluid Mechanics 5 (1959) 113–133

[232] P. Moin, K. Mahesh, 'Direct Numerical Simulation: A Tool in Turbulence Research', Annual Review of Fluid Mechanics 30 (1998) 539–578

[233] R. D. Moser, P. Moin, 'The Effects of Curvature in Wall-Bounded Turbulent Flows', Journal of Fluid Mechanics 175 (1987) 479–510

[234] P. R. Spalart, 'Direct Numerical Simulation of a Turbulent Boundary Layer up to $R_\theta = 1410$', Journal of Fluid Mechanics 187 (1988) 61–98

[235] V. Yakhot, K. R. Sreenivasan, 'Anomalous Scaling of Structure Functions and Dynamic Constraints on Turbulence Simulations', Journal of Statistical Physics 121 (2005) 823–841

[236] D. A. Donzis, P. K. Yeung, K. R. Sreenivasan, 'Dissipation and Enstrophy in Isotropic Turbulence: Resolution Effects and Scaling in Direct Numerical Simulations', Physics of Fluids 20 (2008) 045108

[237] M. Wan, S. Oughton, S. Servidio, W. H. Matthaeus, 'On the Accuracy of Simulations of Turbulence', Physics of Plasmas 17 (2010) 082308

[238] 'Cray XT6m 12-Core 1.9 GHz – TOP500 Supercomputing Sites', accessed 09/03/2013, `http://www.top500.org/system/10552`

[239] S. Sizaret, I. Fedioun, L. Barbanson, Y. Chen, 'Crystallisation in Flow – II. Modelling Crystal Growth Kinetics Controlled by Boundary Layer Thickness', Geophysical Journal International 167 (2006) 1027–1034

[240] R. W. Bilger, 'The Structure of Diffusion Flames', Combustion Science and Technology 13 (1976) 155–170

[241] R. W. Bilger, 'Some Aspects of Scalar Dissipation', Flow, Turbulence and Combustion 72 (2004) 93–114

[242] H. Pitsch, H. Steiner, 'Scalar Mixing and Dissipation Rate in Large-Eddy Simulations of Non-Premixed Turbulent Combustion', Proceedings of the Combustion Institute 28 (2000) 41–49

[243] N. Peters, 'Laminar Diffusion Flamelet Models in Non-Premixed Turbulent Combustion', Progress in Energy and Combustion Science 10 (1984) 319–339

[244] N. Peters, 'Laminar Flamelet Concepts in Turbulent Combustion', Proceedings of the Combustion Institute 21 (1986) 1231–1250

[245] M. Pettit, 'Optimisation and Parallelisation of a Combustion Large-Eddy Simulation Code for High Performance Computation', MSc Dissertation, University of Edinburgh (2010)

[246] G. Coppola, A. Gomez, 'Experimental Investigation on a Turbulence Generation System with High-Blockage Plates', Experimental Thermal and Fluid Science 33 (2009) 1037–1048

[247] G. Coppola, B. Coriton, A. Gomez, 'Highly Turbulent Counterflow Flames: A Laboratory Scale Benchmark for Practical Systems', Combustion and Flame 156 (2009) 1834–1843

[248] L. W. Kostiuk, K. N. C. Bray, T. C. Chew, 'Premixed Turbulent Combustion in Counterflowing Streams', Combustion Science and Technology 64 (1989) 233–241

[249] L. W. Kostiuk, K. N. C. Bray, R. K. Cheng, 'Experimental Study of Premixed Turbulent Combustion in Opposed Streams, Part II – Reacting Flow Field and Extinction', Combustion and Flame 92 (1993) 396–409

[250] E. Mastorakos, A. M. K. P. Taylor, J. H. Whitelaw, 'Extinction and Temperature Characteristics of Turbulent Counterflow Diffusion Flames with Partial Premixing', Combustion and Flame 91 (1992) 40–52

[251] J. C. Vassilicos, J. C. R. Hunt, 'Fractal Dimensions and Spectra of Interfaces with Application to Turbulence', Proceedings of the Royal Society of London Series A 435 (1991) 505–534

[252] B. Mazzi, J. C. Vassilicos, 'Fractal-Generated Turbulence', Journal of Fluid Mechanics 502 (2004) 65–87

[253] D. Hurst, J. C. Vassilicos, 'Scaling and Decay of Fractal-Generated Turbulence', Physics of Fluids 19 (2007) 035103

[254] P. Geipel, K. H. H. Goh, R. P. Lindstedt, 'Fractal-Generated Turbulence in Opposed Jet Flows', Flow, Turbulence and Combustion 85 (2010) 397–419

[255] K. H. H. Goh, P. Geipel, F. Hampp, R. P. Lindstedt, 'Regime Transition from Premixed to Flameless Oxidation in Turbulent JP-10 Flames', Proceedings of the Combustion Institute 34 (2013) 3311–3318

[256] D. Geyer, A. Kempf, A. Dreizler, J. Janicka, 'Turbulent Opposed-Jet Flames: A Critical Benchmark Experiment for Combustion LES', Combustion and Flame 143 (2005) 524–548

[257] D. Geyer, A. Kempf, A. Dreizler, J. Janicka, 'Scalar Dissipation Rates in Isothermal and Reactive Turbulent Opposed-Jets: 1-D-Raman/Rayleigh Experiments Supported by LES', Proceedings of the Combustion Institute 30 (2005) 681–689

[258] B. Böhm, O. Stein, A. Kempf, A. Dreizler, 'In-Nozzle Measurements of a Turbulent Opposed Jet Using PIV', Flow, Turbulence and Combustion 85 (2010) 73–93

[259] D.K. Lilly, 'The Representation of Small-Scale Turbulence in Numerical Simulation Experiments', Proceedings of the IBM Scientific Computing Symposium on Environmental Sciences (1967) 195–210

[260] S. B. Pope, 'Turbulent Flows', Cambridge University Press, Cambridge (2000)

[261] E. Mastorakos, A. M. K. P. Taylor, J. H. Whitelaw, 'Mixing in Turbulent Opposed Jet Flows', in: F. Durst, N. Kasagi, B. E. Launder, F. W. Schmidt, K. Suzuki, J. H. Whitelaw (Editors), 'Turbulent Shear Flows 9', Springer-Verlag, New York (1994) 147–164

[262] G. P. Romano, N. T. Ouellette, H. Xu, E. Bodenschatz, V. Steinberg, C. Meneveau, J. Katz, 'Measurements of Turbulent Flows', in: C. Tropea, A. L. Yarin, J. F. Foss (Editors), 'Springer Handbook of Experimental Fluid Mechanics (Volume 1)', Springer-Verlag, Berlin (2007) 745–849

[263] J. P. Longwell, E. E. Frost, M. A. Weiss, 'Flame Stability in Bluff Body Recirculation Zones', Industrial and Engineering Chemistry 45 (1953) 1629–1633

[264] D. B. Spalding, 'Theoretical Aspects of Flame Stabilization: An Approximate Graphical Method for the Flame Speed of Mixed Gases', Aircraft Engineering and Aerospace Technology 25 (1953) 264–276

[265] B. B. Dally, D. F. Fletcher, A. R. Masri, 'Flow and Mixing Fields of Turbulent Bluff-Body Jets and Flames', Combustion Theory and Modelling 2 (1998) 193–219

[266] B. B. Dally, A. R. Masri, R. S. Barlow, G. J. Fiechtner, 'Instantaneous and Mean Compositional Structure of Bluff-Body Stabilized Nonpremixed Flames', Combustion and Flame 114 (1998) 119–148

[267] A. Uppal, W. H. Ray, A. B. Poore, 'On the Dynamic Behavior of Continuous Stirred Tank Reactors', Chemical Engineering Science 29 (1974) 967–985

[268] K. V. Sharp, R. J. Adrian, 'PIV Study of Small-Scale Flow Structure Around a Rushton Turbine', American Institute of Chemical Engineers Journal 47 (2001) 766–778

[269] V. V. Ranade, M. Perrard, N. Le Sauze, C. Xuereb, J. Bertrand, 'Trailing Vortices of Rushton Turbine: PIV Measurements and CFD Simulations with Snapshot Approach', Chemical Engineering Research and Design 79 (2001) 3–12

[270] J. Derksen, H. E. A. Van den Akker, 'Large Eddy Simulations on the Flow Driven By a Rushton Turbine', American Institute of Chemical Engineers Journal 45 (1999) 209–221

[271] J. Aubin, D. F. Fletcher, C. Xuereb, 'Modeling Turbulent Flow in Stirred Tanks with CFD: The Influence of the Modeling Approach, Turbulence Model and Numerical Scheme', Experimental Thermal and Fluid Science 28 (2004) 431–445

[272] D. A. Deglon, C. J. Meyer, 'CFD Modelling of Stirred Tanks: Numerical Considerations', Minerals Engineering 19 (2006) 1059–1068