

Joint Euler-Lagrange Method for Moving Surfaces in Large-Eddy Simulation

Stefan Wysocki
MEng, MSc

Thesis submitted for the degree of
Doctor of Philosophy
of
Imperial College London

Department of Mechanical Engineering
Imperial College London
South Kensington Campus
London, SW7 2AZ
United Kingdom

August 2012

Anyone who has never made a mistake has never tried anything new
Albert Einstein

Declaration Of Originality

I, Stefan Wysocki, declare that the thesis entitled '*Joint Euler-Lagrange Method for Moving Surfaces in Large-Eddy Simulation*' and the work presented in the thesis is my own, and has been generated by me as the result of my own original research. It has also not been submitted in any other form for another degree or diploma. Information derived from published or unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.

Stefan Wysocki, August 2012

Abstract

Continuous growth of computing power strongly encourages engineers to rely more on computational fluid dynamics for the design and testing of new technological solutions. The fast development of these new tools goes along with the increasing availability of high-performance computers, which are necessary to simulate realistic industrial applications. The presented *immersed boundary* (IB) method is applicable to simple and complex geometries with static and moving boundaries, where fluids interact with the solid structures. The formulation of the method is based on the Eulerian and Lagrangian principles and its key characteristics are its simple formulation and computational efficiency. Furthermore the nature of the method allows the simulations of flows in complex geometries without having to generate complex meshes. The spatial discretization is based on a fixed Cartesian mesh for the Eulerian variables and boundary movements are tracked with Lagrangian particles. Large-Eddy simulations of flows in simple and complex geometries demonstrate the performance of the applied immersed boundary method. Simple cases include the simulation of an isothermal pipe flow and the flow around a sphere. In the first instance, the fluid flows around a static sphere. In the second case the sphere moves relative to the grid for identical flow conditions. Simulations of complex geometries include the investigation of an isothermal and reactive opposed jet flow with perforated and fractal grids. The simulations require cell sizes near the resolution of direct numerical simulations. The injection phase of a piston-cylinder arrangement, assuming constant pressure, is also investigated with the proposed IB method. Good statistical results for first and second moments are achieved for all investigated cases, although the applied grids have to be fine enough to accurately resolve the wall shear stresses. In addition, the concept of using Lagrangian particles has been applied to immiscible flows. Particles are used to improve the accuracy of scalar transport and initial results of simple, two-dimensional test cases are presented.

Acknowledgements

I would like to thank my first supervisor Prof. Kempf for his continuous support and guidance at Imperial College London and after his move to the Universität Duisburg-Essen. He also deserves my gratitude for ensuring my financial support from the EPSRC. I would also like to express my gratitude to Dr. van-Wachem, who took over as supervisor at Imperial College London in my final year. He always had an open door when I needed his advice and he gave me the freedom to finish my work independently. I would also like to thank Dr. Cegla and Dr. Beyrau for the friendly discussions over the years.

A special thanks goes to Mike Pettit, who joined me for the MSc in Edinburgh. I will break even on the snooker score at some point, but as you know, work always takes priority. I would also like to thank Dr. Philipp Geipel, and his very understanding wife Josie, for the fantastic times in South Kensington and Oxford, as well as Dr. Oliver Stein, for his ongoing support and discussions. Last but not least, I would like to show my appreciation to Fabian Denner; I am glad that I managed to convince you to join Imperial. Times with you will not be forgotten and luckily we always found our way home.

A huge thanks also goes to friends, current and 'former' colleagues Dr. Michele Bonnani, Dr. Nakul Prasad, Dr. Roger Robinson, Dr. Konstantina Vogiatzki, Fabrizio Cavallo-Marincola, Ben Franchetti, Tom Jelly, Terence Ma, Yagos Pesmazoglou, Abdel Rabhiou, Sandeep Saha and Nick Vaughan. All of you made my time at Imperial a unique and unforgettable experience.

I would also like to thank Daniel Muldoon, Benjamin Rösner, Christian Hinrichs and my family for their continuous support and always lifting my spirit in difficult times, I honestly appreciate it.

Finally, I would like to express my sincere gratitude to Miriam Rabaçal, who had a huge impact on my life during her short time at Imperial. Thank you for all the conversations, discussion and wonderful times (*Faz o que digo, não o que faço*).

Contents

1	Introduction	13
1.1	Motivation	15
1.2	Thesis Outline	16
2	Introduction to Fluid Dynamics	18
2.1	Conservation of Mass	18
2.2	Conservation of Momentum	19
2.3	General Transport Equation	20
2.4	Turbulence	21
2.4.1	Direct Numerical Simulation	23
2.4.2	Large-Eddy Simulation (LES)	25
2.4.3	Reynolds-Averaged Navier-Stokes	30
2.5	Combustion	32
2.5.1	Premixed Combustion	32
2.5.2	Artificially Thickened Flame	35
3	Numerical Framework	39
3.1	Finite Volume Method	39
3.1.1	Computational Grid	41
3.1.2	Description of Fluxes	42
3.1.3	Particle Transport	45
3.2	Time Discretisation	47
3.2.1	Euler Explicit	47
3.2.2	Euler Implicit	48
3.2.3	Runge-Kutta	48
3.2.4	Time-step width	49
3.3	Predictor-Corrector Algorithm	50
3.4	Pressure Correction	51

3.5	Boundary Conditions	54
3.5.1	Turbulent Inlet Conditions	54
3.5.2	Outflow Conditions	57
3.5.3	Periodic Conditions	57
4	Immersed Boundaries	59
4.1	Overview	59
4.2	Euler-Lagrange Methods	65
4.2.1	Body Force	67
4.2.2	Impermeability	69
4.2.3	Near Wall Region Treatment	72
4.3	Parallelisation	76
5	Test Cases — Static and Moving Surfaces in Flows	79
5.1	Immersed Boundaries — Static Test Cases	79
5.1.1	Pipe Flow	79
5.1.2	Flow around a Sphere — Static	89
5.1.3	Fractal Grid in Opposed Jet Flows	104
5.2	Immersed Boundaries — Moving Test Cases	129
5.2.1	Flow around a Sphere — Moving	129
5.2.2	Piston-Cylinder Arrangement	135
6	Interface Flows and Immiscible Fluids	150
6.1	Euler-Lagrange for Interface Flows	151
6.1.1	Theoretical Background	151
6.2	Interface Flows — Test Cases	154
6.2.1	Block Function	154
6.2.2	Zalesak Disk	155
6.3	Discussion and Conclusion	158
7	Conclusions and Recommendations for Future Work	165
A	The Wiener Process	170
B	Performance Measure and Optimisation of a Computer Tomography Code for High Performance Computing	173

List of Figures

2.1	Sketch of a typical turbulent energy spectrum - Turbulent kinetic energy $E(k)$ versus wave number k	24
2.2	Regime diagram of turbulent premixed combustion by Peters [137]	35
3.1	Illustration of the grid structure and grid notation	42
3.2	A typical 2D domain decomposition across 16 processors. The inflow domain is indicated by the coloured boxes, and shows the overlapping halo region. The red dashed box indicates the domain size (dependent on the prescribed length-scale) that must be stored for the original method of the inflow generator.	58
4.1	Sketch of an immersed boundary, where filled points represent particles to identify wall cells (<i>highlighted grey</i>) on the Eulerian grid	67
4.2	Sketch of an immersed boundary to illustrate the pressure gradient correction for wall faces that only have one neighbouring fluid cell	71
4.3	Sketch of an immersed boundary to illustrate the pressure gradient correction for wall faces that have multiple neighbouring fluid cell	72
4.4	Visualisation of the upwind/central differencing scheme	74
4.5	Structure of Immersed Boundary Method	75
4.6	Normalised simulation times of the LES 'PsiPhi' code with Lagrangian particles for different processor numbers, time measurements are based on the simulation of a flow around a sphere.	78

5.1	Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = 1.0$ mm and two different sub-grid stress models. Results are compared to experimental and numerical results.	85
5.2	Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = 1.0$ mm and $C_s = 0.100$ at different radial angles. Results are compared to experimental and numerical results.	86
5.3	Discretisation of the pipe onto the computational grid (left) and visualisation of mean axial velocity field at the mid point of the pipe for $C_s = 0.1$ (right) for $\Delta = 1$ mm. Maximum positive velocities are shown in white, while black represents a zero velocity.	87
5.4	Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = [1.0, 0.5]$ mm with the Germano model. Results are compared to experimental and numerical results.	88
5.5	Momentum balance of flow over a body, adapted from [95]. . . .	97
5.6	Schematic of the instantaneous axial velocity (top) and pressure (bottom) fields in the near sphere region with symmetry axis indicated by the dash-dotted line for $\Delta = 2$ mm and $Re = 100$. Maximum positive velocities and pressures are shown in white, while maximum negative velocities and pressures are black. . . .	97
5.7	Comparison of the drag coefficient for the flow around a static sphere at different Reynolds numbers for $\Delta = [5, 2]$ mm.	98
5.8	Wake length (l/D), measured from the end of the sphere. Comparison with experimental and numerical data sets	98
5.9	Statistical measurements of first moments of axial velocity behind the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.	99
5.10	Visual comparison of instantaneous axial velocity field at $Re = 300$ on the two grids: coarse (<i>top</i>) and fine (<i>bottom</i>). Maximum positive velocities are shown in white, while maximum negative velocities are black.	100
5.11	Enlarged domain for statistical measurements of first moments of a flow around a static sphere for $\Delta = 5$ mm and $Re = 300$. . .	100

5.12	Sample vector plot with contour lines of instantaneous axial velocity field at $Re = 300$ in the near sphere region on the two grids: coarse (<i>top</i>) and fine (<i>bottom</i>)	101
5.13	Statistical measurements of first moments of radial velocity behind the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.	102
5.14	Statistical measurements of first moments in the along the ordinate axis through the centre of the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.	103
5.15	Schematic of the opposed jet geometry, adapted from Geipel et al. [49].	116
5.16	Schematic of the perforated plate (<i>left</i>) and fractal cross-grid (<i>right</i>) used in the experiments (<i>top</i>) and numerical simulations (<i>bottom</i>). Dimensions are in mm and the cell size for the numerical grid was $\Delta = 0.1$ mm.	117
5.17	Schematic of the numerical opposed jet configuration with symmetry indicated by the dash-dotted lines. Isothermal axial velocity field (top nozzle) and in-plane lateral velocity field (bottom nozzle) in the configuration with a grid resolution $\Delta = 0.1$ mm and bulk velocity $U_b = 4$ m/s. Maximum positive velocities are shown in white, while maximum negative velocities are in black.	118
5.18	Comparison of statistical measurements of first (left) and second (right) moments from a single jet and two opposed nozzles for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.2$ mm and $C_s = 0.065$. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.	119
5.19	Comparison of statistical measurements of first (left) and second (right) moments for different inflow fluctuations, a bulk velocity $U_b = 4$ m/s, $\Delta = 0.2$ mm and $C_s = 0.065$. Three positions of a single jet are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.	120

5.20	Comparison of statistical measurements of first (left) and second (right) moments for different sub-grid scale constants, a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = [0.000, 0.030, 0.065]$. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.	121
5.21	Statistical measurements of first (left) and second (right) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ in an opposed jet configuration for an isothermal flow. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.	122
5.22	Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ along the centreline of the nozzle for an isothermal flow. Results are compared to experimental measurements, with point of reference ($x/D = 0$) being the stagnation plane.	123
5.23	Longitudinal auto-correlation functions for defined positions downstream of the fractal grid in the opposed jet configuration for an isothermal flow.	124
5.24	Integral length-scales and turbulent Reynolds numbers for defined positions downstream of the fractal grid in the opposed jet configuration for an isothermal flow. X is the point of reference (0 mm) for the fractal grid.	124
5.25	Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ along the centreline of the nozzle for a reactive flow. Results are compared to experimental measurements, with point of reference ($x/D = 0$) being the stagnation plane.	125
5.26	Statistical measurements of first (top) and second (bottom) moments for the progress variable in a reactive opposed jet configuration along the centreline of the nozzle. Results are compared to experimental measurements.	126

5.27	Visualisation of an instantaneous progress variable field between the nozzles for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $T_f = 2$. Burnt regions are shown in white, while unburnt regions are shown in black.	127
5.28	Axial velocity field for $Re = 500$ at three different time-steps. The maximum positive velocities is shown in white, while maximum negative velocities are black. The axis of symmetry is shown with the dash-dotted line.	132
5.29	Comparison of the drag coefficient for the static and moving sphere at different Reynolds numbers. The line indicates the theoretical values.	132
5.30	Comparison of the wake length for the static and moving spheres.	133
5.31	Statistical measurements for the first moment of the flow around a moving sphere at different Reynolds numbers. Results are compared to the flow around a static sphere at same Reynolds numbers.	134
5.32	Schematic drawing of the piston-cylinder configuration, based on the coarse computational grid. Symmetry is indicated by the dotted line.	143
5.33	Axial velocity field for two crank angles, 36° (left) and 144° (right), measured from TDC on the fine grid. The maximum positive velocities are shown in white, while maximum negative velocities are black.	144
5.34	Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 1$ mm, and a crank angle of 36° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.	145
5.35	Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 1$ mm, and a crank angle of 144° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.	146

5.36	Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 0.5$ mm, and a crank angle of 36° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.	147
5.37	Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 0.5$ mm, and a crank angle of 144° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.	148
6.1	Graphical Representation of decomposition	160
6.2	Visualisation of a 1-D random walk, based on the Wiener Process, particle has been initialised at the origin (0,0).	160
6.3	Identical diffusion coefficients; Eulerian, Lagrangian and Analytical Solution based on a top-hat function	161
6.4	Comparison of results - original field (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach	161
6.5	Comparison of results, disk returned - original disk (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach	162
6.6	Comparison of results, one revolution - original disk (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach	162
A.1	Random walk for multiple particles initiated at the origin (left) and the particle distribution at various positions along the x-axis (right).	172
B.1	Drawing of the Matrix Burner	178
B.2	Example view of the Matrix Burner test cases, at 45° to the horizontal	182
B.3	Example reconstruction of the matrix burner with the mixed-mode parallelisation at low resolution with 64 views	183
B.4	Efficiency of the parallel ART and MART algorithms; filled symbols represent the MART algorithm, same symbols indicate same test case	187

List of Tables

5.1	Key simulation parameters of examined opposed jet cases, $\sqrt{u'u'}$ indicates the fluctuations at the inflow and C_s is the applied value for the Smagorinsky model	128
5.2	Piston-cylinder dimension and parameters	149
6.1	Simulation parameters for test cases to evaluate the performance of the proposed Euler-Lagrange method for Interface Flows . . .	163
6.2	Block: Comparison of different approaches - Simulation times .	163
6.3	Zalesak disk returned: Comparison of different approaches - Simulation times, RMS and sum of error (SoE)	163
6.4	Zalesak disk, one revolution: Comparison of different approaches - Simulation times, RMS and sum of error (SoE)	164
B.1	Applied test cases	178
B.2	Mixed-mode algorithms - Run Time	185

Nomenclature

<i>Latin Symbols, Uppercase</i>		<i>Units</i>
A	laminar flame surface area	$[m^2]$
A_t	turbulent flame surface area	$[m^2]$
C_d	drag coefficient	$[-]$
C_s	sub-grid scale parameter	$[-]$
C_s^{clip}	clipped sub-grid scale parameter	$[-]$
D	diffusion coefficient	$[m^2s^{-1}]$
D_j	diffusion term in j-direction	$[m^2s^{-1}]$
D_n	opposed jet nozzle diameter	$[m]$
D_s	sphere diameter	$[m]$
E	efficiency factor	$[-]$
L	domain length	$[m]$
L_i	integral length-scale	$[m]$
L_{ij}	resolved turbulent stress tensor	$[m^2s^{-2}]$
N	number of time intervals	$[-]$
R	domain radius	$[m]$
Re	Reynolds number	$[-]$
Re_{crit}	critical Reynolds number	$[-]$
Re_t	turbulent Reynolds number	$[-]$
R_{xx}	autocorrelation function	$[-]$
S_{ij}	strain rate	$[s^{-1}]$
St	Strouhal number	$[-]$
T	generic time interval	$[s]$
T_b	burnt temperature	$[K]$
T_f	flame thickening factor	$[-]$
T_{ij}	test filtered residual-stress tensor	$[m^2s^{-2}]$
T_u	unburnt temperature	$[K]$
T_1	sequential processing time	$[s]$
U_b	bulk velocity	$[ms^{-1}]$
U_τ	frictional velocity	$[ms^{-1}]$
W	Wiener process	$[-]$

<i>Latin Symbols, Lowercase</i>		<i>Units</i>
c	progress variable	[-]
f	body force	$[kgm^{-2}s^{-1}]$
i_g, j_g, k_g	global Cartesian logical co-ordinates	[-]
l_δ	inner layer flame thickness	$[m]$
n	number of computational cells in domain	[-]
r	sphere radius	$[m]$
s_l	laminar flame speed	$[ms^{-1}]$
s_t	turbulent flame speed	$[ms^{-1}]$
s_p	position vector of particle p	$[m]$
t	real time	$[s]$
u_i	velocity component in i-direction	$[ms^{-1}]$
u_p	mean piston velocity	$[ms^{-1}]$
x_i	spatial coordinate in i-direction	$[m]$

<i>Greek symbols, Lowercase</i>		<i>Units</i>
α	volume fraction	[-]
δt	time step/interval	$[s]$
δ_{ij}	Kronecker delta	[-]
δ_l	flame thickness	$[m]$
ϵ	dissipation rate	$[m^2s^{-3}]$
η	Kolmogorov length scale	$[m]$
κ	transfer term	[-]
μ	dynamic viscosity	$[kgm^{-1}s^{-1}]$
ν	kinematic viscosity	$[m^2s^{-1}]$
ν_t	turbulent viscosity	$[m^2s^{-1}]$
ϕ	generic (scalar) field quantity	[-]
$\overline{\phi}$	generic (scalar) field quantity, filtered contribution	[-]
ϕ'	generic (scalar) field quantity, residual contribution	[-]
$\overline{\phi'}$	generic (scalar) field quantity, Eulerian contribution (Ch. 6)	[-]
$\overline{\phi''}$	generic (scalar) field quantity, Lagrangian contribution (Ch. 6)	[-]
ϕ_{rms}	Root Mean Square of a field (Chapter 6)	[-]
ϕ_{SoE}	Sum of Error of a field (Chapter 6)	[-]
ρ	mass density	$[kgm^{-3}]$

$\bar{\omega}$	mean reaction rate	$[kmolm^{-3}s^{-1}]$
τ_{ij}	stress tensor	$[m^2s^{-2}]$

<i>Greek symbols, Uppercase</i>		<i>Units</i>
Δ	filter (cell) width	$[m]$
Δ_t	test filter width	$[m]$
Φ	equivalence ratio	$[-]$
Ω	piston speed	$[rads^{-1}]$

Super- and Subscript

b	burnt
f	cell face quantity
CR	corrected quantity
CS	chemical state
PD	predicted quantity
u	unburnt

Abbreviations

ATF	artificially thickened flames
BDC	bottom dead center
CDS	central differencing scheme
CFD	computational fluid dynamics
CFL	Courant-Friedrichs-Lewy number
CV	control volume
DNS	direct numerical simulation
FVM	finite volume method
HPC	high performance computing
IB	immersed boundary
IC	internal combustion
LDA	laser doppler anemometry
LDV	laser doppler velocimetry
LES	large-eddy simulation
LHS	left hand side

<i>PIV</i>	particle image velocimetry
<i>RANS</i>	reynolds-averaged navier-stokes
<i>RHS</i>	right hand side
<i>TDC</i>	top dead center
<i>TVD</i>	total variation diminishing
<i>UDS</i>	upwind differencing scheme

Chapter 1

Introduction

Continuous growth of computing power strongly encourages engineers to rely more on computational fluid dynamics for the design and testing of new technological solutions. The application of computers in research and development avoids the need for expensive prototypes and difficult experimental measurements. Computational simulations can give a greater insight into the physical and chemical phenomena at locations where reliable and accurate measurements are very hard to obtain, if not impossible.

Computational Fluid Dynamics (CFD) is the analysis of fluid flow systems by computational simulations. These simulations are based on numerical approximations derived from the Navier-Stokes equation. CFD has great potential, as it can be applied to a wide number of industrial applications, which include: analysis of lift and drag forces on surfaces, turbo machinery, ventilation systems, but also meteorology, multiphase flows and internal combustion engines, amongst others.

Computational fluid dynamics was first applied in the aerospace industry in the 1930's as part of their design and Research and Development (*R&D*) [121]. In the past, the limiting factors in CFD development have been the available computer power and model closures. Nowadays, High Performance Computing (HPC) is rapidly growing, which requires the development and improvement of new and existing techniques to model more realistic industrial applications (e.g. internal combustion engine set-ups, airplane turbine cycles). These applications have complex geometries combined with moving boundaries and high

Reynolds numbers. Fine grid resolutions are required to properly resolve the flow field. However, depending on the geometry, the grid generation and quality can be a major issue.

Boundary conforming grids or unstructured grids were originally, and still are, used in simulations with complex geometries [78, 80, 105, 106, 198]. In these approaches, the governing equations are discretized on a curvilinear grid, where the boundary is represented by a continuous line and objects are extracted from the grid. Therefore the boundary location is known and allows for the simplified imposition of boundary conditions and flow solver optimisation to maintain a high degree of accuracy. The mesh generation for complex geometries with multiple objects in the flow are very time consuming [105, 115, 147]. Even the implementation of unstructured meshes for these problems has to be done with care to avoid high variances of aspect-ratios¹ of cells. Simulations with unstructured grids for complex boundaries are also found to be slower on a per-cell basis than simulations with structured grids, decreasing the computational efficiency. Furthermore the motion of boundaries imposes a great challenge for boundary conforming grids, leading to costly re-meshing operations, interpolation errors and mesh deformations near the boundaries [34].

In recent years there has been an increased interest in numerical methods to compute flow fields with complex boundaries (stationary or moving) on fixed Cartesian grids (*non-boundary conforming* grids). The major advantage of these methods is the avoidance of mesh adaptation or deformation. Furthermore Cartesian grids are simple to generate and they are usually computationally more economical [186].

Non-boundary conforming methods, representing the effects of the Immersed Boundary (IB) on the surrounding fluid phase, can be divided into two categories based on the specific treatment of the boundary cells [22]:

- The *Cartesian cut-cell* approach, which is based on the construction of irregular cells near the surface
- The *Immersed boundary* method which enforces wall conditions indirectly through forcing functions (i.e. additional body forces)

¹Ratio of height/length/width of a cell

The Cartesian cut-cell method is applied to inviscid flows and has been extended to viscous flows including moving boundaries [34, 182, 197]. In the Cartesian cut-cell approach, local boundary conformity is achieved by modifying the Cartesian grid cells, which are intersected by the body. The boundary conditions are similar to those of boundary fitted grids, however an iterative solution procedure is required due to the irregular stencil near the boundaries. Furthermore special cell treatment needs to be introduced in the near boundary region, which affects the computational efficiency.

The immersed boundary method was first introduced in the 1970's in order to study the blood flow in the human heart [134] and many different variations have been developed since. This method benefits from its simplicity by enforcing boundary conditions through force functions. In addition, the approach can be easily extended to multiphase flows. A more detailed discussion on modelling boundaries in a flow is given in Chapter 4 in relation with the description of the immersed boundary approach that was applied in this work.

A successful integration of an efficient non-boundary conforming strategy should be able to handle both arbitrary movement and complex body shapes. Combining the two objectives with a robust Cartesian grid solver will open new areas of application. In this work, Lagrangian particles are used to efficiently describe the moving objects on the underlying Eulerian grid.

The idea of using Lagrangian particles has also been applied to improve the accuracy of scalar transport in areas of steep scalar gradients. The increasing availability of computing power in recent years allows for the use of Lagrangian particles in a wide area of applications.

1.1 Motivation

The main objective of the presented study is to develop an immersed boundary method that is valid at low to moderate Reynolds numbers and is suitable for implementation on complex geometries with and without solid boundary movement.

Ideally, an immersed boundary method should be simple to implement, computationally efficient, robust and accurate for laminar and turbulent, isothermal and reactive flows in simple and complex domains, alike. However, trade-offs between these characteristics have to be made, which often depend on the area of application. For example, the modelling of a swimmer in a new wetsuit requires a high degree of accuracy to fully understand the flow in the boundary layer of the swimmer. On the other hand, the near wall regions and corners in the simulation of an industrial combustion chamber are often neglected, if the focus is on the mixing and combustion that normally occur away from walls.

Many different methods to describe boundaries have been proposed in the past (see Chapter 4). Some of the methods require complex mesh deformation or regenerations to improve the accuracy. Other methods are very complex, computationally inefficient or suffer from stability problems. In this work, the presented approach is based on the idea of describing Immersed Boundaries (IB) with Lagrangian particles by blocking cells on the underlying Eulerian grid. The description is simple and efficient if the grids are simple, so that a search operation is not necessary to identify cells in which the particles are located; a simple hashing is sufficient. In this work, the main attention will be given to the simplicity and computational efficiency of the method and apply the approach to static and moving boundary problems in simple and complex geometries. In the presented cases, the moving boundaries are driven by an external force which is not the result of the fluid force acting on the structure. The successful implementation of the IB method will allow for the simulation of engineering problems in complex geometries with moving walls without a significant increase in computational costs.

1.2 Thesis Outline

The thesis has the following structure: Chapter 2 provides the fundamentals required to understand turbulent flows and combustion. The basic equations of fluid dynamics are revised and the phenomenon of turbulence is described. Furthermore three different approaches to model turbulent flows are introduced, while DNS and RANS are outlined briefly, LES is described in more detail as it is the method of choice in this work. The end of chapter 2 outlines

the basics of combustion; the main focus is given to premixed combustion since the work only deals with a premixed combustion case. In chapter 3 the numerical framework and concept of numerical discretization is presented. Chapter 4 focuses on the developed immersed boundary method and describes the implementation into the current numerical framework. Chapter 5 reports on the simulation results of static and moving boundary test cases that were applied to evaluate the performance of the immersed boundary method. A very simple pipe flow was used to evaluate the basic implementation of the method, before applying it to a static sphere in a flow and an fractal grid in an opposed jet flow. The second half of the chapter 5 discusses the simulation results of two test cases with moving boundaries. The first test case is similar to the static sphere, however in this case the sphere moves through the computational domain. The second test case is based on the movement of a piston in a piston-cylinder arrangement. In chapter 6 an alternative application for Lagrangian particles is discussed, where particles are used to improve the accuracy and resolution of scalar transport. The major results and conclusions from the present work are summarized in chapter 7 and suggestions for future work are given.

Chapter 2

Introduction to Fluid Dynamics

This chapter, based on the works [43, 82, 173, 186], briefly outlines the basic concepts of fluid dynamics and its basic equations to govern the flow of an incompressible fluid. The main focus is given to the conservation of mass and momentum equations. Subsequently the importance of turbulence in engineering will be discussed and the key aspects of combustion modes are summarized. Furthermore three common approaches to model turbulent flows are presented, amongst them the method applied in this work, Large-Eddy Simulation (LES). Appropriate references will be given for the interested reader to gain a greater insight into the various sections of this report.

The laws of conservation of mass and momentum describe the motion of fluids at a macroscopic scale and the fluid is regarded as a continuum. If the flow is reactive, further scalar quantities (e.g. enthalpy and species concentration) need to be considered.

2.1 Conservation of Mass

One of the fundamental concepts in fluid dynamics is the conservation of mass, which can be applied to a fixed control volume within the flow field. The concept states that the mass of an isolated system will remain constant over time. Considering any fixed region in the flow of an open system with constant volume, the mass inside the volume can only change by the net-flux into (out of) the volume. The differential form of the continuity equation can be written

as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \quad (2.1)$$

where the first term on the left-hand side (LHS) is the rate of change in time of the density and the second term represents the convective transport term with velocity u_i in direction x_i . As it is written, Eq. (2.1) describes the unsteady mass conservation at a point in a fluid [186]. For low Mach number ($Ma < 0.3$), in-compressibility can often be assumed, hence the density does not change with pressure and the equation simplifies to $\frac{\partial u_i}{\partial x_i} = 0$. This implies that no other mechanism, such as heat release or chemical reaction changes the density. In the context of mixing and combustion, the idea of constant density ($\frac{\partial \rho}{\partial t} \neq 0$) is not suitable.

2.2 Conservation of Momentum

Similar to mass, momentum is a conserved quantity. The conservation of momentum states that the rate of change of the momentum of an object is equal to the sum of forces on an object, it can be expressed as:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} + \rho k_i \quad (2.2)$$

The first term on the left hand side describes the rate of change in momentum at time t and the second term marks the convective transport of momentum. On the right hand side, the first term denotes the stresses due to local deformation and pressure. The second term, ρk_i , describes the external forces (e.g. gravity, centripetal force, electric force,...). Further more the stress tensor τ_{ij} depends on the material property and the assumption of Newtonian Fluid properties. It holds true for gases and liquids like water

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - p \delta_{ij} \quad (2.3)$$

where μ is the dynamics viscosity, p is the pressure and δ_{ij} is the Kronecker delta, which is the index notation of the unity matrix

$$\delta_{ij} = 1 \quad \text{if} \quad i = j \quad (2.4)$$

$$\delta_{ij} = 0 \quad \text{if} \quad i \neq j \quad (2.5)$$

Assuming that gravity (g_i) is the only body force acting on the fluid and by substituting Eq. (2.3) into Eq. (2.2), the Navier-Stokes equations can be formed.

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

The discussion of the surface and body forces will be extended in Chapter 4 in combination with the Immersed Boundary (IB) method.

2.3 General Transport Equation

The equations governing turbulent fluid flow, heat and mass transfer can be cast into a single general form, the general scalar transport equation, providing a framework within which numerical methods can be applied to solve the governing equations. For a conserved scalar ϕ (e.g. $\rho, \rho u_i$) the equation consists of the rate of change of the scalar, the convective transport by the velocity u_j , the rate of diffusion D_ϕ and a source/sink term S_ϕ . The general transport equation can be written as:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi u_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left(D \frac{\partial \phi}{\partial x_j} \right) + S_\phi \quad (2.7)$$

Neglecting the source term ($S_\phi = 0$), the scalar ϕ is only redistributed by the transport process. By setting $\phi \equiv \rho$, it can be shown that the continuity equation (2.1) is a special form of the transport equation (2.7). Assuming that the mass diffusion does not affect the density ($D = 0$) and eliminating the source term ($S_\phi = 0$), the RHS of Eq. (2.7) is equal to zero and the continuity equation (2.1) is obtained. The momentum equation (2.2) and the Navier-Stokes equation (2.6) [190] can be obtained similarly.

To address problems involving mixing or combustion, the governing equations for mixture fraction/concentrations must be solved and the equation becomes

$$\frac{\partial(\rho Y_\alpha)}{\partial t} + \frac{\partial(\rho Y_\alpha u_j)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\rho D_\alpha \frac{\partial Y_\alpha}{\partial x_j} \right) + S_\alpha \quad (2.8)$$

where the mass fraction of each species is represented by Y_α , which describes the ratio of mass of species ' α ' to the overall mass. The chemical source term

is represented by S_α and Fick's law [190] is used to express the diffusion term.

The set of transport equations, perfect gas law and chemical rate expressions are sufficient to describe combustion in turbulent flows, provided that accurate diffusion coefficients are used. However, it is a demanding task to solve the full set of equations, therefore simplifying assumptions are often made to reduce the complexity of describing combustion. For example, in fully premixed combustion the progress of the chemical reaction is described by the balance equation for the *reaction progress variable* c .

$$\frac{\partial(\rho c)}{\partial t} + \frac{\partial(\rho c u_j)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\rho D_c \frac{\partial c}{\partial x_j} \right) + S_c \quad (2.9)$$

The progress variable is a basic parameter to describe the progress of the reaction to turn the fresh into burned gases across the flame and it can be defined with the help of a quantity like temperature.

$$c = \frac{T - T_u}{T_b - T_u} \quad (2.10)$$

where T_u denotes the unburnt temperature, T_b the burnt temperature and T is the local temperature. Details on the modelling of the transport of the reaction progress variable in fully premixed combustion are postponed to section 2.5

2.4 Turbulence

Fluid motion can be differentiated into laminar and turbulent regimes. Basic fluid mechanics is often concerned with laminar flow where the flow moves in ordered layers and the mixing in these flows is the result of diffusion.

In contrast, a turbulent flow is unsteady, chaotic and three dimensional. The flow consists of eddies which appear at random frequency and size. The mixing in a turbulent flow is increased as a result of these random structures throughout the flow. In most engineering applications, turbulent flows are present and hence of great interest to the research community.

The difference in laminar and turbulent flow characteristics was investigated by Reynolds [150], who analysed the onset of turbulence by using dye to visualise

the flow in a pipe. He evaluated the requirements for the transition from laminar to turbulent flows, which are governed by the flow instabilities and the viscous forces. As an outcome, Reynolds established a dimensionless number, later called the Reynolds number (Re), to give a measure for the turbulence in the flow [151]:

$$Re = \frac{UL}{\nu} = \frac{\rho UL}{\mu} \quad (2.11)$$

where U is the characteristic flow velocity, L is the characteristic length, ν is the kinematic viscosity, ρ is the flow density and μ is the dynamic viscosity. Referring to the flow in a pipe, U is the mean flow velocity, L is the pipe diameter and ν is the fluid viscosity. A critical value Re_{crit} for the transition from a laminar to a turbulent flow can be established and in a pipe flow corresponds to $Re_{crit} = 2300$. The critical Reynolds number is a guideline as laminar flow conditions can also be found at higher Reynolds numbers.

The nature of turbulence appears to be random and Reynolds developed another important concept, *Reynolds' decomposition*, to describe turbulence in a statistical form. The concept states that the turbulent velocity is composed of a mean velocity $\bar{u}(x_i, t)$ and a fluctuation component $u'(x_i, t)$ according to

$$u(x_i, t) = \bar{u}(x_i) + u'(x_i, t) \quad (2.12)$$

where the mean velocity is understood as the Reynolds (temporal) average. The concept of Reynolds-Averaged Navier-Stokes (RANS) modelling is based on inserting Eq. (2.12) into the continuity equation (2.1) and momentum equation (2.2) to obtain the temporal average of the flow, which leads to unclosed terms that must be modelled. An outline of the RANS approach is given in section 2.4.3.

Turbulence is often characterised as a chaotic irregular flow, however certain regularities can still be observed. These regularities often depend on the geometry of the flow and may be termed *vortices* or *eddies* (if the structures are of rotational nature). The typical size of such an eddy can be quantified using correlations that determine how the velocity fluctuations at point x are affected by the fluctuations in the vicinity of x (at distance r). For example, the autocorrelation function R_{xx} (normalised auto-covariance) for homogenous,

isotropic turbulence can be written as [186]

$$R_{xx}(r, t) = \frac{\overline{u'(t)u'(r, t)}}{\overline{u'^2(t)}} \quad (2.13)$$

The definition of the *integral lengthscale* of turbulence l_I can be derived by integrating R_{xx} in the r -direction, which represents the typical size of large, energy containing fluid structures, and reads

$$l_I(t) = \int_0^\infty R_{xx}(r, t) dr \quad (2.14)$$

whereas the small scale fluid structures are best discussed in the context of the energy spectrum and Kolmogorov hypothesis below.

Turbulent flows consist of varying structures with fluctuating velocity magnitude and therefore changing energy. Richardson introduced the concept of the energy cascade [152] in 1922, which describes the energy transfer from large to smaller structures. These smaller structures decay further until the smallest structures dissipate the energy into heat. The eddy dissipation rate ϵ is independent of the fluid viscosity and proportional to the integral scales of motion (length l_I , time t_I and velocity u_I scale) [173] and can be estimated as [91]:

$$\epsilon \approx u_I^2/t_I = u_I^3/l_I \quad (2.15)$$

The energy spectrum, sketched in Fig. 2.1, illustrates the concept of the energy transfer from large to smaller structures. In 1941, the concept of the energy cascade was explored by Kolmogorov [91] who also identified the smallest scales, *Kolmogorov micro-scales*. These findings have a strong impact on the numerical simulation of turbulent flows as they express the isotropy and universality of the smallest turbulent scales. A mathematical derivation of a theoretical energy spectrum can be found in the book by Pope [146].

2.4.1 Direct Numerical Simulation

Direct Numerical Simulation (DNS) is potentially the easiest modelling approach to understand. In the ideal situation, turbulent flows would be simulated in great detail - resolving all scales in the flow. Direct Numerical Sim-

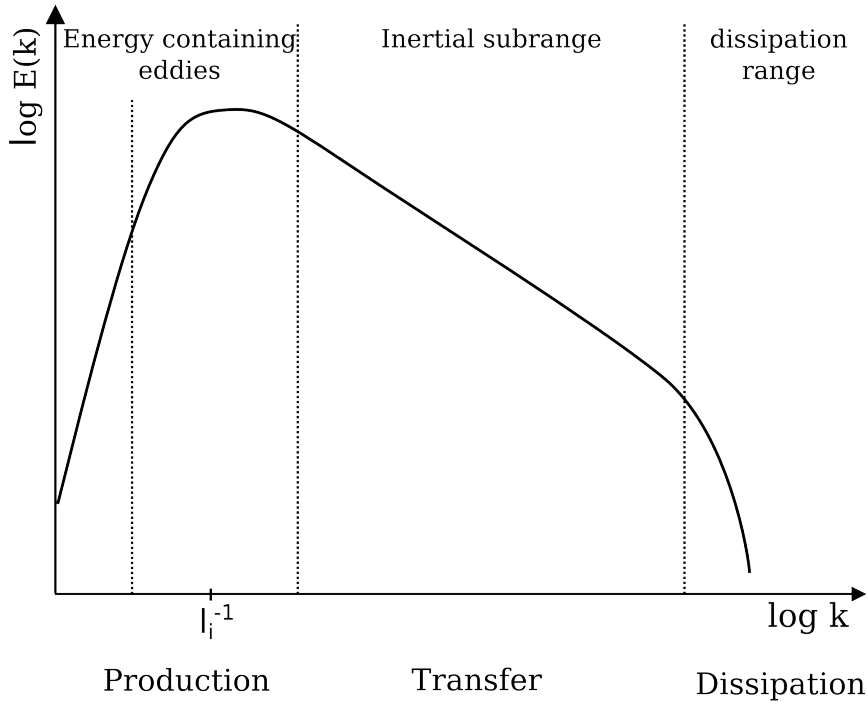


Figure 2.1: Sketch of a typical turbulent energy spectrum - Turbulent kinetic energy $E(k)$ versus wave number k

ulations produce a single realisation of the flow [146]; the method uses the set of equations (*Navier-Stokes*) in space and time to resolve the smallest turbulent eddies and simulate all fluctuations, without relying on any modelling assumptions. Hence the method provides an excellent level of accuracy and description of the flow. The computational grid needs to be fine enough to resolve all turbulent scales [145]. Therefore Direct Numerical Simulations are computationally very costly and until the 1970s the DNS approach was infeasible due to the lack of computational power. Even nowadays, numerical simulations are limited to small and moderate Reynolds numbers. The required resolutions in space for a DNS rises proportionally to the Reynolds number by $Re^{3/4}$ in each dimension of the domain. Considering that turbulent flows are inherently three-dimensional, the number of required grid points for a direct numerical simulation of a turbulent flow can be estimated as:

$$n \approx Re^{(3/4)^3} = Re^{9/4} \quad (2.16)$$

and the ratio of the smallest to the largest time scales varies by $Re^{(1/2)}$ [186]. Poinot and Veynante [143] argue that DNS is still not applicable for practi-

cal flows and simplifications are necessary. However an increase in processing power (if Moore’s law holds) and the development of High Performance Computers (HPC) will increase the applicability of DNS. In addition, Versteeg and Malalasekera [186] maintain the opinion that the estimate of the required grid points can be relaxed, as most of the dissipation takes place at scales that are substantially larger than the order of the Kolmogorov length scale η .

2.4.2 Large-Eddy Simulation (LES)

Large-Eddy Simulation (LES) is becoming increasingly popular, mainly due to an increasing availability of computational power and the method’s ability to benefit from the additional power. The technique can be regarded as an intermediate approach between DNS and RANS. The motivation for LES comes from the disadvantages of the computationally expensive DNS and of the low accuracy of RANS (see Section 2.4.3). In DNS, most effort is spent in resolving the smallest scales, and a general-purpose model in RANS suitable for a wide range of practical applications seems elusive so far [186].

In Large-Eddy Simulation, large scales are resolved explicitly and small scales, smaller than the filter width (often cell size) are modelled. The value of the method is reduced when the key phenomena occur on the small (unresolved) scales, otherwise it presents a promising alternative to DNS and RANS. The large scales are not universal, but strongly dependent on the geometry. The small scales can be regarded as universal, and are traditionally represented by *simple* models. Two of these models (*Smagorinsky* and *Germano*) will be discussed below.

The temporal evolution of spatial averages of the flow quantities are considered in LES by applying a low-pass filter to the governing equations. The filtering separates the larger eddies in the flow from the smaller ones. The LES approach can be summarised in four conceptual steps. The velocity is decomposed by applying a filter operation to remove the small scale information. The next process is the derivation of the equations for the evolution of the filtered velocity field from the Navier-Stokes equation. The residual-stress tensors are then modelled to obtain closure. In the last step the numerical solution of the filtered equations is solved for the filtered velocity component

to provide an approximation to the large-scale motions.

Work to develop Large-Eddy Simulation was carried out by Deardoff [36], Lilly [100] and Smagorinsky [167], amongst others. Their work was motivated by meteorology to develop improved methods for weather predictions. LES methodologies for engineering were first applied by Kraichnan [94] in the 1970s when computing power increased and simulations became affordable. The first Large-Eddy Simulation applications covered turbulent channel flow [35, 163]. In the 1990s and 2000s the research scope broadened to applications dealing with complex geometries and turbulent combustion [2, 45, 68]. Research with Large-Eddy Simulation has continued to date and a considerable amount of literature on LES is available.

2.4.2.1 Filtering

The Large-Eddy Simulation equations are derived from spatial filtering of the Navier-Stokes equations. The filtering decomposes the variable into a filtered (resolved) $\bar{\phi}$ and a residual (sub-grid) ϕ' part, which can be written as

$$\phi = \bar{\phi} + \phi' \quad (2.17)$$

which is not to be mistaken with the *Reynolds Decomposition* in section 2.4. The idea of filtering in LES was first introduced by Smagorinsky [167] and generally described by Leonard [99]. During the filtering all the information of the smaller eddies is destroyed and a sub-grid scale (SGS) model needs to be introduced to model the information of the smaller, filtered eddies.

The filtered variable, obtained from Leonard [99], can be expressed as

$$\overline{\phi(x_i, t)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(x_i, x'_i, \Delta) \phi(x'_i, t) dx'_1 dx'_2 dx'_3 \quad (2.18)$$

where the integration is over the entire flow domain, with a filter function G , a shift x'_i and a filter width Δ . Filter functions include the top-hat function, which is implied in implicit filtering by applying the finite volume method (see Chapter 3). Other filter functions of special interest are the Gaussian filter and the Spectral cut off filter. For a flow of varying density, a density-weighted

filter is often used to simplify the mathematical treatment. This *Favre filtering* can be written as

$$\phi = \tilde{\phi} + \phi'' \quad (2.19)$$

with the density-weighted average $\tilde{\phi} = \overline{\rho\phi}/\bar{\rho}$ and the Favre-fluctuation ϕ'' . This allows the filtered correlation $\overline{\rho\phi}$ in the governing equations to be re-expressed as the product of the Favre filtered quantities $\bar{\rho}\tilde{\phi}$.

The Favre filtering can be applied to the Navier-Stokes equations. The filtering process does not change the continuity equation (2.1), only replaces their unfiltered quantities by the filtered equivalents:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}(\bar{\rho}\tilde{u}_i) = 0 \quad (2.20)$$

The process can be repeated with the Navier-Stokes equation, which gives:

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho}\widetilde{u_i u_j}}{\partial x_j} = \frac{\partial}{\partial x_j}[\bar{\rho}\tilde{\mu}(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}) - \frac{2}{3}\bar{\rho}\tilde{\mu}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij}] - \frac{\partial \bar{p}}{\partial x_i} + \bar{\rho}g_i \quad (2.21)$$

All terms can be written as a function of filtered quantities apart from the convective term. It is important to realise that $\widetilde{u_i u_j} \neq \tilde{u}_i \tilde{u}_j$ and, as such, the left hand side cannot be easily computed. Therefore an approximation for the difference between $\widetilde{u_i u_j}$ and $\tilde{u}_i \tilde{u}_j$ is necessary as it is not closed. The term can be split into

$$\widetilde{u_i u_j} = \tilde{u}_i \tilde{u}_j + \tau_{ij}^{SGS} \quad (2.22)$$

and in the context of LES, τ_{ij}^{SGS} is called the sub-grid stress tensor [146]. Inserting the approximations of equation (2.22) into the filtered momentum equation leads to the modified equation:

$$\frac{\partial \bar{\rho}\tilde{u}_i}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i \tilde{u}_j}{\partial x_j} = \frac{\partial}{\partial x_j}[\bar{\rho}\tilde{\mu}(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}) - \frac{2}{3}\bar{\rho}\tilde{\mu}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} - \bar{\rho}\tau_{ij}^{SGS}] - \frac{\partial \bar{p}}{\partial x_i} + \bar{\rho}g_i \quad (2.23)$$

2.4.2.2 Sub-grid Stress Modelling

Equation (2.23) includes the unknown sub-grid stress tensor τ^{SGS} which requires closure. In the present work the eddy viscosity approach is used, in which the sub-grid contribution is represented by adding the turbulent viscosity $\tilde{\nu}_t$ to the molecular viscosity $\tilde{\mu}$, which allows for the closure of equation (2.23) except for the turbulent viscosity. A detailed derivation of Eq. (2.23) can be

found in the literature [129, 146, 156]. The following paragraphs focus on the two closure methods used in this work. Additional methods are available and the interested reader is referred to Sagaut [156] for more detailed information on closure methods.

Smagorinsky: The Smagorinsky model is the earliest and most commonly used approach. The model follows, as mentioned above, the eddy-viscosity approach and was first proposed by Smagorinsky [167]. Like all eddy-viscosity models, it is based on the notion that the principal effects of the sub-grid Reynolds stresses are increased transport and dissipation, which are phenomena due to the viscosity of laminar flows.

The Smagorinsky approach models the turbulent viscosity contribution $\tilde{\nu}_t$ as a function of a model constant C_s , the filter width Δ and the strain rate of the large scale field $\bar{S}_{ij}$, respectively

$$\tilde{\nu}_t = (C_s \Delta)^2 |\bar{S}| \quad (2.24)$$

where $|\bar{S}| = (\bar{S}_{ij} \bar{S}_{ij})^{1/2}$. Even though C_s is referred to as a model constant, it should be more regarded as a model parameter, which is often adjusted to improve results ($\approx 0.065 - 0.2$). Choosing an appropriate value often depends on the flow geometry and characteristics. For example, Clark et al. [25] use a model constant $C_s = 0.2$ for the case of isotropic turbulence in a channel flow, whereas Forkel [45] prefers a value of $C_s = 0.1$ for the same case.

Even though the Smagorinsky model is a efficient and simple, there are problems with it. The globally applied constant causes problems in the near wall regions where the viscosity should be lower. To overcome this problem, van Driest damping [37], successfully applied in RANS simulations, is often applied. However the damping requires additional computational efforts and hence it is often ignored in highly turbulent flows where the region of interest is far away from a wall.

Germano: The dynamic model was first introduced by Germano [52] in 1991 and is often referred to as the *Germano model*. Many adaptations have been published since, among them work by Lilly and Piomelli [101, 141]. In the

dynamic approach the value of C_s , actually the square of the original quantity, is considered as a field value that depends on the localised flow conditions in time, a test filter width Δ_t , which is larger than the filter (cell) width (typically $\Delta_t = 2\Delta$), and assumes an identical model parameter for both filter widths.

In this work, the implementation by Lilly [101] was used for the dynamic procedure. The idea is to calculate an explicit turbulent stress tensor by implicitly filtering the velocity field. First the test filtered residual-stress tensor is required

$$T_{ij} = \widehat{\tilde{u}_i \tilde{u}_j} - \hat{\tilde{u}}_i \hat{\tilde{u}}_j \quad (2.25)$$

to form the resolved turbulent stress tensor L_{ij} by subtracting the filtered residual stress tensor

$$L_{ij} = T_{ij} - \widehat{\tau_{ij}} = \widehat{\tilde{u}_i \tilde{u}_j} - \hat{\tilde{u}}_i \hat{\tilde{u}}_j \quad (2.26)$$

The elements of L are then resolved components of the stress tensor associated with scales of motion between the test scale and the grid scale [101]. The RHS of Eq. (2.26) can be evaluated and compared with the local closure approximation by subtracting the test scale average

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = 2C\Delta^2|\bar{S}|\bar{S}_{ij} \quad (2.27)$$

from

$$T_{ij} - \frac{1}{3}\delta_{ij}T_{kk} = 2C\Delta_t^2|\hat{\tilde{S}}|\hat{\tilde{S}}_{ij} \quad (2.28)$$

to obtain

$$L_{ij} - \frac{1}{3}\delta_{ij}L_{kk} = 2CM_{ij} \quad (2.29)$$

where $M_{ij} = \Delta_t^2|\hat{\tilde{S}}|\hat{\tilde{S}}_{ij} - \Delta^2|\bar{S}|\bar{S}_{ij}$. The tensors L_{ij} and M_{ij} can then be evaluated based on the filtered velocity field. Applying Lillys' least square approach [101] to Eq. (2.29) allows for the calculation of the localised C_s value by

$$C_s = 0.5 \frac{M_{ij}L_{ij}}{M_{ij}M_{ij}} \quad (2.30)$$

The Germano model benefits from the disappearance of the adjustable model parameter, as it is a self-sufficient approach. However the model is computationally expensive compared to the Smagorinsky model. A localised C_s value

for each position in the flow at every time-step needs to be calculated. Furthermore the model sometimes suffers from instabilities and in this work, negative values of C_s are clipped to avoid negative turbulent viscosity (*'backscatter'*):

$$C_s^{clip}(x_i, t) = \max[C_s(x_i, t), 0] \quad (2.31)$$

The interested reader is referred to the works by Germano, Lilly and Piomelli [52, 101, 141] for a description of the individual model adaptations and a more detailed derivation of the formulae. Due to the models high computational costs and inefficiency, it was only applied to the pipe flow (see Section 5.1.1), in all other cases the Smagorinsky model was preferred.

2.4.3 Reynolds-Averaged Navier-Stokes

In engineering applications, it is important to model practical devices and it is sufficient to capture only the main effects in a flow that are caused by turbulence. The dominant method of choice for these problems in industry has been, and probably will be in the near future, Reynolds-Average Navier-Stokes (RANS) simulations; even though LES is spreading rapidly. The RANS concept was first introduced by Reynolds [150] in the 1880s and the approach focuses on the mean flow and the effect of turbulence on mean flow properties. The starting point is the Reynolds decomposition (see Section 2.4) into a mean and its fluctuation part. The mean part can be interpreted and expressed as a temporal average for statistically stationary flows [145]

$$\overline{\phi}(x_i) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \phi(x_i, t) dt \quad (2.32)$$

where t is the time and T is the averaging time interval; the latter should go towards infinity.

Inserting the Reynolds decomposition into the Navier-Stokes equation allows the derivation of a set of equations for the temporal averages of the velocity $\overline{u_i}$ and pressure field $\overline{p}$. However, this also introduces two new unknown terms, $\overline{u'u'}$ (*Reynolds stresses*) and $\overline{u'\phi'}$ (*turbulent scalar fluxes*) in the averaged conservation equations, which cannot be presented uniquely in terms of mean quantities [43]. Deriving further transport equations for the Reynolds stresses

is possible, although these also include unclosed terms (*Turbulence Closure Problem*). In order to solve the unknown variables, various models have been introduced.

The three most important ones are the k - ϵ model, the k - ω model and the Reynolds-Stress equation models. The k - ϵ model focuses on the mechanisms that affect the turbulent kinetic energy k and the turbulent dissipation ϵ [186]. The turbulent dissipation is the rate at which velocity fluctuations dissipate. In the k - ϵ formulation, the eddy viscosity is determined from a single turbulent length scale, so the turbulent diffusion only occurs at the specified scale, whereas in reality all scales of motion contribute to the turbulent diffusion. The model performs poorly for complex flow involving severe pressure gradients, separation and strong streamline curvature. Furthermore the model is known to lack sensitivity to adverse pressure gradients [120]. The k - ω model is similar to the k - ϵ model, however ω is the turbulent frequency. The k - ω model allows for more accurate near wall treatment, which makes it superior for wall-bounded and low Reynolds number flows. Both models are often referred to as 'two equation models' and they are the most widely used and validated methods. The Reynolds-Stress models use a separate transport equation for each component of the stress tensor, which captures the directional effects of rotating flows better.

RANS allows for relative coarse computational grids and satisfactory results with well-established methods can be achieved. Furthermore most of the turbulence is modelled, allowing for efficient calculations. In general, the RANS approach is the mainstay in engineering numerical simulation due to its modest computational cost and a lot of research has been done with RANS, resulting in well validated models, even though the overall accuracy is not always satisfactory.

2.5 Combustion

Combustion processes can be characterised by three modes based on the introduction of fuel and oxidiser into the reaction zone; these are non-premixed, partially premixed and premixed combustion. The latter is used in the present work and will be discussed in greater detail below. In non-premixed combustion, fuel and oxidiser are completely separated when entering the combustion chamber and the flame stabilises around the stoichmetric point in the mixing layer. Non-premixed combustion is often regarded as safe, as large formations of unburnt mixtures of fuel and oxidiser are prevented. A majority of flames in industry are non-premixed (e.g. gas turbines, furnaces and diesel engines), even though this approach suffers from higher pollutant levels. Partially premixed combustion is a hybrid between premixed and non-premixed combustion, fuel and oxidiser are introduced separately but are then partially mixed by the transport processes before the flame has propagated to them. For example, fuel is issued from a nozzle into air and for some fuels with a large enough exit velocity, the flame lifts off the rim of the nozzle. Therefore there is room below the flame base for fuel and oxidizer to premix. The interested reader is referred to other work [51, 173, 181] for information on non-premixed and partially premixed combustion.

2.5.1 Premixed Combustion

Premixed combustion is applied, for example, in spark ignition engines or in advanced gas turbine combustors, which require a homogenous mixture of fuel and oxidiser before ignition. Whilst a non-premixed flame does not propagate, a premixed flame moves normal to its flame surface towards the unburnt mixture. Therefore the laminar burning velocity is an important quantity which depends on pressure, reactant temperature, equivalence ratio¹ and the fuel and oxidiser that are used.

Since turbulent flames are investigated in this work, the effect of turbulence on combustion also needs to be discussed. While a laminar flame remains uniform,

¹The equivalence ratio relates the fuel to air ratio of the mixture to the fuel to air ratio at stoichiometry

turbulence wrinkles the flame, increasing the active flame surface area

$$A_t \simeq A \frac{u'}{s_l} \quad (2.33)$$

The consumption of reactants is therefore enhanced [33] and can be expressed as

$$\frac{s_t}{s_l} \simeq \frac{A_t}{A} \quad (2.34)$$

Equation (2.34) links mass flux through the flame front area (A_t) to the mass flux through the cross section (A), assuming that the reactant density (ρ_u) is constant. Damköhler identified two regimes of turbulent premixed combustion, involving large scale and small scale turbulence [33]. The idea of characterising the regimes has been extended to more advanced premixed combustion diagrams, based on three non-dimensional numbers, which are discussed below. The turbulent Reynolds number Re_t can be defined as

$$Re_t = \frac{u' L_i}{\nu} = \frac{u' L_i}{s_l \delta_l} \quad (2.35)$$

where u' is the turbulent intensity, L_i is the integral length-scale, ν is the kinematic viscosity and the RHS relates the turbulence to the flame quantities.

The ratio of flame scales to the Kolmogorov scales can be defined by the two Karlovitz numbers. The first relates the chemical time-scale τ_c of the laminar flame to the Kolmogorov time-scale τ_K . The second number defines the ratio of the reaction zone thickness to the Kolmogorov length-scale [51]

$$Ka_1 = \frac{\tau_c}{\tau_K} = \left(\frac{\delta_l}{\eta} \right)^2 \quad (2.36)$$

$$Ka_2 = \left(\frac{l_\delta}{\eta} \right)^2 = \gamma^2 Ka_1 \approx 0.01 Ka_1 \quad (2.37)$$

where δ_l is the typical flame thickness, l_δ is the inner layer flame thickness and γ is the ratio of the inner layer thickness to the flame thickness ($\gamma = l_\delta/\delta_l \approx 0.1$).

The last characteristic number is the Damköhler number, which describes the relation between the chemical and turbulent time-scales, as shown in equa-

tion (2.38)

$$Da = \frac{\tau_t}{\tau_c} = \frac{L_i s_l}{u' \delta_l} \quad (2.38)$$

A Damköhler number lower than unity indicates that the turbulent eddies interact strongly with the flame structure. For $Da > 1$ the chemical reaction is faster than the mixing time and for $Da \gg 1$ combustion is characterised by the fast chemistry.

Figure 2.2 shows the premixed combustion diagram, as described by Peters [137]. The diagram is based on the comparison of typical length scales and velocity scales, which allows the characterisation of premixed combustion into a number of different regimes of turbulent-chemistry interaction. The following paragraph briefly introduces each regime; more detailed descriptions on combustion diagrams and flame structures are given in the works by Borghi and Poinot [15, 144] and [143, 173].

- The laminar regime, located in the bottom left corner of the diagram, is characterised by a turbulent Reynolds number lower than unity. Flames in this regime are expected to always burn stably.
- In the wrinkled flamelet regime, the flame front is slightly wrinkled by turbulence and the flame propagation is still dominated by the laminar burning velocity, as the turbulence intensity is lower than the laminar burning velocity. The regime is limited by the horizontal line $u'/s_L = 1$.
- Flames may be strongly wrinkled in the corrugated flamelet regime. So the active flame area increases through the wrinkling of the flame, enhancing the process of chemical reaction. The regime is limited by the Klimov-Williams criterion ($Ka_1 < 1$).
- In the thin reaction zones regime, small eddies can enter the preheat zone, but are not strong enough to penetrate the inner flame layer. Therefore the mass transport in the preheat zone is affected by turbulence as well as governed by diffusion.
- Turbulent eddies enter the inner flame layer in the broken reaction zones regime, which leads to an increased distribution of heat from the inner

layer to the preheat zone, affecting the radical species production. As a results, combustion may no longer be stable and flame extinction can theoretically occur.

The spatial structure of the flame is another important parameter, which is often characterised by the *flame thickness* and the temperature profile. Poinot and Veynante [143] define the flame thickness as the distance over which the progress variable c changes from 0 (*unburnt*) to 1 (*burnt*). In this work the description of the combustion process is based on the transport of the process variable c and the model to describe the transport is introduced below.

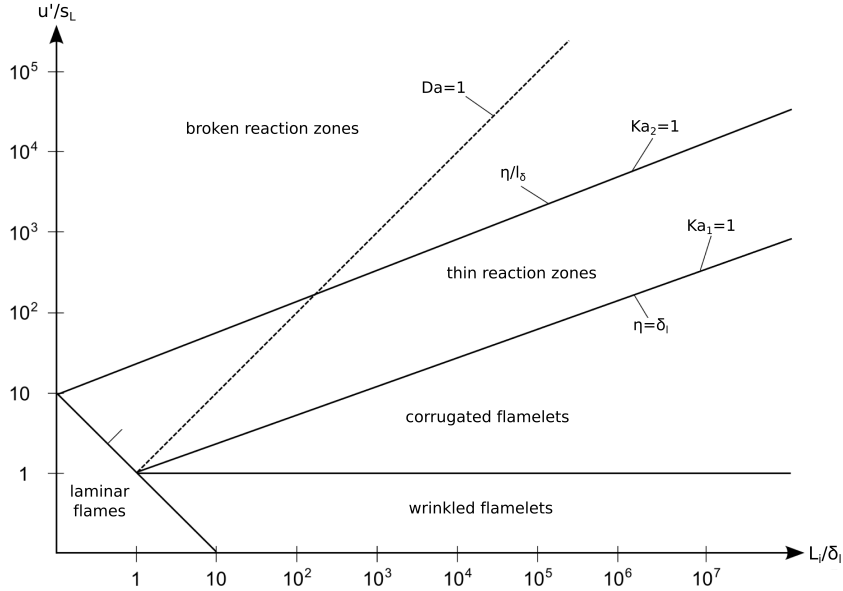


Figure 2.2: Regime diagram of turbulent premixed combustion by Peters [137]

2.5.2 Artificially Thickened Flame

One of the major issues in modelling premixed flames using LES is that the flame thickness δ_l is usually about 0.1 – 1 mm, which is often below the LES filter width. This means that the flame cannot be properly resolved, leading to numerical problems. Many approaches have been suggested to overcome this problem. The most commonly used methods are:

- Eddy-Break-Up [170]
- Eddy-Dissipation [110]

- Filtered Progress Variable [140]
- Flame Front Tracking [87, 119, 168]
- Flame Surface Density [13]
- Thickened Flame Approach [18, 21, 27]

In this work, the thickened flame (TF) approach was used to model the turbulence-chemistry interaction. From a numerical point of view, the model has the advantage of describing the reaction rate using Arrhenius expressions [27]. However, the flame structures have to be adequately resolved.

In the TF model, the computed flame front is artificially thickened to resolve it on the coarse LES grid without altering the laminar flame speed [70]. The laminar flame speed s_l and the flame thickness δ_l can be expressed as:

$$s_l \propto \sqrt{D\bar{\omega}} \quad (2.39)$$

$$\delta_l \propto \frac{D}{s_l} \quad (2.40)$$

where D is the molecular diffusivity and $\bar{\omega}$ is the mean reaction rate. Increasing the flame thickness δ_l by a flame thickening factor (T_f) is achieved by multiplying the diffusivity by T_f . A sufficiently large thickening factor allows for the flame to be resolved on the LES grid. However, to maintain original flame speed (s_l), the source term must then be divided by T_f . Thickening the flame also leads to a change in the interaction between the turbulence and chemistry, because the Damköhler number is reduced by a factor of T_f

$$Da = \frac{\tau_t}{\tau_c} = \frac{L_i}{u'} \frac{s_l}{T_f \delta_l} \quad (2.41)$$

Accordingly, the flame sensitivity to turbulence is decreased and the sensitivity to strain is increased [144]. The thickened flame model benefits from its simple implementation and, due to the Arrhenius law, it is possible to deal with more complicated chemical kinetics, as well as effects that are associated with ignition and flame-wall interaction processes. The primary disadvantage of the artificially thickened flame model is the generally increasing insensitivity of the modelled flame to the turbulence motions due to the artificial thickening of the flame brush. An efficiency function has therefore been introduced to

overcome the difficulty of the decrease in the Damköhler number and correct the flame response [27, 118]. In this work, the chosen efficiency factor E is based on the description by Colin and the interested reader is referred to his paper [27] for more information. The small scale structures in the flame can also not be represented with the thickened flame model.

The transport equation for the progress variable, modelled with the ATF approach, becomes

$$\frac{\partial(\bar{\rho}\tilde{c})}{\partial t} + \frac{\partial(\bar{\rho}\tilde{c}u_j)}{\partial x_j} = \frac{\partial}{\partial x_j}(\bar{\rho}T_f E D \frac{\partial \tilde{c}}{\partial x_j}) + \frac{\bar{S}_c}{T_f} \quad (2.42)$$

where the source term $\bar{S}_c$ is the reaction rate corrected by the efficiency factor E . Recent examples of the application of the ATF model include work by Freitag [47], Roux et al. [155] and Sommerer et al. [169].

As mentioned above, the artificially thickened flame model benefits from its simplicity and allows for larger cell sizes, as the flame thickness is artificially thickened. The reaction rate can also be estimated from chemical tables instead of using Arrhenius laws, allowing for more detailed chemical kinetics. However, the efficiency function might underestimate the action of the turbulence wrinkling of the flame. An alternative approach to model the reaction rate is using the Flame Surface Density (FSD), which is defined as the flame surface area per unit volume. Similar to the thickened flame model, the flame surface density approach uses the progress variable transport equation, but the reaction rate is closed either by the algebraic expression or solving the computationally expensive flame surface density transport equation [38, 142]. The flame surface density model measures the flame front wrinkling due to turbulence and it allows to model chemistry features and turbulence-chemistry interaction. Even though the method allows to observe the small scale structures in the flame, the chemistry is defined by only the laminar flame speed. In the opposed jet set-up, the flame position depends on the turbulence-chemistry interaction, which makes the flame surface density model less suitable.

An opposite approach to the thickened flame model, the flame front tracking approach treats the flame as a surface with zero thickness. In the approach the flame propagates freely towards the unburnt mixture. A common model is the

G -equation formulation where the flame front is represented with a constant value G_0 of the level set function G . The level set function can be chosen to be a surface of constant temperature or reaction progress variable. It takes a value of $G = 0$ at the flame front, $G < 0$ in the unburned mixture and $G > 0$ in the burned gases [142]. Similarly to the artificially thickened flame model, the G -equation formulation has been developed to overcome the problem of flame resolution. Nonetheless in the thin reaction zones regime, where the flame is broadened by turbulence, the flame structure is not resolved and has to be modelled. In addition, the assumption of an intact flame front makes the model unsuitable for the broken reaction zones [38, 173]. The G -equation formulation was not used, as the thickened flame model is computationally more efficient when using Arrhenius law with one-step chemistry.

Unlike in the thickened flame and flame surface density models, chemistry kinetics are neglected in the Eddy Break Up (EBU) model. The approach is mainly applicable to flows at high Reynolds and Damköhler numbers, where the flame is most governed by the turbulent field and chemical kinetics are negligible. In the eddy break up model, combustion is described by a single step global chemical reaction and the model assumes an infinitely thin flame. However, some species, such as CO , cannot be represented by fast chemistry and therefore cannot be predicted correctly with the eddy break up model [173]. To incorporate the significance of the fine structures in a turbulent reacting flow in which combustion chemistry is important, the Eddy Dissipation Model (EDM) has been introduced. The model is a modified version of the eddy break up approach to model the mass fraction occupied by the fine structures. However, the assumption of irreversible, one-step chemistry results in no coupling between the chemical reaction and the turbulent mixing [128].

Chapter 3

Numerical Framework

The following chapter outlines the characteristics of the Finite Volume Method (FVM) and spatial discretisation. The FVM is described on the basis of the general transport equation, along with the approximation of the convective and diffusive fluxes for the Euler-Lagrange approach used in this work. The computational grid, its advantages and disadvantages, will be described and the chosen notation will be introduced. In section 3.2 the time discretisation and time advancement are outlined, before describing the role of the pressure-solver for the present conservative transport scheme. A general overview of the boundary conditions and wall treatment is given in section 3.5, whereas the specific details of the applied immersed boundary method is given in chapter 4.

3.1 Finite Volume Method

As discussed in Chapter 2, the general transport equation can be used to describe the evolution of scalar fields in turbulent flows. However it is not always feasible, nor necessary, to compute a continuous solution. In the well established Finite Volume Method (FVM), the domain is split into a finite number of elements and the governing equations are integrated for each control volume (CV). The volume integrals in the equation are then converted to surface integrals by applying the divergence theorem to evaluate the terms as fluxes at the surfaces of each finite volume. The flux entering is equal to the flux leaving the control volume. An advantage of the FVM is its easy adaptability to unstructured meshes. The interested reader is referred to the works by Schäfer [162] or Versteeg and Malalasekera [186] for a more detailed description of the FV

method.

The principle of the FVM will be demonstrated with the transport equation for a generic scalar ϕ ; the momentum equation and mixture-fraction transport can be derived in a similar manner. The transport equation (2.3) for a scalar quantity ϕ reads:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi u_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left(D \frac{\partial \phi}{\partial x_j} \right) + S_\phi \quad (3.1)$$

and when integrated over a finite volume (control volume) ΔV can be written as

$$\int_{\Delta V} \frac{\partial \phi}{\partial t} dV + \int_{\Delta V} \frac{\partial \phi u_j}{\partial x_j} dV = \int_{\Delta V} \frac{\partial}{\partial x_j} \left(D \frac{\partial \phi}{\partial x_j} \right) dV + \int_{\Delta V} S_\phi dV \quad (3.2)$$

The convective and diffusive terms in Eq. (3.2) need to be rewritten as fluxes over the surfaces ΔA of ΔV by applying the divergence theorem [5], which relates the volume integral and surface integral by

$$\int_{\Delta V} \frac{\partial \psi}{\partial x_j} dV = \int_{\Delta A} \psi n_j dA \quad (3.3)$$

where ψ is a generic field quantity and n_j is the outward pointing normal vector of the surface ΔA . Substituting Eq. (3.3) into Eq. (3.2) yields

$$\frac{\partial}{\partial t} \int_{\Delta V} \phi dV + \int_{\Delta A} (\phi u_j) n_j dA = \int_{\Delta A} D_j(\phi) n_j dA + \int_{\Delta V} S_\phi dV \quad (3.4)$$

where the term $D_j(\phi)$ is proportional to the gradient of ϕ with the constant of proportionality being the diffusion coefficient D_ϕ

$$D_j(\phi) = \left(D_\phi \frac{\partial \phi}{\partial x_j} \right)_j \quad (3.5)$$

Solving Eq. (3.4) would still allow for an analytical solution for simplified cases. However, applying it to a finite volume has already removed information from the differential form, which applies an infinite number of points. In the next step, approximations for the volume and surface integrals are introduced to obtain an algebraic form of the governing equations. In this work the product of the cell volume and the quantity at the cell midpoint is used to estimate

the volume integral as follows:

$$\int_{\Delta V} \psi dV \approx \psi_M \Delta V \quad (3.6)$$

The approximation of the surface integrals is based on the sum of all quantities at the cell faces multiplied by the face area and the face normal pointing outwards.

$$\int_{\Delta A} \psi n_j dA \approx \sum_f \psi_f n_f \Delta A \quad (3.7)$$

Applying these approximations to Eq. (3.4) and assuming cubic control volumes (where the summation of Eq. (3.7) is simply carried out over the six square faces f) results in

$$\frac{\partial}{\partial t}(\phi_M \Delta V) + \sum_f \phi_f u_f n_f \Delta A_f = \sum_f D_f(\phi) n_f \Delta A_f + S_{\phi, mid} \Delta V \quad (3.8)$$

where n_f is the cell face normal. As quantities are often stored at the cell centres, interpolation of these values to the cell faces is necessary to apply Eq. (3.8). These will be discussed separately for the convective and diffusive fluxes for the Euler and Lagrange approach in the following sections, after a brief introduction of the computational grid and coordinate system.

3.1.1 Computational Grid

The present work utilises an in-house code called *PsiPhi*, which uses a structured Cartesian grid. The grid consists of cubic cells of length Δ ($\Delta_x = \Delta_y = \Delta_z$), where all information is stored at the cell centre. Figure 3.1 shows an example of a two-dimensional grid, where w , e , n , s denote the cell faces of the neighbouring cells to the west, east, north and south, respectively. In a three-dimensional grid there are also t (top) and b (bottom) cell faces. To fully describe the mesh, the centres of adjacent cells are labelled with W , E , N , S , T , B according to their position. The notation will be used during the discussion of the discretisation schemes for the convective and diffusive fluxes.

A structured grid offers the advantage of avoiding costly division operation for non-uniform cell lengths and simple addressing in memory, resulting in efficient loop vectorisation when compiling and running the program. Furthermore the

coupling of the Eulerian and Lagrangian fields is simplified and straight forward, as particle positions can be easily translated into grid coordinates. Using one cell-size is also consistent with the use of one filter width in LES. However a constant cell size means that the grid must be fine enough to model critical small regions correctly, but coarse enough to limit computational time.

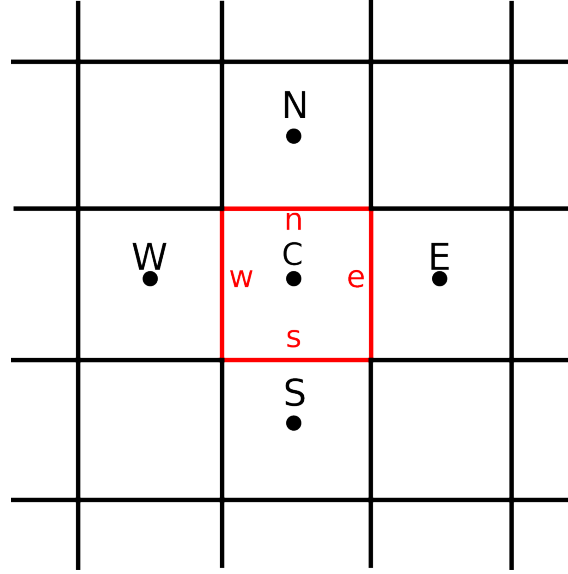


Figure 3.1: Illustration of the grid structure and grid notation

3.1.2 Description of Fluxes

This section will give a brief description of the approximation methods for the convective term. Furthermore an approximation for the diffusion term of the Eulerian approach will be discussed. The discussion of the Lagrangian diffusion is deferred to chapter 6 as it is redundant in the presented immersed boundary approach.

Different approaches have been taken for the convection of the Eulerian and Lagrangian transport. The Eulerian field will be evaluated by approximating the convective and diffusive terms over the cell faces and the Lagrangian particles will be transported by a simple equation of motion.

3.1.2.1 Convective Fluxes

As discussed in section 3.1, the convective flux $\sum_f F_{C,f}$ over cell face f of a generic quantity ϕ is given by:

$$\sum_f F_{C,f} = \sum_f \phi_f u_f n_f \Delta A_f \quad (3.9)$$

Assuming an orthogonal grid, the convective flux F_e over the eastern face of cell C in the Eulerian approach can be expressed as

$$F_e = \phi_e u_e \Delta A_e \quad (3.10)$$

The velocity u_e can be calculated by dividing the momentum by the density and needs to be interpolated to the centre of the surface A_e . The value of ϕ_e also requires an approximation by interpolation, where the method of estimation influences the behaviour of the convection scheme. Three differencing schemes (*Central Differencing scheme - CDS*, *Upwind Differencing Scheme - UDS* and *Total Variation Diminishing - TVD*) were applied for the Eulerian convection.

In CDS, the convective terms are approximated by linear interpolation between two neighbouring cells in order to compute the cell face values of a property ϕ . On a one-dimensional domain with equidistant rectangular cells it leads to

$$\phi_e = \frac{\phi_C + \phi_E}{2} \quad (3.11)$$

The central differencing scheme is second-order accurate, simple to implement and computationally efficient. However, the central differencing scheme ignores the flow direction, which can cause instabilities. The resulting oscillations are unrealistic for mixture fractions and progress variable, where the range limit is between zero and one. Outside this range, the mixture fraction and progress variable have no physical meaning and therefore CDS should be used with care or higher order accurate methods should be implemented.

The Upwind Differencing Scheme or 'donor cell' differencing scheme is suitable for convection dominated flows because it accounts for the flow direction. The cell face value ϕ_w is always equal to the upstream value of the cell face.

$$\phi_w = \begin{cases} \phi_W & \text{for } u_w > 0 \\ \phi_C & \text{for } u_w < 0 \end{cases} \quad (3.12)$$

UDS is first order accurate and is used in early CFD codes as the scheme suppresses numerical oscillations, however it can produce erroneous results from *false* or *numerical* diffusion, and therefore more sophisticated methods had to be developed.

The Total Variation Diminishing (TVD) scheme can be seen as an advancement by using the stability advantage of UDS in combination with the high-order accuracy of CDS. Many TVD schemes are second-order schemes that suppress numerical diffusion and oscillation. Total variation diminishing schemes weigh the UDS and CDS contributions based on a flux limiter function $B(r)$ and can be written as

$$\phi_e = \phi_C + \frac{B(r)(\phi_C - \phi_W)}{2} \quad (3.13)$$

assuming a positive velocity field pointing from west to east. In this work, the non-linear CHARM limiter function was used

$$B(r) = \frac{r(3r+1)}{(r+1)^2} \quad \text{if } r > 0 \quad (3.14)$$

$$B(r) = 0 \quad \text{if } r < 0 \quad (3.15)$$

where the gradient ratio r is given by

$$r = \frac{\phi_E - \phi_C}{\phi_C - \phi_W} \quad (3.16)$$

The CHARM limiter function benefits from accuracy, smoothness and stable behaviour. Therefore the TVD scheme was used in combination with the CHARM limiter function for the convective transport of the mixture fraction and progress variable. Alternative flux limiter functions and their characteristics can be found in the book by Versteeg and Malalasekera [186].

3.1.2.2 Diffusive Fluxes

Diffusion can be defined as the net transfer of mass from a region of higher concentration to a region of lower concentration. Molecular diffusion is the net

transport of molecules by random motion [6]. Simple cases in one dimension can be mathematically described as

$$\frac{\partial}{\partial t}\phi(x, t) = \frac{\partial}{\partial x}(D\frac{\partial}{\partial x}\phi(x, t)) \quad (3.17)$$

with the generic scalar ϕ and the diffusion coefficient D , as described in equation (3.1). The LHS of Eq. (3.17) defines the local change of the scalar and D defines the diffusivity of the scalar at point x at time t .

Similar to the convective flux, an approximation for the diffusive fluxes over cell face f in Eq. (3.8) is required.

$$\sum_f F_{D,f} = \sum_f D_f(\phi)n_f\Delta A_f \quad (3.18)$$

where the term $D_f(\phi)$ is proportional to the gradient of ϕ , as shown in Eq. (3.5). For the approximation of $D_f(\phi)$, a second order accurate CDS approximation is used. Assuming unity cell width, in one dimension, this corresponds to the following discretisation for the eastern cell face:

$$D_e(\phi) \approx D_\phi \frac{\phi_E - \phi_C}{\Delta x} \quad (3.19)$$

where D_ϕ is the diffusion coefficient and Δx is the distance separating the two points. Using Eq. (3.19) and assuming an orthogonal grid, the diffusive flux can be written as

$$F_{D,e} = D_e(\phi)n_e\Delta A_e \approx D_\phi \frac{\phi_C - \phi_E}{\Delta x}\Delta A_e \quad (3.20)$$

The equation allows for the second order approximation of the diffusive term on orthogonal grids and is used in this work as part of the PsiPhi code.

3.1.3 Particle Transport

Lagrangian particles can be added to the simulations to describe numerical particles in the flow. The change in particle locations are described by the transport equation. Hence, the particle locations are functions of position and time; so any particle position can be described with a simple differential equation and the position of each particle can be tracked as changes are computed.

The use of a cubic Cartesian grid simplifies the tracking of particles without the need of long and computationally expensive search algorithms to determine the particle position. Knowing the exact particle location and velocity allows particles to be transported by applying a simple equation of motion that is computationally inexpensive and simple to implement, as shown in Eq. (3.21).

$$\vec{x}(t) = \vec{x}(t_0) + \int_{t_0}^{\Delta t} \vec{v} \delta t \quad (3.21)$$

where $\vec{x}$ is the location and v is the velocity. If the particle motion is influenced by the flow field, the particle velocity in a three dimensional domain is calculated from tri-linear interpolation. In this work particles are used to describe objects which move independently of the flow field; hence a particle velocity is prescribed. The particles are advanced in time with the same time integration scheme as the Eulerian fields, in this work either the third order low-storage Runge-Kutta or Euler explicit (see Section 3.2). As the wall movement is driven by an external force, independent of the flow field, only one way coupling is considered.

3.2 Time Discretisation

The numerical solution for the partial differential equations needs to be advanced in time in order to describe the unsteady turbulent flow. The following sections will focus on the Euler explicit scheme (EE), Euler Implicit scheme (EI) and the Runge-Kutta scheme (RK). Other schemes are available, such as the Leap-Frog method (LF). However, the focus will lie on the EE and RK methods as these were used during this work.

As a first step, the transport equation (2.7) for a generic scalar quantity ϕ , is rewritten as

$$\frac{\partial \phi}{\partial t} = -\frac{\partial \phi u_j}{\partial x_j} + \frac{\partial}{\partial x_j} \left(D \frac{\partial \phi}{\partial x_j} \right) + S_\phi = \mathfrak{S}(\phi) \quad (3.22)$$

where the convection term has been moved to the right hand side and $\mathfrak{S}(\phi)$ is used to represent the effects of convection, diffusion and sources/sinks. The total simulation time is split into a number of time intervals referred to as time steps m , each lasting a duration of δt . With this in mind, the more convenient form of Eq. (3.22), written as

$$\frac{\partial \phi}{\partial t} = \mathfrak{S}(\phi) \quad (3.23)$$

is then discretized in time by the schemes outlined in the following subsections.

3.2.1 Euler Explicit

The simplest method for the time integration is the Euler Explicit scheme. The solution is computed based on the previous known time step by the expression

$$\phi_{m+1} \approx \phi_m + \mathfrak{S}(\phi_m) \delta t \quad (3.24)$$

Though the method is computationally cheap, it is only first order accurate ($O(\delta t)$). Therefore the solution may include large errors, unless a small time-step width is used. Furthermore the EE scheme is generally unstable with the Navier-Stokes equation if the value of the Courant-Friedrichs-Lewy (CFL) limit (see Section 3.2.4) is greater than unity. Hence the use of the EE scheme should be avoided, as discussed in more detail by Ferziger and Perić [43].

3.2.2 Euler Implicit

The one step scheme can be stabilised by discretising $\mathfrak{F}(\phi)$ at the new time-step $(m + 1)$. This method is called the Euler Implicit (EI) scheme.

$$\phi_{m+1} \approx \phi_m + \mathfrak{F}(\phi_{m+1})\delta t \quad (3.25)$$

The first order accurate EI scheme is more stable and allows for time-steps larger than unity without becoming unstable. However it is dissipative, more complex and computationally more expensive. Additional computing effort is required to solve a system of equations for the unknown values at the new time step.

3.2.3 Runge-Kutta

The Runge-Kutta scheme is a sub-step scheme, where additional points between the time-steps δt_m and δt_{m+1} are used. The scheme is computationally more expensive in comparison to the EE scheme, however its simplest form is of second order accuracy, consisting of two steps. The first step uses a half step as a predictor step (Eq. 3.26), which is based on the Euler explicit scheme. The second step is a mid-point corrector (Eq. 3.27), as shown below.

$$\phi_{m+\frac{1}{2}} \approx \phi_m + \mathfrak{F}(\phi_m)\frac{\delta t}{2} \quad (3.26)$$

$$\phi_{m+1} \approx \phi_m + \mathfrak{F}(\phi_{m+\frac{1}{2}})\delta t \quad (3.27)$$

The more popular form of the Runge-Kutta scheme is a fourth order scheme, where the first two steps use an explicit Euler predictor and an implicit Euler corrector. This is followed by a mid-point predictor and Simpson's rule final corrector [43].

In this work, a third order low-storage Runge-Kutta scheme with weighted sub-time steps was used [148]. The applied scheme has lower memory requirements than the fourth order form and it is more accurate than the simplest form. The scheme has also been applied successfully in other studies [130, 139, 173, 194]. The applied scheme consists of three sub-steps and each sub-step advances by explicit Euler time integration of the previous intermediate step until a

time-step is completed. The sub-steps are weighted as follows

$$\delta t_1 = \frac{1}{3}\delta t, \quad \delta t_2 = \frac{5}{12}\delta t, \quad \delta t_3 = \frac{1}{4}\delta t$$

and the Runge-Kutta scheme can be written as

$$\phi_{m+\frac{1}{3}} \approx \phi_m + (\alpha_1 \mathfrak{S}(\phi_m))\delta t_1 \quad (3.28)$$

$$\phi_{m+\frac{3}{4}} \approx \phi_{m+\frac{1}{3}} + (\alpha_2 \mathfrak{S}(\phi_{m+\frac{1}{3}}) + \beta_2 \mathfrak{S}(\phi_m))\delta t_2 \quad (3.29)$$

$$\phi_{m+1} \approx \phi_{m+\frac{3}{4}} + (\alpha_3 \mathfrak{S}(\phi_{m+\frac{3}{4}}) + \beta_3 \mathfrak{S}(\phi_{m+\frac{1}{3}}) + \frac{\beta_2}{\alpha_2} \mathfrak{S}(\phi_m))\delta t_3 \quad (3.30)$$

where the coefficients in Eq. (3.28 - 3.30) have the following values

$$\alpha_1 = \frac{1}{3}, \quad \alpha_2 = \frac{15}{16}, \quad \alpha_3 = \frac{8}{15}, \quad \beta_2 = -\frac{75}{144}, \quad \beta_3 = -\frac{51}{81}$$

3.2.4 Time-step width

To ensure stability and accuracy of explicit time integration, such as the Euler Explicit, the flow information should be transported to neighbouring cells without 'jumping across cells' in a single time-step. This can be achieved by applying the Courant-Friedrichs-Lewy condition to the time-step width. The accuracy increases with a reduction in time-step width, similar to spatial discretisation. Courant et al. [29] suggested that the CFL condition can be evaluated from the velocity u , time-step width δt and the cell dimensions Δ and it can be expressed as follows

$$CFL = \frac{u_{max}\delta t_{max}}{\Delta} \Rightarrow \delta t_{max} = \frac{\Delta}{u_{max}}CFL \quad (3.31)$$

Therefore the maximum time-step width is evaluated based on the cell dimensions and the velocity and the information must not be transported further than a distance corresponding to the CFL limit times the cell width. If the time-step width is larger than permitted by the CFL condition, the simulation will diverge. No distinction between the different axes needs to be made due to the use of cubic cells.

3.3 Predictor-Corrector Algorithm

The accurate scalar transport is important for Large-Eddy simulations in order to conserve mass and momentum. Even though a Runge-Kutta scheme is used for the advancing of the flow fields, continuity has not been considered yet. Therefore the *PsiPhi* program uses a predictor-corrector scheme to conserve mass and momentum. A density (mass) error due to transport and wall movement is predicted to then correct the momentum field accordingly. The scheme was developed by Kempf [82] and it was shown to work well in the past [85, 139, 195]. The algorithm is briefly outlined in the following paragraphs; a detailed description can be found in the work by Kempf [82].

In the predictor-step, the transport equations for the filtered density $\bar{\rho}$ and filtered product of a scalar and density $\bar{\rho}\tilde{\phi}$ are solved based on the velocity field of the previous time-step. The transport process then leads to the predicted values $\bar{\rho}^{PD}$ and $\bar{\rho}\tilde{\phi}^{PD}$ and the related scalar $\tilde{\phi}$ can be derived by dividing $\bar{\rho}\tilde{\phi}^{PD}$ by $\bar{\rho}^{PD}$. In this description, the transported scalar is used to represent the mixture fraction $\tilde{f}$ for isothermal flows and the progress variable $\tilde{c}$ for premixed combustion. Nonetheless, the filtered scalar $\tilde{\phi}$ will be used, to keep the further description of the predictor-corrector algorithm more generic.

In the next step of the scheme, the chemical sub-model is used to relate the scalar $\tilde{\phi}$ to the corresponding chemical state and the target density from the sub-model is denoted $\bar{\rho}^{CS}$. At this point, the target density differs from predicted density because the velocities were advanced without considering the pressure term at the end of the previous time step and without considering density changes due to chemical reactions. Therefore the velocity field needs to be adjusted so that the transport with the corrected velocities will result in the target density $\bar{\rho}^{CS}$ instead of the predicted density $\bar{\rho}^{PD}$. The necessary corrections of the momentum and velocity fields are based on a pressure correction algorithm, which is described in section 3.4.

In the following correction step, $\bar{\rho}\tilde{\phi}$ are transported again using the corrected velocities. The resulting transported density is now much closer to $\bar{\rho}^{CS}$ and if necessary, this can be extended to an iterative process to further reduce the error. The stability of the predictor-corrector algorithm is improved by

applying a relaxation factor α to the density correction. This can strongly reduce the fluctuations in the density field and thus smoothen the density field with convection [82]. In comparison to the standard projection method for constant density flows, in which the divergence of the velocity field is calculated to apply the correction, the density and mixture fraction/progress variable is transported in the described method. These transported values and the values from the chemical progress are then used to predict the density error, on which the correction is calculated.

3.4 Pressure Correction

As mentioned above, the predictor-corrector algorithm requires corrected momentum and velocity fields to ensure that mass does not accumulate. In the presented work, a projection method [23, 82] was used to calculate the pressure field, which was subsequently applied to adapt the momentum and velocities to fulfil continuity. The projection method will be described based on the unfiltered momentum equation, where the viscous term, convective term and source/sink term are denoted by T_v , T_c and T_s , respectively. The momentum equation, in its differential form, then reads

$$\frac{\partial(\rho u_i)}{\partial t} = T_v + T_c + T_s - \frac{\partial p}{\partial x_i} \quad (3.32)$$

The following approach is equally valid for the filtered LES equations. Integration of Eq. (3.32) over a time step width δt yields the momentum at the new time level $t + \delta t$.

$$(\rho u_i)_{t+\delta t} = (\rho u_i)_t + \int_t^{t+\delta t} (T_v + T_c + T_s) d\tau - \int_t^{t+\delta t} \frac{\partial p}{\partial x_i} d\tau \quad (3.33)$$

At this point the pressure is unknown, so the third term on the RHS is neglected in the first step and the equation can be rewritten for the predicted momentum $(\rho u_i)^{PD}$

$$(\rho u_i)^{PD} = (\rho u_i)_t + \int_t^{t+\delta t} (T_v + T_c + T_s) d\tau \quad (3.34)$$

The change (*predicted correction*) in momentum $\Delta(\rho u_i)$ can be expressed as

$$\Delta(\rho u_i) = (\rho u_i)_{t+\delta t} - (\rho u_i)^{PD} \quad (3.35)$$

and subtracting Eq. (3.34) from Eq. (3.33) yields

$$\Delta(\rho u_i) = - \int_t^{t+\delta t} \frac{\partial p}{\partial x_i} d\tau = - \frac{\partial}{\partial x_i} \int_t^{t+\delta t} p d\tau \quad (3.36)$$

where for the RHS, it is assumed that the spatial derivative does not vary with respect to time integration. Therefore Eq. (3.36) can be rewritten as

$$\Delta(\rho u_i) = - \frac{\partial P}{\partial x_i} \delta t \quad (3.37)$$

with the mean pressure P over the time interval δt

$$P = \frac{1}{\delta t} \int_t^{t+\delta t} p d\tau \leftrightarrow \int_t^{t+\delta t} p d\tau = P \delta t \quad (3.38)$$

Based on Eq. (3.37), the *Poisson* equation for pressure can be derived by taking the spatial derivative

$$\frac{\partial \Delta(\rho u_i)}{\partial x_i} = - \frac{\partial^2 P}{\partial x_i^2} \delta t \quad (3.39)$$

In Eq. (3.39), both sides are unknown, however an expression for the momentum term on the LHS can be formulated by writing a continuity equation for the density correction $\Delta \rho = \rho_{t+\delta t} - \rho^{PD}$ and the momentum correction $\Delta(\rho u_i)$.

$$\frac{\partial \Delta \rho}{\partial t} + \frac{\partial \Delta(\rho u_i)}{\partial x_i} = 0 \quad (3.40)$$

Equation (3.39) can then be rewritten as

$$\frac{\partial^2 P}{\partial x_i^2} \delta t = \frac{\partial \Delta \rho}{\partial t} \approx \frac{\rho_{t+\delta t} - \rho^{PD}}{\delta t} \quad (3.41)$$

where the RHS of equation (3.41) is known at the end of the prediction step, as $\rho_{t+\delta t}$ ¹ is derived from the chemical state and ρ^{PD} is known from the transport with the uncorrected velocity field. Integrating the known equation yields the

¹noted as $\bar{\rho}^{CS}$ in Section 3.3

pressure field required for the correction of the momentum field. Once the pressure field is known the corrected momentum field can be calculated from

$$\Delta(\rho u_i) = -\frac{\partial P}{\partial x_i} \delta t \quad (3.42)$$

At the end of the prediction step, the correction is applied to the momentum field to yield the correct momentum at time level t

$$\Delta(\rho u_i)_{t+\delta t} = \Delta(\rho u_i) + (\rho u_i)^{PD} \quad (3.43)$$

The same procedure is applied to the velocity field to ensure the coupling of the momentum and velocity field, rather than calculating the momentum and then interpolate the velocities. A more detailed description of the predictor-corrector scheme has been given by Stein [173].

3.5 Boundary Conditions

CFD problems are all defined in terms of initial and boundary conditions, which have to be specified by the user correctly to get meaningful results. For a finite volume method, the most common boundary conditions are:

- Inlet
- Outlet
- Wall
- Symmetry
- Periodicity

In this section, the focus of discussion of the boundary conditions will be given to the inlet, outlet, and periodic boundary conditions, as these were used in the presented work. The detailed discussion of wall boundary conditions is deferred to chapter 4. For information on symmetric boundary conditions, the interested reader is referred to the book by Versteeg and Malalasekera [186].

For the opposed jet (see Section 5.1.3) and piston-cylinder assembly (see Section 5.2.2) test cases, *Dirichlet* boundary conditions were used at the inlet, in which a value of a given quantity ϕ (e.g. velocity) is specified at the boundary of the domain as a known value ϕ_{inlet} so that

$$\phi = \phi_{inlet} \tag{3.44}$$

Setting a value for the inflow velocity in the LES simulation is non-trivial, especially for turbulent flows. However non-laminar flows require some turbulent fluctuations, which can be *artificially* generated. The *artificially* generated turbulent field at the inflow should satisfy statistical properties, which are similar to the experimental inflow.

3.5.1 Turbulent Inlet Conditions

The simplest, and often preferred method, is to superimpose random fluctuations (in terms of random noise) onto a mean velocity field. However, the

random noise for each cell at the inflow is spatially not correlated and the small fluctuations will dissipate within a few cells of the domain. Klein et al. [90] suggested a method to generate turbulent inflow conditions that satisfy a prescribed length-scale and Reynolds stress tensor, the method was later extended by Kempf et al. [84] to arbitrary grids.

The method generates three fields of random noise r^a , one for each velocity component ($a = [u, v, w]$), to which a digital filter is applied. Each field has to be created greater in cross-section than the inflow plane, and a stream-wise extension that exceeds $U_b t$, where t is the time simulated. The random fields are convoluted with a discrete low-pass filter $B_{i,j,k}$ to obtain the filtered fields R^a

$$R_{I,J,K}^a = \sum_{i=-N}^N \sum_{j=-N}^N \sum_{k=-N}^N B_{i,j,k} r_{I+i,J+j,K+k}^a \quad (3.45)$$

The coefficients $B_{i,j,k}$ must be chosen such that a prescribed integral length-scale L_i is recovered by filtering the random noise to satisfy the modelled auto-correlation function R_{xx} for fully-developed, homogeneous turbulence [9], expressed in logical coordinates

$$R_{xx}(i, j, k) = \exp\left(-\frac{\pi(i^2 + j^2 + k^2)}{4l^2}\right) \quad (3.46)$$

where l is the normalised length-scale $l = L_i/\Delta$. The filter operation is then iteratively applied to the fields of random noise until the prescribed turbulent length-scale is achieved. The fields are renormalised with zero mean and variance of one to avoid the reduction in amplitude by the filtering process. To satisfy the prescribed Reynolds stress tensor, the three independent signals are then superimposed according to a procedure suggested by Lund et al. [107]. This leads to a fluctuation field, which is added to the mean velocity and the process is repeated for each new inflow slice in the simulation.

Furthermore the filter can be expressed as a tensor product of one-dimensional filters ($B_{i,j,k} = b_i b_j b_k$), which permits an efficient implementation at low computational costs by subsequently applying the filter in each direction.

$$R_{I,J,K}^a = \sum_{i=-N}^N b_i \sum_{j=-N}^N b_j \sum_{k=-N}^N b_k r_{I+i,J+j,K+k}^a \quad (3.47)$$

Recently, the method was further advanced by Kempf, Wysocki and Pettit [86] to improve its parallel efficiency and memory requirements. Nowadays, modern CFD codes run on hundreds (even thousands) of CPUs by parallelising the code and using domain decompositions. These decompositions are then distributed to individual processors and information of the block boundaries is communicated between them, including information for the generation of the artificial inflow turbulence to ensure continuity of the inflow conditions. The required communication for the inflow data can become computationally expensive, introducing unnecessary overheads and reduce the parallel efficiency. Furthermore, storing the filtered fields can significantly effect the memory requirements of the code. These problems can be avoided by exploiting the *deterministic* characteristic of computer-generated random number series.

For each local domain of the size $[S_i, S_j, S_k]$, the creation of turbulent inflow slices requires a domain of size $[1 + 2N, S_j + 2N, S_k + 2N]$. A *halo* region of width N is included in all directions, to enable filtering of all of the points within the inflow plane (Fig. 3.2). There is therefore an overlap of N inflow cells with each neighbouring domain. Normally the values in the halo region would be obtained from communication with the neighbouring domain. However the communication becomes obsolete if the random number generator is seeded with deterministic random numbers that are obtained as an integer function of the global cell co-ordinates $[i_g, j_g, k_g]$. Therefore every processor can obtain the same random number for the same point in physical space, as the logical co-ordinates only depend on the physical location in the global domain. Overall the cost for creating individual random numbers is slightly increased, however highly costly communication is avoided, improving the overall computational efficiency.

The previously described modifications make the creation of pseudo-turbulent velocity fields during the simulation initialisation obsolete, as inflow data can easily be generated at the simulation run time. Each inflow slice requires a domain of size $[1 + 2N, S_j + 2N, S_k + 2N]$. Once each slice of random numbers has been filtered in the j - and k -directions, these slices are 'stacked up' and filtered in the i -direction to produce a field of normalised filtered random numbers, which is then scaled to obtain the desired turbulent kinetic energy. After a time δt has been simulated, the 'oldest' inflow slice is removed from

the stack and a new inflow slice (filtered in the j - and k -direction) is added, before the stack is filtered again in the i -direction.

3.5.2 Outflow Conditions

The outflow boundary conditions are often specified with a von Neumann condition. The conditions prescribe a constant c_{outlet} in terms of the gradient of the variable ϕ in the direction of the normal vector n_j of the boundary

$$\frac{\partial \phi}{\partial x_j} n_j = c_{outlet} \quad (3.48)$$

In this work, a zero gradient condition ($c_{outlet} = 0$) with enforced outflow (clipping of negative velocities) was used to improve simulation stability. An alternative to the zero gradient condition is the linear gradient condition, which is based on the gradient of the last 2 cells in the domain.

3.5.3 Periodic Conditions

The third type of boundary conditions used in this work are periodic or cyclic boundary conditions [186]. To apply periodic boundary conditions, the scalar quantities entering the domain at the inlet are equal to the scalar quantities leaving the domain at the outlet. These conditions are also trivial to implement on the structured grid, by equating the values of each variable at the nodes just upstream of the inlet plane to the values just downstream of the outlet plane. For example, the periodic boundary condition allows for the simulation of a flow in an infinitely long pipe on a relative short computational domain (see Section 5.1.1).

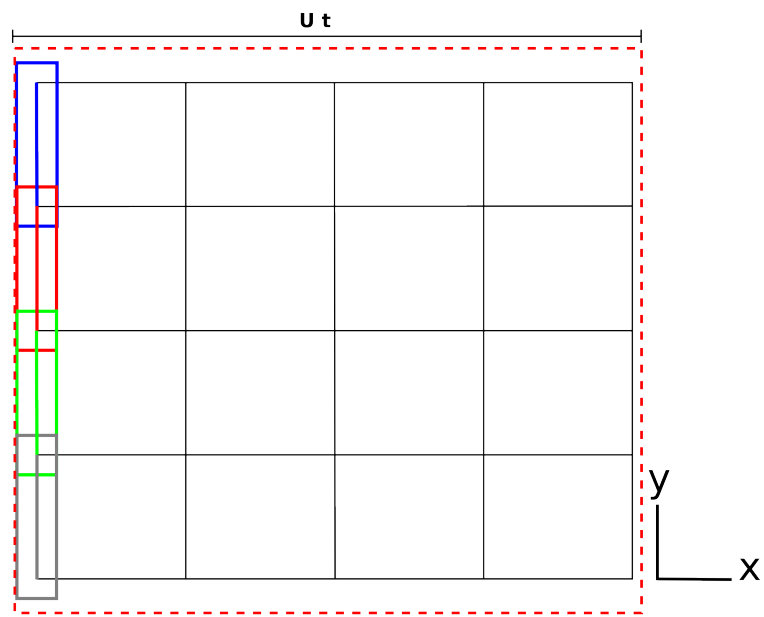


Figure 3.2: A typical 2D domain decomposition across 16 processors. The inflow domain is indicated by the coloured boxes, and shows the overlapping halo region. The red dashed box indicates the domain size (dependent on the prescribed length-scale) that must be stored for the original method of the inflow generator.

Chapter 4

Immersed Boundaries

The immersed boundary method is described in the following chapter. An overview of related boundary methods is given at the beginning of the chapter, which focuses on Immersed Boundary (IB), Marker-and-Cell (MAC) and Particle-in-Cell (PIC) methods, as they are directly related to the presented work. Alternative methods, not based on the Navier-Stokes equation such as the Lattice Boltzmann method for moving boundaries, are also mentioned. Subsequently the applied IB method is outlined and a numerical description of the forcing term added to the governing equations is given. The chapter concludes with a brief discussion on the methods applicability to high performance computing.

4.1 Overview

In the immersed boundary method, the governing equations, configured for complex geometries, can be solved on simple meshes by applying *forcing* conditions in specific areas where boundaries are present. Many researchers have applied the IB approach in the previous decades and in this overview, the main focus will be given to flows with solid immersed boundaries. For overviews of liquid-liquid or liquid-gas boundaries, the interested reader is referred to Anderson et al. [3] and Scardovelli and Zaleski [161]. The primary advantage of immersed boundary methods is the non-body conforming grid, which simplifies grid generation greatly. Furthermore the method is also computationally very efficient and it is favoured for its simplicity in dealing with moving boundaries

as grid refinements can be avoided. However, it is difficult to enforce accurate boundary conditions, especially along curvature boundaries cutting a cell. The method is also often limited to first order accuracy.

The term *immersed boundary* was first introduced by Peskin [134] in 1972, who used the approach to model blood flow in the heart. The two-dimensional Cartesian grid did not conform with the geometry of the heart and a novel IB procedure was used to impose the boundaries. A singular force, in form of a delta function, was added to the Navier-Stokes equation for the modelling of the boundary conditions. However, the method suffered from mass loss by having permeable walls. Another major disadvantage of the method was the smearing of the boundaries by applying the forcing term over 3 – 4 neighbouring cells to improve stability. This causes a problem if the boundaries move as a result of the fluid flow, however can be easily solved if the boundary is driven by an external force. Later, Peskin [135] and McQueen and Peskin [117, 136] applied the IB method successfully to flows in the heart in three-dimensional domains.

Since the successful application by Peskin, many subsequent IB methods have been developed. To the author’s knowledge, the first application of the IB approach to un-deformable immersed boundaries was carried out by Basdevant and Sadourny [8] and Goldstein et al. [60]. Basdevant and Sadourny studied unsteady flows around circular cylinders in two-dimensional domains for different Reynolds numbers ($Re \leq 1000$). Goldstein et al. applied a feedback forcing term to simulate the start-up flow around static circular cylinders and three-dimensional turbulent channels. Both groups used the IB approach in conjunction with spectral methods and the forcing term was applied over multiple cells to reduce numerical oscillations near the wall and therefore improve the numerical stability. Later, Saiki and Biringen [157] used a similar forcing approach to Goldstein et al. [60] to compute the flow around a rotating circular cylinder. They applied a 4th-order central finite-difference approximation for the forcing term which reduced the oscillations experienced by the other groups. However the three methods share one major disadvantage. The forcing term is applied over multiple cells and requires two model constants to calculate the forcing term. These constants also need to be tuned to the problem. Furthermore, simulations of unsteady flow problems require very small

time-steps, which limits the numerical efficiency. Satisfactory results for unsteady flows around a circular cylinder for low to moderate Reynolds numbers were also presented by Briscolini and Santangelo [17].

Mohd-Yosuf [124] developed an alternative forcing approach without the need of smoothing near the boundaries and the stability problems. Furthermore, parameters no longer need to be adjusted to geometries. He applied the approach to laminar flows in a three-dimensional channel and good results were achieved. Similar to the work by Mohd-Yosuf [124], Fadlun et al. [42] introduced a direct forcing non-boundary conforming approach in which the forcing term is defined on the first grid node outside the boundary by linear approximation of the velocities. However, the solidity of the walls was problematic in complex geometries. Therefore, Majumdar et al. [111] suggested the use of a quadratic two-dimensional stencil, which was applied to a couple of laminar flow problems, to improve the solidity. More recently Verzicco et al. [187] applied the forcing approach by Mohd-Yosuf [124] in the framework of a finite difference LES code to investigate a piston-cylinder arrangement and results were promising. A more general scheme, based on the work by Fadlun et al. [42] and Majumdar et al. [111], was investigated by Balaras [7] to derive the direct forcing term with a highly accurate interface tracking scheme. The approach has only been applied to simple geometries with low Reynolds number flows and the distribution of tracking points is a complex task.

A higher-order accurate immersed boundary scheme was proposed by Cortez and Minion [28], who used a smoothed projection in combination with a delta function to compute the forcing term. However, the application of the delta function leads to the same smearing of the boundaries as in [117, 136]. Lai and Pesking [97] tried to improve the original IB method [135] to second-order accuracy by calculating the Dirac Delta function with a grid independent fixed smooth function. In practice, the method remains first-order accurate because the Dirac Delta function causes jumps at the interface. Two implicit second-order accurate methods have been compared by Mark and van Wachem [112]. The first approach uses triangle vertices as control points and velocities at non-grid conforming boundaries are interpolated. The second method is based on Mohd-Yusof's forcing approach [124], but rewritten to be fully implicit, resulting in no flux over the immersed boundaries and no oscillations near the

boundary. The methods mainly differ in the convergence rate and stability, with the second being more stable.

Although the term *immersed boundary* was first introduced by Peskin [134], similar ideas existed before. One of them is the Particle-in-Cell (PIC) method, which was first introduced in the 1950's by Harlow et al. [41, 63], and strictly speaking is not a pure boundary method. Particles are used to carry material position, mass and species in order to simulate time-varying behaviour of materials undergoing distortions. The poor translation from particle properties to the underlying Eulerian grid leads to low accuracy as a result of numerical diffusion. Many attempts have been made to solve these issues, but none succeeded [65]. A semi-successful advancement of the PIC method is the Fluid-Implicit-Particle (FLIP) method [16]. A full Lagrangian representation is used for the fluid, which eliminates the diffusion. A good level of accuracy was achieved, especially with increasing number of particles, however the method is computationally very expensive.

The second method that should be mentioned is the Marker-and-Cell (MAC) method, which was developed by Harlow and Welsh [66, 191] in the 1960's. Based on the PIC method, it was developed for free surface flows. The interface is treated as a free surface, tracked by virtual Lagrangian particles and pressure boundary conditions are imposed across the surface. However, the method was greatly limited to simple domains and small time-step widths for stability reasons. The successor of the the MAC method is referred to as AB-MAC (Arbitrary Boundary MAC) method [188], which is a generalised form that can also deal with moving boundaries. Rider and Kothe [153] presented an implicit marker and cell method based on the early development of the original PIC and MAC methods. They used particles to track the fluid phase and velocities were calculated from bilinear interpolation. The method has been applied to compressible flow problems, but is limited by the high computational cost of transporting the large required number of particles.

The Cartesian grid method, another class of methods, is often used to simulate viscous flows with embedded solid boundaries on Cartesian grids [10, 26, 34]. The Cartesian grid methods rely on non-body fitted meshes without using a force term to impose the correct boundary conditions. The grid cells at

the body interface are modified according to the intersection with the underlying grid. Although, depending on the body and the geometry, a large number of cut interface cells may be created and these require special treatment. Having similar capabilities as the IB methods, Udaykumar et al. [183] and Ye et al. [197] used the Cartesian grid method to study unsteady viscous flows. Gilmanov and Sotiropoulos [58] proposed a hybrid Cartesian/immersed boundary method to simulate a flapping wing and a fish. They applied unstructured grids and achieved a solution with second-order accuracy. However, the method requires the generation of very complex grids to obtain good results.

Alternative classes for the simulation of immersed boundaries in steady and unsteady flows are the front tracking Arbitrary Euler-Lagrange (ALE) method, the level-set method and the Lattice Boltzmann method. The front tracking ALE method is used by Jaeger and Carin [77] and Hirt et al. [71] in combination with unstructured finite element meshes. The mesh is continuously adapted by the nearest node to the interface to maintain a sharp solution. Osher and Sethian [132] proposed the level-set method and the interface is simply captured by a contour of a scalar function. The method works well for simple geometries, especially with respect to moving and merging surfaces and the interested reader is referred to the books by Sethian [165, 166] for more information. Overall the approach has a higher numerical error than the front-tracking methods and mass conservation is also a problem.

Cui et al. [32] and Lallemand and Luo [98] apply the Lattice Boltzmann method to problems with moving solid surfaces in fluids. They use a simple bounce-back scheme to mimic particle boundary interaction for no-slip boundary conditions by reversing the particle momentum near walls. The method is second-order accurate and has been used in other studies [59, 69, 108].

The concept of immersed boundaries is closely linked to the fundamental aspects of flow behaviour in the near wall regions in applications where viscous effects have a strong impact on the region of interest. More recently, an increasing popularity in using LES for the simulations of highly unsteady engineering flows motivates the use of boundary methods. However, scalability of IB methods, complicated implementations with tuneable parameters and high

computational costs of methods pose limitations to their applicability.

The presented immersed boundary method has been developed for Large-Eddy simulations of engineering flows at low to moderate Reynolds numbers, where boundary layer effects are negligible in the region of interest. The simplicity of the method allows for computationally efficient simulations of isothermal and reactive flows in complex geometries, while maintaining a good level of accuracy in the main flow region. Furthermore the technique also works effectively for moving objects, with no significant increase in computational costs.

4.2 Euler-Lagrange Methods

Flows in engineering problems are often constrained by walls, which can be anything from a straight pipe to a complex turbine assembly. The constraints need to be represented accurately and wall effects need to be incorporated in order to gain an accurate flow representation. As mentioned before, unstructured grids can be used for a more accurate consideration of wall effects in complex geometries. However, these grids are often difficult to generate and adversely affect the computational efficiency. Structured grids, on the other hand, are more efficient and less complex to work with.

In this section, the Euler-Lagrange method for immersed boundaries is presented. The primary advantages of IB methods are based on the fact that the grid generation is greatly simplified, the methods are computationally efficient, no grid refinement near the boundaries is required and the approach has been successfully applied for low to moderate Reynolds number flows [42, 60, 124]. Resolving the boundary layer accurately in higher Reynolds number flows is more challenging, especially if the boundary surface is not aligned with the grid lines. However in many engineering problems the region of interest is often away from the surfaces, especially in reactive applications such as the opposed jet flow (see Chapter 5), an internal combustion engine, the combustion process in a gas turbine or coal combustion in power plants. The Navier-Stokes equation is reviewed to account for the force term and the section concludes with a short discussion of the method's suitability for high performance computing.

In this work, wall positions are tracked with Lagrangian particles on an underlying Eulerian grid. In other words, particles are used to distinguish between fluid and wall cells. Mapping a particle position to a cell on the Eulerian grid identifies the cell as an immersed boundary and appropriate boundary conditions are imposed if the boundary cell has neighbouring fluid cells, as illustrated in Fig. 4.1. The grey coloured cells identify a solid object and the region outside the object is the *free fluid* region. The red line indicates the cell faces where appropriate boundary conditions have to be imposed to ensure impermeability. At this point it should be mentioned that the term *wall* also refers to the interior of a solid object, and not just the interface between the

liquid and solid state.

Particle positions can be directly translated (simple hashing) to the Eulerian grid by converting the physical particle position to the Cartesian grid coordinates, according to

$$[i, j, k] = [\frac{x}{\Delta_x}, \frac{y}{\Delta_y}, \frac{z}{\Delta_z}] \quad (4.1)$$

where $[x, y, z]$ are the axial positions of a particle p and Δ is the cell width in that dimension. The use of an equidistant Cartesian grid ($\Delta_x = \Delta_y = \Delta_z$) simplifies the translation from particle positions to the Eulerian grid and makes the identification of cells with particles inside straight forward and computationally efficient. For example, Mittal and Iaccarino [123] need to use a search algorithm to track their Lagrangian boundary position on the unstructured Eulerian grid, introducing additional computational overheads. The use of an unstructured grid would also increase the complexity of interpolation schemes as cell sizes have to be taken into account. Unlike in other methods, the proposed method has only two cell types, wall and fluid cells. Although cells in the near wall region are subjected to special treatment to ensure impermeability of walls.

Furthermore, the tracking and mapping of particles is only necessary for moving wall problems. For static problems, the particle positions remain constant, making the operation of particle tracking obsolete, which improves the computational efficiency significantly. The mapping of particles onto the Eulerian grid can also be computationally expensive, especially if a large percentage of the domain contains Lagrangian particles. In order to avoid unnecessary computations, especially for non-moving boundary problems, an immersed boundary (IB) helper field is used to improve the computational efficiency for the treatment of the near wall region, which is discussed in more detail in the following subsections. The IB-helper field takes value unity for fluid cells and zero for wall (particle) cells and only needs to be updated at the beginning of the simulation (static) or each time-step (moving boundary problems). This allows for computational efficient array arithmetic on the transported scalar fields without having to repetitively map particles to the Eulerian grid.

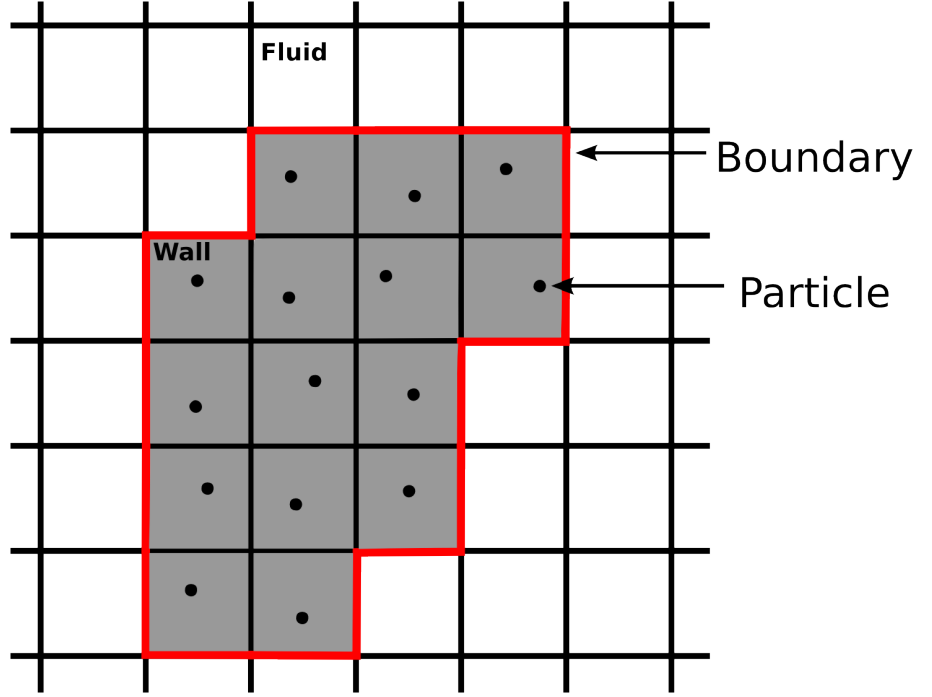


Figure 4.1: Sketch of an immersed boundary, where filled points represent particles to identify wall cells (*highlighted grey*) on the Eulerian grid

4.2.1 Body Force

A term for the body force f is introduced into equation (2.6) to model the effects of the body on the flow. The term is then evaluated to assign a desired velocity distribution over the boundary face within the blocked walls. At this point it should be stressed that the forcing term does not necessarily imply a force acting on the fluid; but can generally represent the action of the immersed boundary on the flow [74, 134]. Adding the force term to the simplified Navier-Stokes equation yields

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = \mu \frac{\partial^2 u_i}{\partial x_j^2} - \frac{\partial p}{\partial x_i} + f_i \quad (4.2)$$

Rearranging Eq. (4.2) for the body force f leads to

$$f_i = \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} - \mu \frac{\partial^2 u_i}{\partial x_j^2} + \frac{\partial p}{\partial x_i} \quad (4.3)$$

and to get the target velocity $u_{i,tg}$ for the wall cells to ensure the correct force on the fluid at the end of each time step, the forcing term, in discretised form,

must be:

$$f_i = \frac{\rho u_{i,tg} - \rho u_i}{\delta t} + T_{s,i} \quad (4.4)$$

where $T_{s,i}$ is used to represent the convective and viscous terms and the pressure gradient from Eq. (4.3). In the numerical description, the same effect is achieved by simply resetting the velocity in a wall cell to the target value.

In principle there is no restriction on the boundary shape and type of force used. For example, Goldstein et al. [60] and Saiki and Biringen [157] apply feedback forcing, based on the difference between the flow and the wall that enforces a zero velocity gradient across the wall face. The interested reader is referred to Fadlun et al. [42] for a more general discussion on the different types of forcing approaches. In the presented work, a more direct forcing approach is used, which is based on the wall (particle) velocity and no implicit forcing term needs to be computed. In static, non-permeable walls, the momentum (forcing) in a cell is set to zero. For moving walls, the '*forcing*' term is explicitly defined by the particle velocity (i.e. the wall velocity) and the fluid density.

$$u_{tg} = u_p \quad (4.5)$$

The forcing term can be directly applied to the Eulerian field as all particle positions, and therefore boundary positions, are known. So the forcing is directly imposed on the boundary without any dynamic process and the boundary holds at every time-step.

If the boundary does not coincide with a cell face (complex and/or moving boundary problems), an interpolation scheme is required to determine the contribution of the body force. In this work, the weighting based scheme is used, in which the factors are calculated from the volume fraction occupied by the body with respect to the volume of the cell

$$\rho u_{i,tg} = \alpha \rho u_p + (1 - \alpha) \rho u_i \quad (4.6)$$

where α is the volume fraction of the cell occupied by the solid body. The volume fraction is determined from the distance of the particle to the cell face.

4.2.2 Impermeability

The pressure correction algorithm needs to be adjusted to ensure impermeable walls by setting the wall pressure gradient ∇P_{wf} to zero. Furthermore, diffusion across the wall faces should also be avoided (Eq. 4.7).

$$\nabla P_{wf} = 0 \quad \text{and} \quad F_{D,wf} = 0 \quad (4.7)$$

In this work, the solution to the problem of ensuring impermeable walls is kept simple, so no additional terms are introduced into the pressure correction, which avoids possible stiffness problems.

The immersed boundary field is used for the correction of the pressure, as wall cell faces with neighbouring fluid cells are easily identified, allowing for efficient computation of the appropriate boundary conditions in these regions. As mentioned above, the pressure gradient across wall faces has to be zero to ensure impermeability of the walls. This is achieved by copying the value of the pressure term from the neighbouring fluid cell to the first cell within the boundary, so that

$$P_w = P_f \quad \Rightarrow \quad \nabla P_{wf} = 0 \quad (4.8)$$

For example, referring to Fig. 4.2, the pressure P from cell N is copied to cell C to get a zero pressure gradient across the northern face of cell C . To efficiently correct the pressure gradients at wall faces without lengthy search algorithms, the pressure corrections are calculated for the entire domain. The corrections are then multiplied by the IB-helper field to set unnecessary modifications to zero. These calculations are more efficient than search algorithms and can be optimised by the compiler through the use of array notation, justifying the small increase in memory requirements to store the IB field. At the end, the corrected wall pressure values are then mapped back onto the pressure field, ensuring zero pressure gradients across the boundary faces.

In a multi-dimensional domain, a boundary cell can have more than one neighbouring fluid cell, as illustrated in Fig. 4.3. The pressure gradient therefore needs to be corrected for multiple faces of the boundary cell. An obvious solution would be to treat each face individually, however this approach is

computationally expensive and does not comply with the idea of developing an efficient immersed boundary method. So an averaging approach, as shown in Eq. (4.9), is used to compute the pressure term of the wall cell C , where IB corresponds to the equivalent cell value of the immersed boundary field, and an equidistant grid is assumed.

$$P_C = \frac{P_N IB_N + P_S IB_S + P_W IB_W + P_E IB_E + P_T IB_T + P_B IB_B}{\max(1.0, (IB_N + IB_S + IB_W + IB_E + IB_T + IB_B))} \quad (4.9)$$

For example, referring to Fig. 4.3 again, the pressure value of cell C is calculated by taking the average of the pressure values from the W and N cells, as pressure contributions from wall cells are ignored. At this point it should be noted that the wall is only impermeable if $P_W = P_N$; otherwise the pressure gradient across the cell faces is not necessarily zero, allowing fluid to penetrate the wall. During this work, the error was found to be negligible for the investigated cases, especially considering the computational efficiency of the approach. Higher-order schemes with weighting factors can be used to improve the accuracy of the pressure gradients across all faces.

A similar idea of using the IB -helper field has been applied for the correction of the diffusive fluxes in the wall regions. The diffusive fluxes are calculated in each direction for the entire domain, using vector notation to allow for compiler optimisation. The calculated fluxes are then multiplied by IB -helper fields, so that no fluxes are applied over the walls. For example, the helper field for the x -direction takes the minimum value of (IB_i, IB_{i+1}) . Yet again, using the helper fields minimise the computational costs, as looping through particle positions is avoided to identify boundary faces.

Scalar quantities, such as mixture fraction and progress variable, should also not penetrate the walls. To avoid this problem for moving walls, treatment of the scalar quantities, similar to the pressure correction, is required at the immersed boundary. In this work, a von Neumann condition is applied for the scalar quantities across the wall faces, such that

$$\nabla \phi_{wf} = 0 \quad (4.10)$$

Unlike in the pressure correction, the scalar quantities from the fluid cell cannot

simply be copied to the wall cell as the scalar density $\phi\rho$ is transported, instead of the scalar ϕ . Furthermore, the difference in density between the fluid and the wall ($\rho_w \neq \rho_f$) needs to be taken into account. Therefore, the scalar density $\phi_w\rho_w$ has to be corrected, according to Eq. (4.11), so that $\phi_w = \phi_f$.

$$\phi_w\rho_w = \rho_w \frac{\phi_f\rho_f}{\rho_f} \quad (4.11)$$

Yet again, the chosen approach is straight forward, computationally efficient and applicable to arbitrary boundary shapes. Nonetheless, a similar problem, as discussed for the correction of the pressure gradient, arises when the boundary cell has more than one neighbouring fluid cell that needs to be considered for the correction of the scalar quantity. For consistency, the same averaging approach is used to calculate the scalar quantity of the boundary cell to ensure that $\nabla\phi_{wf} = 0$.

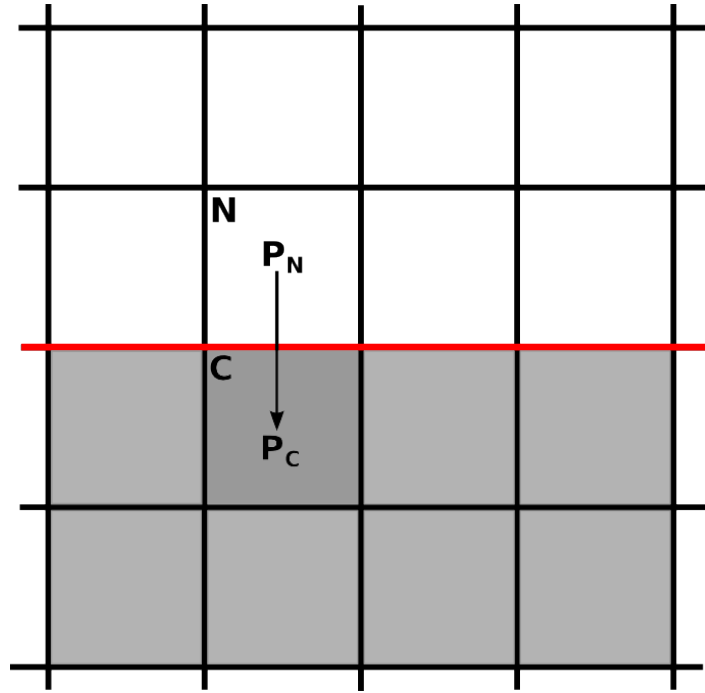


Figure 4.2: Sketch of an immersed boundary to illustrate the pressure gradient correction for wall faces that only have one neighbouring fluid cell

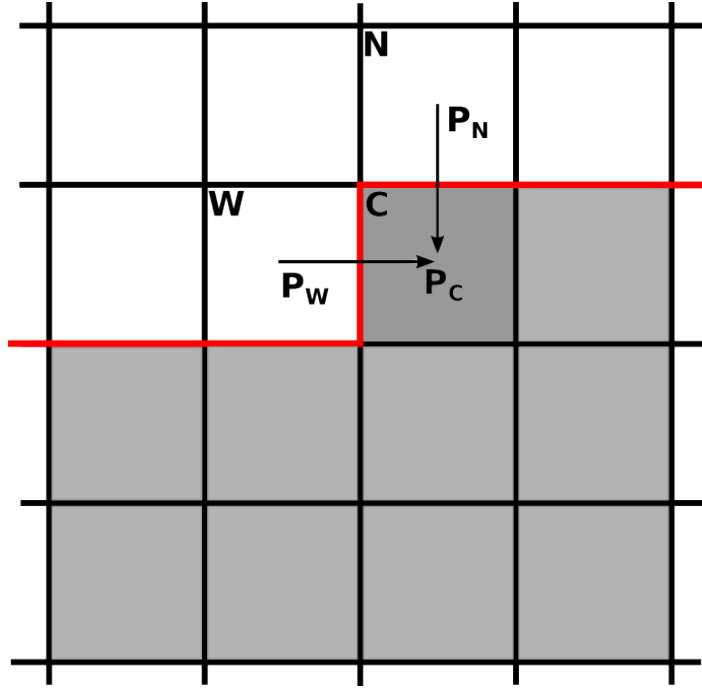


Figure 4.3: Sketch of an immersed boundary to illustrate the pressure gradient correction for wall faces that have multiple neighbouring fluid cell

4.2.3 Near Wall Region Treatment

The following section focuses on the treatment of the cells in the immediate vicinity of solid boundaries. This includes the treatment of freshly cleared cells for moving boundary problems and the stabilisation of the flow field at the immersed boundaries. A summary of the key characteristics of the immersed boundary method is given at the end of the section.

Freshly cleared cells are an important issue that should also be discussed. These cells were in the interior of the body and are exposed to the fluid on the following time-step as a result of boundary movement. Therefore, none of the cleared cells have a fluid history, leading to a temporal discontinuity in the flow field. Nonetheless, a value for the momentum in each cell is required in order to satisfy the momentum equation. A linear interpolation scheme with weighting factors is used to create history in the freshly cleared cells from the momentum in the neighbouring cells. The weighting factors are based on the distance of the wall to the empty cell. Udaykumar et al. [182] address this problem in greater detail by applying a cell-merging formulation in conjunction with a quadratic interpolation among neighbouring grid nodes in the

flow. However more complex interpolation schemes are computationally more expensive and affect the computational efficiency.

A blended upwind/central differencing routine (U-CDS), which is a combination of the upwind- and central differencing schemes, is used for the convection of the velocity fluxes to improve the numerical stability at the immersed boundaries. To briefly recapitulate, the central differencing scheme is accurate, but it suffers from numerical instability, especially when immersed boundaries move through the domain. The upwind differencing scheme is more stable, but very diffusive. Based on these characteristics the upwind differencing scheme is used for the convection of the fluxes if a cell in the fluid region has one or more neighbouring wall cells, otherwise the convective fluxes are calculated using the central differencing scheme. The blended upwind/central differencing scheme, also illustrated in Fig 4.4, for a generic scalar can be written as

$$\phi = (1 - IB^{filt}) * \phi_{UDS} + IB^{filt} * \phi_{CDS} \quad (4.12)$$

where IB^{filt} is the filtered immersed boundary helper field and more details on it are given below.

The use of the blended upwind/central differencing scheme requires the identification of the near-boundary cells. The method should be simple and computationally efficient, since wall locations change in moving boundary problems. Furthermore, the boundary positions are known from the particle locations and the immersed boundary helper field, making the identification of the neighbouring fluid cells straightforward. Fluid cells with neighbouring wall cells are identified by filtering the IB field according to

$$IB_{i,j,k}^{filt} = \frac{1}{27} \sum_{ii=-1}^1 \sum_{jj=-1}^1 \sum_{kk=-1}^1 IB_{i+ii,j+jj,k+kk}^{temp} \quad (4.13)$$

where $IB_{i,j,k}^{temp}$ is the minimum value of the neighbouring cells in the IB field, i.e. $IB_{i,j,k}^{temp}$ is zero if the fluid cell has one or more neighbouring wall cells, otherwise it takes the value unity. Having identified the near boundary cells allows then the application of the blended upwind/central differencing scheme. The approach is general and can therefore be applied to arbitrarily complex

bodies; simulation results using this approach are presented in chapter 5. In general, the scalar transport for the immersed boundary method can be summarized in six distinctive phases as shown in Fig. 4.5.

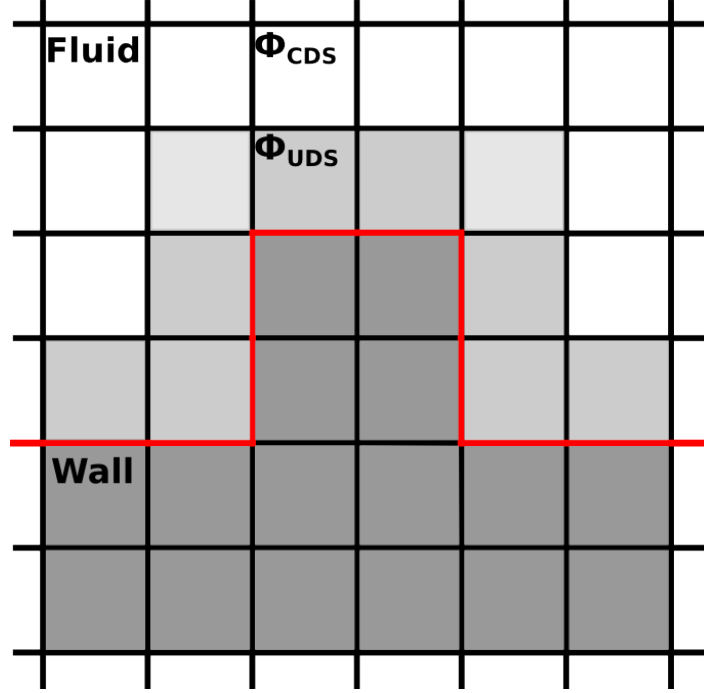


Figure 4.4: Visualisation of the upwind/central differencing scheme

To recapitulate, one of the key characteristics of the immersed boundary method is its simplicity and efficiency. The use of a Cartesian grid allows for the direct translation (simple hashing) of particle positions onto the underlying Eulerian grid. Complex search algorithms are therefore not necessary to identify cells containing particles. Furthermore, the use of the immersed boundary helper fields allows for compiler vectorisation of array calculations. These calculations are computationally more efficient than looping through arrays or remapping the particle positions for each calculation. The disadvantage of the IB helper fields is that they require additional memory to store the information, however the memory footprint is negligible. The immersed boundary characteristics mentioned above, including the use of simple interpolation schemes, make the method computationally very efficient in comparison to alternative immersed boundary methods. Even though, most of the advantages are enabled by the use of a simple grid.

Structure of Immersed Boundary Method

1. Predictor step

Scalar transport: Advance the transport equation for $\bar{\rho}$ and $\bar{\rho}\tilde{\phi}$ using the uncorrected velocities from the previous time-step to yield predicted scalar values $\bar{\rho}^{PD}$ and $\bar{\rho}\tilde{\phi}^{PD}$

2. Evaluation of chemical state

- (a) Calculate the value of the scalar $\tilde{\phi}$ by dividing $\bar{\rho}\tilde{\phi}^{PD}$ by $\bar{\rho}^{PD}$ and
- (b) Calculate the target density $\bar{\rho}^{CS}$ by using the chemistry model to evaluate the chemical state based on the scalar value $\tilde{\phi}$

3. Projection

- (a) Calculate the density error $\Delta\bar{\rho}$ from the difference of the target $\bar{\rho}^{CS}$ density and the predicted density $\bar{\rho}^{PD}$
- (b) Correct pressure gradients for current wall position to ensure impermeability
- (c) Use the density error to calculate the momentum and velocity corrections

4. Correction step

- (a) Scalar transport: Use the corrected velocities to advance the transport equation for $\bar{\rho}$ and $\bar{\rho}\tilde{\phi}$ yielding the corrected scalar values $\bar{\rho}^{CR}$ and $\bar{\rho}\tilde{\phi}^{CR}$
- (b) Momentum transport: Advance the momentum equations to provide the momentum prediction $\bar{\rho}\tilde{u}_i^{PD}$ for the next time-step

5. Wall movement

- (a) Particle transport: Advance all particles to yield the new particle positions
- (b) Momentum update: Correct the momentum to yield the prediction $\bar{\rho}\tilde{u}_i^{CR,PD}$ for the new boundary positions
- (c) Update immersed boundary helper field for the pressure correction in the next time step (if applicable)

6. Finalise time step

- (a) Treatment of the boundaries of the computational domain
- (b) Calculate the turbulent viscosity ν_t

Figure 4.5: Structure of Immersed Boundary Method

4.3 Parallelisation

The increasing availability of computing power at decreasing costs allows for the use of supercomputers in scientific research. The large machines consist of hundreds, even thousands, of multi-core chips or GPGPUs (general purpose computing on graphics units) which have gained popularity in the field of High Performance Computing (HPC) in recent years. However, the parallelisation of a scientific code, to achieve efficient computational run times, can be difficult and time consuming. Nonetheless, the parallelisation of these codes allows the increase of the problem size and complexity while minimising the time to reach a solution by using the available computing power more efficiently.

In the present work, the Eulerian domain is decomposed in three dimensions and the solution of the global domain is calculated in parallel. The particles corresponding to each sub-domain are stored alongside the Eulerian fields in the local memory of the CPUs. A halo-swap is performed at the end of every time-step¹ to exchange boundary values between processors. Storing particles locally might lead to a load imbalance between processors as most particles could end-up on one processor. However this was not a problem in the current work because geometries and wall movements were known at the beginning of the simulations and appropriate decompositions were chosen. Furthermore, typically one particle was used for each wall cell, minimising the memory footprint. An alternative approach would be the decomposition of the Eulerian domain onto processors and have designated processors solely assigned to the particle transport. The main disadvantage of this approach is that all particles need to be communicated to their specific processor location at the end of every time-step. The communication introduces an overhead which can have a significant impact on the computational efficiency. The interested reader is referred to Appendix B and the work by Pettit [138] and Wysocki [193] for more information on parallelising and optimising the PsiPhi code and another scientific code the author has worked on. More general information about parallelising scientific codes can be found in the work by Berthou and Colombet [11] and Jackson et al. [75, 76].

Weak scaling, which defines how the solution time varies with the number of

¹Sub-time-step if the Runge-Kutta scheme is applied

processors for a fixed problem size per processor, was used to evaluate the performance of the 'PsiPhi' code with the immersed boundary method. The performance was evaluated for two different problem sizes with and without moving surfaces on a fixed number of processors². The investigated cases are based on the flow around a sphere for different grid resolutions and number of particles, although the output of data files was turned off. The Eulerian grid consisted of 216×10^3 (*low*) and 1×10^6 (*high*) cells per processor. The sphere was represented with 461 and 2109 particles per processor, respectively. The simulations were timed on (1, 2, 4, 8, 12) processors over a duration of 100 time-steps. A cubic domain decomposition was chosen for optimum parallel performance. The evaluation focused on the weak scaling performance as CPUs and memory should be fully utilised. Ideally the processing time remains constant, but the necessary communication introduces overheads, affecting the overall run time. Furthermore 'on the node' (local) communication is usually faster than internode communication. The normalised results in Fig. 4.6, where T_1 is the sequential run time, suggest that the internode communication has a significant effect on the computational efficiency. The data has to be sent through the interconnect, which has a latency and limited bandwidth. Nonetheless, the parallel efficiency did not drop below 64% in any of the investigated cases. The higher workload cases are computationally more efficient as simulations are less dominated by the communication. Overall, none of the cases show perfect weak scaling characteristics. The difference in run time between the static and moving cases arises from the fact that the particle transport routine is only called for simulations with moving boundaries.

Overall the IB method is suitable for large parallel simulations. The performance can be improved by maximising the work load on each processor while minimising the communication between processors.

²2.4GHz AMD Opteron quad-core processors, 2GB RAM per processor

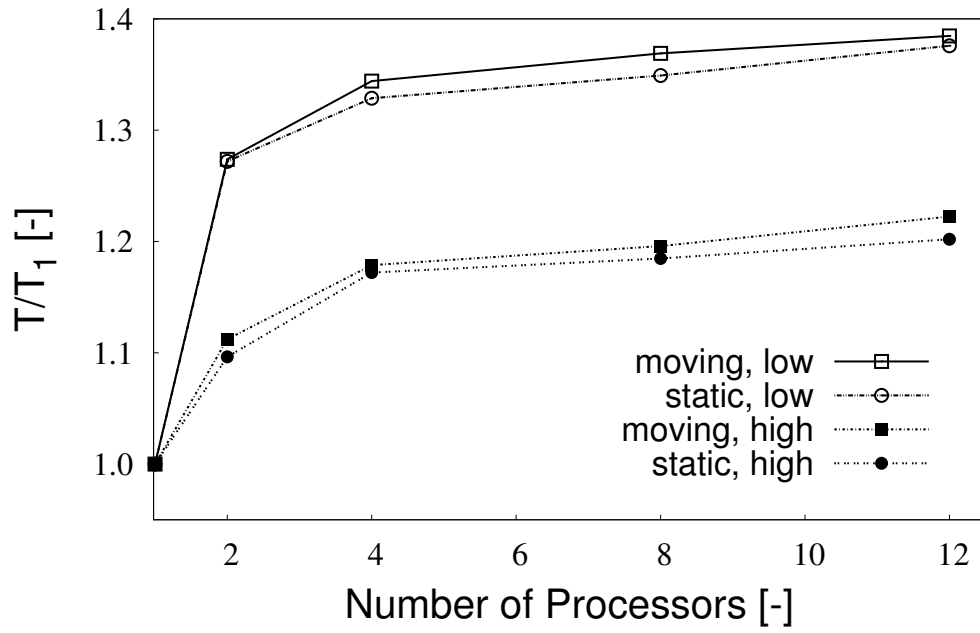


Figure 4.6: Normalised simulation times of the LES 'PsiPhi' code with Lagrangian particles for different processor numbers, time measurements are based on the simulation of a flow around a sphere.

Chapter 5

Test Cases — Static and Moving Surfaces in Flows

This chapter describes the evaluation of the immersed boundary method. Simulations of relevant cases and results are presented and discussed. The chapter is split into two main sections; the first focuses on static boundary problems, the second discusses simulation results with moving immersed boundaries.

5.1 Immersed Boundaries — Static Test Cases

Static test cases were investigated to assess the accuracy of conservation of mass and momentum of the code. Three test cases, with varying complexity, were selected. A simple pipe flow was used to test the original implementation of the immersed boundary (IB) method. The accuracy of the immersed boundary method was determined by modelling the flow around a sphere. The drag coefficient and length of the recirculation zone behind the sphere were calculated and compared to data from the literature. The case is well documented and regarded as a standard reference to evaluate boundary methods. The immersed boundary method was also applied to a complex flow in an opposed jet configuration with perforated plates and fractal grids to demonstrate its applicability to more complex domains.

5.1.1 Pipe Flow

The flow in a straight pipe can be easily modelled at low computational expense. It allows for simple, but extensive validation of momentum and energy

conversation and the evaluation of grid dependencies as well as the accuracy and efficiency of sub-grid scale models.

5.1.1.1 Introduction and Specific Objectives

Assuming an isothermal, incompressible flow, a fully developed flow profile, independent of the inflow conditions, will form in an infinitely long pipe. Hence, the mean flow is homogeneous in the axial flow direction. In a laminar flow, the velocity profile becomes parabolic, and can be described as a second-order polynomial. In a highly turbulent flow, the wall boundary layer thickness decreases, thus, a fuller velocity profile develops, taking a logarithmic shape towards the centre of the pipe. From a statistical point of view, the flow is one-dimensional, but all velocity components undergo turbulent fluctuations. The flow will be simulated in a three-dimensional domain, using the LES flow solver described in chapter 3.

In general, it is not feasible to model the flow in a pipe of infinite length, so a section of the pipe has been simulated by applying periodic boundary conditions, to mimic an infinitely long pipe. A minimum pipe length of $L = 8D$ is often suggested [184] as an appropriate configuration to guarantee a fully developed flow, where L is the length of the pipe and D is the pipe diameter. Such a configuration was investigated in this work.

The Reynolds number for the simulation of the pipe flow was based on the bulk velocity U_b and the frictional velocity U_τ :

$$Re = \frac{2U_b R}{\nu} = 40,000 \quad \text{and} \quad Re_\tau = \frac{U_\tau R}{\nu} = 1050 \quad (5.1)$$

where ν is the kinematic viscosity. The frictional velocity is also referred to as the characteristic velocity at the wall and can be defined as

$$U_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (5.2)$$

In Eq. (5.2) τ_w represents the wall shear stress and ρ is the fluid density. In order to establish the correct momentum, which decreases due the shear stresses at the wall, a pressure gradient equivalent to the wall shear stresses

was calculated [171] and added as a source term to the momentum equation (Eq. 2.2).

$$\frac{\partial \langle \bar{p} \rangle}{\partial x} = \frac{-2\rho U_\tau^2}{R} \quad (5.3)$$

5.1.1.2 Numerical Simulations

The overall simulated domain size was $320 \times 40 \times 40$ mm with a filter (cell) width of $\Delta = [1.0, 0.5]$ mm, resulting in equivalent grids of $320 \times 40 \times 40$ cells and $640 \times 80 \times 80$ cells, respectively. The Smagorinsky and Germano sub-grid stress models (explained in Section 2.4.2) were used, with $C_s = [0.065, 0.100]$, to evaluate the mean velocity profile and fluctuations.

In order to resolve the near wall stresses, an appropriately fine grid was required. This usually leads to high computational costs for an equidistant grid. In this work, the emphasis of the investigation was the overall prediction of the mean velocity profile and the fluctuations. The area near the wall is clearly affected by the Cartesian grid with cubic cells, directly influencing the effect of the correction of the momentum equation. An excessive distance from the first cell to the wall leads to a larger error in the calculation of the wall shear stress. However, a very fine grid suffers from additional computational costs, an increase in computational run time and over-resolving of the flow field in the centre of the pipe. A compromise was identified to resolve the grid sufficiently for an accurate representation of the mean flow while keeping the computational costs to a minimum. The corresponding values for the wall distance (y^+) for both grids are $y^+ = [17.2, 8.6]$, respectively. Salim and Cheah [159] determined a value of $y^+ < 30$ to be sufficiently accurate for turbulent flows and $y^+ < 10$ for laminar flows. To fully resolve the viscous sublayer the wall y^+ value should be lower than five.

5.1.1.3 Results and Discussion

The following section discusses the simulation results for a simple pipe flow. The predictions of the first and second moments of the axial velocity across the radius of the pipe are compared to numerical results by Forkel [45] and experimental data by Hishida et al. [72]. In the simulations, Forkel [45] uses a Smagorinsky constant of $C_s = 0.1$ for the simulations and an unstructured

cylindrical grid. The main purpose of this study was the evaluation of the IB method and the conservation of mass and momentum based on a coarse computational grid at minimal computational effort. Furthermore, the simplicity of the case allows the assessment of the Germano sub-grid stress model implementation at relatively low computational costs.

Figure 5.1 shows the normalised mean velocity profiles and the fluctuations at the midpoint of the computational domain for the different sub-grid stress models applied. The presented mean velocity profile and fluctuation plots are normalised by the frictional velocity U_τ . At this point it should also be mentioned that the shorthand notation u' is used for the standard deviation $\sqrt{u'u'}$ throughout this chapter. From the figure, it is evident that the different sub-grid stress models produce significantly different mean velocity profiles. A Smagorinsky constant of $C_s = 0.065$ leads to an over-prediction in the mean velocity in the pipe. The low C_s value leads to a reduction in viscosity of the flow, over-predicting the pressure gradient correction and therefore increasing the mean velocity. The simulation with $C_s = 0.100$ shows the best agreement with the experimental data in the near wall region, even though an under-prediction of the mean velocity at the centre of the pipe was found. The dynamic Germano sub-grid stress model shows the best predictions near the centre of the pipe. However, the mean velocity of the flow in the near wall region is over-predicted. A C_s value of 0.08 is suggested for optimum results. In general, differences in the mean flow rate between experimental and numerical results can originate from incorrectly calculated frictional velocity, as also pointed out by Forkel [45].

None of the simulations captured the fluctuation peak in the near wall region well, suggesting that the area is under-resolved and a grid refinement would be required. Nonetheless, a similar trend in comparison to the measurements by Hishida et al. [72] could be established. Furthermore, the simplicity of the Smagorinsky sub-grid stress model influences the predictions as a viscosity constant is applied across the entire field, ignoring the shear layer effect in the near wall region. A lowered constant leads to a decrease in viscosity and predictions could be improved by tuning the constant accordingly. The Germano model under-predicts the fluctuations across the radius of the pipe. However, the coarseness of the grid limits the accuracy in the near wall region. Overall,

the presented fluctuations show a good agreement with the experimental data.

Figure 5.2 shows the radial velocity profiles at different angles relative to the axes at the midpoint of the pipe with the Smagorinsky model ($C_s = 0.1$). The mean radial velocity profiles and corresponding fluctuations along the ordinate (y) and applicate (z) axes are indistinguishable, confirming the axis-symmetry of the flow. Furthermore, the radial profiles at 45 degrees to the ordinate and applicate axes, named *positive* and *negative* diagonal also show similar predictions for the first and second moments. Nonetheless, a small difference between the diagonal and axial profiles can be found, resulting from the difference in the computational grid in the near wall region and from discretisation parallel/diagonal to the mesh, as it is visible on the RHS of Fig. 5.3. The grid points along the axial radii differs from the grid points in the diagonal radii, influencing the predictions of the first and second moments. Refining the mesh or applying unstructured grids for a smoother representation of the pipe walls would improve the predictions along all radii.

As suggested above, a grid refinement improves the statistical measurements of first and second moments, especially in the near wall regions. Figure 5.4 shows the normalised mean velocity profiles and the fluctuations for the simulations with the Germano model for two different filter (cell) widths $\Delta = [1.0, 0.5]$ mm. The mean velocity profile shows improvements at the centre of the pipe; as expected the biggest improvements are in the region near the wall. The fluctuations are generally well captured across the pipe radius and significant improvements can be seen in the near wall regions, where the peak is predicted well. Simulation results show a significant improvement in comparison to previous numerical investigations.

The predictions of the mean velocity and fluctuations are satisfactory for the purpose of the immersed boundary method, as in combustion applications near wall effects are of less significance than in boundary dominated flows. Nonetheless, results benefited from a higher grid resolution, especially in the near wall region.

5.1.1.4 Conclusion

The flow in a pipe has been investigated to evaluate the initial implementation of the immersed boundary method and the dynamic Germano model into the Large-Eddy Simulation code 'PsiPhi'. Normalised mean velocity profiles and fluctuations have been compared to experimental and numerical data sets available. The mean velocity profiles were captured well and fluctuations show a better agreement with the experimental data than the numerical results by Forkel [45]. However, an under-prediction of the fluctuations was found on the coarse grid in the near wall region, indicating that the flow in these regions is under-resolved, as the y^+ values are greater than 5. Nonetheless, inside combustors, the wall treatment only needs to consider displacement effects on the flow, but rarely calculate friction/drag in boundary layers.

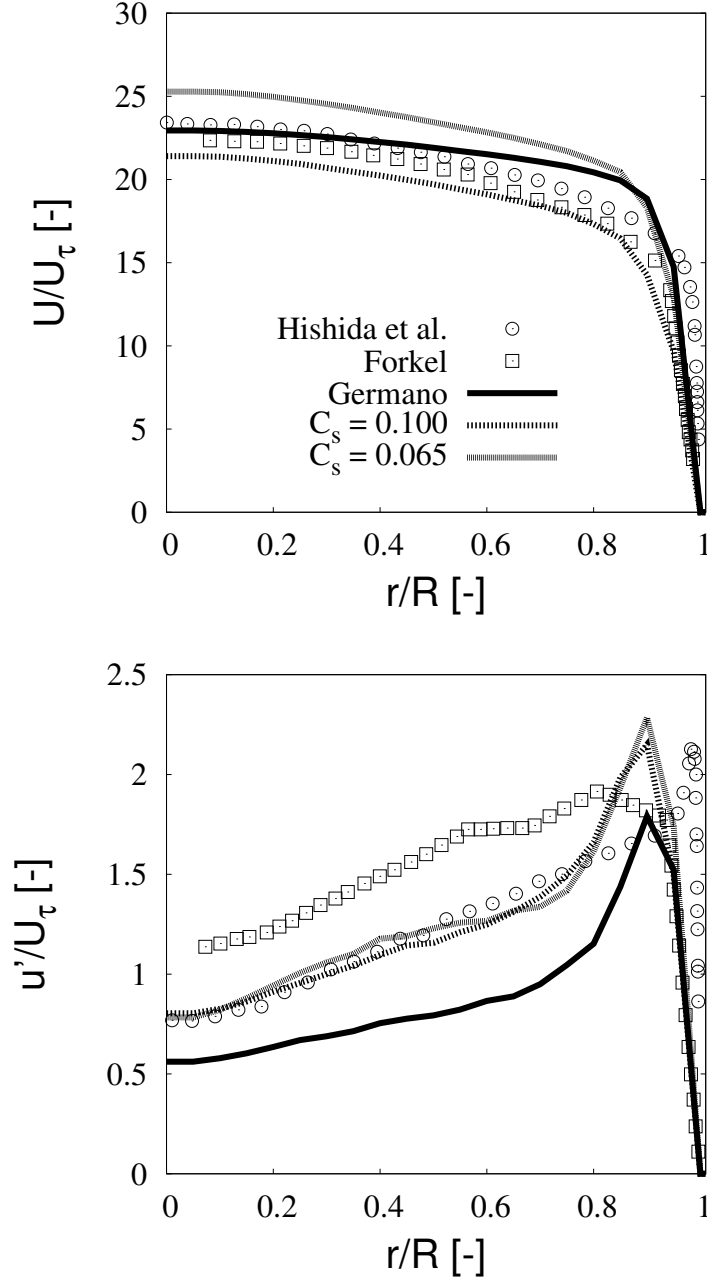


Figure 5.1: Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = 1.0$ mm and two different sub-grid stress models. Results are compared to experimental and numerical results.

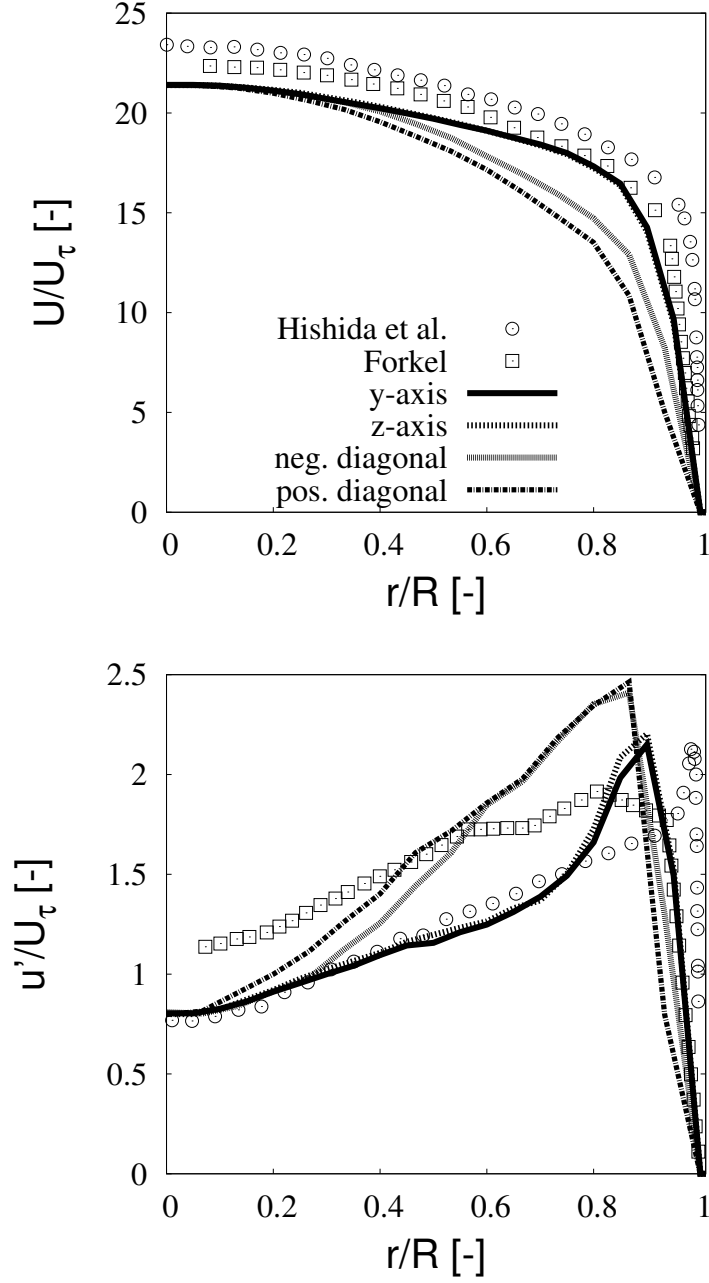


Figure 5.2: Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = 1.0$ mm and $C_s = 0.100$ at different radial angles. Results are compared to experimental and numerical results.

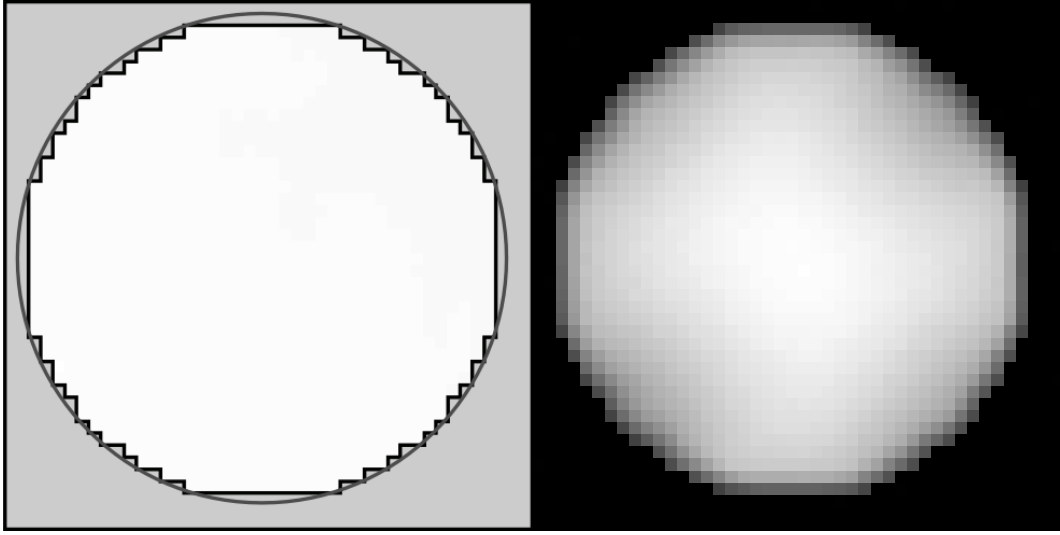


Figure 5.3: Discretisation of the pipe onto the computational grid (left) and visualisation of mean axial velocity field at the mid point of the pipe for $C_s = 0.1$ (right) for $\Delta = 1$ mm. Maximum positive velocities are shown in white, while black represents a zero velocity.

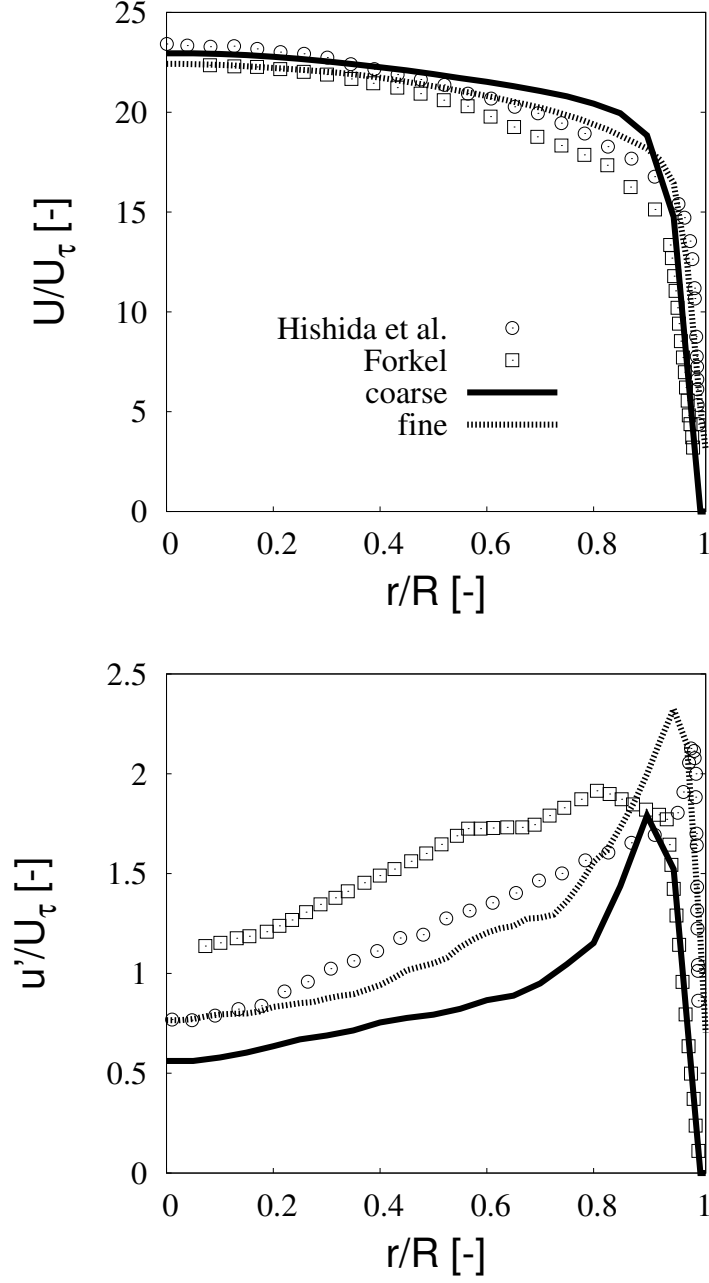


Figure 5.4: Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 14.47$ m/s, $\Delta = [1.0, 0.5]$ mm with the Germano model. Results are compared to experimental and numerical results.

5.1.2 Flow around a Sphere — Static

Viscous flow around a stationary sphere can be considered as a simplification of the general family of immersed bluff-body flows. The instabilities arising from the flow past a sphere are known to generate fully three-dimensional unsteady flow fields despite the symmetry of the body. The behaviour of the flow past a sphere has been studied by numerous researchers, a brief overview is provided below. The results are used to validate the implemented immersed boundary method.

5.1.2.1 Introduction and Specific Objectives

Since the pioneering work of Oseen [131], the flow past a sphere has been of great interest. The first detailed experimental investigations into the flow behaviour around the object were recorded in the 1950s. Taneda [177] investigated the wake and flow around a stig-mounted sphere for Reynolds numbers $5 < Re < 300$. The author determined that the flow separation at the sphere occurs at $Re = 24$ and axis-symmetric vortex rings form. Furthermore another transition in the flow was identified at $Re = 130$, which became known as the von Kármán vortex street. Improved measuring techniques were used by Nakamura [126], to carry out similar experiments. These data sets agree well with the results of Taneda. The contributions by Magarvey and Bishop [109] and, more recently, by Wu and Faeth [192] found the flow to be stable with Reynolds numbers up to 210. Magarvey and Bishop used dye visualisation on free falling drops of an immiscible liquid in water. It was later shown by Natarajan and Acrivos [127] that the work on free falling drops can be linked to the work on solid spheres.

Experimental work focusing on $Re > 300$ was carried out by Achenbach [1] and Sakamoto and Haniu [158]. Achenbach [1] reports that the flow loses its axial symmetry at $Re \approx 300$, but maintains a planar symmetry around the plane on which the shedding process is initiated. Sakamoto and Haniu [158] found another transition in the flow for Reynolds numbers greater than 420, at which point the flow becomes almost chaotic. Kim and Durbin [89] showed results of frequencies present in the wake for $Re < 420$.

In the numerical framework, Tomboulides [179] applied Large-Eddy Simulation to study flow past a sphere for $Re < 270$. The results agreed well with the experimental measurements by Magarvey and Bishop [109]. Furthermore, Tomboulides and Orszag [180] carried out a computational investigation of viscous incompressible flow past a sphere for $25 < Re < 1000$ using Direct Numerical Simulations with spectral type methods to investigate the mechanisms of the transition to turbulent flows. Gebing [48] solved the unsteady compressible-flow equations for a sphere in flows with $20 < Re < 1000$ and a Mach number $Ma = 0.4$. The results showed a loss of symmetry at $Re = 300$ and the flow becomes unsteady and almost chaotic for $Re > 400$. Different flow regimes for $Re < 300$ were examined by Johnson and Patel [79] to analyse the flow dynamics and structures that form behind the sphere. Flows past a sphere with Reynolds numbers $Re \leq 5000$ were investigated by Fornberg [46], who used an iterative approximation approach (Newton-Raphson method). Fornberg [46] applied boundary fitted meshes and results agree well with experimental measurements.

Fadlun et al. [42] simulated the flow past a sphere to validate an immersed boundary method, which uses feedback forcing at the surfaces that do not coincide with the grid. The computational efficiency was found to strongly depend on the forcing term. The sphere test case was also used for the validation of the sharp interface immersed boundary method for compressible viscous flows by Ghias et al. [57]. Simulations focused on Reynolds numbers up to 3900 and local grid-refinement was used. The results show good agreement with the experimental data.

For the validation of the immersed boundary method, a uniform flow with velocity U_b at different Reynolds numbers Re was imposed at a certain distance upstream from the sphere with diameter D .

Based on the two-dimensional analysis by Kundu and Cohen [95], a control volume, as shown in Fig. 5.5, can be used to determine the drag force F on the sphere. The control volume consists of the rectangular region $ABCD$ with a sphere positioned along the centreline of the horizontal axis. The sides AB and DC are chosen far enough from the sphere so that the pressure at these sides is approximately equal to the undisturbed pressure p_∞ . The right-hand

side (BC) of the control volume is also chosen far enough from the sphere for the streamlines to be almost parallel at this position. Therefore the pressure variation along the face BC is negligible, so the pressure can be assumed to be constant and equal to the undisturbed pressure p_∞ . The surface forces therefore cancel out and based on these assumptions, the only force acting on the boundary of the fixed volume is F , which is the force acting on the body of the sphere.

For a steady flow around an impermeable sphere, the axial momentum component reduces to the net flux rate of momentum through the boundaries of the control volume. Applying the assumptions above, the expression for the force (drag) on the fluid in the negative x -direction may then be written in terms of velocity difference over the inlet and outlet

$$F = \rho \int_{-h}^h u(U_\infty - u)dy \quad (5.4)$$

which can be evaluated from the measured velocity profile. A more detailed description of the theory is outlined in the book by Kundu and Cohen [95].

However, the idealised configuration with infinite axial distance and constant pressure cannot be assumed in computational fluid dynamics; hence the change in pressure also needs to be taken into account. Equation (5.4) therefore becomes

$$F = \rho \int_{-h}^h u(U_\infty - u)dy + \int_{-h}^h p(P_\infty - p)dy \quad (5.5)$$

assuming that streamlines in the control volume are parallel to the horizontal surfaces AB and DC of the control volume and periodic boundary conditions are applied, allowing viscous terms to be neglected. The drag force in the axial direction can then be evaluated based on the change of pressure and the net outflux of momentum through the surfaces. If the vertical distance between the sphere and the control volume is too short and streamlines are not parallel to the horizontal surfaces of the control volume, the viscous terms also need to be accounted for.

Having determined the drag force of the object, the drag coefficient C_d is

calculated as

$$C_d = \frac{2F}{\pi \rho r^2 U_\infty^2} \quad (5.6)$$

where r is the sphere radius and ρ the fluid density. The empirical solution for comparison is based on the derivation by Oseen [131] and can be written as

$$C_d = \frac{24}{Re} (1 + 0.15 Re^{0.687}) \quad (5.7)$$

Equation (5.7) holds for Reynolds numbers $0 < Re < 1000$ and a C_d value of approximately 0.44 is approached at higher Reynolds numbers. [131].

5.1.2.2 Numerical Simulations

A domain of size $2 \times 0.48 \times 0.48$ m, equivalent to $50D_s \times 12D_s \times 12D_s$, was used for the three-dimensional numerical simulations of a flow past a sphere. Two different cell sizes ($\Delta = [5, 2]$ mm), referred to as *coarse* and *fine* grids, respectively, were applied to evaluate the accuracy of the immersed boundary method. The chosen grid sizes resulted in meshes of 3.6×10^6 and 57.6×10^6 cells, respectively. The sphere was located 10 sphere diameters ($D_s = 0.04$ m) downstream of the inlet plane.

The current study focused on Reynolds numbers ($Re = [50, 100, 200, 300, 500]$) in the steady and unsteady flow regime. Dirichlet boundary conditions were applied at the inflow plane and von Neumann conditions were set at the outlet of the domain. Periodic boundary conditions were applied at the ordinate and applicate boundaries. The turbulent viscosity was estimated with the Smagorinsky model ($C_s = 0.173$) to keep computational costs low and the sphere was represented with immersed boundaries. Figure 5.6 shows a section of the axial velocity and pressure field at $Re = 100$ in the near sphere region. The colour of the sphere represents a velocity of zero. A similar colour is seen representing the negative velocities in the wake of the sphere due to the low Reynolds number flow.

5.1.2.3 Results and Discussion

The simulation results for the static sphere are presented and discussed in the following section. The wake length¹ was compared to the numerical results by Fadlun et al. [42] and Fornberg [46]. Fadlun et al. [42] apply a non-uniform mesh whilst Fornberg [46] uses boundary fitted meshes to simulate the flow around a sphere. Both groups assumed axis symmetry in their flow, which is unrealistic for higher Reynolds numbers ($Re \geq 300$) because the flow develops three-dimensionality in the wake. Nonetheless, results for the higher Reynolds number flows are also compared to data obtained by Fornberg [46] to assess the axis-symmetric hypothesis. The drag coefficients C_d are compared to the theoretical values derived by Oseen [131]. Furthermore, only the mean axial velocity profiles will be presented for quantitative comparison. To the author's knowledge the only experimental data for ($Re = 300$) is available [79] for the selected Reynolds numbers.

Figure 5.7 shows the calculated drag coefficients for the modelled flows at different Reynolds numbers in comparison to the theoretical values by Oseen [131]. Good results were achieved for the lower Reynolds number flows. In these flow regimes the corresponding values for the wall distance y^+ are within the viscous sublayers range ($y^+ < 5$). However, results contain a larger discrepancy from flow simulations at higher Reynolds numbers, up to 12% on the coarse grid and 6% on the fine grid. At $Re > 300$, the flow is unsteady and first signs of turbulence behind the sphere can be identified. The differences result from the coarseness of the grids, as the boundary layer flow is under-resolved ($y^+ > 5$ on the coarse grid). The best results are obtained at $Re = 200$. The predictions of the drag coefficients are satisfactory, considered that no wall model was used, likely leading to a bigger error. The y^+ values also indicate that the coarse grid is not sufficient to accurately predict the boundary layer flow at higher Reynolds numbers. Nonetheless, the main purpose of this methodology is to develop an immersed boundary method for combustion applications, where the near wall flow is often neglected.

In general, the wake length is better predicted on the fine grid, as shown in Fig. 5.8. At lower Reynolds numbers, the results agree well with the predicted

¹The wake length is defined from the end of the sphere to the stagnation point ($u = 0$)

values by Fadlun et al. [42] and Fornberg [46]. Larger deviations are found in the region of higher Reynolds numbers (i.e. $Re = 500$). In these regimes the flow is three-dimensional and axis symmetry is lost. The fact that Fadlun et al. [42] and Fornberg [46] apply a symmetry assumption which is not always justified, may play a role in the observed under-prediction of the wake length. The under-prediction of the wake length strengthens the argument of refining the grid for a detailed analysis of the flow in the wake of the sphere at higher Reynolds numbers.

The mean axial velocity profiles in the wake of the sphere, normalised by the bulk velocity U_b , are presented in Fig. 5.9 for a quantitative analysis between the coarse and fine grid results. It is clear that the difference between the two grids is minimal for lower Reynolds number flows ($Re \leq 200$). Hence, the grid resolution for these cases is regarded as sufficient for the purpose of this study. The higher Reynolds number flows show larger deviations in the flow behaviour behind the sphere. Mittal [122] carried out an extensive study of the flow behind the sphere, reporting the development of an unsteady flow at $Re = 270$, where the wake becomes wavy. The phenomena of waviness is captured well on the fine grid ($Re = 300$), as shown in Fig. 5.10, leading to a good prediction of the mean axial velocity profile. The predictions immediately downstream of the sphere are in good agreement with the experimental measurements of Johnson [79], supporting the estimation of the separation bubble. However a general under-prediction of the mean axial velocity profile was found further downstream on the coarse grid at $Re = 300$.

An investigation of the flow field further downstream has shown that the axial velocity reaches the measured value by Johnson [79] (see Fig. 5.11). It may be argued that the delayed recovery of the flow along the centreline is the result of a poor prediction of the turbulence, limiting the waviness of the flow on the coarse grid. The visualisation of the contour plot for the coarse grid (Fig. 5.12) shows that the flow is strongly deflected upstream of the sphere, leading to an early separation of the flow. On the other hand, the vector plot for the fine grid shows a stronger vortex in the wake of the sphere. More fluid is sucked towards the centre of the domain, which supports the predictions of the mean axial velocity profiles. A similar flow behaviour was also observed for the flow of $Re = 500$.

Figure 5.13 shows the mean radial velocity profiles in the wake of the sphere, normalised by the bulk velocity U_b for the coarse and fine grid. For most of the Reynolds numbers, the profiles are identical. A difference can be found for $Re = 500$ where the difference in grid resolution has an effect on the radial velocity components as a result of the boundary layer effect. Furthermore the mean axial velocity profiles normal to the axial velocity field are shown in Fig. 5.14. Yet again, a deviation can be seen at $Re = 500$. The difference across the sphere between the fine and coarse grid

The flow separation is strongly dependent on the boundary layer and a wall model or high grid resolution should be used to accurately resolve the boundary layer flow. The recirculation bubble behind the sphere was well predicted in all cases and the results are satisfactory for the purpose of this study. Nonetheless higher grid resolutions should be used when analysing the flow field in the near wall region.

5.1.2.4 Conclusion

The flow around a sphere is well documented and can be regarded as a common test case to validate new numerical methods. Large-Eddy Simulations were carried out to determine the accuracy of the proposed immersed boundary method for steady and unsteady flows at different Reynolds numbers. Simulation results were compared to numerical results for the drag coefficient, wake length and mean axial velocity profiles by Fadlun et al. [42], Fornberg [46] and Johnson [79]. Generally the results for the drag coefficient and wake length agreed well at lower Reynolds numbers ($Re \leq 200$). An over-prediction of the drag coefficient and an under-prediction of the mean axial velocity along the centreline results from the coarse filter width.

Differences in the mean axial velocity profiles are more noticeable at higher Reynolds numbers, where the flow enters the unsteady regime. However, it must be considered that the data sets from the literature assume axis-symmetric flows along the centreline, which does not capture the full three-dimensionality of the flow in the wake of the sphere. Furthermore, axial veloc-

ity predictions at $Re = 300$ agree well with the data by Johnson [79]. Overall, the results encourage the use of the immersed boundary method in combination with LES, so that more complex problems can be investigated.

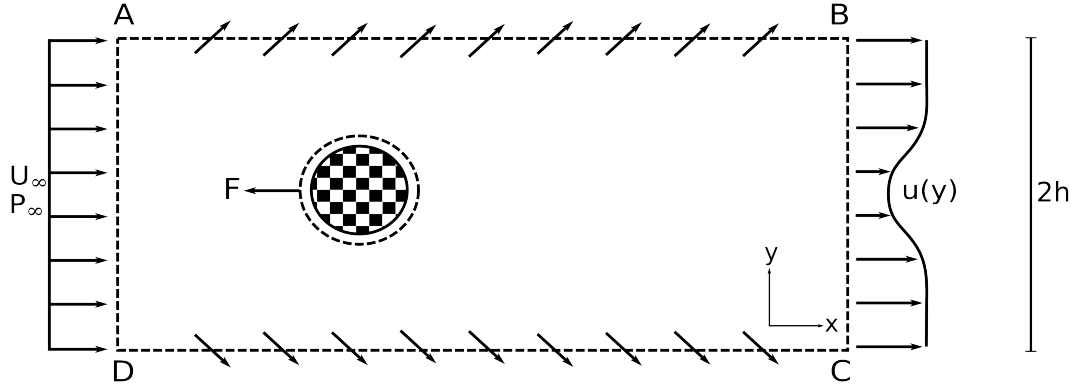


Figure 5.5: Momentum balance of flow over a body, adapted from [95].

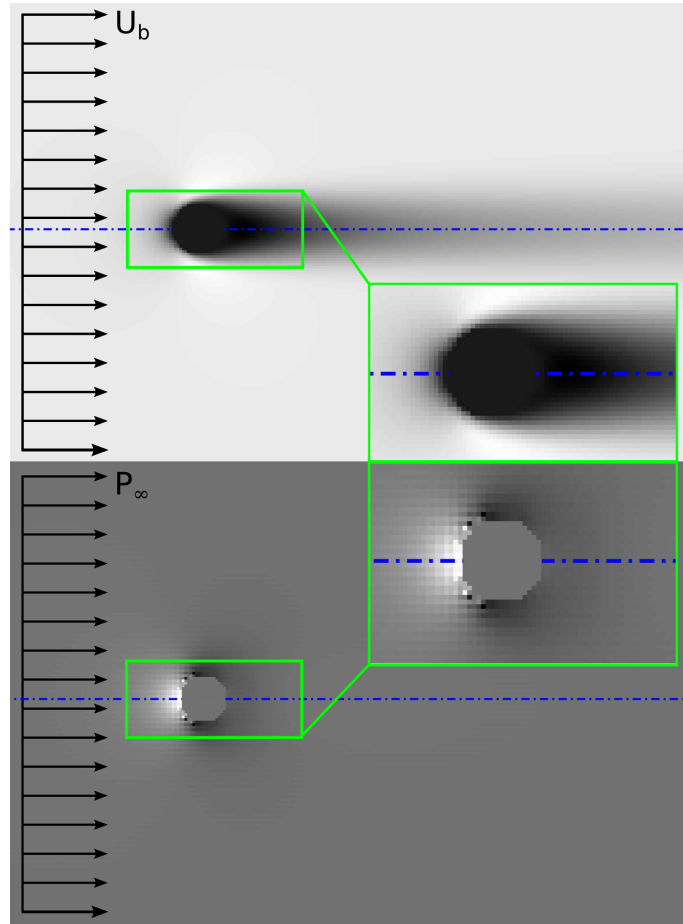


Figure 5.6: Schematic of the instantaneous axial velocity (top) and pressure (bottom) fields in the near sphere region with symmetry axis indicated by the dash-dotted line for $\Delta = 2$ mm and $Re = 100$. Maximum positive velocities and pressures are shown in white, while maximum negative velocities and pressures are black.

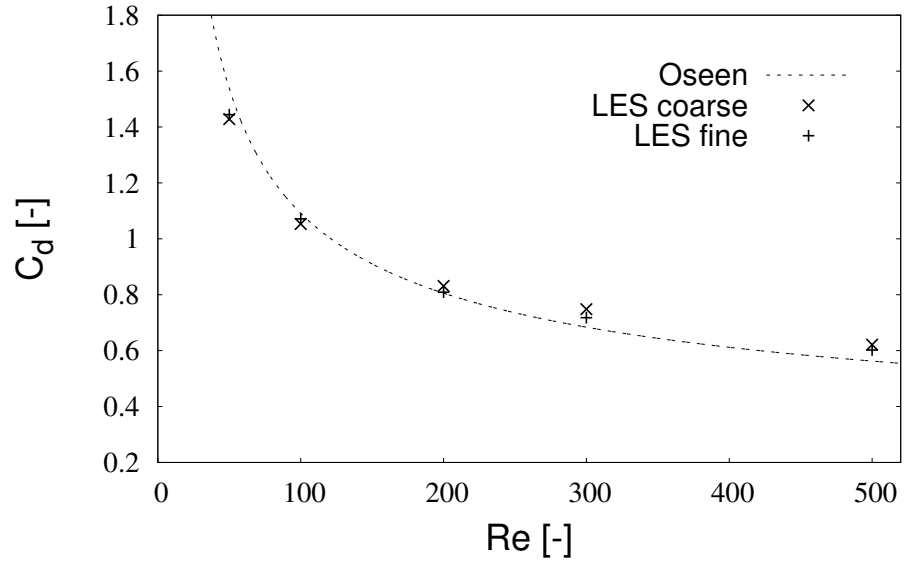


Figure 5.7: Comparison of the drag coefficient for the flow around a static sphere at different Reynolds numbers for $\Delta = [5, 2]$ mm.

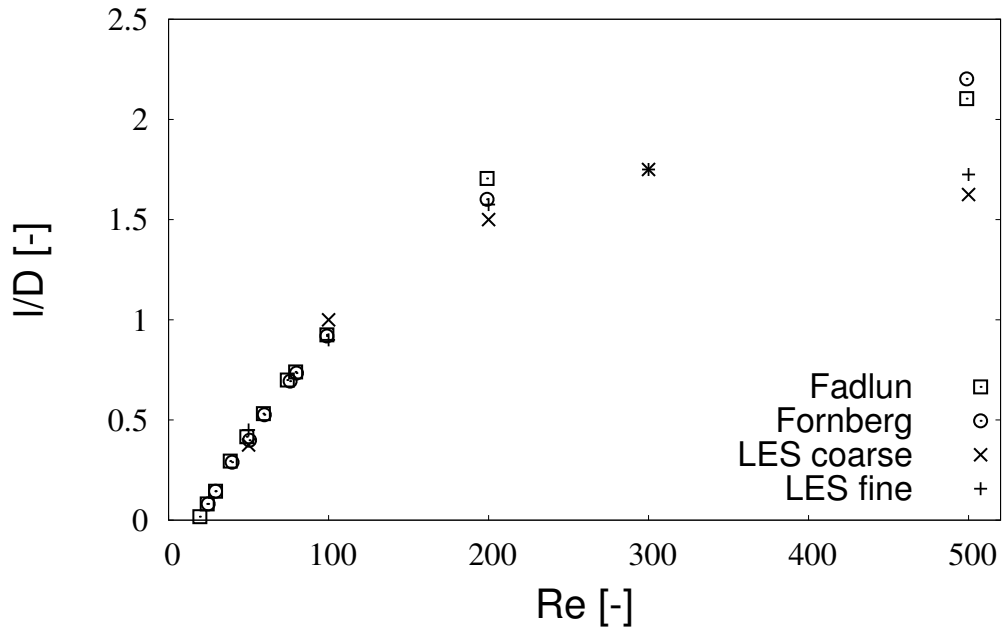


Figure 5.8: Wake length (l/D), measured from the end of the sphere. Comparison with experimental and numerical data sets

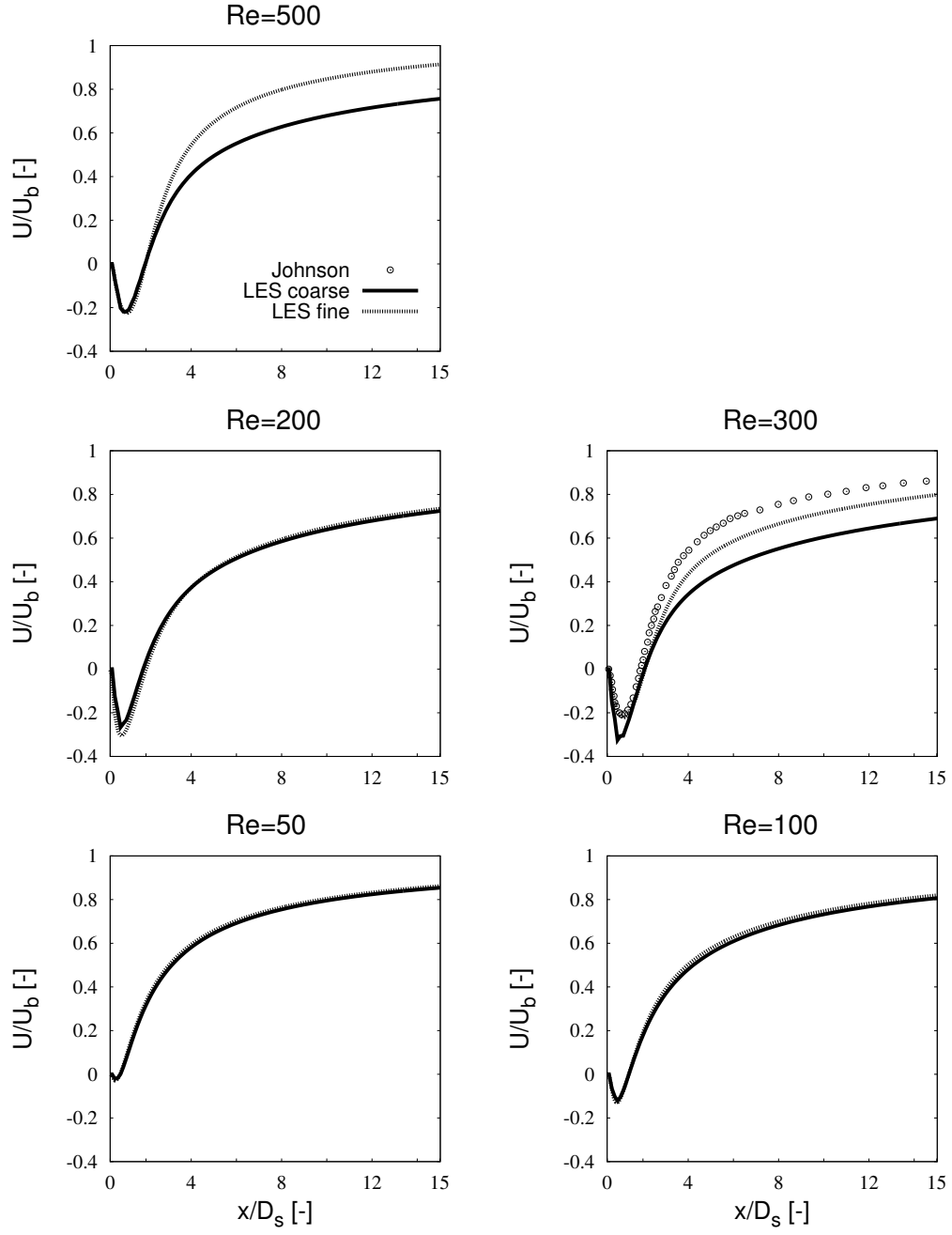


Figure 5.9: Statistical measurements of first moments of axial velocity behind the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.



Figure 5.10: Visual comparison of instantaneous axial velocity field at $Re = 300$ on the two grids: coarse (*top*) and fine (*bottom*). Maximum positive velocities are shown in white, while maximum negative velocities are black.

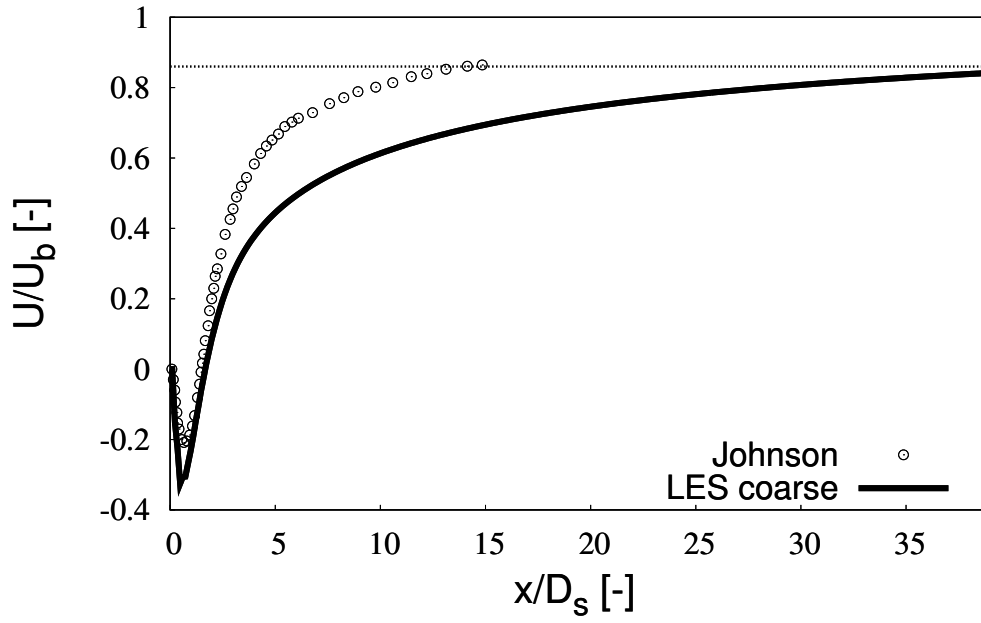


Figure 5.11: Enlarged domain for statistical measurements of first moments of a flow around a static sphere for $\Delta = 5$ mm and $Re = 300$.

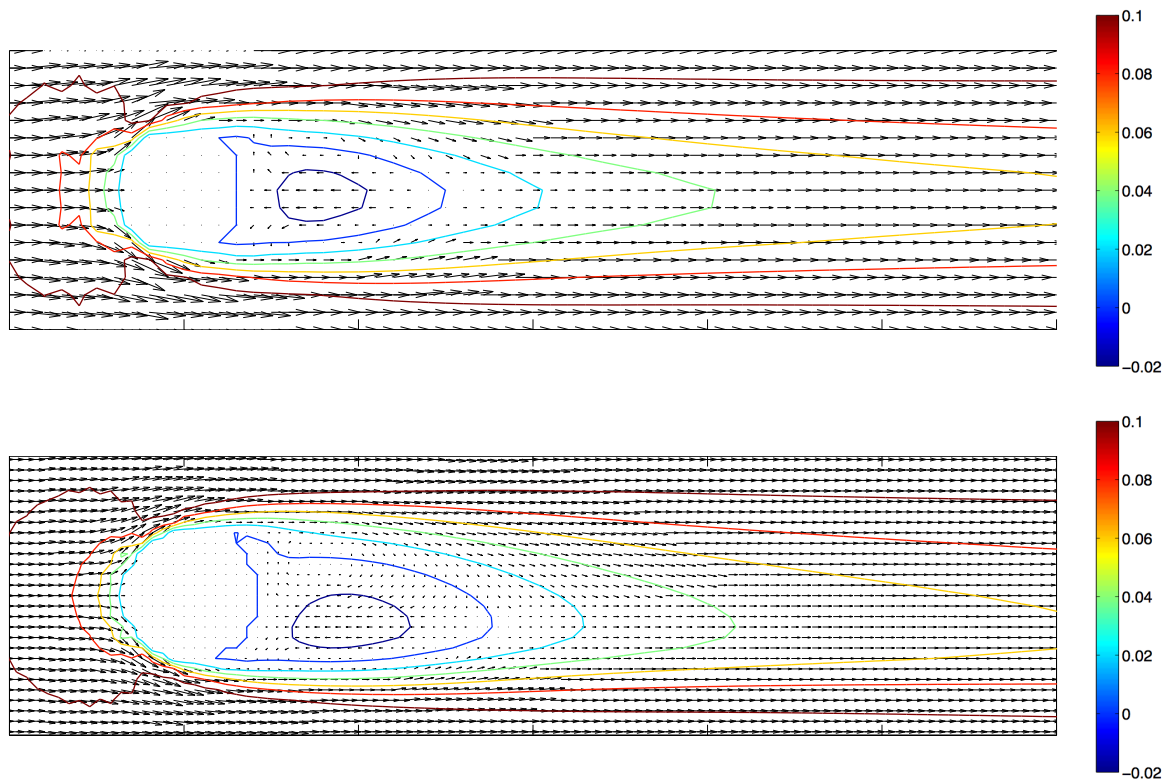


Figure 5.12: Sample vector plot with contour lines of instantaneous axial velocity field at $Re = 300$ in the near sphere region on the two grids: coarse (*top*) and fine (*bottom*)

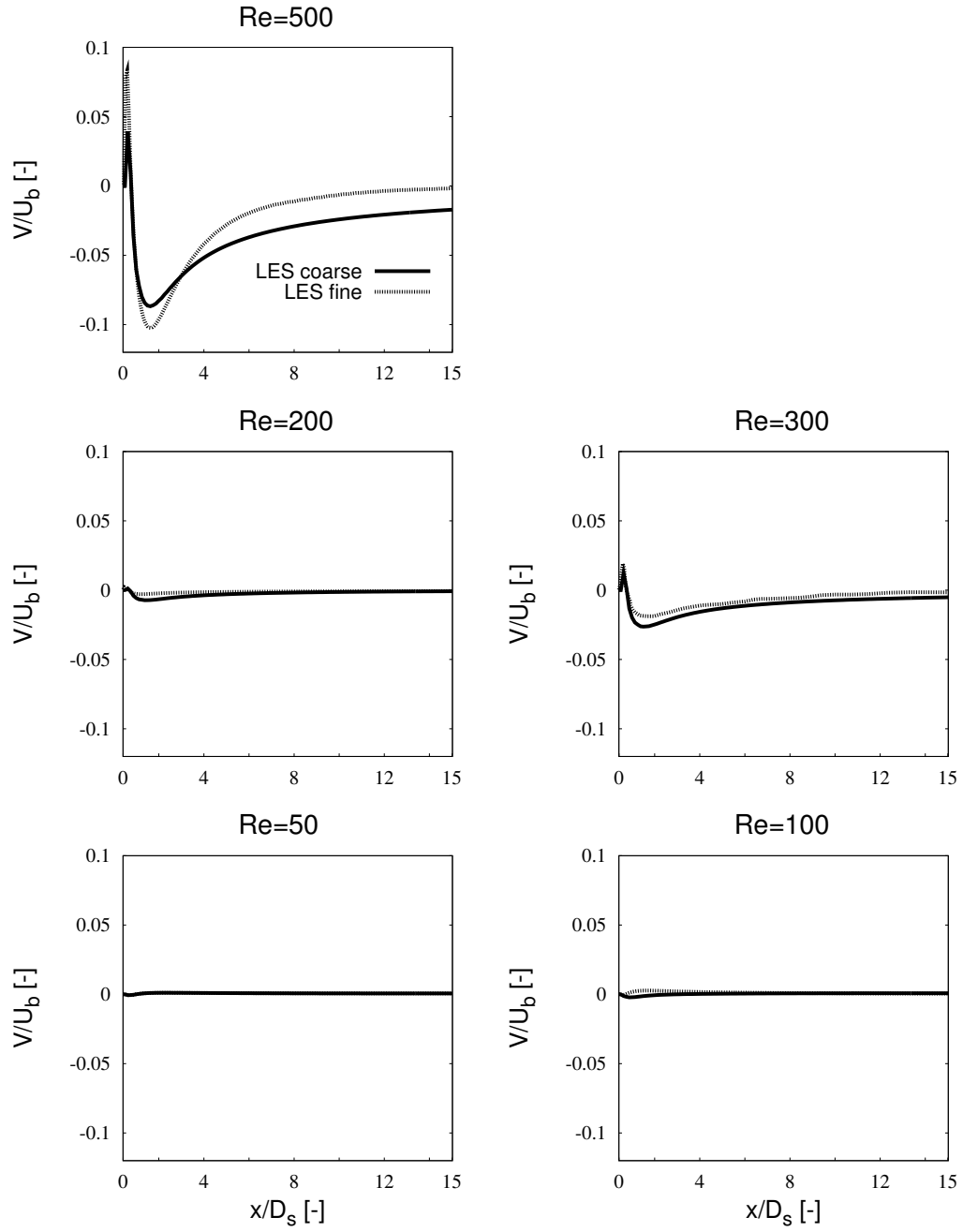


Figure 5.13: Statistical measurements of first moments of radial velocity behind the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.

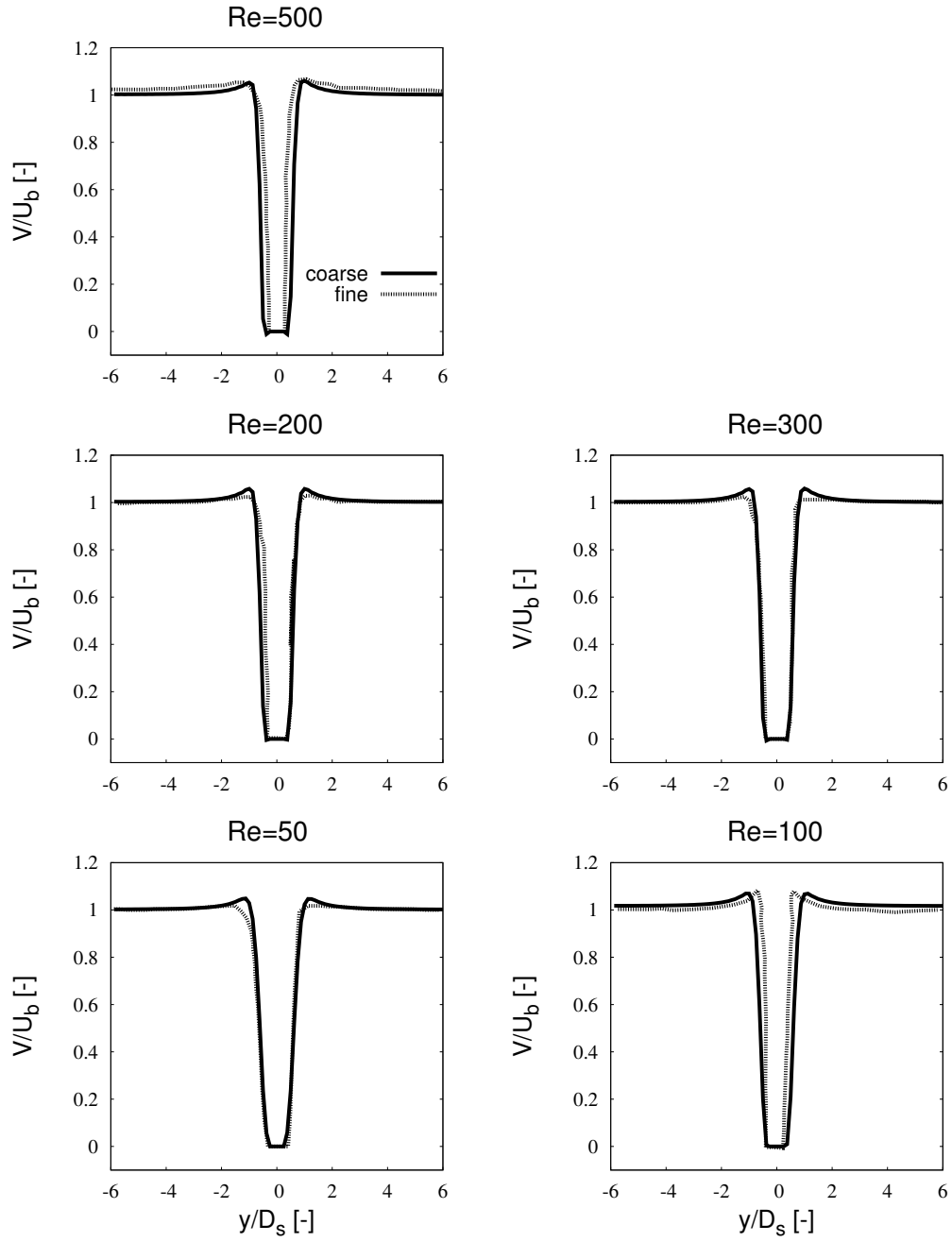


Figure 5.14: Statistical measurements of first moments in the along the ordinate axis through the centre of the sphere. Five mean velocity profiles for Reynolds numbers $Re = [50, 100, 200, 300, 500]$ are shown.

5.1.3 Fractal Grid in Opposed Jet Flows

The work on Turbulent Opposed Jet flows is discussed in this section. A summary of previous experimental and numerical investigations of reactive and non-reactive opposed jet flows is presented. The burner geometry and experimental set-up are introduced, followed by details of the numerical modelling of the opposed jet using LES along with the boundary conditions. The simulation results are then shown and discussed, before drawing conclusions from the presented work.

5.1.3.1 Introduction and Specific Objectives

Opposed jet geometries have been extensively used to study isothermal and reacting laminar and turbulent flows. The compactness of the flow domain and planar flame shape are beneficial to experimental and numerical studies alike. The set-up provides good optical access to the stagnation plane and flame for laser-based measurement techniques used to obtain two-dimensional velocity and scalar data, as well as for intrusive measurement techniques (e.g. thermocouples, hot wire anemometry, etc.). Furthermore, the effects of heat loss to a cold surface (e.g. flame holder, impinging plates, etc.) are avoided by aerodynamically stabilising the flame between the two nozzles. However, the turbulent flow in the opposed jet geometry is often limited to moderate Reynolds numbers ($Re < 15,000$), before flame blow-off occurs.

The opposed jet set-up has been subject to numerous experimental and numerical investigations of both laminar and turbulent flows, as well as non-premixed, partially-premixed and fully premixed combustion. A vast amount of literature is available and the following is a brief summary of previous work relevant to the current investigation.

Isothermal and reactive opposed jet flows have been investigated experimentally and numerically by several authors. Kostiuk et al. [92] used Laser Doppler Velocimetry (LDV) for isothermal velocity measurements in non-reacting, turbulent opposed flows and reported similar axial mean flows for a range of turbulence and flow conditions. The same authors also investigated twin propane flames close to extinction by LDV and discovered that the flame brushes move

closer together, merge and eventually blow-out as the bulk inlet velocity increases [93]. Mastorakos [113] used the geometry for experimental investigation of various modes of combustion (non-premixed, partially and fully premixed) using LDV and fine-wire thermocouples. The temperature characteristics of partially premixed counter-flows and the scalar dissipation rate of non-premixed opposed jet flames at extinction were measured by Mastorakos et al. [114], and work on the opposed jet configuration followed by Sardi et al. [160] and Geyer et al. [56]. Lindstedt and Luff [102] studied lean premixed opposed jet flames using Particle image velocimetry (PIV) and thermocouples and compared the accurate experimental statistics along the burner axis to RANS simulations, applying eddy viscosity and second moment closure models. The previous work focused on flame structure and extinction behaviour of turbulent opposed jet flames.

Rolon et al. [154] used LDV to study laminar opposed jets of air and they showed that the axial and radial velocity gradients were constant at the axis and the stagnation plane. Opposed water jets were investigated by Stan and Johnson [172] and they found an increase in turbulence intensity at the stagnation plane, resulting from highly unsteady motion of the jet impingement region. Their jets were aligned horizontally as opposed to the conventional vertical alignment. Experimental data was also compared to RANS simulations with the same configuration. Kurosoy and Whitelaw [96] used hot wire, pressure probes and LDV to investigate isothermal and premixed turbulent opposed flows at small nozzle separation. They found an increasingly non-uniform radial velocity profile at the nozzle exit with small nozzle separation. The pressure measurements also showed an increase in static pressure with smaller nozzle separations. Lindstedt et al. [103] studied isothermal and reactive opposed jet flows by using Particle Image Velocimetry (PIV) and RANS simulations to characterise velocity and strain rates with different turbulent models. Perforated plates were used to ensure turbulent conditions. The work by Lindstedt et al. [103] was continued by Geipel [51], who investigated different fractal grids to increase the turbulent Reynolds number without increasing the bulk strain rate.

In the framework of Reynolds-Averaged Navier-Stokes (RANS) simulations, a number of studies have been published by Craft et al. [30, 31], Lindstedt

and Vaos [104], Chou et al. [24] and Eckstein et al. [39]. These studies are mainly based on one- and two-dimensional simplifications of turbulent stagnation point flows. In most cases the mean quantities were predicted more accurately than fluctuating components and the Reynolds stress turbulence models usually performed better than the eddy viscosity approaches. Simulations often struggle with the boundary conditions and effects of jet asymmetries. Craft et al. [30, 31] compared the performance of four turbulence models ($k - \epsilon$ and three second moment closures) based on a jet impinging onto a wall, including wall effects. Lindstedt and Vaos [104] focused their numerical work on premixed flames stabilised in stagnating turbulence using second moment closure for velocity and scalar statistics. Eckstein et al. [39] analysed scalar mixing in turbulent opposed jet flows using one-dimensional Monte-Carlo simulations. It was found that the results strongly depended on the chosen data sets for comparison.

In terms of LES of opposed jet flames, Kempf et al. [83] applied Large-Eddy Simulation to opposed jet flames to simulate the isothermal flows and non-premixed flames experimentally investigated by Mastorakos [113] and Geyer [54, 55]. One burner nozzle was supplied with air, the other with methane. The numerical predictions agree well with the experimental data. More recently Stein et al. [174] studied isothermal flows in the opposed jet configuration of Lindstedt et al. [103] in great detail and the comparison of the results from LES and experiments shows good agreement, both qualitatively and quantitatively. Pettit et al. [139] carried out an LES of Gomez' opposed jet burner and found good accordance of the simulation with available experiments. Further details of opposed jet simulations may be found in the work of Geipel, Geyer and Stein [51, 54, 174].

The majority of opposed jet development work aims to increase turbulence levels without increasing the strain rate. A significant increase in the turbulent Reynolds number, while maintaining a constant bulk strain rate, was achieved by Geipel [49], through the use of a fractal turbulence-generating grid positioned downstream of a conventional perforated plate. The generated turbulence appears to be fully developed and isotropic near the nozzle exit plane. The scaling and decay of fractal-generated turbulence has been investigated by Vassilicos et al. [73, 116, 164, 185]. The use of fractal grids in the flow

allows for greater turbulence generation in comparison to applying a simple perforated plate. This is the result of different length-scales being present in the flow close to the fractal grids, reducing the length to achieve isotropic turbulence.

In the present work, Large-Eddy Simulation is used to acquire a more complete understanding of the mechanisms that provide the higher level of turbulence, by allowing inspection of the flow field development within and in-between the nozzles of the opposed jets for isothermal and reactive configurations. Furthermore a successful simulation of the opposed jet set-up shows the applicability of the immersed boundary method to complex geometries. The flow in the opposed jet configuration is very sensitive and the complex structures of the fractal grid have to be modelled accurately to investigate the flow within the nozzles and the region of interest. Even on unstructured grids, a high grid resolution is required to accurately represent the flow phenomena. Statistical measurements of first and second moments of velocities and progress variable in the turbulent opposed jet configuration, including a perforated and fractal plate, are compared to experimental measurements obtained by Geipel et al. [51]². In order to accurately represent the complex geometry of the turbulence generators, grid resolutions typical of a Direct Numerical Simulation are required.

5.1.3.2 Experimental Configuration

The opposed jet burner consists of two identical nozzles in vertical alignment designed by Geyer [53]; a schematic drawing of the nozzles is shown in Fig. 5.15. A nozzle separation $H = 30$ mm was used in the experimental work, such that $H/D_n = 1$ (where D_n is the diameter of the nozzle). In the current configuration turbulence is generated within each nozzle using both a perforated plate and a fractal grid, located 60 mm and 50 mm upstream of the nozzle exit plane respectively. The perforated plate holes have a diameter of 4 mm and a separation of 5.2 mm, yielding a blockage of 45%, similar to the ones used by Mastorakos et al. [113]. The fractal bars range from 2 mm to 0.5 mm in width, resulting in an overall blockage of 65%. A schematic drawing of the

²The reactive results, and therefore the corresponding LES data, is currently not published [50].

plate and the grid is shown in Fig. 5.16. Geipel [49] performed in-nozzle PIV measurements with a glass nozzle of identical dimensions to the conventional steel nozzle, based on the technique of Böhm et al. [14], to record axial mean velocity profiles and fluctuations at distances between 20 and 40 mm downstream of the fractal grid. The fuel mixture for the reactive measurements was methane with an equivalence ratio $\Phi = 0.8$ and bulk velocity $U_b = 4$ m/s. Further information on the experimental configuration is given by Geipel et al. [51].

5.1.3.3 Numerical Simulations

The presented simulations of the opposed jet flows and flames were carried out with the LES code *PsiPhi*, which is described in Chapter 3. Two different domains, single jet (*SJ*) and opposed jet (*OJ*), were used in the modelling of the turbulent opposed jet flows. The single jet domain consists of a single nozzle with a solid wall at the stagnation plane. This was used for a parametric study since the sensitivity of simulation results to inflow conditions and other numerical parameters were previously reported [173] for measurements with perforated plates only. The parameter study focused on isothermal flows for the single jet domain. Using the shortened domain leads to a significant reduction in computational efforts and costs. The opposed jet domain was used for the analysis of the in-nozzle velocity field, including a time series analysis, for an isothermal flow, and the integral length-scale was evaluated. Furthermore, the full geometry was modelled for the reactive simulations to fully capture the flow behaviour between the nozzles. The ATF model, with a thickening factor of $T_f = 2$, was used to describe the premixed combustion process, as discussed in Section 2.5.

The single jet domain is a shortened simple approximation of a single nozzle of the opposed jet geometry, without the contraction or co-flow, and with a wall at the stagnation plane. In LES, turbulent flows are recognised as a three-dimensional problem and no symmetry assumptions are permitted. The overall domain size was $90 \times 32 \times 32$ mm, and included a 11 mm region upstream of the perforated plate. The opposed jet domain had overall dimensions of $180 \times 32 \times 32$ mm and consisted of the two opposing nozzles. The co-flow was omitted here too, because the main area of focus is the flow field in the

nozzles behind the fractal grid.

Two different filter (cell) widths of $\Delta = [0.2, 0.1]$ mm were used for both domains and are referred to as the *normal* and *fine* grids, respectively. Thus, leading to meshes with a minimum of 11.8×10^6 cells and a maximum of 184×10^6 cells. The fine grid resolution, which is close to the Kolmogorov length-scale (estimated as $\eta \approx 0.1$ mm [49, 173]), was required to represent the fractal grid accurately. Hence, the simulations can be termed *near DNS*. Figure 5.17 shows the computational domain of the opposed jet configuration. Transient Dirichlet boundary conditions were applied at the inflow planes and von Neumann conditions with enforced outflow (clipping of negative values to 0 to improve simulation stability) were applied at the outlets of the domains. The Smagorinsky model (see Section 2.4.2) was used to estimate the subgrid viscosity, as it offers good numerical properties and is computationally cheaper than the memory-intensive Germano sub-grid scale model. Immersed boundary conditions were used to represent the nozzle walls, the perforated plate and the fractal grid.

The current study focused on cases with a bulk velocity $U_b = 4$ m/s and an overview of the key parameters of the numerical simulations is presented in Table 5.1. Five different cases were investigated during the numerical work on the opposed jet configuration: the effect of a wall at the stagnation plane on the in-nozzle flow field (isothermal cases only); the influence of the turbulence fluctuations at the inlet on the velocity field within the nozzle; the effect of the sub-grid scales on the simulation results based on the Smagorinsky model; and the analysis of the isothermal and reactive velocity flow field in the nozzle and between the jets.

5.1.3.4 Results and Discussion

The detailed analysis of the velocity field in the nozzle and at the nozzle outlet is based on the opposed jet configuration shown in Fig. 5.17. In this case the statistical measures of first and second moments for the velocity field are compared to experimental data obtained by Geipel et al. [49, 51] across the nozzle diameter and along the nozzle centreline. Furthermore the longitudinal auto-correlation and integral length-scale are evaluated at various points along the

centreline of the nozzle. Inspection of Fig. 5.17 provides an initial indication of the physics of the flow. The axial velocity field clearly shows the formation of many small, high-velocity jets, with a strong recirculation zone behind the central bar of the fractal grid. Rapid jet break-up results from unstable shear layers formed in the wakes of the grid. The strongest turbulence was found in this region, as indicated by the peak in-plane lateral velocities. Small-scale structures then dissipated downstream, while the decreasing magnitude of lateral velocities suggests a reduction in turbulence levels. In the case of the opposed jet, the velocity data is normalised by the bulk velocity U_b and the axial coordinate x by the nozzle radius R , unless otherwise stated.

Single Jet vs. Opposed Jet The single jet geometry was compared to the opposed jet configuration, to ensure that the flow fields within the nozzle were identical to avoid effects of the wall or second jet that might influence the simulation results. The simulation grid had a spacing of $\Delta = 0.2$ mm with identical inflow conditions and boundary treatment. Figure 5.18 shows a comparison of the normalised axial mean velocity profiles and the fluctuations at three locations $x = [20, 30, 40]$ mm downstream from the fractal grid. It is clearly noticeable that the mean velocity profiles are indistinguishable. The difference in the velocity fluctuations is negligible. Statistical measurements could benefit from a higher sampling count and longer run times should minimise these differences. However, it is clear that single nozzle simulations are sufficient for the parameter study of isothermal cases. In Fig. 5.18 it is also very noticeable that the centreline velocities are under-predicted and jet peaks can still be found at the first point of interest. The peaks in the mean axial velocity profile result from a delayed jet break up, which also influences the velocity fluctuations at $x = 20$ mm. The delayed jet break-up may be attributed to the behaviour of the Smagorinsky model, which might reduce the fluctuations as the flow passes through the holes in the fractal grid, thus increasing the stability of the jets.

Inflow Study: It has been previously reported [173], that the flow field in the opposed jet geometry is dependent on the inflow conditions. Therefore the effects of the turbulent intensities at the inlet have been investigated.

Stein [173] used a perforated plate layout as the inlet condition, whereas in the present simulations a perforated plate and fractal grid were used for the increase of turbulence in the flow field. Increasing the prescribed turbulence intensity at the inlet leads to a higher initial disturbance. The investigated values of u' ranged from $u' = [0.1, 0.2, 0.3]$ m/s and results of the normalised mean axial velocity profiles and fluctuations are shown in Fig. 5.19. It is clear that the mean axial velocity profiles and fluctuations are identical for all three parameter values and the downstream measurements are independent of the inflow conditions upstream of the turbulence generating plates. The fluctuations show good agreement with the experimental results, especially towards the nozzle exit plane. A repeat of the test with different prescribed integral length-scales also showed no change in the mean velocity profiles and fluctuations. Therefore it can be assumed that the flow field characteristics are strongly dependent on the turbulence generators, but independent of the flow upstream, so that this region does not need to be simulated. At the same time the experiments are very reproducible.

Subgrid Study: As mentioned above, in the present work the eddy viscosity approach is applied to represent the sub-grid contribution to the turbulent energy balance by adding a turbulent viscosity to the molecular viscosity. In this case, the Smagorinsky model was used. Simulations often suffer from the arbitrariness of the model constant, no criteria for parameter selection exist. An alternative is the Dynamic Germano model, but it is very memory intensive and computationally slow. Considering the grid resolution applied to the case, a serious increase in run time would result. Therefore, a sub-grid scale parameter study was carried out on the fine grid ($\Delta = 0.1$ mm). The applied model values ranged from $C_s = 0 - 0.065$ and simulation results of the normalised axial mean velocity profiles are shown in Fig. 5.20. A value of 0 meant that the LES simulation could be regarded as a DNS under the condition that the flow field is resolved beyond the Kolmogorov length-scale. A lower Smagorinsky constant leads to a decrease in viscosity, which resulted in higher fluctuations in the flow. The axial mean velocity profiles match the experimental results well at all considered positions. However, the statistical convergence of the fluctuations needs to be reconsidered, although even at this stage a considerable effect of C_s was found. Based on albeit early statistics, the fluctuation

levels vary for C_s of 0.065 and 0.000, where a low but non-vanishing C_s of 0.030 appears optimal on this grid. The grid requires further refinement for a Direct Numerical Simulation in order to fully resolve the flow field.

Isothermal Study: Figure 5.21 shows the normalised mean axial velocities and fluctuations of the production run at three positions ($x = [20, 30, 40]$ mm) downstream from the fractal grid. Overall, an excellent agreement is observed between the simulation and the experiment, particularly for the velocity fluctuations at all three locations. An under-prediction in mean axial velocity around the centreline is found nearer to the fractal grid. This under-prediction may be accounted for by the dependence of predicted velocity profiles within the nozzle on the position at which the jets from the fractal grid break up: in this case, it is speculated that a delayed break-up resulted in more pronounced peaks in the mean velocity profile close to the grid, as well as an elongation of the recirculation zone downstream of the central cross. This is also indicated by the slight over-prediction in velocity fluctuations at $x = 20$ mm, where the spikes in u' correspond to the oscillating shear layers at the edges of these extended jets. The cause of the delayed break-up may be attributed to the behaviour of the Smagorinsky model, the effect of which may be to reduce fluctuations as the flow passes through the holes in the fractal grid, thus increasing the stability of the jets as they form downstream.

The normalised mean axial velocity profile and fluctuations along the centreline of one of the nozzles in the opposed jet configuration are shown in Fig. 5.22. Overall a good agreement is observed between the simulation and experimental data for the mean profiles. The statistical measurements for the second moments along the centreline do not agree with the original experimental data, which also shows a deviation from the equivalent point in the radial profile. Preliminary, confidential data also showed that a different inlet set-up and seeding material has a strong effect on the fluctuations in the experiments, but not the mean profiles [50]. These findings require further investigations and the results will be published in the near future by the research group of Prof. R.P. Lindstedt.

Analysis of time series recorded along the centreline provides further insight

into the development of the flow within the nozzle. Longitudinal auto-correlation functions at seven locations downstream of the fractal grid have been calculated using the Taylor hypothesis, and are shown in Fig. 5.23. The time-scale over which axial velocity at a given point retains some correlation with itself increases monotonically downstream of the grid. This suggests that larger eddies form whilst smaller ones dissipate towards the nozzle exit, as it would be expected and the flow becomes homogenised.

The integration of the auto-correlation functions provided an estimate of the integral length-scale for the flow at each considered position. By multiplying the time-scale with its corresponding mean velocity, an estimate of the integral length-scale L_i was calculated. It should be noted that this analysis assumed isotropy, which may not hold near to the fractal grid. Hence, values near the grid only provide a qualitative indication of how the length-scale evolves. The turbulent Reynolds number at each position is also approximated and these results are presented in Fig. 5.24. Inspection of the results confirms that the integral length-scale increases approximately linearly, from around 1.2 mm at a position 5 mm downstream of the fractal grid, to just over 4.0 mm at a point 2.5 mm beyond the nozzle exit. The turbulent Reynolds number, which may be used as a measure of the turbulence intensities, increases rapidly to a peak value in excess of 200, just downstream of the fractal grid. This resulted from elevated fluctuations within the recirculation zone behind the central cross. The value of Re_t then dropped rapidly over the next 15 mm as these fluctuations attenuated. Along the remainder of the nozzle, a gradual decrease in u' was matched by a corresponding increase in L_i , resulting in an approximately constant value of Re_t . Beyond the nozzle exit, the turbulent Reynolds number was found to be 107, similar to the value $Re_t = 109$ determined from experimental data.

Reactive Study: The mean axial velocity profile and fluctuations along the centreline between the nozzles for the reactive configuration are shown in Fig. 5.25. The mean axial velocity profiles showed the largest deviation from the experimental data in the nozzle exit regions and results improve across the reaction zone, towards the stagnation plane. The under-prediction of the axial velocities was already noticeable in the isothermal simulations, where a

delayed jet break up from the fractal grids was thought to cause the low axial velocities. The statistical measurements for the second moments show deviations from the experimental data. Nonetheless, similar line characteristics can be established. The peak in the centre of the reaction was slightly shifted as a result of the flame flapping.

Figure 5.26 shows the mean progress variable and the progress variable variance along the centreline between the two nozzles. The presented results were taken from calculations running up to a real time of 0.3 s and approximately 120 samples were collected. The mean progress variable indicates a good prediction of the mean reaction zone and turbulent burning velocity. The statistical data for the progress variable variance is in good agreement with the experimental data. This indicates a good prediction of the flame flapping phenomenon, which results from reasonably good capture of the turbulence-chemistry interaction. A snapshot of an instantaneous progress variable field is also shown in Fig. 5.27. The nozzles are in vertical alignment and the yellow bars indicate the nozzle outlets. Unburnt fuel is shown in black, while burnt regions are shown in white.

5.1.3.5 Conclusion

The fractal grid, inspired by the research of Vassilicos et al. [73, 116, 164, 185], has been investigated experimentally in an opposed jet flow by Geipel et al. [51]. Highly-resolved Large-Eddy Simulations were performed to gain a more complete understanding of the flow, and to examine the ability of LES to simulate turbulence production downstream of the fractal grids. These have shown to deliver increased turbulent Reynolds numbers compared to cases employing conventional perforated plates only. Simulation results were compared to in-nozzle measurements of an isothermal and reactive test case. The numerical and experimental data generally agree for the first two moments of axial velocity for all nozzle locations considered. An under-prediction of mean axial velocity on the centreline near the fractal grid is possibly caused by the influence of the applied eddy viscosity sub-grid model leading to a delayed jet break-up.

Time series of axial velocities have been analysed to determine

auto-correlation functions, integral time-scales and length-scales and turbulent Reynolds numbers. This analysis provides some insight into the evolution of the flow, as the time series are recorded at seven positions along the centreline downstream of the fractal grid. The integral length-scale is found to increase approximately linearly and turbulent Reynolds numbers peak in the highly turbulent recirculation zone behind the central cross of the fractal grid, before dropping to an approximately constant value. The predicted value of Re_t slightly downstream of the nozzle exit matches well with that determined from experiments.

The artificially thickened flame model was used for the simulation of a reactive opposed jet flow with a methane flame. The model is a simple and cost-effective method, which can capture the turbulence-chemistry interaction reasonably well, resulting in accurate predictions of the mean progress variable and progress variable variance.

The combination of highly-resolved simulations and in-nozzle measurements has confirmed that the numerical prediction of in-nozzle turbulence is sufficiently detailed to allow accurate simulation of reactive opposed jet cases. In addition, the immersed boundary method is considered suitable for modelling flows in complex geometries without moving surfaces, while achieving a good accuracy of first and second statistical measurements.

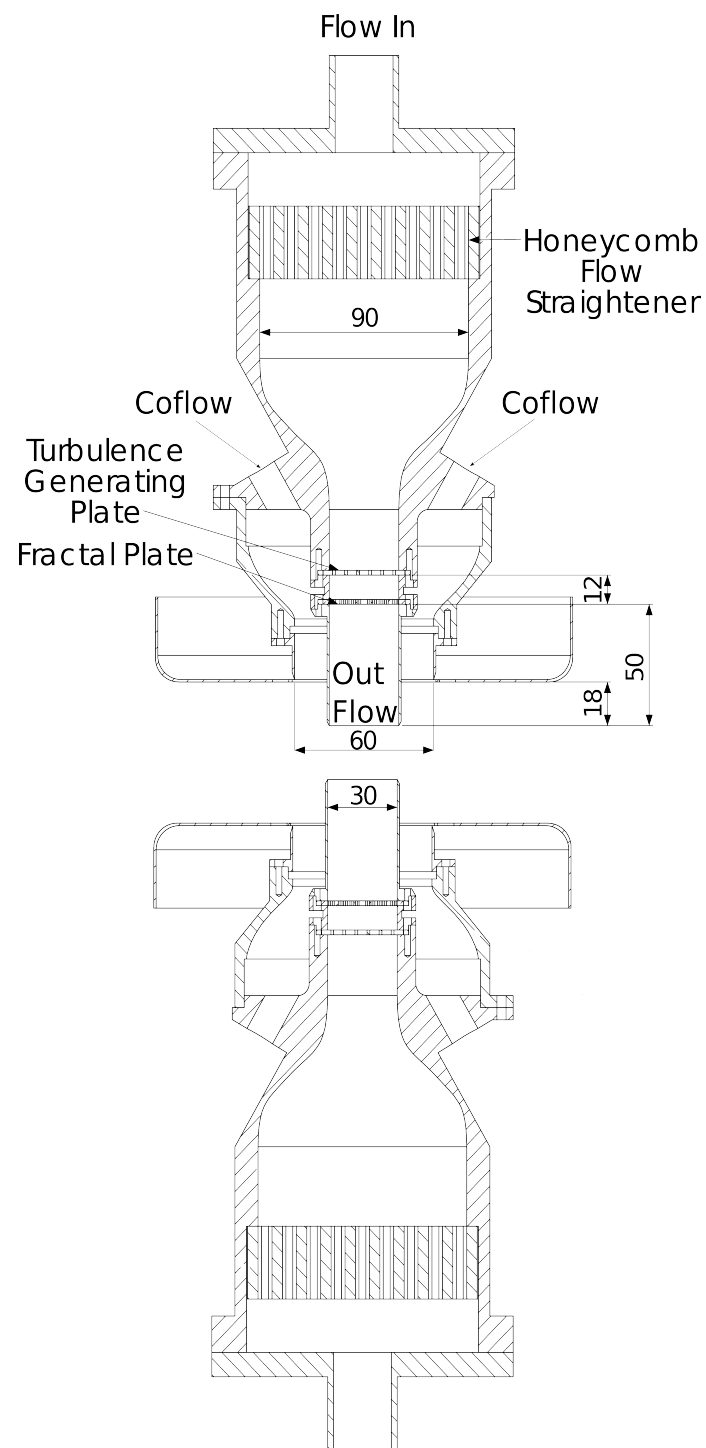


Figure 5.15: Schematic of the opposed jet geometry, adapted from Geipel et al. [49].

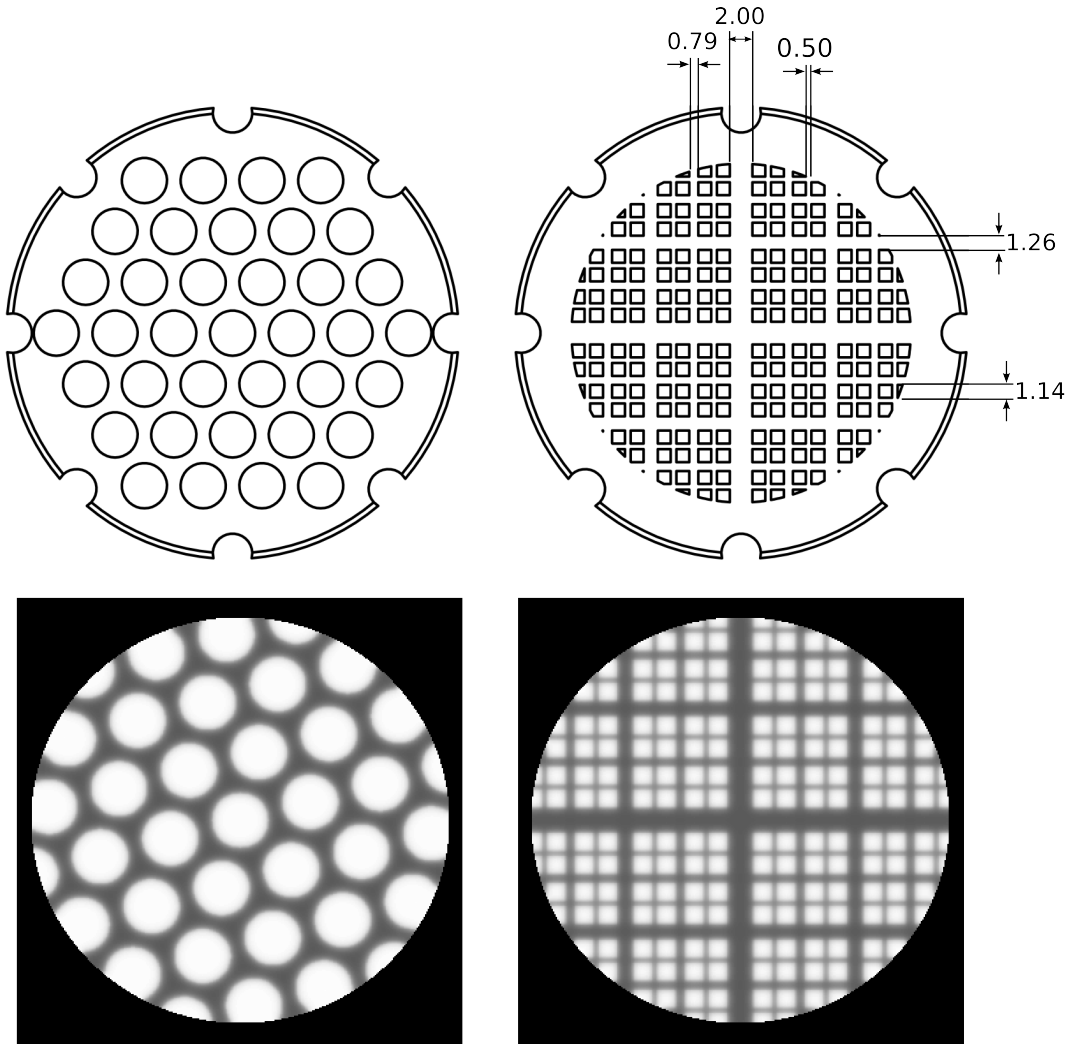


Figure 5.16: Schematic of the perforated plate (*left*) and fractal cross-grid (*right*) used in the experiments (*top*) and numerical simulations (*bottom*). Dimensions are in mm and the cell size for the numerical grid was $\Delta = 0.1$ mm.

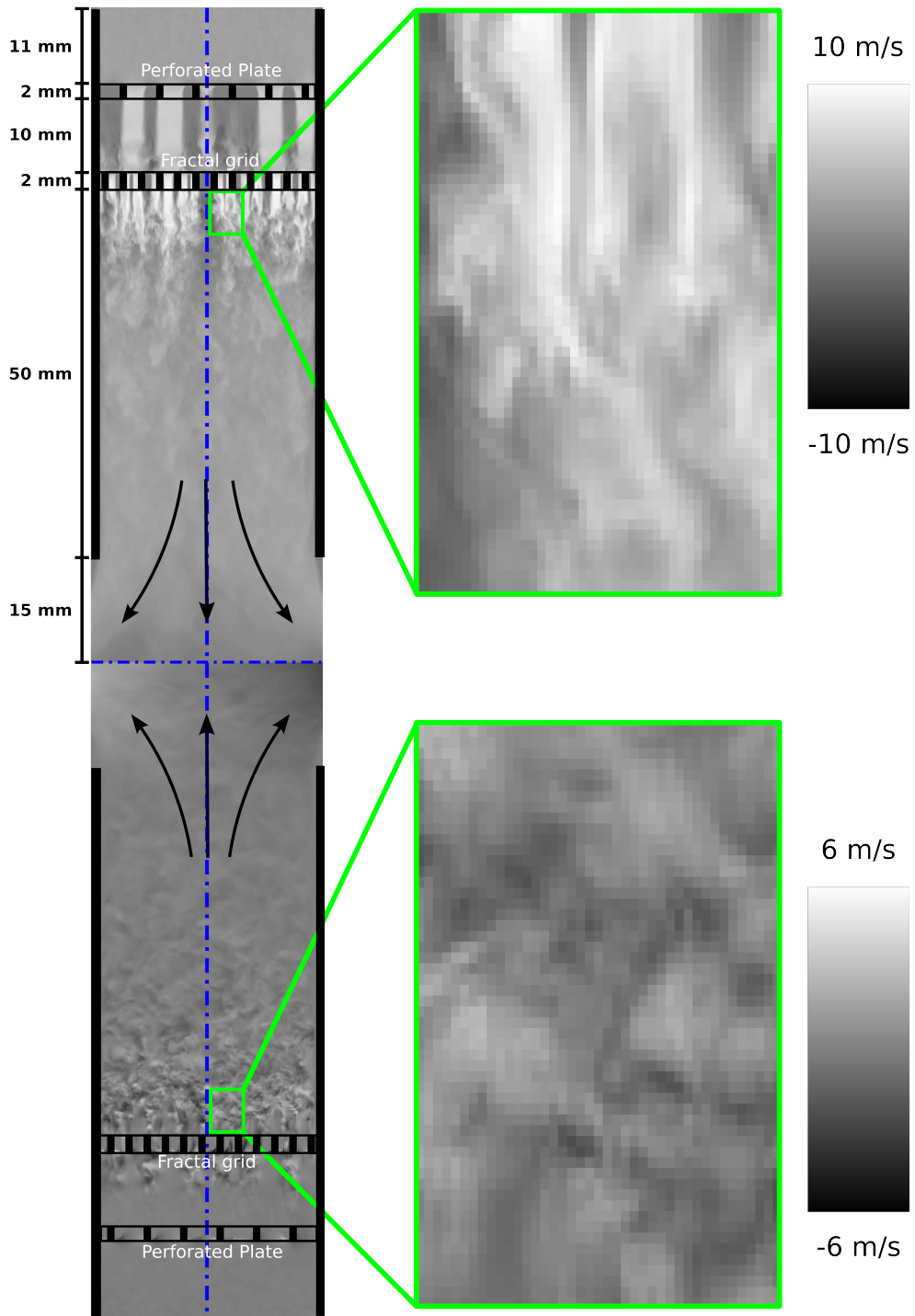


Figure 5.17: Schematic of the numerical opposed jet configuration with symmetry indicated by the dash-dotted lines. Isothermal axial velocity field (top nozzle) and in-plane lateral velocity field (bottom nozzle) in the configuration with a grid resolution $\Delta = 0.1$ mm and bulk velocity $U_b = 4$ m/s. Maximum positive velocities are shown in white, while maximum negative velocities are in black.

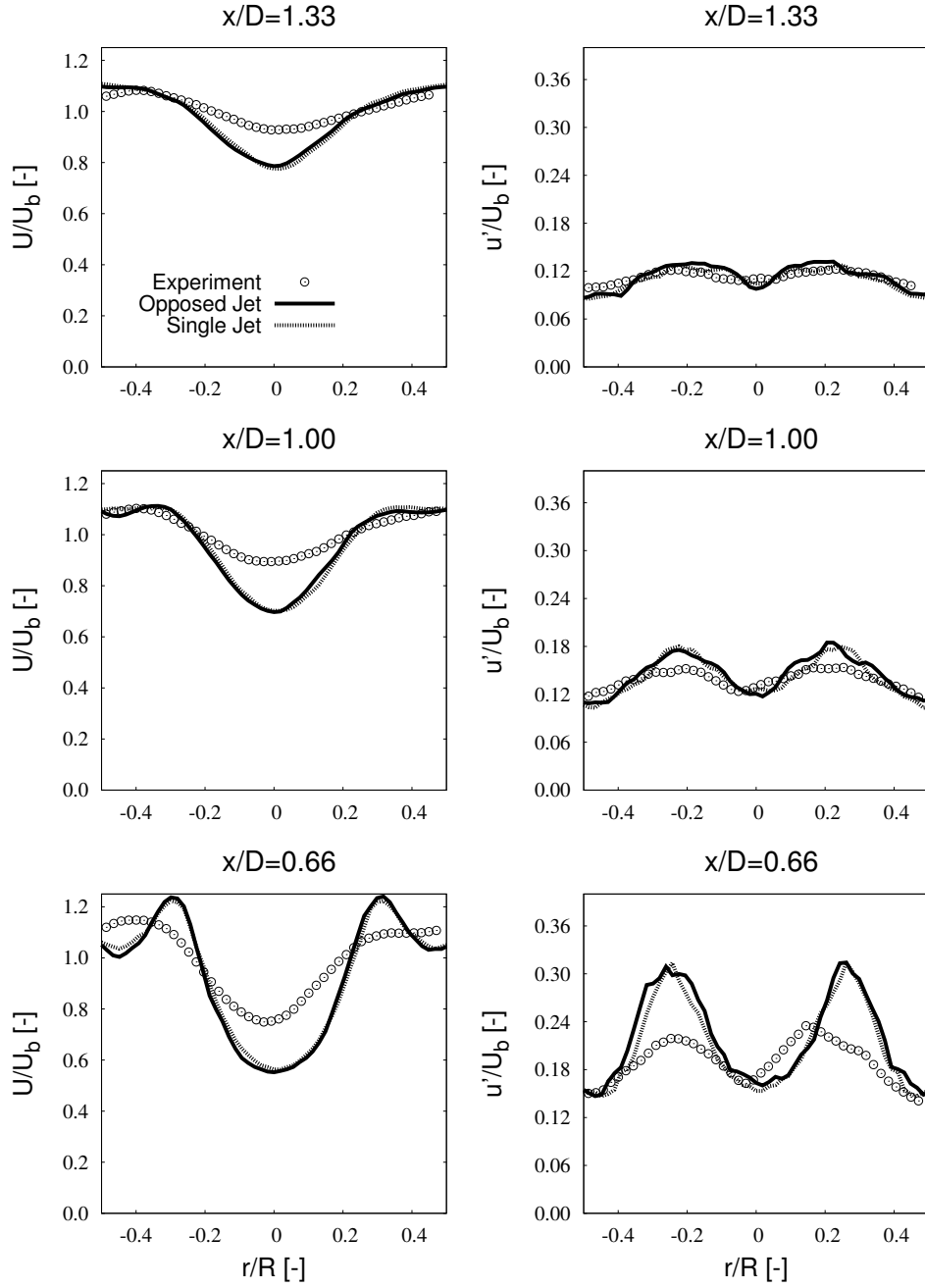


Figure 5.18: Comparison of statistical measurements of first (left) and second (right) moments from a single jet and two opposed nozzles for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.2$ mm and $C_s = 0.065$. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.

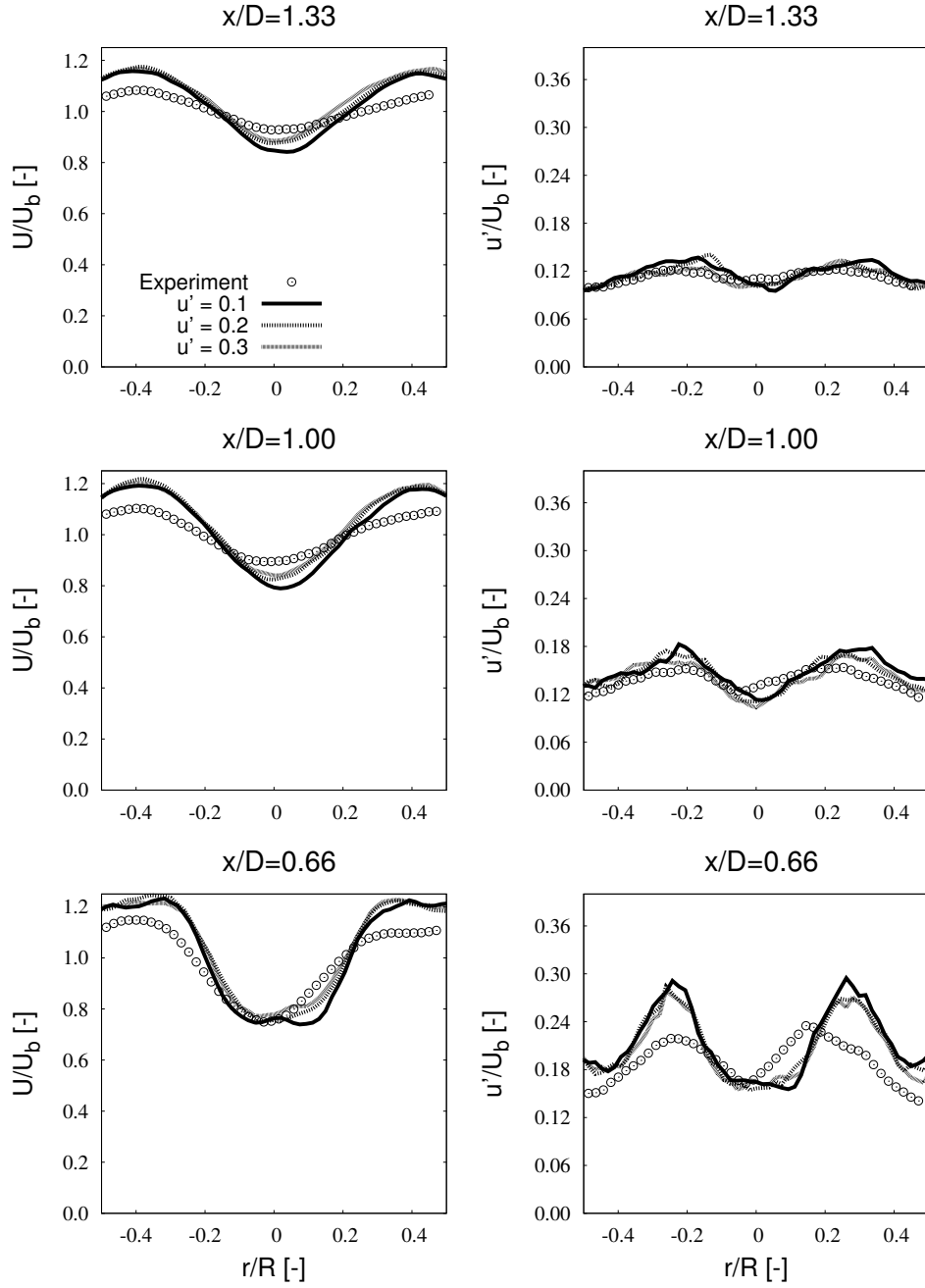


Figure 5.19: Comparison of statistical measurements of first (left) and second (right) moments for different inflow fluctuations, a bulk velocity $U_b = 4$ m/s, $\Delta = 0.2$ mm and $C_s = 0.065$. Three positions of a single jet are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.

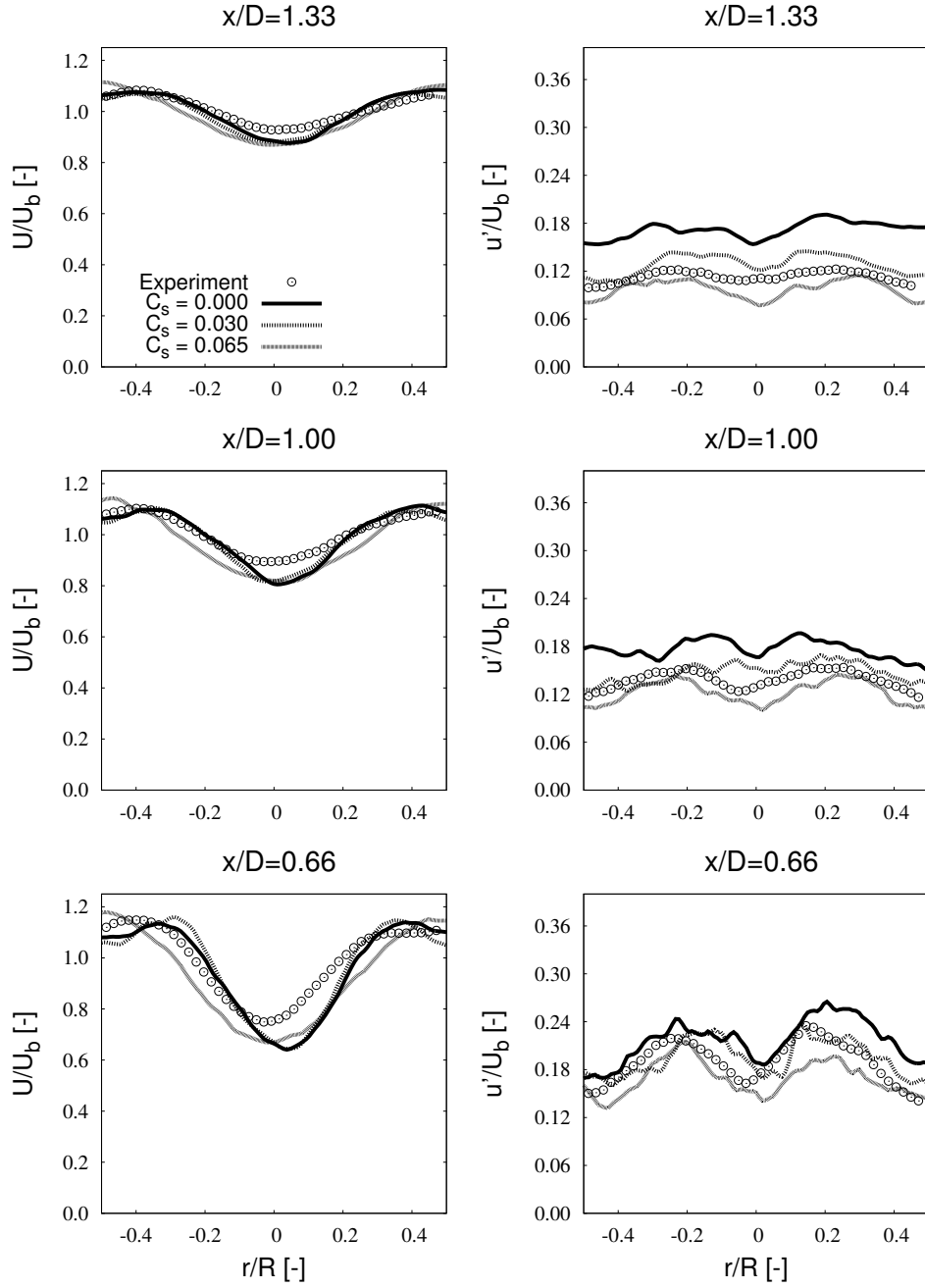


Figure 5.20: Comparison of statistical measurements of first (left) and second (right) moments for different sub-grid scale constants, a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = [0.000, 0.030, 0.065]$. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.

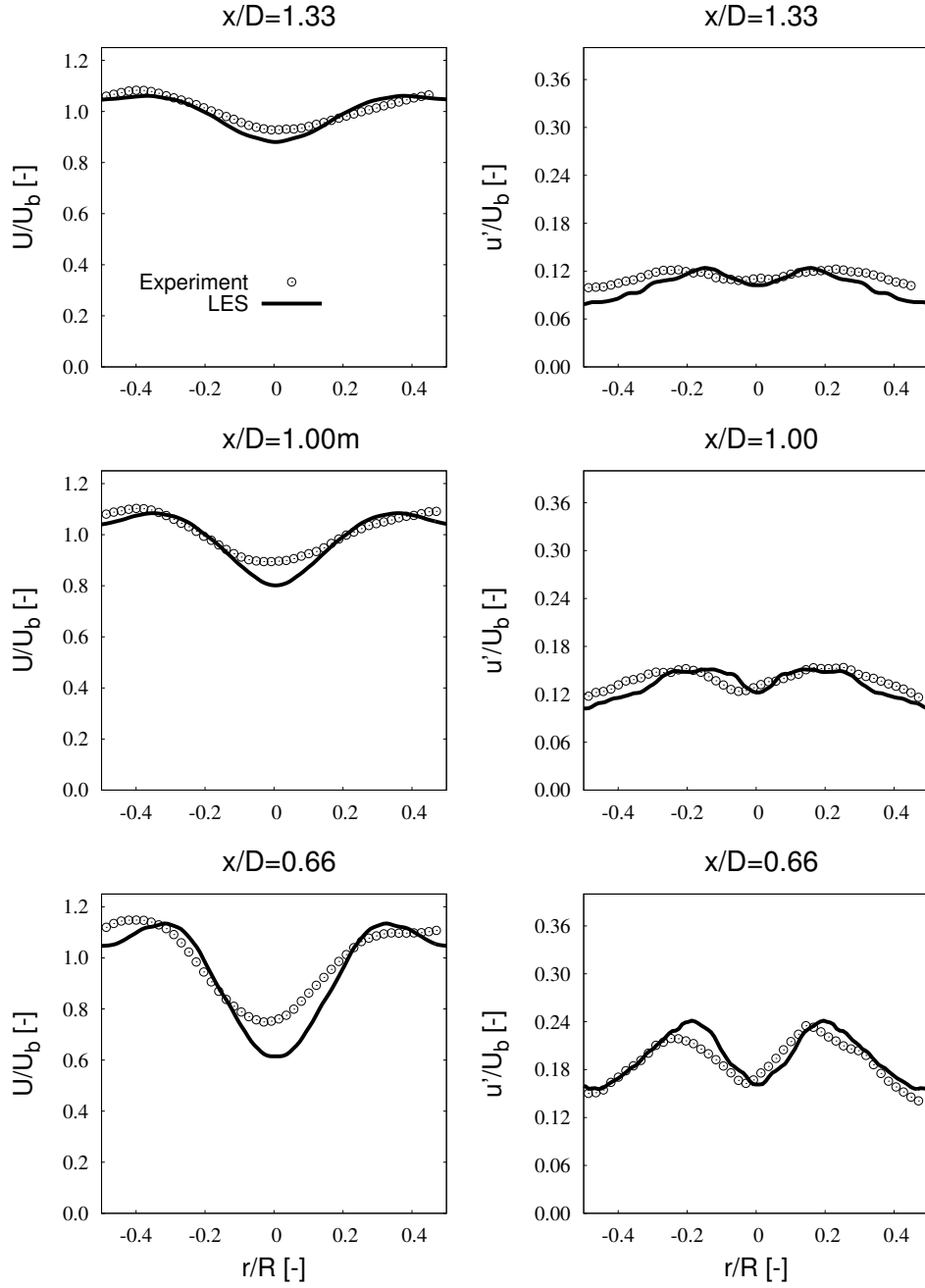


Figure 5.21: Statistical measurements of first (left) and second (right) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ in an opposed jet configuration for an isothermal flow. Three positions are compared to in-nozzle measurements at defined locations $x/D = [0.66, 1.00, 1.33]$ downstream of the fractal grid.

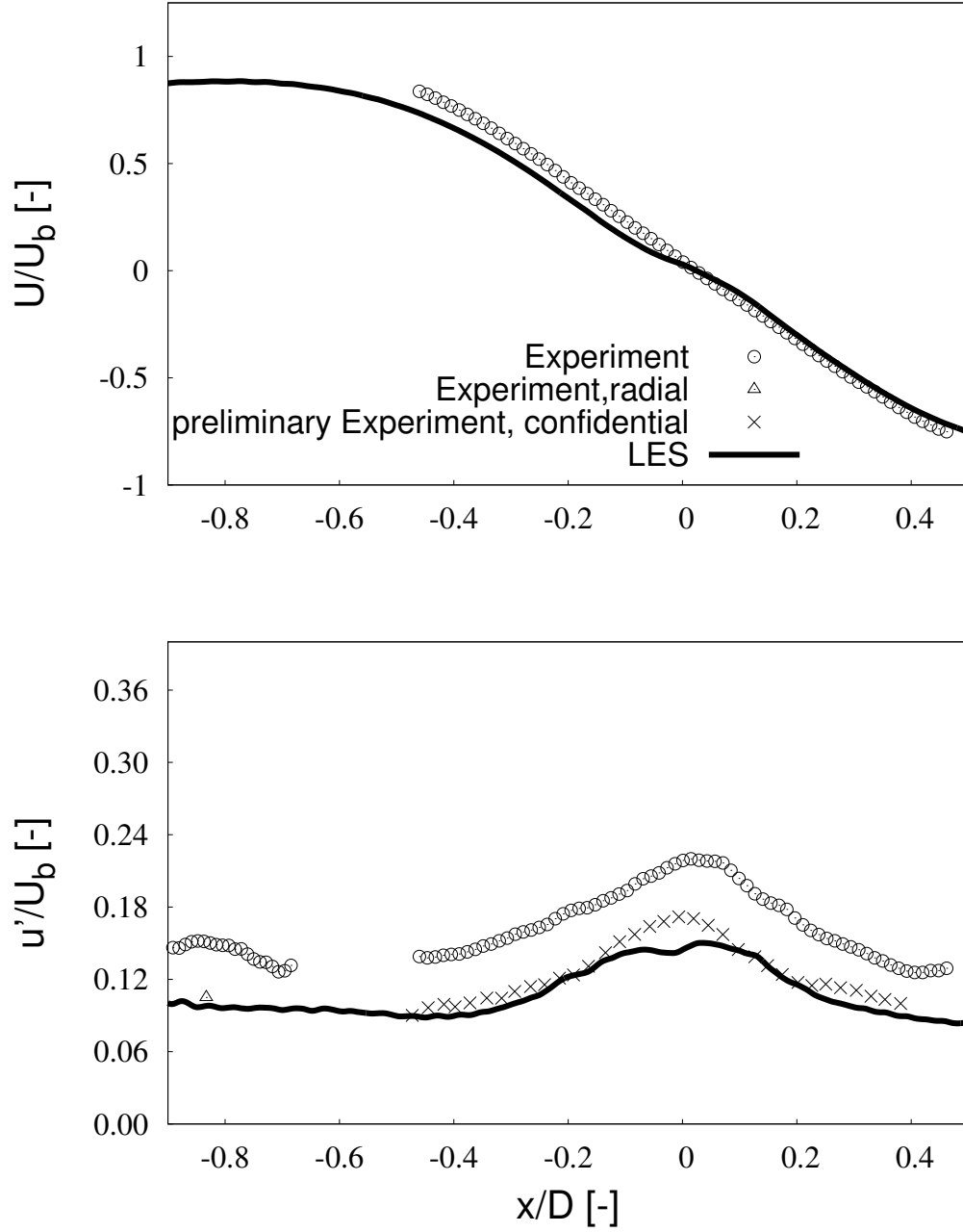


Figure 5.22: Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ along the centreline of the nozzle for an isothermal flow. Results are compared to experimental measurements, with point of reference ($x/D = 0$) being the stagnation plane.

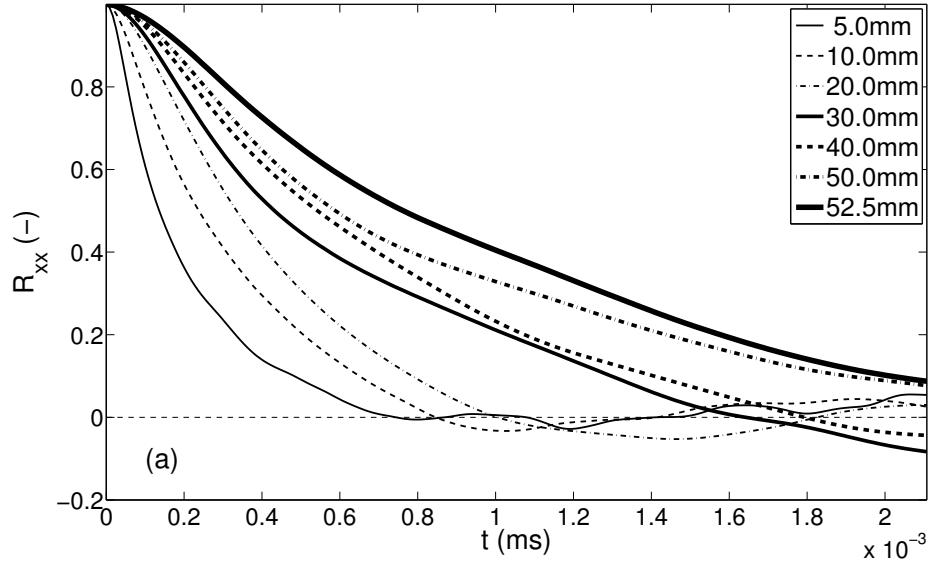


Figure 5.23: Longitudinal auto-correlation functions for defined positions downstream of the fractal grid in the opposed jet configuration for an isothermal flow.

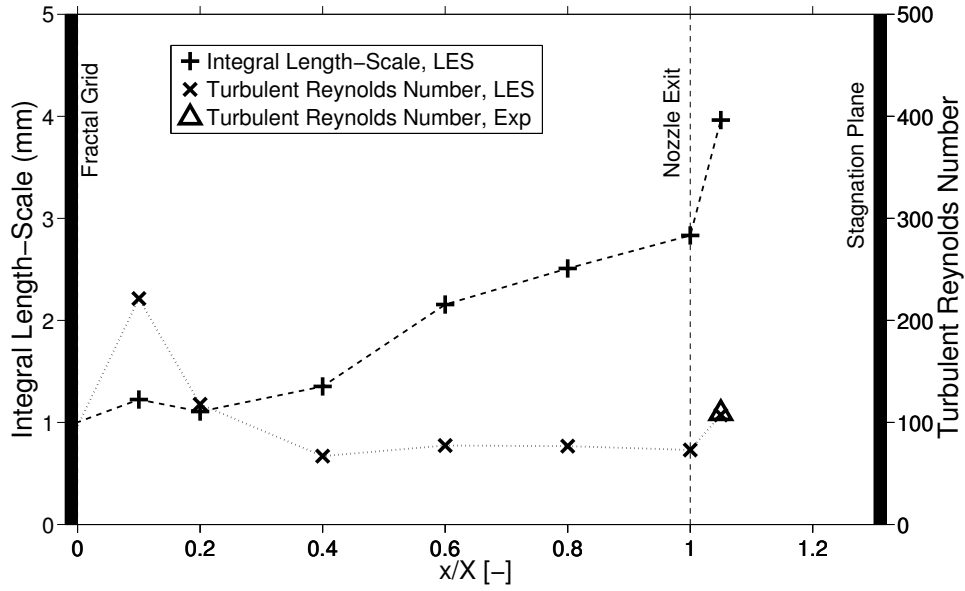


Figure 5.24: Integral length-scales and turbulent Reynolds numbers for defined positions downstream of the fractal grid in the opposed jet configuration for an isothermal flow. X is the point of reference (0 mm) for the fractal grid.

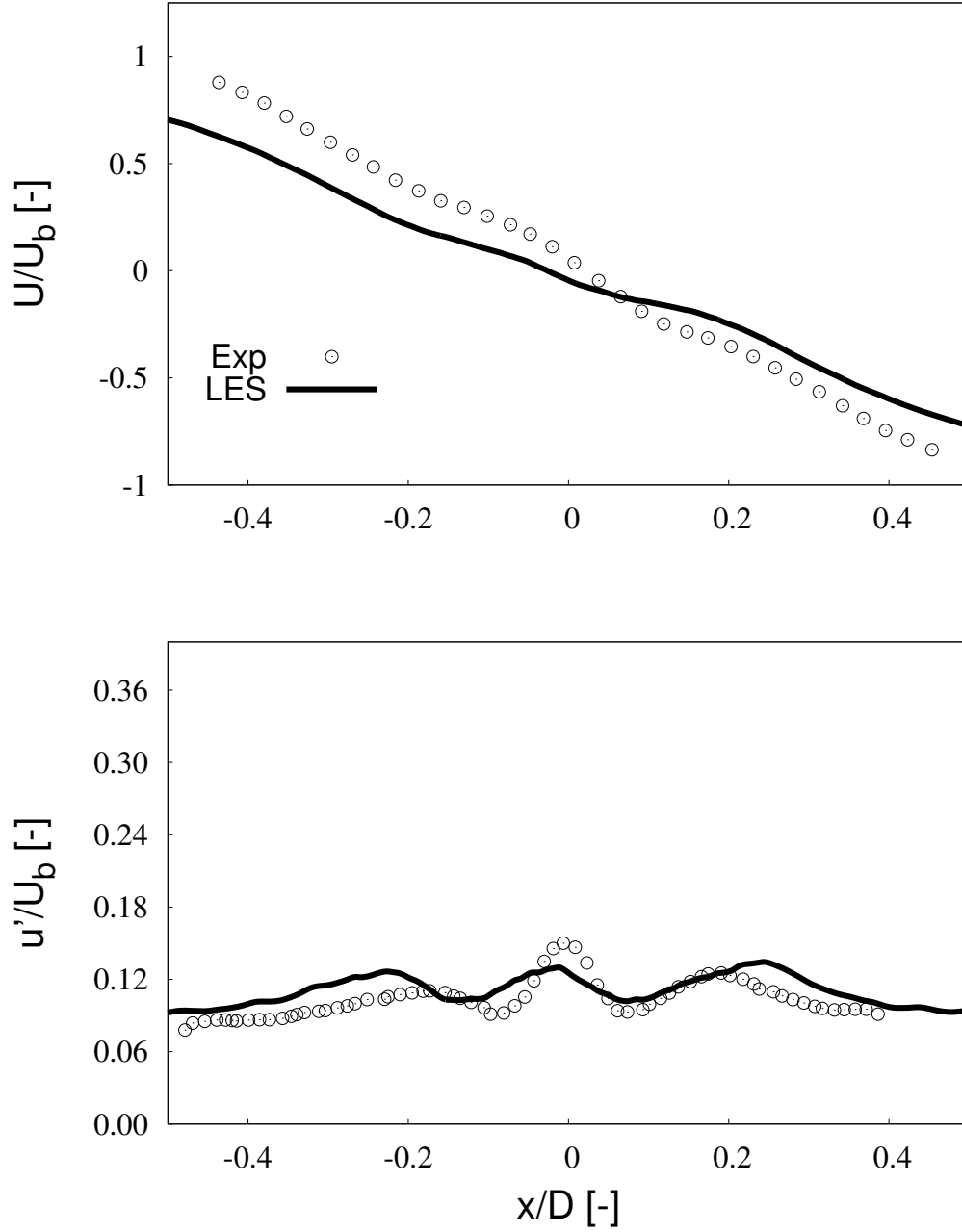


Figure 5.25: Statistical measurements of first (top) and second (bottom) moments for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $C_s = 0.030$ along the centreline of the nozzle for an reactive flow. Results are compared to experimental measurements, with point of reference ($x/D = 0$) being the stagnation plane.

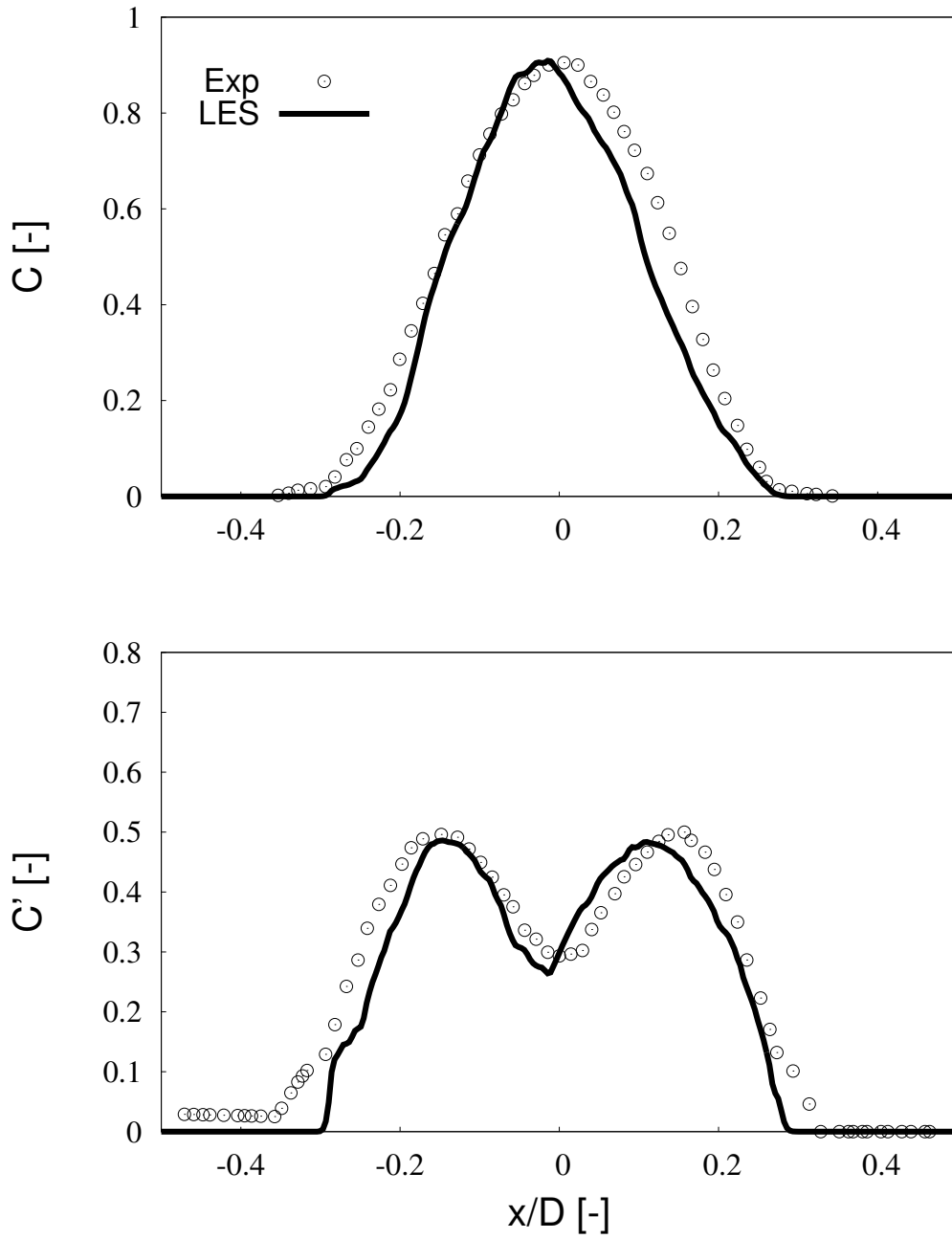


Figure 5.26: Statistical measurements of first (top) and second (bottom) moments for the progress variable in a reactive opposed jet configuration along the centreline of the nozzle. Results are compared to experimental measurements.



Figure 5.27: Visualisation of an instantaneous progress variable field between the nozzles for a bulk velocity $U_b = 4$ m/s, $\Delta = 0.1$ mm and $T_f = 2$. Burnt regions are shown in white, while unburnt regions are shown in black.

Table 5.1: Key simulation parameters of examined opposed jet cases, $\sqrt{u'u'}$ indicates the fluctuations at the inflow and C_s is the applied value for the Smagorinsky model

Case	Δ (mm)	$\sqrt{u'_i u'_i}$ (m/s)	C_s (-)	Cells ($\times 10^6$)	Simulated Time (ms)	Typical Cost (kCPUh)	No. of CPUs
SJ0.2 _{<i>svO</i>}	0.2	0.2	0.065	11.8	300	1.15	4
OJ0.2 _{<i>svO</i>}				23.6		2.59	8
SJ0.2 _{<i>T10</i>}		0.1					
SJ0.2 _{<i>T20</i>}	0.2	0.2	0.065	11.8	300	1.15	4
SJ0.2 _{<i>T30</i>}		0.3					
SJ0.1 _{<i>c00</i>}			0.000				
SJ0.1 _{<i>c30</i>}	0.1	0.2	0.030	92.2	244	13.82	36
SJ0.1 _{<i>c65</i>}			0.065				
OJ0.1 _{<i>full</i>}	0.1	0.2	0.030	184.4	230	31.10	72
OJ0.1 _{<i>react</i>}	0.1	0.2	0.030	184.4	300	44.62	108

5.2 Immersed Boundaries — Moving Test Cases

The following section describes two validation cases for the immersed boundary method with moving surfaces in isothermal flows. The first case focuses on a sphere that moves through the computational domain. Flow conditions identical to the static sphere case of Section 5.1.2 are applied. Furthermore, a piston-cylinder arrangement for incompressible flows is investigated. The configuration has been chosen to evaluate the performance of the proposed method in a complex domain with moving boundaries. The simulations of a piston-cylinder arrangement may then allow the investigation of the flow in areas that are difficult to measure experimentally.

5.2.1 Flow around a Sphere — Moving

The results of a flow around a static sphere at different Reynolds numbers were presented in Section 5.1.2. The same flow conditions were applied to a sphere in motion relative to the grid. The same three-dimensional flow characteristics form, allowing a comparison of results between the static and the moving sphere. Ideally the modelling results should be indistinguishable. Section 5.1.2 provides an overview of previous experimental and numerical studies on flows around a sphere.

5.2.1.1 Numerical Simulations

A detailed overview of the key parameters is given in Section 5.1.2. For the simulations of the moving sphere, a domain of size $50D_s \times 12D_s \times 12D_s$ was used and the sphere travelled periodically through the domain. The investigation focused on the *coarse* filter width ($\Delta = 5$ mm) to evaluate the functionality of the moving boundaries. The investigated Reynolds numbers were in the range of $50 \leq Re \leq 500$.

5.2.1.2 Results and Discussion

The statistical data obtained for the flow around a moving sphere is compared to the results presented in Section 5.1.2, as identical flow regimes have been

considered. The comparison includes the values for the drag coefficient, length of the separation bubble in the wake of the sphere and the normalised mean axial velocity profiles. Instantaneous velocity fields at three different time-steps are shown in Fig. 5.28. The images, from right to left, show the development of the flow field as the sphere starts to move through the domain. The turbulent wake is clearly visible in the third image.

Figure 5.29 shows the drag coefficients for the moving and static sphere in comparison to theoretical values by Oseen [131]. It is instantly visible that the drag coefficients of the moving sphere are lower in error, especially at higher Reynolds numbers. The error does not exceed 8%, whereas the largest error for the static sphere was 12%. The reduced error is a result of the momentum correction in the immersed boundary method and results should be considered with caution. The IB method is first order accurate and momentum corrections in freshly cleared cells can introduce an error in the flow field. This may positively or negatively affect the neighbouring cells and overall solution. A moving particle is unlikely to be in exactly the same position as a static particle when taking statistics. This effect became more visible when inspecting the length of the separation bubble and mean axial velocity fields.

The wake lengths for the different Reynolds number flows are shown in Fig. 5.30. The calculated wake lengths agree well with the numerical prediction of the flow past the static sphere and the predicted values by Fadlun et al. [42] and Fornberg [46] for $Re \leq 200$. A small deviation in the wake length, caused by the sphere's movement, can be seen in the result for $Re = 300$. The high particle velocity influences the freshly cleared cells, as these are filled by trilinear interpolation from neighbouring cells. This has an effect on the wake of the sphere. The same phenomenon is also noticeable at $Re = 500$, where the wake of the moving sphere is longer in comparison to the static results. Overall, the prediction agrees much better with the results by Fadlun et al. [42], but it should be considered with caution as they assume axis symmetry along the centreline to reduce the computational domain, which does not account for the three-dimensionality of the flow in the wake. The differences in results at $Re = 500$ can be reduced by using a higher order interpolation scheme, to reduce the effect of the discontinuities from the freshly cleared cells.

At low Reynolds numbers ($Re \leq 200$), the normalised mean axial velocity profiles along the centreline agree well with the predictions of the flow around the static sphere (Fig. 5.31). Minimal differences can be identified in the region behind the sphere, however these should mainly result from momentum correction. A noticeable difference can be found in the wake region of the sphere for $Re = 300$ and $Re = 500$, with a good agreement further downstream. This strengthens the argument that the momentum blending in the freshly cleared cells has an effect on the wake of the sphere, especially at higher Reynolds numbers. Similar to the static results, the mean axial velocity was under-predicted in the vicinity of the wake for $Re = 300$. Furthermore, wake velocities on the coarse grid are much lower in comparison to the data by Johnson [79]. However, the grid refinement for the static sphere showed a significant improvement. Overall, the results show a good agreement with the results of the flow around the static sphere and similar flow phenomena were identified. Especially at lower Reynolds numbers, the results for the first moment of the axial velocity, wake lengths and drag coefficients are almost identical, indicating that the immersed boundary method can be used for static and moving boundary problems alike.

5.2.1.3 Conclusion

Large-Eddy Simulations were carried out to investigate the flow around a moving sphere at different Reynolds numbers and the prediction of the wake-length and mean axial velocity show a good agreement with previous work on the flow around a static sphere. Small differences in the flow field can be identified in the immediate vicinity of the moving boundary, which results from the treatment of the freshly cleared cells. The immersed boundary method is consistent for static and moving immersed boundaries and it is applicable to different flow problems. The simplicity of the method allows for fast computations with a satisfying accuracy. Overall, the results are encouraging and support the use of the immersed boundary method for complex geometries with moving surfaces.

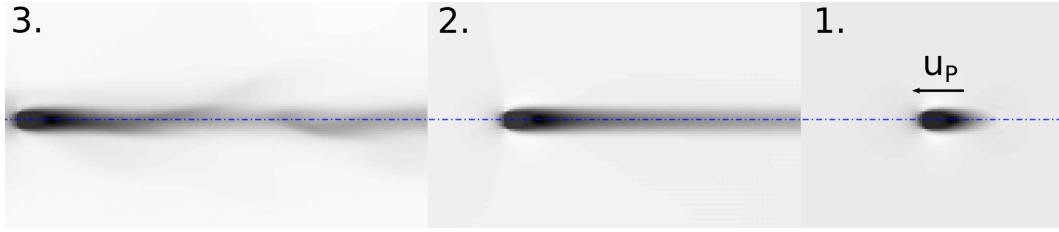


Figure 5.28: Axial velocity field for $Re = 500$ at three different time-steps. The maximum positive velocities is shown in white, while maximum negative velocities are black. The axis of symmetry is shown with the dash-dotted line.

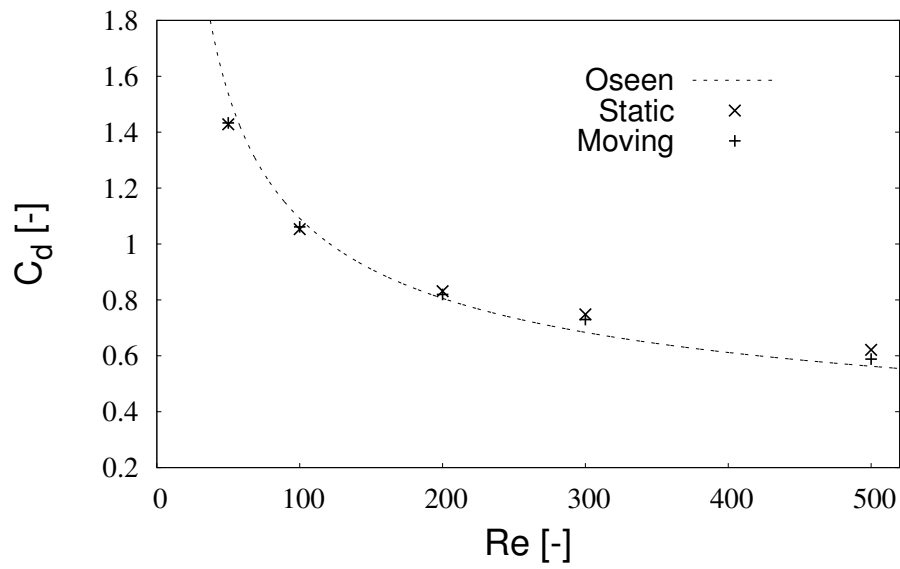


Figure 5.29: Comparison of the drag coefficient for the static and moving sphere at different Reynolds numbers. The line indicates the theoretical values.

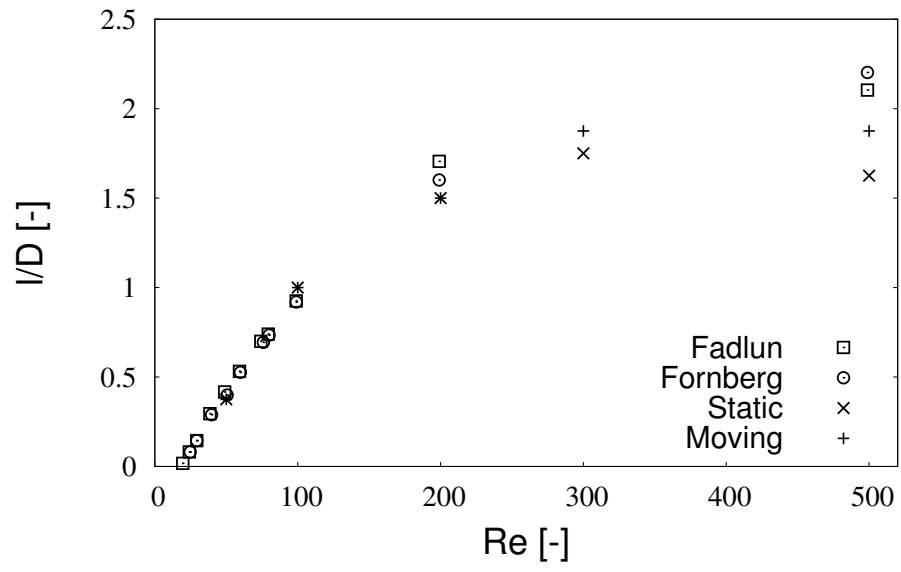


Figure 5.30: Comparison of the wake length for the static and moving spheres.

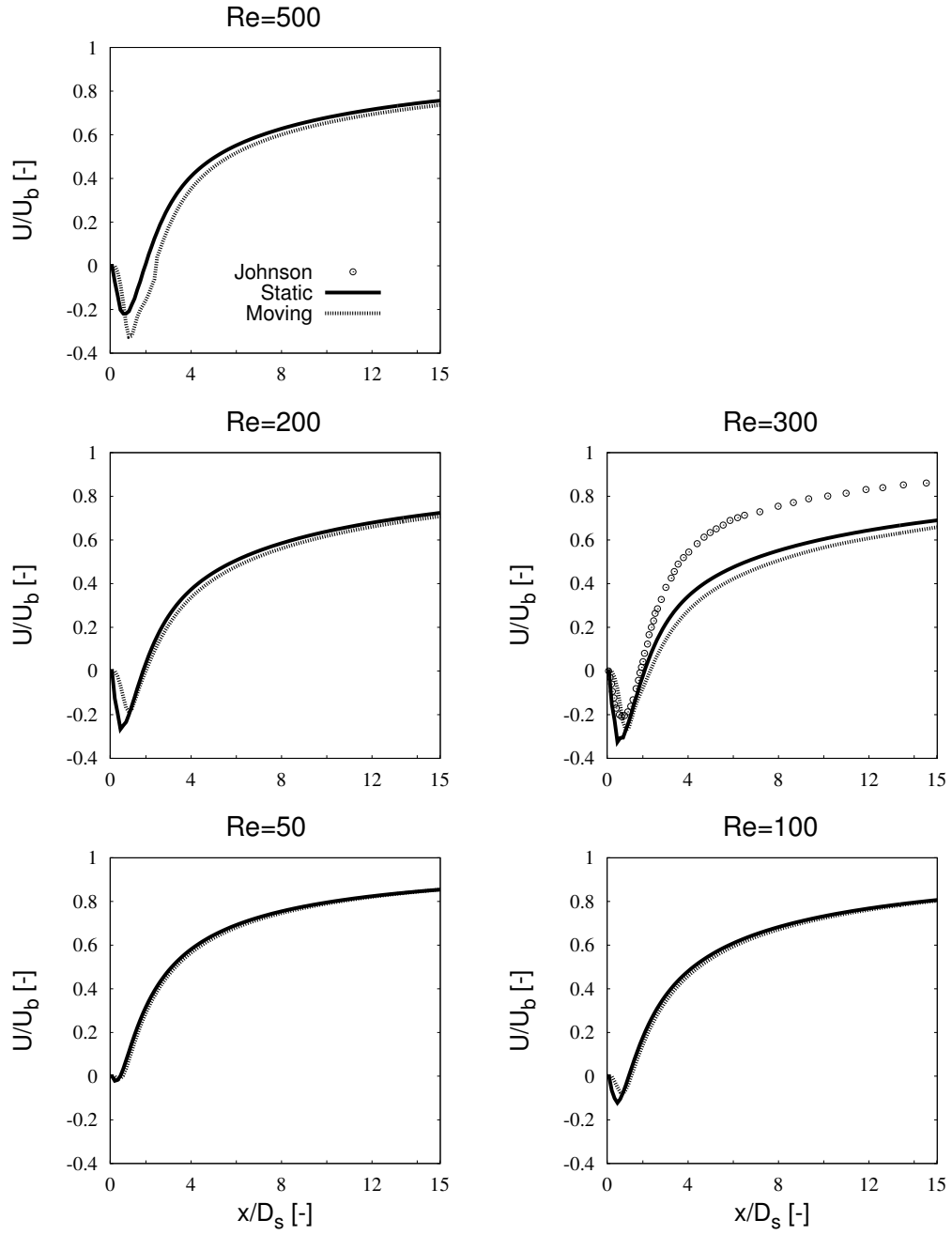


Figure 5.31: Statistical measurements for the first moment of the flow around a moving sphere at different Reynolds numbers. Results are compared to the flow around a static sphere at same Reynolds numbers.

5.2.2 Piston-Cylinder Arrangement

The simulation results for the computation of a motored piston-cylinder assembly using LES is discussed in this section. A brief summary of previous experimental and numerical investigations of the applied geometry is given and specific objectives of this work are outlined. The experimental piston-cylinder assembly is introduced, followed by details on the numerical modelling using LES with moving immersed boundaries in a code for fluid flows at constant density. At the end of the section simulation results are discussed.

5.2.2.1 Introduction and Specific Objectives

Internal Combustion (IC) engines are of great interest to human kind, the most obvious example being car engines. The development and improvement of IC engines has been an ongoing process for many decades. The main goals in the development of IC engines have been the reduction in fuel consumption while increasing the power output and improving the cleanliness of the emissions. Trade-offs between these characteristics often need to be made. In this work, the geometry of a piston-cylinder arrangement is modelled to show the capabilities of the immersed boundary method in complex geometries with medium Reynolds number flows. The successful application of the IB method in LES allows for future contributions in the development of IC engines by gaining a greater insight of the flow field and combustion process in areas that are difficult and expensive to measure.

Measurement techniques for complex geometries are either intrusive (directly affecting the flow), or modifications to the engine need to be made, which can alter the flow and combustion process. Due to the complexity of the geometry, investigations often focus on specific parts of the combustion cycle (e.g. injection, compression, expansion, flow field analysis). Hence, a vast amount of literature is available and only a review is given here. A more detailed review on experimental and numerical investigations of internal combustion engines can be found in the work by Arcoumanis [4]. Discussions on engine specific issues can be found in the review by El Tahry and Haworth [175, 176].

Morse et al. [125] used Laser-Doppler Anemometry (LDA) to measure phase

averaged first and second moment radial profiles of axial velocities. Measurements were taken at 36° and 144° crank angle after cylinder top-dead-centre (TDC) in 10 mm increments. In the experiments the piston was driven by an external motor. The inlet valve was fixed in one position and a small annular gap was left between the valve and the cylinder head, ignoring the compression phase. Xu et al. [196] applied LDA for the development of a new type of combustion chamber in an automobile engine and Kang and Baek [81] applied LDV to analyse tumble formation and decay in a four-valve engine. Particle Image Velocimetry (PIV) was used by Reeves et al. [149] in the analysis of in-cylinder flows and they present a number of PIV vector maps for various crank angles.

Initial numerical investigations of internal combustion engines were limited to Reynolds-Averaged Navier-Stokes simulations because Large-Eddy and Direct Numerical simulations were not affordable at the time. Han and Reitz [62] used RANS with the $k - \epsilon$ model to simulate variable-density engine flows for an compressible engine layout. Computations focused on the compression/expansion of the flow within the cylinder. The authors compare the main flow characteristics from their simulations with experimental data. The flow processes with the combustion chamber, using RANS modelling were also investigated by Gosman [61]. Khalighi et al. [88] modelled the combustion process in a four-valve engine with the RANS approach and phase averages of many cycles are presented. The absence of turbulent fluctuations affected the results, however the mean profiles are predicted well.

In the context of LES, Haworth et al. [67, 68] used unstructured deforming meshes to explore the suitability of LES to simulate IC-engine geometries. At the time of the study, the available computing power was a limiting factor. However it was found that LES provided more complete information in comparison with RANS. Thobois et al. [178] used LES for the prediction of aerodynamics in IC engines. The work focused on the diesel engine intake ports under steady flow conditions. Results showed a satisfactory agreement with experimental data. A piston-cylinder arrangement was employed by Verzicco et al. [42, 187] in the validation and evaluation of an immersed boundary method for a Finite Difference Method. A satisfactory agreement between numerical results and experimental data was achieved.

In the present work, the geometry of the piston-cylinder arrangement is based on the experimental work by Morse et al. [125]. The geometry forms the basis for an excellent test case to evaluate the presented immersed boundary method. The geometry is complex and includes moving surfaces for flows at medium Reynolds numbers. Statistical measurements of first and second moments are compared to the experimental measurements, although small alterations to the geometry were made. These will be discussed in more detail with the numerical simulation parameters. A successful prediction of the velocity profiles at different crank angles would encourage the further development and application of the proposed method.

5.2.2.2 Experimental Configuration

The configuration chosen to validate the present numerical procedure is a simplified piston-cylinder assembly with a fixed valve, based on the model used by Morse et al. [125]; a schematic drawing is shown in Fig. 5.32. The cylinder head and piston are flat (*pancake chamber*) with a 3 : 1 compression ratio. The inlet is 33 mm in diameter and the cylinder has a bore of 75 mm. All relevant dimensions and operating conditions are summarized in Table 5.2. In the experiments the piston was externally motored allowing fluid to enter or exit the combustion chamber and vice versa as the piston moved. The valve was fixed, so that there was no compression phase. The piston is driven by a simple harmonic motion at speed of 200 rpm (≈ 21 rad/s). Morse et al. [125] used LDA to measure phase-averaged first and second moments of axial velocities. Profiles are available in 10 mm increments for a crank angle of 36° and 144° after TDC.

5.2.2.3 Numerical Simulations

The numerical simulations of the piston-cylinder assembly have been carried out with the *PsiPhi* code and the proposed immersed boundary method (see Chapter 4). The overall domain length is $264 \times 80 \times 80$ mm; the domain was extended below the Bottom Dead Centre (BDC) to improve numerical stability. Two different filter (cell) widths ($\Delta = [1.0, 0.5]$ mm) were used for the modelling of the piston-cylinder arrangement, leading to meshes with 1.7×10^6 cells and 13.5×10^6 cells, respectively. The coarse filter width was used for

the initial testing and evaluation of the boundary method, whereas the fine grid was applied for a more detailed representation of the flow field. As in the experimental configuration, the compression phase was neglected, leaving the valve in a fixed position. The current study focused on a mean piston velocity of $\overline{u_p} = 0.4$ m/s, averaged over half a cycle, resulting in a Reynolds number of 2000. Transient Dirichlet boundary conditions were applied at the inflow plane and von Neumann conditions at the outlet (where negative velocity values were clipped). The Smagorinsky model ($C_s = 0.2$) was used to estimate the turbulent viscosity in the flow. The walls were described by the immersed boundary method, following the suggested value of Verzicco et al. [187].

The numerical domain contains a small alteration to the original experimental set-up, besides extending the simulated domain, a small gap (1 cell) was left between the piston and the cylinder wall to improve the numerical stability of the simulation; similarly to the set-up by Verzicco et al. [187]. The implications of this alteration will be discussed in detail with the simulation results. The investigation and detailed analysis of the flow field within the cylinder are of secondary importance to the proof of the concept. The successful simulation of the arrangement allows for contributions to the development of internal combustion engines in the future. Despite the change in geometry, similar trends should be established. Two piston cycles were simulated before numerical phase-averages of first and second moments were taken at specified crank angles over a period of six piston cycles.

5.2.2.4 Results and Discussion

The flow inside the piston engine is highly turbulent, unsteady and three dimensional. Simulations must be able to cope with all of these aspects [187]. At time $t = 0$, the piston is at top dead centre and starts descending towards BDC. Then fluid enters the combustion chamber from the outside through the gap between the valve and the cylinder wall. Due to the small opening, high speed jets are created which impinge on the cylinder wall before fluid separates due to viscous effects (Fig 5.33). The jets reach their highest velocity ($u_{jet} \approx 13u_p$) at $t = \pi/2$ and the flow is dominated by the vortices in the centre of the chamber which increase in size as the piston continues moving down. At $t = \pi$, the flow is dominated by complex structures emerging from the

interaction of the recirculating vortices that formed in the combustion chamber. Similar observations were made by Verzicco et al. [187]. When the piston starts moving back up from BDC, the fluid is pushed out of the cylinder. A new cycle starts at $t = 2\pi$ after the piston reaches TDC. The flow below the piston is neglected, as it results from the imposed boundary conditions and the piston movement and does not interact with the primary flow.

Simulation results are compared to the experimental data by Morse et al. [125] and numerical data by Verzicco et al. [187]. Verzicco et al. apply the assumption of axis-symmetry along the centreline in their simulations after studying a full three-dimensional domain at lower Reynolds number to observe that the flow remains symmetric for the first two cycles. However, investigations of the flow field showed that the axis-symmetry is not necessarily preserved at the given flow conditions. It was also observed that the position of the jet break-up is strongly dependent on the the vortex formations, which themselves depend on the piston movement. Therefore a fully three-dimensional domain was used for the current investigations. Animations of different sections allow for a detailed investigation of the flow field inside the chamber.

Figure 5.34 shows a comparison between the experimental and numerical statistical measurements of the first and second moments of the axial velocity at a crank angle 36° after TDC. The velocities were normalised by the mean piston speed and the radial distance was normalised by the cylinder radius R . The profiles have been taken at 10 mm increments downstream of the inlet valve. The results are phase averaged over six piston cycles. The mean velocity profile shows a good agreement with the experimental and numerical data. At this crank angle the small gap around the piston edge does not have any visible effect on the velocity field. The fluctuation profiles would certainly benefit from additional averaging and a finer grid resolution. The fluctuations are considerably lower in the region near the cylinder wall. A wall model or increased resolution in that region is required to improve predictions, especially as wall temperatures have a significant effect on the combustion process of IC engines. The variation in the numerical data sets results from the different grid structures used. In this work, a structured cubic grid was employed, whereas Verzicco et al. [187] used an unstructured grid with high resolution in the near cylinder head region. Their exact grid resolution is unknown, although simi-

lar problems in the predictions of fluctuations can be found in the results of Verzicco et al. [187].

The same positions of statistical measurements for an crank angle of 144° are shown in Fig. 5.35. The first position at 10 mm shows a large variation of the mean velocity profile. The jet is shifted towards the centre and a higher mean velocity was recorded. This is a result of the delayed jet break up in the flow and modifications in the Smagorinsky constant might lead to small improvements. The fluctuations show a great degree of variance in comparison the the experimental data. One of the reasons for the large variations in fluctuations is the use of the relatively coarse filter width ($\Delta = 1$ mm) and the dissipative nature of the sub-grid scale model. Great efforts could be made to tune the value of C_s to improve the results, but the overall improvement might be minimal. Furthermore the high fluctuations near the wall at $x = 30$ mm are an artefact of the gap between the piston and cylinder wall. The gap allows for some mass to by-pass the cylinder, which is defrimental in an internal combustion engine. To ensure that all mass stays in the cylinder, the gap should be closed and ideally a compressible flow solver would be used to model the compression phase as well. Nonetheless the key characteristics of the flow have been captured on the coarse grid and encourage further development and application of the method.

Figure 5.36 shows the phase-averaged first and second moments of the velocity field for a crank angle of 36° after TDC. A refined mesh with cell width of $\Delta = 0.5$ mm was used for the simulations. The predictions of the first and second moments 10 mm downstream of the valve agree well with the experimental measurements [125] and the numerical predictions [187]. However an under-prediction in fluctuations is found in the near wall region, similar to the coarse grid results. The statistical measurements further downstream ($x = 20$ mm) indicate an early jet break up and fluctuation levels are very low in comparison to the experimental data. At $x = 30$ mm, the data for first and second moments agrees well with the results by Morse et al. [125]. This is to be expected as the piston is close to the measurement position and the flow is almost undisturbed at this point in time. It is also interesting to note that the modification of the geometry does not have a significant effect on the results.

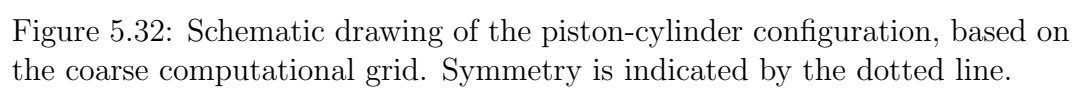
Statistical measurements of the velocity field were also taken on the finer grid for a crank angle of 144° after top dead centre and results are shown in Fig. 5.37. The plots show a much thinner and longer jet at $x = 10$ mm in comparison to the experimental measurements. This indicates a delayed jet break up which is influenced by the sub-grid scale model. A similar phenomenon was identified in the flow field of the opposed jet flow (see Section 5.1.3). The mean jet is also positioned further towards the centre of the cylinder, which is a result of the vortex forming in the centre. The strong influence of the vortex can also be identified at the other two locations further downstream, where negative velocities are found in the centre of the cylinder. As a result the inlet jets are drawn towards the centre and away from the walls. The statistical measurements for the fluctuations are generally in good agreement with the experimental predictions in the centre of the cylinder. However, large variations are found in the wall regions, especially for the latter two sampling points. It is known that the Smagorinsky model usually does not perform well in these regions, resulting in wrongly modelled sub-grid stresses near the wall. In the simulations the jets also do not impinge on the walls as strongly as it was observed in the experiments and the fluctuations increase in these areas as the jets are drawn further towards the centre of the cylinder.

In general, the results of the piston-cylinder arrangement are encouraging for both grid resolutions. Nonetheless, the simulations with the different grid sizes share characteristics that influenced the results. Statistical results would benefit from additional sampling, as six engine cycles are not enough. Furthermore both grid resolutions can be regarded as too coarse to accurately predict the flow in the centre of the cylinder at higher crank angles. This is particularly true considering that the development of the flow field is strongly dependent on the inlet jets and the resulting vortex formations. The effect was noticed more clearly in the simulation with the finer grid. A LES filter width of $\Delta = 0.1$ mm ($\approx 1.7 \times 10^9$ cells) is suggested for the detailed analysis of the flow inside the chamber to accurately resolve the flow characteristics. In addition, the statistical measurements for the second moments can be improved by using a more advanced sub-grid stress model, such as the Germano model [52], and a wall model to accurately predict the flow in the near wall regions. It should also be mentioned that the predictions of the second moments have been plotted with a thinner line to illustrate that they leave room for improvement.

5.2.2.5 Conclusion

A piston-cylinder arrangement, based on the experimental work by Morse et al. [125], has been investigated. Large-Eddy Simulation was used for the numerical work to evaluate the applicability of the immersed boundary method to complex geometries with moving surfaces. Simulation results were compared to experimental in-cylinder measurements of isothermal flows, neglecting the compression phase.

First moment statistical measurements agree well at both crank angles for all measured positions. The statistical second moments suffer from the low sampling rate and a relatively coarse grid. The flow is very inhomogenous in space and a single value for the turbulence model might not be sufficient to describe the flow dynamics, which agrees with the findings by Verzicco et al. [187]. An investigation of the flow field showed that the jet break up and resulting vortex formation is strongly dependent on the piston movement and the jet modelling in the near wall regions. The coarseness of the grid also leads to high fluctuations in the flow of the jets. These areas would benefit from a grid refinement to better resolve the flow field. Furthermore the alteration in the geometry has an effect on the results, especially in the near wall region of the cylinder. Hence, a detailed investigation of the flow field for reacting flows certainly requires the closure of the piston gap. The investigation of a full cycle of an internal combustion engine should also be carried out with an compressible flow solver to include the compression phase of the engine cycle. Nonetheless the results encourage the use of the immersed boundary method on further investigations of the piston-cylinder arrangement in order to analyse the flow in greater detail.



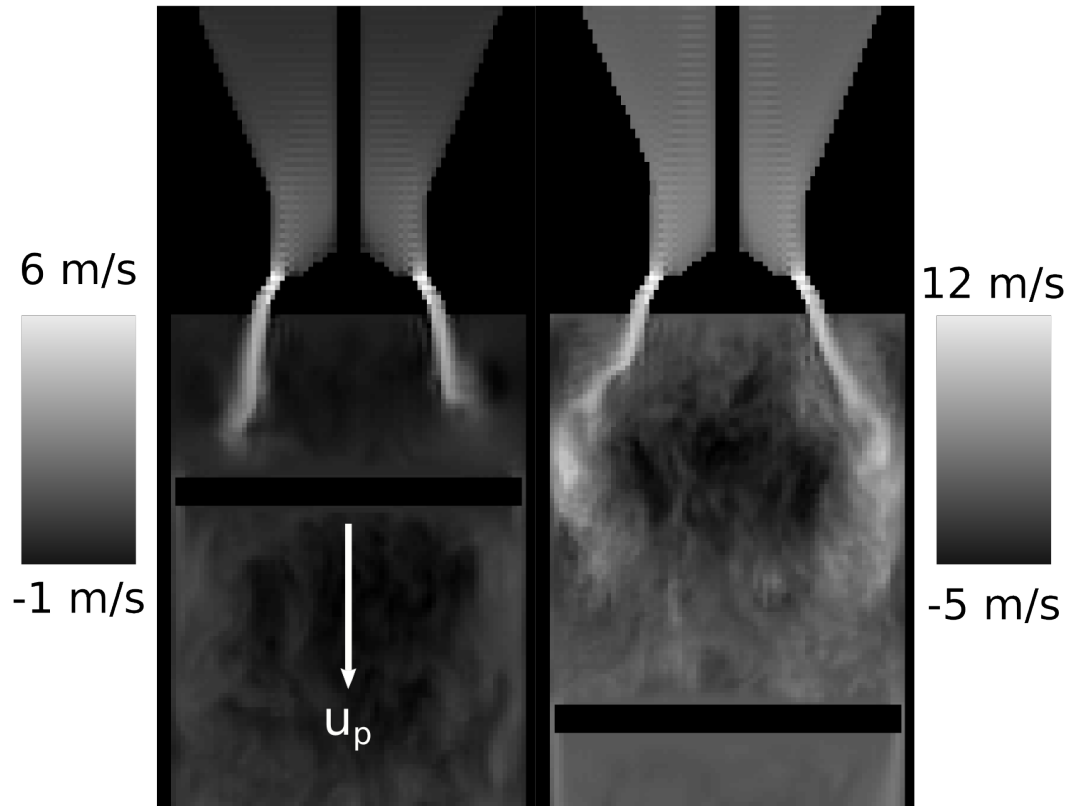


Figure 5.33: Axial velocity field for two crank angles, 36° (left) and 144° (right), measured from TDC on the fine grid. The maximum positive velocities are shown in white, while maximum negative velocities are black.

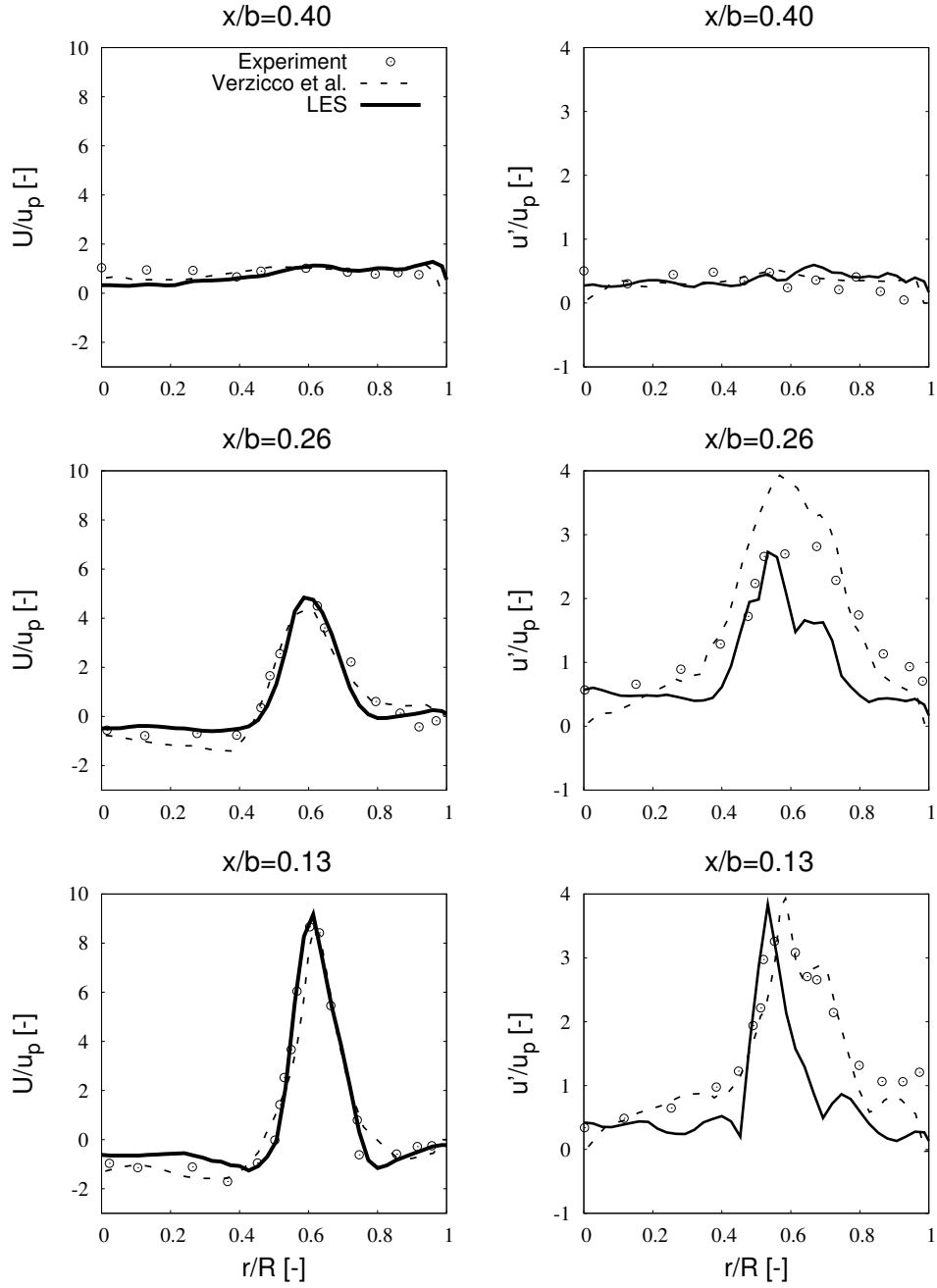


Figure 5.34: Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 1$ mm, and a crank angle of 36° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.

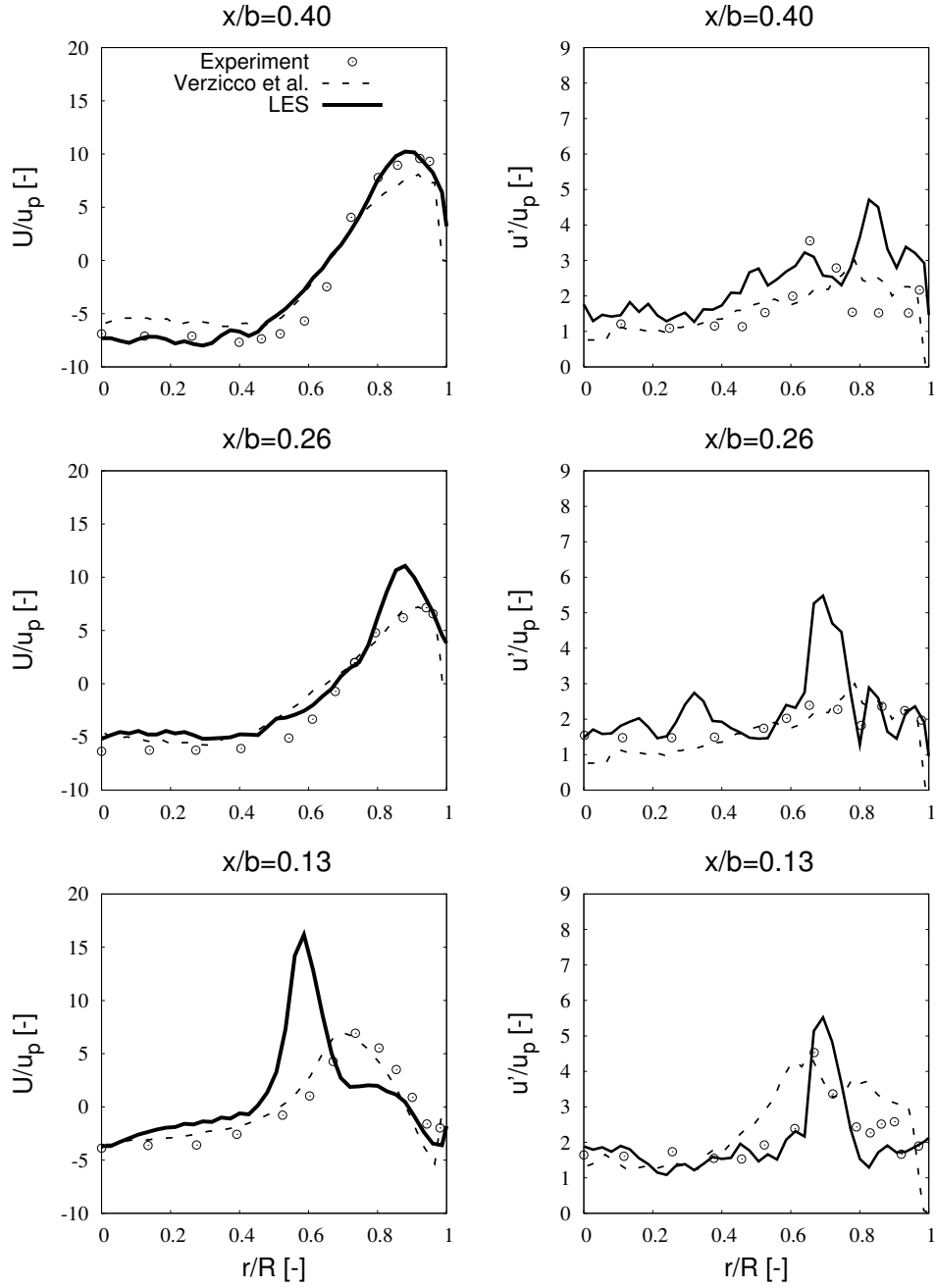


Figure 5.35: Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 1$ mm, and a crank angle of 144° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.

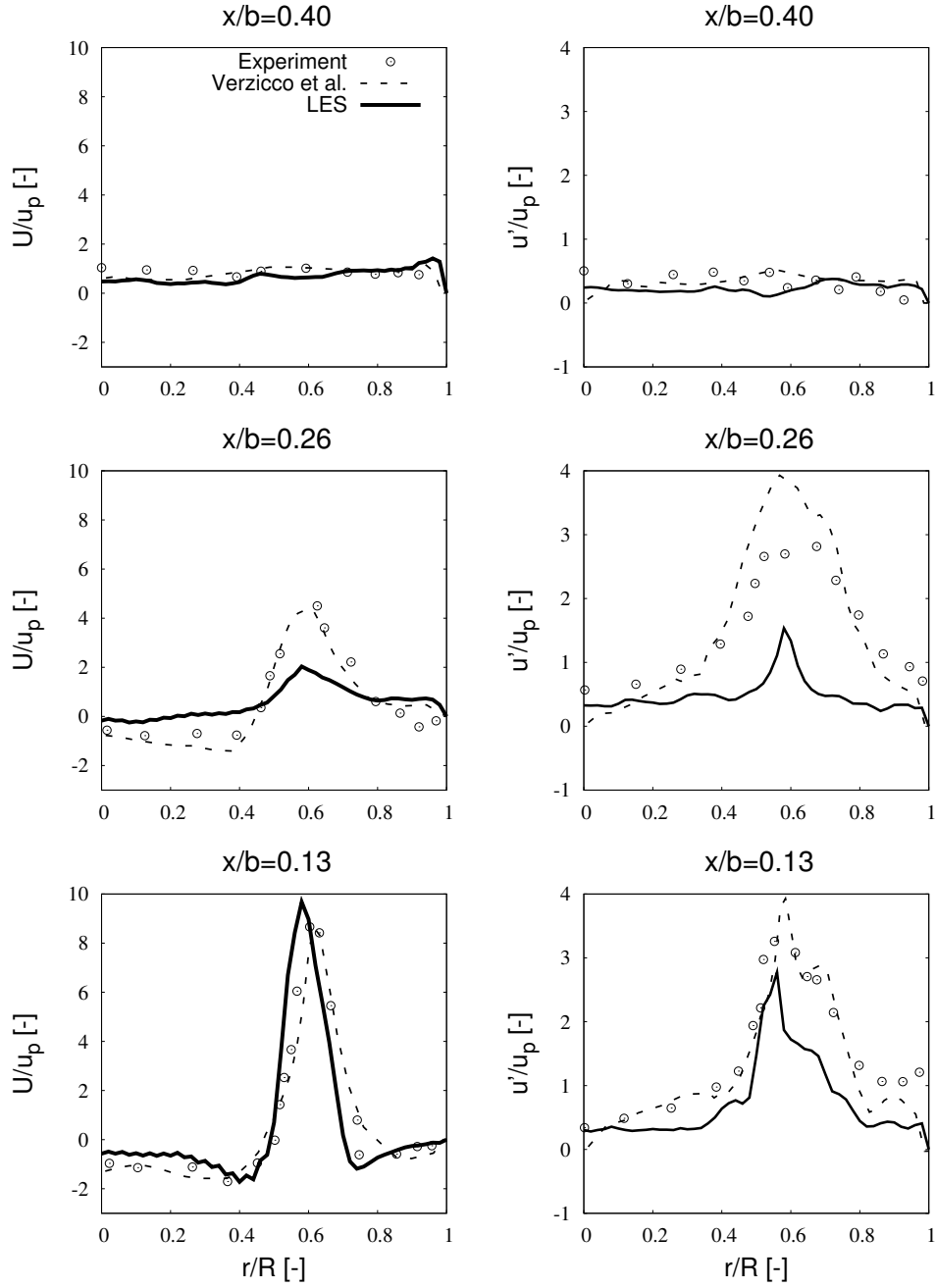


Figure 5.36: Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 0.5$ mm, and a crank angle of 36° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.

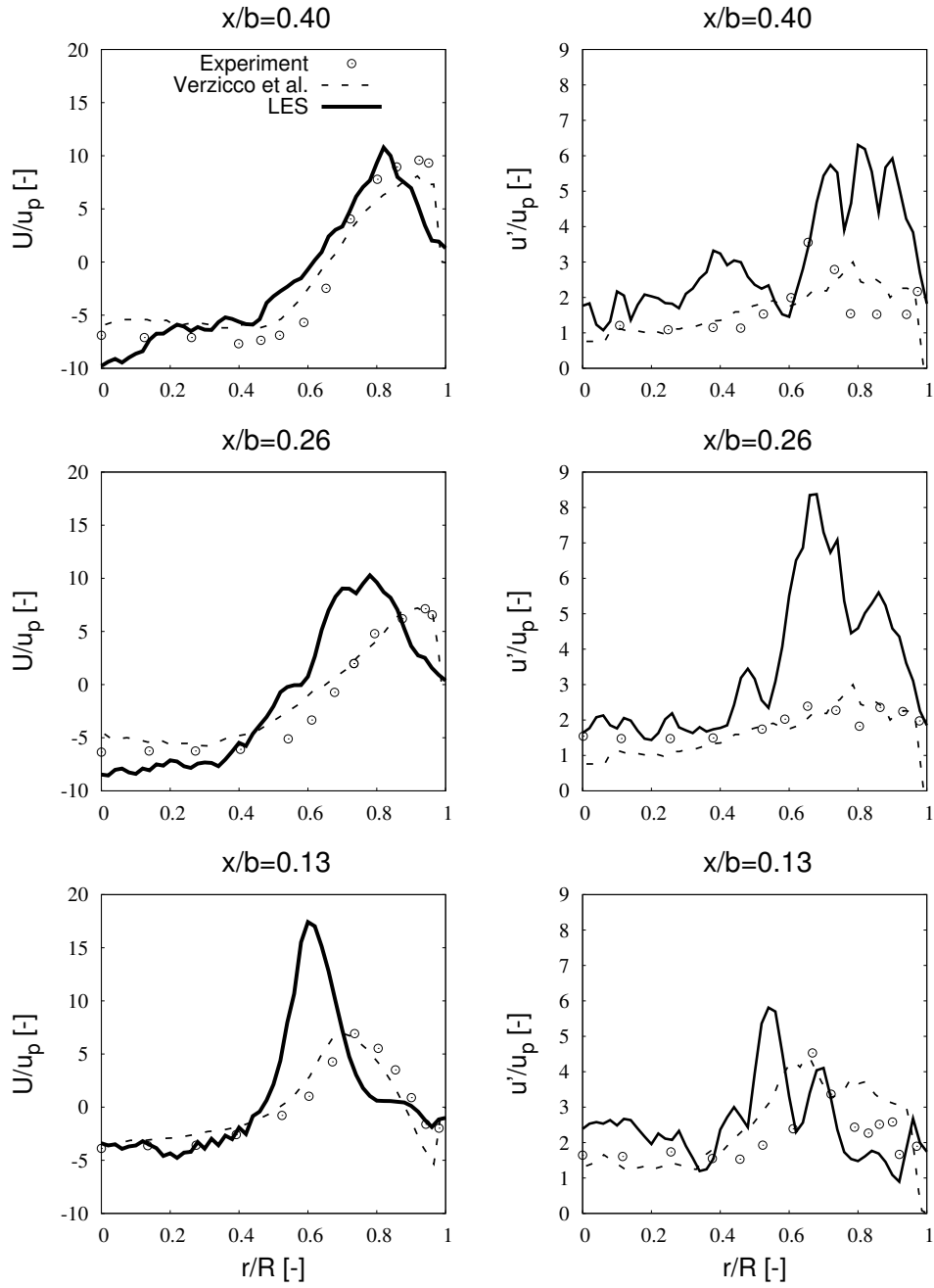


Figure 5.37: Statistical measurements of first (left) and second (right) moments for a mean piston velocity $u_p = 0.4$ m/s, $\Delta = 0.5$ mm, and a crank angle of 144° . Three positions are compared to in-nozzle measurements at defined locations $x/b = [0.13, 0.26, 0.40]$ downstream of the inlet valve.

Table 5.2: Piston-cylinder dimension and parameters

Ω rpm	Re (-)	b (mm)	c (mm)	d (mm)	g (mm)	h (mm)	r_a (mm)	r_e (mm)	r_s (mm)	r_v (mm)	r_w (mm)	s (mm)	s_0 (mm)
200	2000	75	30	16.5	40	6.5	20.8	33	3	16.8	13	60	11

Chapter 6

Interface Flows and Immiscible Fluids

Immersed boundary methods are one of the applications for Lagrangian particle methods. Another area to apply particle methods are interface flows and thin mixing layers. A wide range of turbulent length scales need to be either resolved or described by closure models to accurately describe the scalar fields. This can be achieved by local grid refinement, however, these algorithms are often complicated and difficult to implement and parallelise. Furthermore the short time-step width imposed on the refined region is enforced on the entire computational domain. Therefore alternative methods are sought to describe the physics of the flow.

The proposed alternative Euler-Lagrange method has the potential to improve accuracy and resolution of scalar transport by combining advantages of the Eulerian and Lagrangian approach in comparison to a purely Eulerian method, while maintaining a moderate computational cost. To the authors knowledge, this particular technique has not been exploited before in the field of combustion and interface flows. The project was started in collaboration with a colleague (M. W. Pettit) at Imperial College London, who is continuing to develop the method. Therefore only initial results, based on personal contributions, for two dimensional test cases are shown.

Similar approaches to the proposed Euler-Lagrange technique have been developed; for example the hybrid particle-level set method by Enright et al. [40] or

the hybrid Monte Carlo PDF method for turbulent combustion simulations by Pope [145]. In a broader sense, the proposed method can also be compared to the particle-in-cell (PIC) method (see Chapter 4), developed by Harlow [64], to explore shock interactions with material interfaces and the deformations that could occur during hypervelocity impacts. The method was originally applied in two dimensions, and it became rather popular for plasma simulations [12].

6.1 Euler-Lagrange for Interface Flows

The proposed method is aiming to combine advantages of the Eulerian approach with advantages of the Lagrangian particle method. A scalar field is split up into a smooth, low-frequency field and a high-frequency field. The smooth, low-frequency field is described and transported as an Eulerian field and the high-frequency field is described and transported using Lagrangian particles. Therefore the Lagrangian contribution can be regarded as a correction to the resolved Eulerian field and particles are solely introduced in areas of steep gradients. The reason being that small scales are often constrained to a corrugated surface in space, for example a mixing layer or an interface. In terms of computational combustion problems, these areas are found in both premixed (thin flame-front) and non-premixed flows (steep gradients around the nozzle).

6.1.1 Theoretical Background

As mentioned above, the scalar field is split into a smooth, filtered Eulerian field and a high-frequency Lagrangian field, and can be expressed as:

$$\phi = \bar{\phi} + \phi' \quad (6.1)$$

where $\bar{\phi}$ is the Eulerian contribution and ϕ' represents the Lagrangian contribution, not to be mistaken with the Reynolds decomposition in Chapter 2. The Eulerian scalar field is obtained from low-pass filtering and the difference between the original field ϕ and the filtered field $\bar{\phi}$ is the contribution ϕ' of the small scales. One could say that the Lagrangian field results from 'high-pass' filtering. Figure 6.1 shows a graphical representation of the concept. The Eulerian contribution and the Lagrangian contribution add up to form the original

field (in this case a step function). Smooth regions are well presented by the Eulerian approach and Lagrangian particles are only necessary in areas with steep gradients.

Substituting Eq. (6.1) into the general transport equation (2.7), it can therefore be written as

$$\frac{\partial \bar{\phi}}{\partial t} + \frac{\partial \phi'}{\partial t} + \frac{\partial \bar{\phi} u_j}{\partial x_j} + \frac{\partial \phi' u_j}{\partial x_j} = \frac{\partial D_j(\bar{\phi})}{\partial x_j} + \frac{\partial D_j(\phi')}{\partial x_j} \quad (6.2)$$

The source term has been neglected in Eq. (6.2) and the equation is satisfied when the following two equations are also satisfied.

$$\frac{\partial \bar{\phi}}{\partial t} + \frac{\partial \bar{\phi} u_j}{\partial x_j} = \frac{\partial D_j(\bar{\phi})}{\partial x_j} + \kappa \quad (6.3)$$

$$\frac{\partial \phi'}{\partial t} + \frac{\partial \phi' u_j}{\partial x_j} = \frac{\partial D_j(\phi')}{\partial x_j} - \kappa \quad (6.4)$$

where κ is the transfer term to convert particles to the Eulerian phase and vice versa. Hence a transport equation can be solved for the large-scale contribution $\bar{\phi}$ independently of the equation for the small-scale contribution ϕ' . In this work a classical Eulerian solver is used for the large scale solution, and a Lagrangian (particle) solver for the small scales. It should also be stressed that the above contribution assumes that the flow field u_j is *fully resolved*.

Lagrangian Diffusion

Even though diffusion was not considered in the presented cases, it affects the scalar transport and therefore the Lagrangian diffusion needs to be modelled. Particle diffusion can be simulated with a random walk based on the Wiener process (Fig. 6.2), assuming that all particle positions are known. The particle diffusion is restricted by the domain dimensions and a normalised random number. The method requires a continuous function with independent increments and finite small time-step δt . Mathematically the process can be

described as:

$$s_p^{t+1} = s_p^t + W_t \quad (6.5)$$

where s is the position of particle p at time t and W_t is a Wiener process. W_t can be defined as a mathematical model of the Brownian motion of a free particle with friction being neglected,

$$W_t = (\delta t)^{1/2} \sum_{i=1}^n \xi(i) \quad (6.6)$$

where ξ is an independent standardised normal random variable. For an infinite small time-step and in its simplest form, the Wiener variable W_t can therefore be represented by a normal (*Gaussian*) distribution with zero mean and variance proportional to the time difference [145].

To ensure that the fields remain coupled, both fields (Eulerian and Lagrangian) have to diffuse at the same rate. Figure 6.3 shows a top-hat function that has been numerically diffused by applying the Eulerian diffusion, Lagrangian diffusion and an analytical diffusion. Identical diffusion coefficients have been used and both fields diffuse at the same rate. Therefore, it can be assumed that the large high and low frequency field are correctly coupled. A more detailed description of the coupling, based on the work by Cant and Mastorakos [20], is given in Appendix A.

6.2 Interface Flows — Test Cases

The proposed approach was evaluated based on two test cases (a block function and a Zalesak disk [199]) and results are compared to the pure Eulerian and pure Lagrangian approaches. The block function with an vortical velocity field produces 'thin structure' around the central block, which demonstrate any improvement in accuracy of the joint approach. The Zalesak disk test case gives a good indication of conserving flow details. A low storage Runge-Kutta scheme (see section 3.2.3) was applied for the time integration in all tests. Long simulations were performed to reduce error in time measurements and all test cases have been evaluated on their accuracy and computational efficiency. The simulation parameters for the two cases are summarized in Table 6.1.

6.2.1 Block Function

The block function creates nine evenly spaced squares in the domain and an irrotational vortical velocity field was applied. A 4th-order central differencing scheme (CDS4) and a Total Variation Diminishing (TVD) scheme was used for the convection of the Eulerian contribution. The Lagrangian approach was initialised with 1.6×10^6 particles and the joint approach was used with a 4th-order CDS for the Eulerian field and 3×10^5 particles for the Lagrangian field. The particle number was determined from the sum difference of the original field and the Eulerian field, and a prescribed particle density. The simulation times are summarized in Table 6.2.

The central differencing scheme, of 4th order, is computationally efficient, however oscillations are introduced which produce unbounded results (Fig. 6.4b). These oscillations can cause instabilities, especially in flows with low diffusivity. The 'thin structures' at the centred block are not well preserved, and oscillations would increase further with longer simulation times. Furthermore, the oscillations are unrealistic for the transport of scalar quantities mixture fraction and progress variable where the range limit is between 0 and 1. Outside this range, the mixture fraction and progress variable have no physical meaning. The TVD scheme is computationally more expensive, although oscillations are avoided (Fig. 6.4c). The 'thin structures' around the centre block are also not well-preserved due to numerical diffusion.

The Lagrangian approach is the most accurate method (Fig. 6.4d), but also the most computationally expensive. The computational time was approximately five times longer than for the TVD scheme, as $\sim 1.6 \times 10^6$ new particles positions had to be calculated in each time step. Accuracy can be improved further at the cost of computational efficiency. In general, the 'thin structures' around the centre block are well preserved.

The joint Euler-Lagrange approach preserved the 'thin structures' at the centre (Fig. 6.4e) and oscillations are suppressed at medium computational cost. The results are promising, especially considering the computational costs and simulations times.

6.2.2 Zalesak Disk

The Zalesak disk is commonly used to test proposed flux-corrected transport algorithms and critical flux limiting stages in multi-dimensions [199]. The case has since been adopted as a common test for interface tracking and level set methods. The case is based on a 'solid' cylinder with a cut out notch, which adds some complexity to the test field. The cylinder is placed in a constant-vorticity field with anti-clockwise rotation, and the rotational velocity is set such that one revolution around its center point takes 628 time steps. The case is ideal to test the accuracy of existing numerical schemes and a comparison to the proposed method can be made. The simulation parameters are summarized in Table 6.1.

In this work, two versions of the case were simulated. First, the disk completed one revolution, before returning to its original position. The case was evaluated with the Euler CDS4 and CHARM TVD scheme, Lagrange and the proposed joint Euler-Lagrange approach. Table 6.3 shows the simulation times and the accuracy of each method. The root-mean-square (RMS) was evaluated based on the difference between the original and final field, as shown in Eq. (6.7). The RMS values give a quantitative indication on how well the shape of the disk is preserved. Similarly the sum of error (*SoE*) is the sum of the absolute values of the original field minus the final field (Eq. 6.8), where n defines the total number of cells in the domain. The SoE was used to determine the accuracy

of each approach. Ideally, the sum of error and the root mean square values are zero.

$$\phi_{RMS} = \sqrt{\frac{1}{n}(\phi_1^2 + \phi_2^2 + \dots + \phi_n^2)_{orig.}} - \sqrt{\frac{1}{n}(\phi_1^2 + \phi_2^2 + \dots + \phi_n^2)_{final}} \quad (6.7)$$

$$\phi_{SoE} = \sum_{i=1}^n |\phi_i|_{orig.} - \sum_{i=1}^n |\phi_i|_{final} \quad (6.8)$$

The original and transported fields of the Zalesak disk are shown in Fig. 6.5. Immediately apparent is the numerical diffusion resulting from transport by CDS4 and TVD. The diametric extent of the disk has been increased for the CDS4 ($\approx 2\%$) and the TVD ($\approx 3\%$) approach, as both Eulerian methods suffer from diffusion. The notch is poorly represented, especially in case of the TVD scheme. This might indicate that the CDS4 approach is more accurate than the TVD, however the central differencing scheme is known to introduce numerical oscillations, which are not visible in Fig. 6.5b. For the given conditions, the CDS4 approach is more accurate than the TVD scheme because any introduced oscillations are undone by returning the disk to its original position.

The Lagrangian approach is lowest in error ($\approx 1\%$) and the particles recover the sharp edges well (Fig. 6.5d). A close inspection of the field shows that it is almost identical to the original. However the seeding of the particles is important and the field value should not exceed unity, which would lead to unphysical results in some scalar fields. Furthermore the particle approach suffers from computational costs as a result of the impossibility of efficiently vectorising routines.

The graphical result of the field for the joint method is shown in Fig. 6.5e. A close inspection of the field reveals that even though the notch and sharp edges are well preserved (70% fewer particles in comparison to the Lagrangian method), the approach still suffers from the numerical oscillations and diffusion of the Eulerian method, which is simply not smooth enough. To avoid the unwanted expansion of the field, more filter operations can be applied to increase the contribution of the high-frequency field. However, increasing the number of particles increases the computational cost. Therefore a fine balance

between accuracy and computational efficiency should be used to benefit from the Euler-Lagrange approach. Overall the joint method shows significantly improved results in comparison to the Eulerian approach (1.3% error) at moderate additional computational cost.

The above results can be misleading because a flow usually does not return to its original condition and the test case should be seen as an exception of the good performance of the central differencing scheme with unphysical flow conditions. To get a more realistic behaviour of the different approaches, the tests were repeated with a single disk rotation.

The results of the single disk rotation are summarized in Table 6.4, and it is instantly clear that the oscillations (swirls behind the disk in Fig. 6.6b) introduce a substantial numerical error. Furthermore the Eulerian approach suffer from numerical diffusion (glow-like look of the transported disk) and as a result the sharp edges are not preserved and the area of the disk increases. The edges of the notch soften and it is clear that the Eulerian transport is insufficient for steep changes in scalar gradients. Overall the CDS4 transport scheme has a larger error ($\approx 20\%$) than the chosen TVD scheme.

Yet again, the smallest error was found with the Lagrangian approach ($\approx 0.6\%$). The sharp edges are well preserved, especially in the area near the notch. The transported field is almost identical to the initial field and an increase in particle density would improve the results further. However the Lagrangian approach is the computationally most expensive method and scalar fields can be unphysical.

The proposed joint Euler-Lagrange method shows good results at moderate computational costs. All sharp edges are well represented. The numerical oscillations of the CDS4 scheme are none existing due to the filtering of the original disk and therefore reducing the sharp edges in the Eulerian field. However, the Eulerian discretization scheme might have to be chosen appropriately for different test cases. In addition, the accuracy can be improved by increasing the number of filter operations at the beginning, subsequently increasing the Lagrangian contribution.

6.3 Discussion and Conclusion

The proposed Euler-Lagrange method has been compared to the pure Eulerian and pure Lagrangian approach. The accuracy and computational efficiency of the three methods were evaluated based on two-dimensional test cases, a block function and the Zalesak disk. The discretisation with CDS4 proves to be the least computational expensive approach that was chosen, but it is also the least accurate method. The sharp edges in both test cases became distorted and the fields were smoothed as a result of false diffusion. Furthermore the scheme suffers from numerical oscillations which lead to very high errors in the mean. The TVD scheme shows similar characteristics, although the overall errors are smaller compared to CDS4, with exception where the field returns to its original position. However the computational cost of the TVD scheme is twice as high with a limited improvement in accuracy.

The Lagrangian approach provides an extremely low error, which can be reduced further with an increase in particle density. Although, too many particles can lead to unphysical results in some scalar fields (mixture fraction, progress variable). The computational costs are also significantly higher (three to four times as high as CDS4).

Finally, the joint Euler-Lagrange approach has shown to provide low errors in all test cases. The approach was used with a third of the number of particles in comparison the Lagrangian approach, reducing the computational cost significantly. Further investigations are required into three-dimensional turbulent flows where scalar fields change continuously. As one of the strengths of the proposed method is the particle distribution to only areas of steep gradients, particle reinitialisation is of significant importance. Re-filtering the field when particles are reinitialised will introduce a continuous error. To avoid this problem, the second derivative of the scalar field can be used to identify areas of steep gradients. If the derivative is above a certain threshold, particle are initialised. Moreover, the application of the new method is currently limited to fully resolved flows. On the other hand, sub-grid scales are modelled in LES; hence the particle contribution also needs to be linked to the sub-grid scale models for fully bound results.

Overall it has been shown that the approach offers good potential to improve the accuracy and resolution of scalar transport by combining the advantages of the Eulerian and Lagrangian approach while maintaining moderate computational costs. As mentioned above, the further development of the method is conducted by a fellow PhD student.

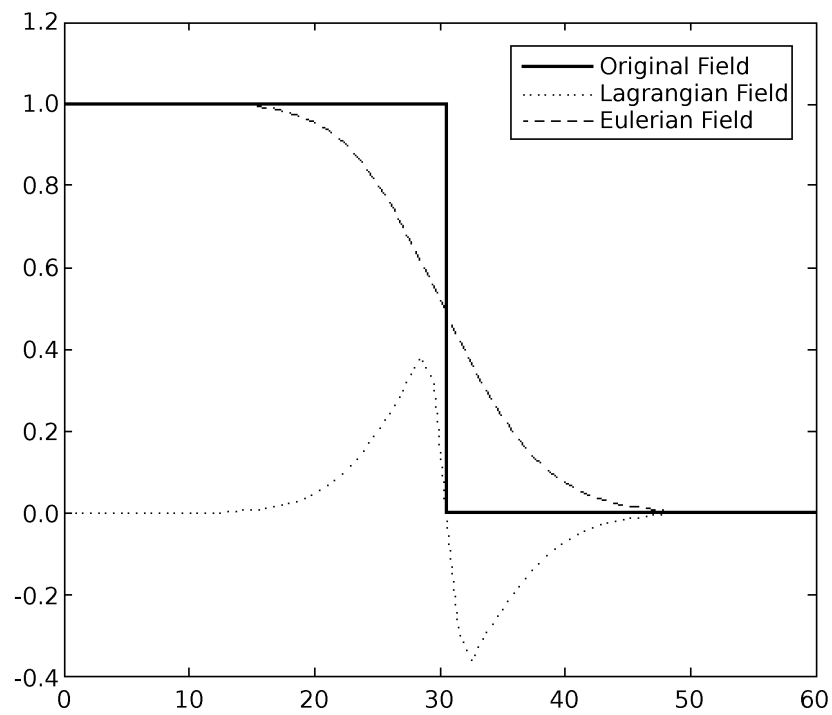


Figure 6.1: Graphical Representation of decomposition

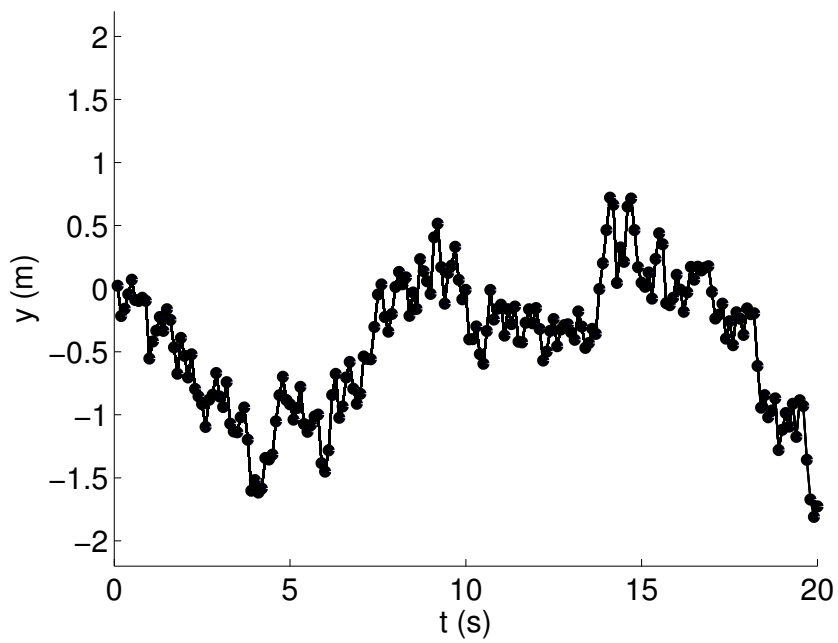


Figure 6.2: Visualisation of a 1-D random walk, based on the Wiener Process, particle has been initialised at the origin $(0,0)$.

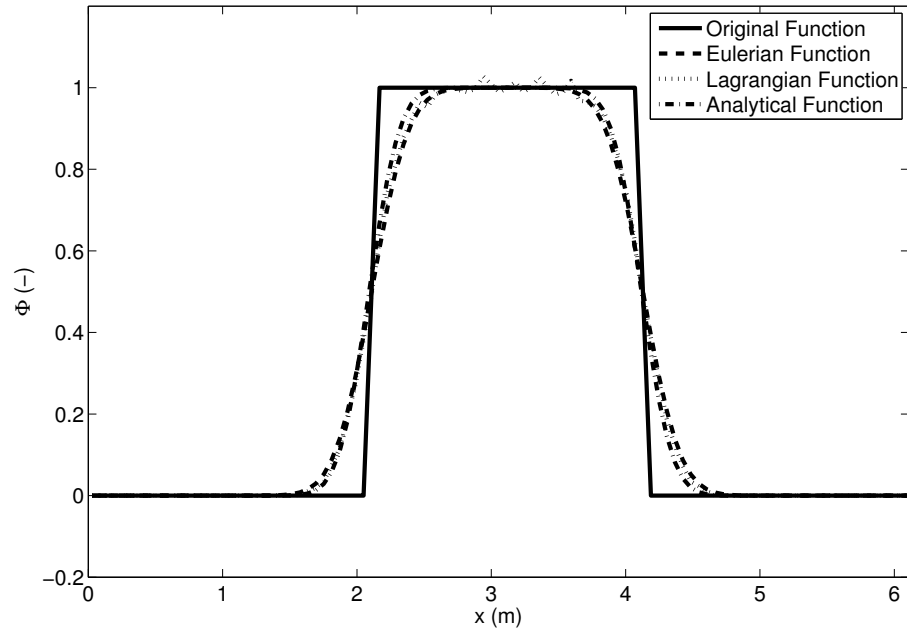


Figure 6.3: Identical diffusion coefficients; Eulerian, Lagrangian and Analytical Solution based on a top-hat function

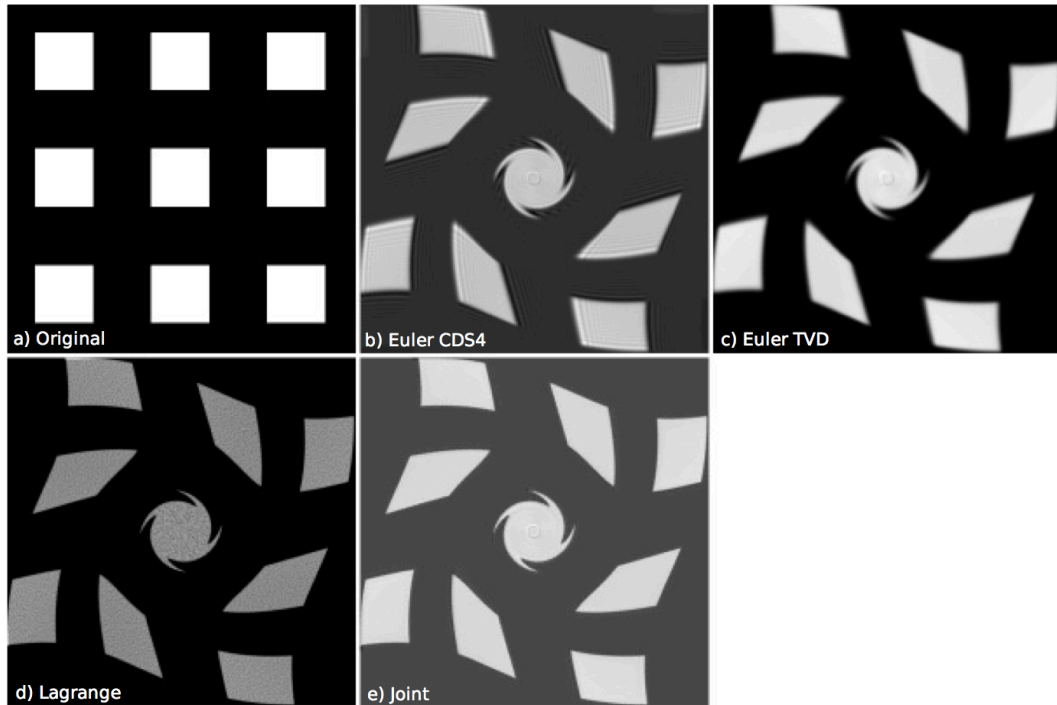


Figure 6.4: Comparison of results - original field (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach

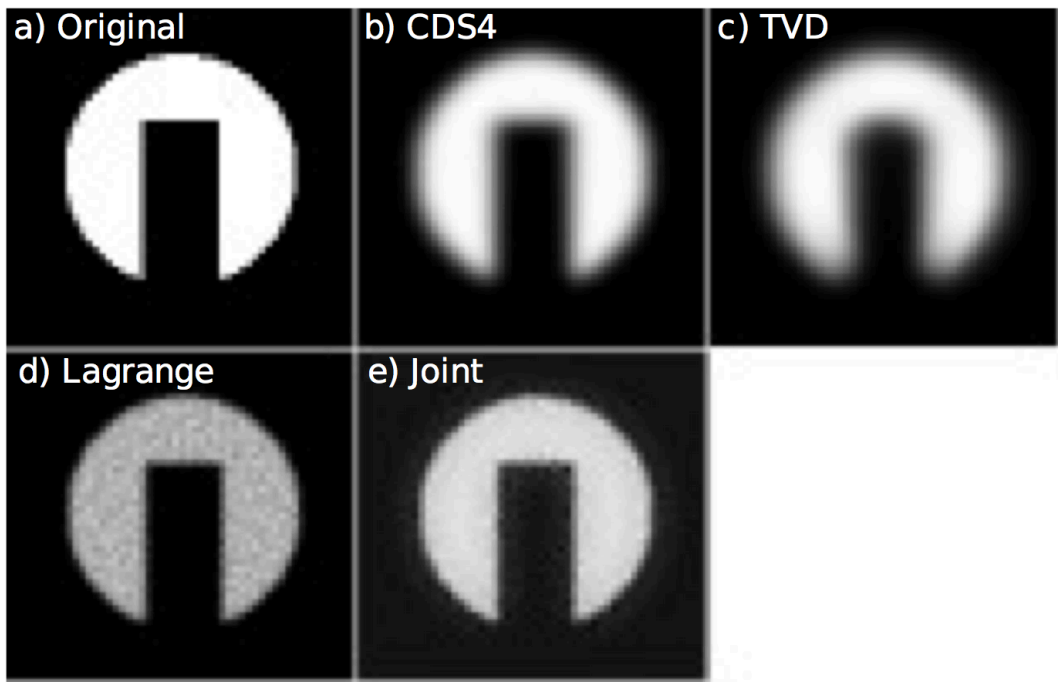


Figure 6.5: Comparison of results, disk returned - original disk (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach

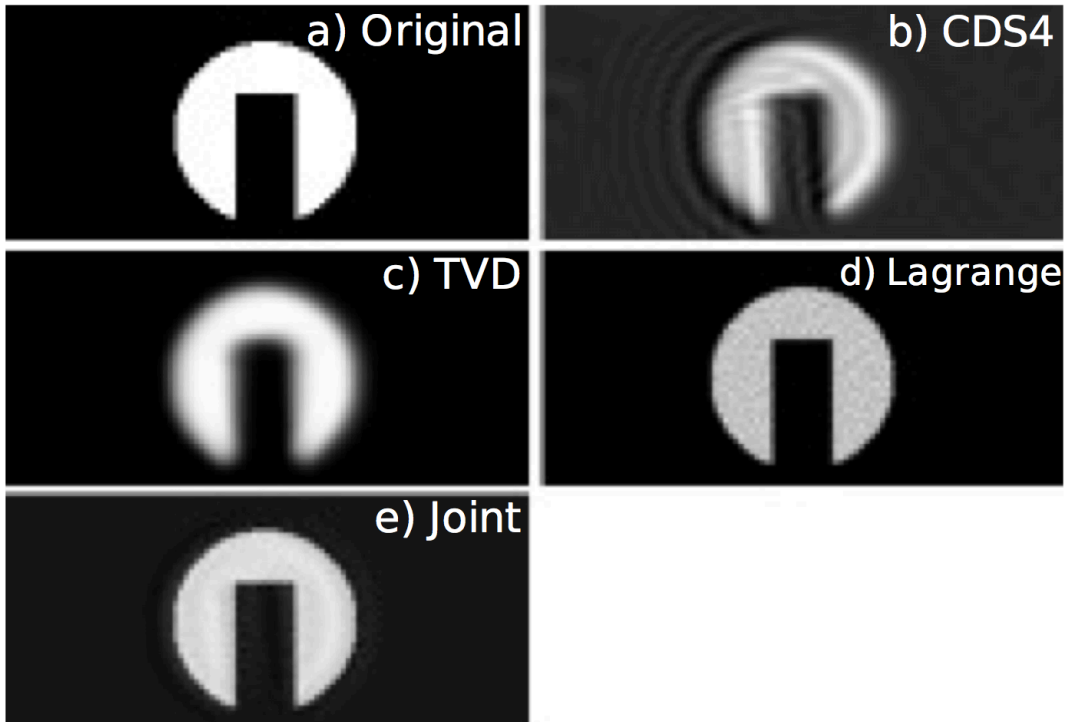


Figure 6.6: Comparison of results, one revolution - original disk (a), Eulerian (CDS4 (b) and TVD (c)), Lagrangian (d) and Joint (e) approach

Table 6.1: Simulation parameters for test cases to evaluate the performance of the proposed Euler-Lagrange method for Interface Flows

	Block function	Zalesak (return)	Zalesak (1 rev.)
Domain Size	400 ²	200 ²	200 ²
Time steps	8000	12560	6280
Time-step width	0.1	0.1	0.1
Diff. coefficient	0.1	0.1	0.1
Velocity Field	Rot. Vortical	Rot. Vortical	Rot. Vortical
Time Integration	3 rd R-K	3 rd R-K	3 rd R-K

Table 6.2: Block: Comparison of different approaches - Simulation times

Approach	Diff. Scheme (-)	Cells (-)	Particles ($\times 10^6$)	Time (s)
Eulerian	CDS4	400x400	0	409.9
Eulerian	TVD-CHARM	400x400	0	1073.3
Lagrangian	None	0	1.64	5454.3
Joint	CDS4	400x400	0.31	1171.1

Table 6.3: Zalesak disk returned: Comparison of different approaches - Simulation times, RMS and sum of error (SoE)

Approach	Diff. Scheme (-)	Cells (-)	Particles ($\times 10^3$)	Time (s)	x_{rms} (-)	x_{SoE} (-)
Eulerian	CDS4	200x200	0	86.7	0.0747	753.1
Eulerian	TVD-CHARM	200x200	0	194.8	0.0901	1094.6
Lagrangian	None	0	110.6	485.6	0.0476	405.6
Joint	CDS4	200x200	37.6	217.4	0.0514	522.9

Table 6.4: Zalesak disk, one revolution: Comparison of different approaches - Simulation times, RMS and sum of error (SoE)

Approach	Diff. Scheme (-)	Cells (-)	Particles ($\times 10^3$)	Time (s)	x_{rms} (-)	x_{SoE} (-)
Eulerian	CDS4	200x200	0	45.4	0.0812	1071.1
Eulerian	TVD-CHARM	200x200	0	86.3	0.0793	839.6
Lagrangian	None	0	110.6	136.9	0.0334	270.9
Joint	CDS4	200x200	37.6	62.8	0.0362	338.7

Chapter 7

Conclusions and Recommendations for Future Work

The presented project considered an Euler-Lagrange method for moving surfaces in Large-Eddy Simulation for incompressible flows. Immersed boundaries (solid objects) were represented and tracked by Lagrangian particles on an underlying Cartesian grid, on which the flow is advanced with a conventional Eulerian solver. Simplicity and computational efficiency, while achieving good predictions of first and second statistical moments, are key characteristics of the presented IB method. Large-Eddy simulations of non-moving boundary problems were carried out for a pipe flow and the flow around a sphere to evaluate the conservation of mass and momentum and accuracy of the method. The approach was then applied to an opposed jet flow with perforated plates and fractal grids with a very complex shape, based on the work by Geipel [51], to investigate the effect of fractal grids on the flow inside the nozzles. The immersed boundary method for moving objects was validated with the flow around a moving sphere, before investigating a piston-cylinder arrangement for an incompressible set-up. In addition, the idea of employing Lagrangian particles in an Eulerian framework was also applied to multiphase flows to increase the accuracy of scalar transport.

Lagrangian particles are used to identify and track immersed boundaries on an Eulerian grid. Diffusion, momentum and pressure terms are corrected near

the boundaries to ensure impermeability of walls. The 'fluid history' for *freshly cleared* cells, which are a results of boundary movement, is calculated by trilinear interpolation to satisfy the momentum equation. The parallelisation and optimisation of the method allow its application on large multi-core computers to investigate flows in complex domains. Furthermore the turbulent inflow generator was modified to improve its computational efficiency and avoid expensive communication between processors for the generation of inflow data. Good inlet conditions are important as they affect the flow further downstream. The dynamic Germano model also offers an alternative for the calculation of the sub-grid stresses in comparison to the static Smagorinsky model, as the contribution is localised for each position in the flow.

The flow in a pipe was simulated to evaluate the conservation of mass and momentum of the 'PsiPhi' flow solver in combination with the immersed boundary method and investigate the effect of two sub-grid scale models on the flow field. Results were analysed in terms of velocity statistics for first and second moments and compared to numerical results by Forkel [45].

A more detailed validation of the immersed boundary method was based on the well-documented flow around a static and moving sphere. Different flow regimes were investigated and results of the drag coefficient, wake length and mean axial velocity profiles were compared to numerical DNS and LES studies which used unstructured grids. Good agreements were achieved for lower Reynolds number flows for all investigated flow characteristics, however the coarseness of the initial grid had an effect on the predictions of higher Reynolds number flows. The statistical measurements of the moving sphere agree well with the static predictions, however the treatment of freshly-cleared-cells, which leave discontinuities in the flow field, effects the wake velocities near the sphere to a small degree.

The immersed boundary method was also applied to a turbulent opposed jet flow. The geometry was based on the work by Geipel [51], who used fractal grids in addition to the conventional perforated plates. Isothermal and reactive flows were simulated on grid sizes similar to those used in Direct Numerical Simulations. General data of first and second moments of axial velocity for isothermal simulations agree well with experimental measurements and the

predicted turbulent Reynolds number matches the value by Geipel. The artificially thickened flame model was used for the transport of the progress variable for the simulation of a methane flame with equivalence ratio of 0.8. The statistical measurements of first and second moments of the progress variable show good predictions when compared with the experimental data, encouraging the investigation of areas that are difficult to measure experimentally. Overall, the description of the complex fractal grid by immersed boundaries led to very good and accurate results, demonstrating the power of the IB method.

The geometry of piston-cylinder arrangement was based on the experimental set-up by Morse et al. [125]. Statistical measurements of first and second moments of axial velocities at two different crank angles were compared to experimental and numerical data. Mean velocity profiles show a good agreement with the experimental data sets, whereas the statistical measurements for the second moments would have benefited from a higher sampling count and a finer grid resolution.

In general, the coarseness of the grids affected the accuracy of results, as larger cell sizes were usually insufficient to resolve the domain accurately, leading to poor predictions of the flow properties. In the simulations of the spheres, the drag coefficient and wake length for higher Reynolds number flows were wrongly predicted on the coarser grid. The flow is strongly influenced by the wall effects and needs to be properly resolved. Similarly, the grid resolution in the opposed jet study was an important factor to resolve the turbulent flow inside the nozzle correctly.

A simple forcing approach is used in the immersed boundary method, which allows the simulation of complex geometries without having to generate complex meshes. This was advantageous for the simulation of the opposed jet and the piston-cylinder arrangement, which are both complex. The fractals in the grid of the opposed jet have to be resolved to accurately predict the flow. Similarly, the resolution of inflow jets in the piston-cylinder arrangement affect the flow inside the cylinder. Generating unstructured grids for these domains can be a complex task, especially when boundaries move.

A key benefit of the proposed method is that the equidistant Cartesian grid al-

lows for fast computational arithmetic and compiler optimisation. The method was also parallelised and optimised for large-scale supercomputers without having to implement major changes. This allowed for the analysis of the turbulent opposed jet and piston-cylinder arrangement, which were mentioned above.

Even though results of the presented IB method are encouraging, especially considering the computational efficiency, the presented immersed boundary approach has certain limitations. The simple forcing term makes the method first order accurate near walls and it would benefit from a higher order accuracy in tracking the moving boundaries. In addition a fourth-order interpolation scheme would reduce the effect of the discontinuities, which are an artefact of moving boundaries.

The accuracy of simulations can generally be improved by refining the grid. However a grid refinement increases the computational costs, by a factor of 8 for cubic equidistant grids if the cell size is halved. Therefore it should be evaluated if the simulation benefits from a refinement. It is also important to resolve the boundary correctly in flows that are dominated by the wall effects. Instead of refining the grid in these regions, wall stress models can be used to model the flow near the wall. The simplest approach introduces the van Driest damping in the near wall region [19] to account for the wall stresses. More sophisticated models, such as the zonal two-layer approach, are often based on well established RANS models, except that they are applied in an instantaneous sense [189].

The presented results look promising and encourage the implementation of the immersed boundary method in a compressible flow solver to widen the application range. For the investigation of compressible problems, other aspects have to be taken into account. Constant density can no longer be assumed and therefore the energy equation and evolution law of entropy are required, in addition to the conservation of mass and momentum, for the description of the Eulerian flow field. Furthermore for supersonic flows, the time-step width is very small, increasing the computational times significantly to simulate an equal length of real flow time. Nonetheless, the implementation of the actual immersed boundary method should not differ from the presented method.

Currently, the immersed boundary method only considers one-way coupling between the fluid and the solid boundary, in other words the force exerted on the fluid by the boundary. However, in dynamic systems, where the fluid can deform the boundary or effect its movements (for example the blood flow through the heart [134]), two-way coupling has to be considered to account for the fluid force on the boundary. The flow field around the object needs to be evaluated to calculate the force acting on the boundary and hence, evaluate the forces' effect on the particle movement to predict the new boundary position.

The immersed boundary method can also be extended to porous objects. Instead of correcting the pressure across the object boundaries to ensure impermeability, the routine needs to be adjusted to calculate the pressure and flow within the object. Particles could be given an additional class to assign higher or lower densities to specific areas.

In summary, the presented immersed boundary approach works well for cases, where the region of interest is not the boundary layer flow and the flow is not dominated by the flow in the boundary layer, such as combustion chambers, chemical reactors or mixers. The presented results are encouraging and offer the opportunity to develop the approach further.

Appendix A

The Wiener Process

The following section outlines the Wiener process in more detail. The analysis is based on the work by Cant and Mastorakos [20].

The description of particle diffusion should be easily computable and evolve similar to a classical, continuum diffusion process. A time interval T is considered, which can be divided into equal sub-intervals $\delta t = T/N$, where N is the number of sub-intervals. Let $\xi_{(n)}$, with $(n = 1, 2, \dots, N)$, denote N independent normal variables with zero mean $\langle \xi_{(n)} \rangle = 0$ and variance equal to the time difference. The particle diffusion can then be described with a Wiener process (W_t) , which can be expressed as:

$$W_{t_n} = (\delta t)^{1/2} \sum_{i=1}^n \xi(i) \quad (\text{A.1})$$

where $t_n = n\delta t$ and initial condition $W_{t_0} = 0$. The Wiener variable is independent of n and all random variables are uncorrelated. Alternatively, the increment of W_{t_n} at time t can be defined by

$$W_{t+\delta t} - W_t = (\delta t)^{1/2} \xi_{(n)} \quad (\text{A.2})$$

which is equal to the square root of the time-step multiplied by a standardised Gaussian random variable $\xi_{(n)}$ [145]. For a time interval $(t_d - t_c)$ with $(d > c)$, the mean and variance of the difference $W_{t_d} - W_{t_c}$ can be deduced as

$$\langle W_{t_d} - W_{t_c} \rangle = (\delta t)^{1/2} \sum_{i=c+1}^n \langle \xi(i) \rangle = 0 \quad (\text{A.3})$$

and

$$\langle [W_{t_d} - W_{t_c}]^2 \rangle = (\delta t) \sum_{i=c+1}^n \sum_{j=c+1}^n \langle \xi(i)\xi(j) \rangle = \delta t(d - c) \quad (\text{A.4})$$

Thus the difference $W_{t_d} - W_{t_c}$ can be represented by a Gaussian random variable with zero mean and variance equal to the time difference [145], as shown in Fig. A.1. The figure shows the simulation results of a random walk in one dimension for twenty particles, where all particles are initiated at the origin $(0, 0)$.

In other words a Wiener process can be described as a random walk with random step size. During the random walk, a particle 'jumps' a distance $\Delta y = (\delta t)^{1/2} \xi_{(n)}$ at each time step δt between $(t_d - t_c)$ with equal probability to the top or bottom [133] and the probability P of a particle to be at position y after time t is given by

$$P(y, t) = \frac{1}{\sqrt{(4\pi tk)}} e^{(-\frac{y^2}{4tk})} \quad (\text{A.5})$$

which is a Gaussian distribution with $\sigma = \sqrt{(2tk)}$, $\mu = 0$ and diffusion rate k [44].

To demonstrate that the Wiener process is also a solution of the diffusion equation, the Gaussian distribution approach is chosen. For simplicity a one-dimensional case with no convection, a constant diffusion rate D and constant density ρ is considered to differentiate the right-hand side of equation A.5. The first order derivative of the Gaussian distribution, with respect to t , is

$$\frac{df}{dt} = \frac{y^2 e^{(-\frac{y^2}{4tk})}}{8kt^2 \sqrt{\pi} \sqrt{tk}} - \frac{k e^{(-\frac{y^2}{4tk})}}{4kt \sqrt{\pi} \sqrt{tk}} \quad (\text{A.6})$$

and the second derivative of the Gaussian distribution, with respect to y , is

$$\frac{d^2 f}{dy^2} = \frac{y^2 e^{(-\frac{y^2}{4tk})}}{8k^2 t^2 \sqrt{\pi} \sqrt{tk}} - \frac{e^{(-\frac{y^2}{4tk})}}{4kt \sqrt{\pi} \sqrt{tk}} \quad (\text{A.7})$$

Inserting the results from A.6 and A.7 into the diffusion equation

$$\frac{df}{dt} = D \frac{d^2 f}{dy^2} \quad (\text{A.8})$$

with $D = k$, results in

$$\frac{y^2 e^{(-\frac{y^2}{4tk})}}{8kt^2 \sqrt{\pi} \sqrt{tk}} - \frac{ke^{(-\frac{y^2}{4tk})}}{4kt \sqrt{\pi} \sqrt{tk}} = \frac{y^2 e^{(-\frac{y^2}{4tk})}}{8kt^2 \sqrt{\pi} \sqrt{tk}} - \frac{ke^{(-\frac{y^2}{4tk})}}{4kt \sqrt{\pi} \sqrt{tk}} \quad (\text{A.9})$$

The diffusion equation holds, hence the Wiener process is also a solution of the diffusion equation and can be used to describe particle diffusion.

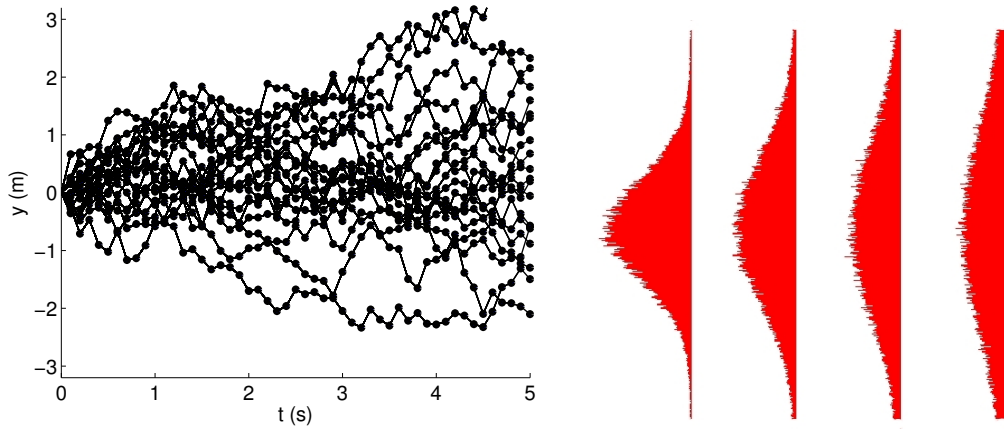


Figure A.1: Random walk for multiple particles initiated at the origin (left) and the particle distribution at various positions along the x-axis (right).

Appendix B

Performance Measure and Optimisation of a Computer Tomography Code for High Performance Computing

The structure of the presented research project was a little bit different to the usual postgraduate research project as it included a MSc in High Performance Computing (HPC) from the University of Edinburgh. The additional study ensured the improvement of computational techniques and algorithm development, especially with regards to HPC.

The following Chapter gives a brief overview of the Masters project. The work has been included to illustrate the importance of high performance computing in research nowadays, such as Computational Fluid Dynamics (CFD), and that algorithmic and computational improvements can lead to significant reductions in computational time. It should be made clear that the MSc project should not be regarded as additional work to the present project, but that the current work benefited from the additional training received during the MSc. Optimising parallel routines for a CFD code can be given almost as much importance as the development of the CFD code itself for large industrial applications. Furthermore the presented diagnostic tool is suitable in combination with Large-Eddy simulations to analyse premixed flames.

Scientists and engineers have been interested in characteristics and efficiency of combustion for a long time. The improved combustion process allows cars to travel further with the same amount of fuel and power stations to run more efficient. Furthermore an improved combustion process often means a reduction in CO_2 emissions and prevent in-reversible long term effects such as climate change, besides the fact that fossil fuels are diminishing. Alternative methods exists, but they are often not economically liable for the majority of energy users. Until these methods have been improved, fossil fuels will remain the main energy source and this requires measure to make the most of the remaining supplies.

Introduction

Time resolved three-dimensional measurements of turbulent flames are required to further understand the combustion process and to support advanced simulation techniques. Understanding the combustion process allows the development of more sophisticated burners, which require less fuel and are more efficient.

A sensor technique applied by Floyd et al. [9] uses Computed Tomography of Chemiluminescence (CTC) to reconstruct the structure of the turbulent flame brush. The technique uses an iterative Computed Tomography algorithm to reconstruct a 3D chemiluminescence profile of a flame from multiple instantaneous line-of-sight light emission measurements (pictures) taken at different viewing angles around the flame. The quality of the recovered field strongly depends on the number of images (*view*), where a higher number of images leads to a better resolution.

Experimental techniques for time resolved whole volume measurements are often very costly and limited by visual access. As a result the application of these techniques to the field of combustion are limited, although they supply important information to compliment simulation techniques such as RANS (Reynolds-Averaged Navier-Stokes) and LES (Large-Eddy Simulation) to gain a greater insight into the combustion process.

Alternative used approaches are often Laser based and they can provide high

resolution quantitative information of line and planar measurements, but often struggle with 3D measurements. Notable exceptions are stacked multiplexed planar laser sheet set-ups by Böhm et al. [3] and Nygren et al. [17] to provide three-dimensional data. However these techniques require the detector position to be normal to the plane of interest, which makes measurements on some flame configurations difficult to measure. Furthermore expensive, often custom built, high speed cameras are required for these techniques. Computed tomography has also been coupled with laser absorption, for example [2, 5], to get precise species concentrations. However the measurements were often limited to low resolution and a single plane due to the cost of the equipment and access requirements. The interested reader is referred to [8, 9] for an more extensive list of methods and applications of laser based methods.

Computed Tomography of Chemiluminescence allows for greater flexibility in camera position, especially in 3D measurements where cameras no longer need to be co-planar, as shown by Floyd et al. [9]. Due to its flexibility in camera position, CT has been a preferred method in combustion research in recent years, examples include the work by Feng et al. [7] and Gillet et al. [10] who used the method to localise integral measurements. Higher 3D resolutions often require more complex system set-ups and are therefore more considered for flow visualisations. 3D flame chemiluminescence has been investigated by Ishino and Ohiwa [14], who focused on turbulent flame flickering by using expensive custom build cameras. Floyd [8] measured the chemiluminescence of a turbulent opposed jet flame with standard cameras to reconstruct a three-dimensional instantaneous scalar field of the flame. More applications of computed tomography in the field of combustion are discussed in the work by Floyd et al. [9, 21].

Computed Tomography

Computed tomography is essentially a mathematical inversion technique which is used to reconstruct images from one or two dimensional projections. Three dimensional data is returned from integral (line-of-sight) measurements taken at different angles around a chosen object. The idea of computed tomography was first introduced by Radon in 1927, nonetheless it was not broadly applied until the 1970s when the EMI scanner was invented. The basic technique of

computed tomography has not changed since its early developments, but it remains an active field of research. Today CT is employed in many different disciplines such as medicine, seismology and electron microscopy, to name the most common areas of application.

The first attempts of applying computed tomography in the field of combustion were made in the early 1980's, but success was often limited. The resolution of the reconstruction depends on the number of views, the resolution of each view and signal to noise ratio. Spatial resolution can be achieved by simultaneous measurements and therefore many detectors and cameras are required. With improving technologies, the advantage of computed tomography are the moderate costs and the use of standard cameras with flexible camera positions. The reconstruction is always discrete and it will be an approximation of the original field, where the error diminishes with increasing resolution and number of views. More information on computed tomography can be found in the book *Principles of Computerized Tomographic Imaging* by Kak and Slaney [15]. The book covers the mathematical principles of the technique and the Algebraic Reconstruction Technique.

Chemiluminescence

Chemiluminescence is non-intrusive and the result of a chemical reaction that emits light which is not resulting from heat [9]. The reaction occurs in a narrow region of the flame, hence spatially resolved volume information can be obtained relatively cheap. During the chemical reaction an excited molecule decays to the ground state by emitting a photon of light. The wave length of light depends on the specific transition that occurs and can be ultraviolet, visible or infrared light. More detailed information about on the process of the chemical reaction is outlined in the book *The Spectroscopy of Flames* by Gordon [11]. Its non-intrusive process makes it very suitable for the study of turbulent combustion - especially in premixed combustion with no or little soot [1, 4, 18]. These studies looked at the local equivalence ratios and heat release, even though results can be affected by absorption and scattering of emitted light by the flame before it reaches the detector. Additional areas of use of chemiluminescence are listed in the literature [22].

Specific Objectives

Floyd [9] applied an Algebraic Reconstruction Technique (ART) algorithm to reconstruct the 3D field. Despite of the good reconstruction, computed tomography algorithms are often limited by the computational efforts required to reconstruct the field, due to their iterative nature. However, nowadays, modern computers allow the exploitation of parallel computing by using multi-core chips to reduce the time to solution. The idea of parallelising scientific codes has been around for decades and the following example illustrates the possibilities of improving the efficiency of a scientific code by applying various techniques. Similar methods were applied to the scientific code that was developed and used in the main report in order to minimise computational effort and time to solution, while maintaining maximum code flexibility [19].

The following paragraphs describe the use of modern computers to exploit parallel computing paradigms using multi-core chips to reduce the time to solution. Two algorithms, ART and MART (Multiplicative Algebraic Reconstruction Technique), were parallelised to exploit these architectures. Three parallel programming paradigms were applied for the parallelisation of the algorithms; these are the commonly known shared memory, distributed memory and mixed-mode models. The later model is based on the shared and distributed models where the architecture of multiple multi-core chips is exploited to maximise performance and minimise time to solution. The algorithms will be discussed briefly and important data dependencies of the sequential algorithm will be highlighted; a more detailed discussion of the algorithms can be found in [21].

The parallel implementations are evaluated with three test cases, which are based on a known synthetic field, called a *Phantom*, of the matrix burner used by Floyd [9]. The test cases vary in image resolution and number of views as shown in Table B.1.

CTC Experimental Method

The reconstruction process of the three-dimensional field consists of two parts; the camera arrangement for the measurements of the multiple simultaneous views and the reconstruction of the intensity field based on the views by ap-

Table B.1: Applied test cases

Test Case	Resolution (number of pixel)	Number of views
lowres_highview	64x18	64
highres_lowview	142x40	10
highres_highview	142x40	64

plying the reconstruction algorithm. A detailed description of the experimental method and the equipment used for the matrix burner can be found in [8, 9]. Floyd also applied the method to an opposed jet setup, but this work will use the matrix burner as a setup due to its simplicity of the flame structure.

A matrix burner was used as a test case, the burner consists of rows of two small concentric nozzles with inner diameter of 1 mm and 3 mm. The burner plate consists of 21 nozzles, which are arranged as shown in Fig. B.1. The inner nozzle supplies the fuel and the outer nozzle the oxidiser.

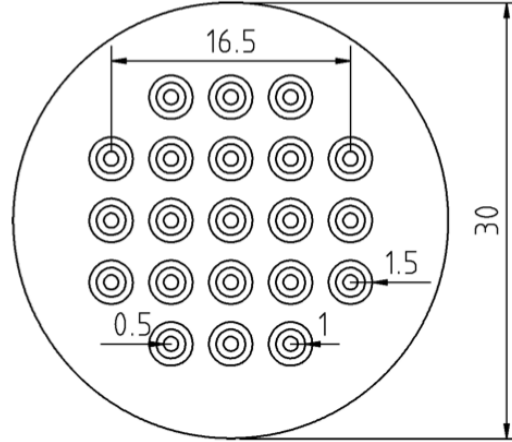


Figure B.1: Drawing of the Matrix Burner

3D CT Processing

The obtained views can be processed using the CT algorithm to yield the 3D reconstruction. In the presented work, the parallel ART and parallel MART algorithms were used for the reconstruction, as they work well for combustion applications as well as limited and noisy datasets [9]. Numerous other CT algorithms exist and they can usually be divided into two categories: *ana-*

lytical and *iterative*. Most analytical algorithms are based on the concept of back projection and require many low noise views for a successful reconstruction [15]. These algorithms, in particular Filtered Back Projection (FBP) are often applied in medical applications where good optical access is guaranteed. The iterative algorithms are usually more robust in the presence of noise and therefore are preferred in physical/chemical science. Although time to convergence with iterative algorithms is generally slower.

The original ART algorithm was developed by Gordon [12] and many different forms have been formulated since, one of which is the multiplicative ART algorithm that has also been used in this work. In simple terms, the iterative algorithms back project a diminishing error into the reconstruction domain until a solution criteria is met. The convergence criteria is necessary because in many situations, it is not possible to find the exact reconstruction of the object, due to noise and limitations in the number of views available. More details about the algorithms are given in [8, 12, 21].

In the ART algorithm, the error is generated by comparing each measured projection value with an equivalent projection taken through a current iteration's estimate of the object $f^{(h)}$. The normalised error is then back projected into the reconstruction domain using a contribution factor for stability, before the next projection is addressed. The next iteration will start once all projections in all views have been addressed. In this work, the relaxed, additive version of the ART algorithm, which is shown in Eq. (B.1), parallelised,

$$f^{(h+1)} = f^{(h)} + \beta \omega_{qp} \frac{I_{qp} - \omega_{qp} * f^{(h)}}{\omega_{qp} * \omega_{qp}} \quad (\text{B.1})$$

where β is a relaxation factor, f the reconstruction object at iteration h , I is a column vector containing the integral values p of all projection measurements q and ω is the weighting factor.

According to Eq. (B.1), each view contributes individually to the reconstruction. Regardless of the mathematical description of the algorithm, the convergence speed can be improved by updating pixel values in place. Therefore the correction of a pixel depends on a previous estimate of a previous angle in the same iteration. For example, the value of a pixel P_{q+1} at angle θ_q becomes

P_q in the calculation of P_{q+1} for the following angle θ_{q+1} . Therefore the correction of the estimate is progressive within an iterations as well as between iterations. The solution is regarded as converged if the difference of the object in two successive iteration is below some proportion of the field, as discussed below.

The Multiplicative Algebraic Reconstruction Technique (MART) is derived from the ART algorithm, which has been introduced above, and is often considered as a successor. The ART algorithm has the disadvantage that negative corrections are possible and reconstructions often show artefacts or tracers [6]. To avoid this problem, the multiplicative ART algorithm was suggested [20] and applied as shown in Eq. (B.2).

$$f^{(h+1)} = f^{(h)} * \left(1 - \frac{\omega_{qp}}{\omega_{qp} * \omega_{qp}} \left(1 - \frac{I_{qp}}{\omega_{qp} * f^{(h)}}\right)\right) \quad (\text{B.2})$$

As the name suggests, the algorithm does not apply an additive, but multiplicative correction to the reconstruction. The correction is based on the ratio of the recorded projection intensity and the reconstruction intensity from the previous iteration. The advantage of the MART algorithm is that it can diminish the influence of noise in the projection data and guarantees a meaningful solution when the initialisation is a constant. Identical to the ART algorithm, the MART algorithm considers the correction of a pixel at a time, with an iteration being completed after all projections have been considered. Compared to the ART algorithm, the MART algorithm does not allow for negative corrections, hence once a reconstruction pixel has been set to zero, it will remain at zero due to the multiplicative nature of the algorithm. As a consequence an initial estimate of $f(0) = 0$ is not reasonable, hence the mean value derived from the projections or $f(0) = 1.0$ are acceptable initialisations for the MART algorithm.

Data Dependencies

Data dependencies in a sequential algorithm need to be determined before parallelising the algorithm. Generally these dependencies should not be broken, as results might change. However, the reconstruction problem is usually defined by a set of under-determined equations and the solution is an estimate to the

real solution, as opposed to the real solution. Therefore the violation of the data dependencies might change a particular solution, but not necessarily reduce the quality of the reconstruction itself.

The dependencies between iterations cannot be broken. An estimate for $f^{(h)}$ has to be found first before estimating the solution for the next iteration $f^{(h+1)}$, which limits the options of possible parallelisations.

The dependency of projections should also be considered. In the sequential algorithm, views are considered in sequence, although the consideration of views does not need to be in order, as shown by Guan and Gordon [13]. Their suggested multilevel access scheme considers projections at 90 degree separation with which they managed to achieve a considerable speed up for the sequential algorithm. Due to the approximative nature of the solution and the findings by Guan and Gordon, it is legitimate to consider views in parallel.

The final dependency to consider arises as each ray in a particular projection is considered. In most cases each ray width only considers one voxel value of a particular row or column in the reconstruction matrix. Hence the calculation of each voxel is independent of all other voxels in the parallelisation.

As a result, there are three options for parallelising the algorithm: over projections, over rays and over both (mixed-mode version only).

Conversion Criterion

The algorithm requires a conversion criterion in order to determine when the reconstruction of the field is sufficient. Normally the algorithms are known to converge when sufficient views are available to reconstruct the object. However, point of convergence is less than trivial, due to noise in the measured data and there is no outstanding choice for convergence criterion. The most suggested approach in the literature regards the solution as converged when the difference between two successive iterations is less than some value Δ_c . However this option is not ideal for very noisy datasets, as convergence fluctuates. In this project the solution is regarded as converged when the change in a reconstruction is below some proportion of the field for three successive

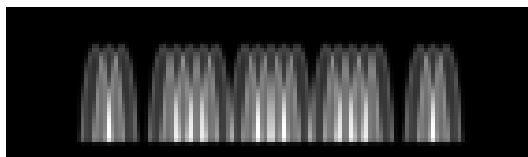


Figure B.2: Example view of the Matrix Burner test cases, at 45° to the horizontal

iterations as shown in Eq. (B.3). This criterion works well for the ART and MART algorithms and is more effective in the presence of noise.

$$f^{(h)} - f^{(h-1)} < f^{(h)} * \Delta_c \quad (\text{B.3})$$

Results and Discussion

The experimental work and reconstruction of the sequential algorithm will not be discussed here. The interested reader is referred to Floyd et al. [8, 9] for more detailed information on the CTC method, the experimental results of the matrix burner and the turbulent opposed jet. Furthermore the following section will only highlight key results from the mixed mode version of the parallelisation. The aim of this chapter is to highlight the advantages of an efficient parallelisation in order to reduce time to solution. Details of the parallelisations, results for the three programming paradigms and a detailed discussion can be found in [21].

Performance measurements for the evaluation of the parallel algorithms were carried out on the three different test cases of the matrix burner. The views had a resolution of 68×18 or 142×40 pixels, with 64 or 10 views as shown in Table B.1. A sample view of the phantom field of the matrix burner is shown in Fig. B.2.

The reconstruction domains were of size $64 \times 64 \times 18$ for the first test case and $142 \times 142 \times 40$ for the other two test cases. The reconstructions were performed with the parallel implementations of the ART and MART algorithms for all three parallel paradigms. A reconstruction was regarded as converged when the condition in Eq. (B.3) was met, where Δ_c was chosen as $1e^{-6}$. An example of the converged reconstruction, based on the lowres_highview test case and the mixed-mode parallelised ART algorithm, is shown in Fig. B.3.

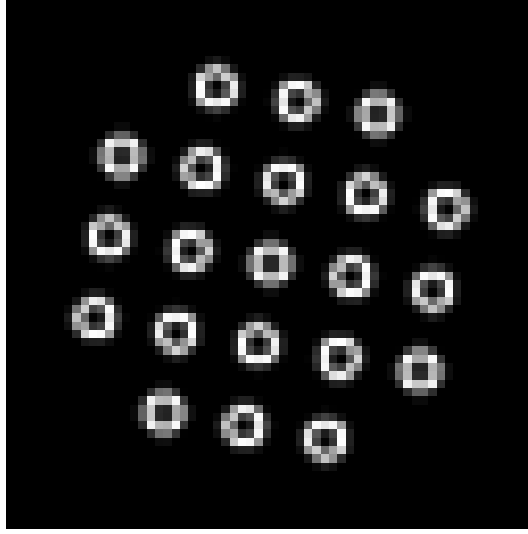


Figure B.3: Example reconstruction of the matrix burner with the mixed-mode parallelisation at low resolution with 64 views

The correctness of the reconstruction was assessed visually and numerically. Considered that all test cases are based on high and low resolution phantoms of the matrix burner, means the reconstruction is known. This allows for visual inspection and comparison of the results, which is a first indication to the correctness of the results. However visual inspection does not guarantee a correct solution. Therefore the correlation coefficient of the phantoms and their respective reconstructions on single and multiple processors, according to Eq. (B.4), was calculated.

$$r_{xy} = \frac{\sigma_{xy}^2}{\sigma_{xx}^2 * \sigma_{yy}^2} \quad (\text{B.4})$$

A correlation coefficient of unity results in a perfect reconstruction of the phantom. Therefore it can be assumed that the reconstruction was successful. During the project, all results were in the range of $0.96 < r_{xy} < 1.00$.

In the distributed memory model, data is stored locally by each processors and data needs to be explicitly communicated between processors. As a result each processor reconstructs the field based on its local data sets. At the end of each iteration, the sets are averaged over the number of processors to gain a global reconstruction. The performance improvements showed a great deal of

variance, which were often limited by the communication latencies and work load imbalances between processors, depending on the data decomposition. The work-load imbalance can be improved by choosing appropriate data decompositions, however the communication overheads are still significant. The best reduction of time to conversions was achieved on 8 processors. The time to solution was reduced by a factor of 2 with an efficiency of 25% for the ART algorithm. The parallelisation of the MART algorithms results in better performance improvements. A reduction in time to solution by a factor of 5 was achieved on 6 processors. Therefore the efficiency is at 83%.

In the shared memory implementation, all data is stored in global memory and accessed by all processors. Hence, no explicit communication is required. However, sharing memory between processors also involves risks of over-writing important data that has not been read by all processors, which can lead to *race conditions* and the solution might not converge. It has to be ensured that race conditions will not occur during the reconstruction, which obviously limits the performance improvement of the shared memory implementations of the ART and MART algorithms. It should be pointed out that these limitations might not apply to other test cases [16]. Although Melvin does not specifically mention race conditions and hence he might not be aware of their occurrence. For the ART algorithm, the best reduction of time to conversions was achieved on 6 processors for all test cases. The time to solution was reduced by a factor of 2.5 with an efficiency of 40%. The MART algorithm scaled better and performance improvements of a factor 3.5 were recorded on 4 processors, which is in an efficiency of 88%.

Nowadays computer chips consist of 2 or more CPUs on a single chip (multi-core chips) and workstations with up to 16 quad-core chips are no rareness in research. Message passing often suffers from load imbalance and communication overheads, but it offers good scalability and portability. The shared memory scalability is often limited by the available memory, although synchronisation overheads are often negligible. The aim of the mixed-mode version is to combine the advantages of both methods. The mixed-mode might be more efficient than MPI alone, without being limited by the memory (Open-MP). MPI will be used for the communication between the nodes and Open-MP to synchronise threads through local memory. The run times of the ART and

Table B.2: Mixed-mode algorithms - Run Time

Proc.	Threads	Run Time (s)					
		64x18x64		142x40x10		142x40x64	
		ART	MART	ART	MART	ART	MART
1	1	22.5	15.1	248.4	82.3	176.8	117.4
1	2	12.1	7.7	125.6	53.5	95.0	69.8
2	2	9.5	3.9	113.0	34.8	90.2	77.0
2	3	6.7	2.7	78.2	31.9	87.3	58.3
2	4	5.9	1.9	94.8	25.2	61.9	47.0
2	8	6.4	3.3	80.2	24.8	52.4	47.1
4	2	8.1	1.9	126.1	26.9	101.1	58.9
4	4	6.3	1.4	107.8	22.0	84.6	48.7

MART algorithms are shown in Table B.2. The run time of the ART algorithm drops continuously with an increasing number of processors, however the efficiency drops below 40% for all test cases run on eight or more processors, independent of the decomposition. The inefficient jobs can be justified in situations where very fast results are needed. Although running these jobs on a big supercomputer would cost expensive resources for a minimal convergence advantage. The results of the parallel MART algorithm look very promising, especially for the two smaller test cases. Good speed ups and efficiencies were achieved, which is desirable in parallel computing. The efficiency of the parallel algorithms for the mixed-mode versions is shown in Fig. B.4.

Overall the parallel MART algorithm scales better than the parallel ART algorithm. The parallel ART algorithm requires more additional iterations than the equivalent MART algorithm due to the averaging of results. Furthermore a relaxation factor is applied in the ART algorithm to improve stability in convergence. The relaxation factor limits the correction in each time step, whereas the nature of the MART algorithm allows for greater correction which limits the impact of the averaging. The MART algorithm also benefits from its multiplicative nature with regards to zero values. Once a voxel has been set to zero, it will remain at zero. This limitation does not exist in the ART algorithm; the algorithm actually allows for negative corrections. In parallel this might lead to collisions between values on different processors. The ART algorithm performs better with test cases of high number of views. This theory has only been tested on three similar test cases and requires further investigation by

considering a test case with a medium number of views.

As expected, the MPI implementation suffers from communication overheads. In the chosen implementation, results are communicated after each iteration. This leads to an insufficient work load / communication ratio and should normally be avoided. The Open-MP implementation allows for the synchronisation in memory, therefore the performance does not suffer from communication latency. However the performance is still influenced by an uneven work load balance, as only non-zero elements are considered in the reconstruction.

Furthermore the Open-MP version is superior in the small grain parallelisation. The mixed-mode version was implemented to use the advantages of both shared and distributed memory implementations. The mixed-mode showed good speed ups and allows for greater scalability with good performance results compared to the pure MPI and Open-MP versions. With an increasing popularity in multi-core chips, the mixed-mode version should be tested and developed further for test cases with higher resolutions ($\sim 640 \times 480$ pixels).

Good speed ups were achieved in tests with low number of processors. The efficiency often dropped with an increasing number of processors (~ 8) due to the decreasing work load on each processor by increased communication. Hence the use of a high number of CPUs needs to be questioned for the given problems. The parallel implementations seem more suitable to work station clusters than super computers.

Conclusion and Future Work

Modern computers allow the exploitation of parallel computing and hence two CT algorithms (ART and MART) were parallelised for different memory systems. A shared memory parallelisation was developed using the Open-MP directives; whereas the distributed memory version uses a message passing interface (MPI) to communicate between processors. Modern computer chips consist of multiple cores and to take advantage of that architecture a mixed-mode version was created to maximise performance improvements and reduce time to solution.

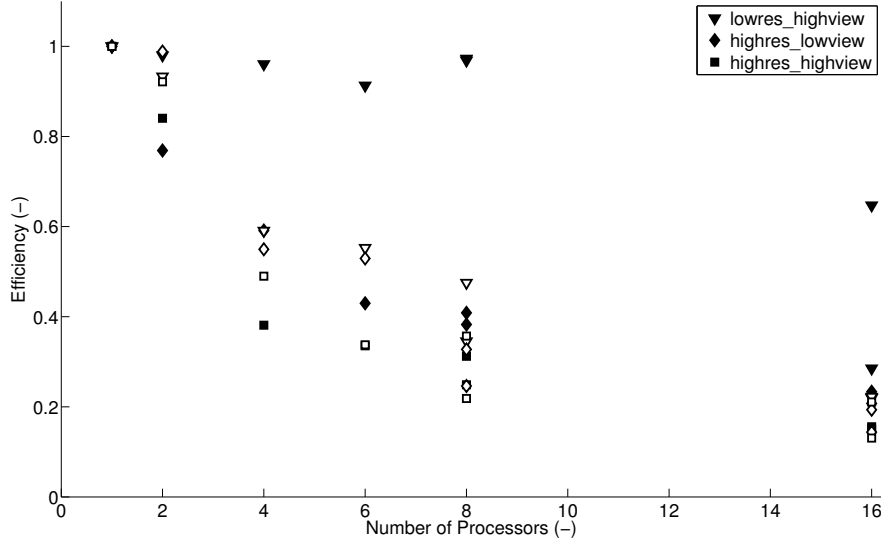


Figure B.4: Efficiency of the parallel ART and MART algorithms; filled symbols represent the MART algorithm, same symbols indicate same test case

Three test cases were used to test the different implementations. A small test case consisting of a high number of views at low resolution, a medium test case consisting of a low number of views at high resolution and a large test case which had a high resolution and a high number of views.

The MPI implementation of the algorithm showed a great deal of variance in the results. The gain of executing the algorithm in parallel was lost in communication overheads and additional iterations required as a result of the reconstruction averaging. Good reductions in run times were achieved with the parallel version of the MART algorithm and the ART algorithm on a low number of processors. Speed ups at higher number of processors are possible at the cost of low efficiency.

Theoretically the Open-MP implementation should be ideal for this kind of problem, however the implementation suffered from barriers which ensure the avoidance of race conditions. Yet again the efficiency dropped at higher numbers of processors and the scalability is limited by the available shared memory.

The mixed-mode model results were better in comparison to the distributed and shared memory models. Good speed ups were achieved with moderate ef-

iciencies. The implementation scales better than the shared memory version and is well suited for modern computer architectures with multi-core chips. However, the performance is still limited by the communication overheads and the averaging of results that is introduced by the MPI.

The two algorithms behaved differently. The ART algorithm seems to suffer from a relaxation factor and higher number of iterations, whereas the MART algorithm is more stable with noisy data sets. It also leads to faster convergence times, sequentially as well as parallel.

Overall the parallelisation is probably not suited for super computers due to the limitations of the algorithms and their inefficiency. The main limitations were the additional iterations due to the averaging and the communication/synchronisation overheads. More processors often resulted in higher communication overheads. Good performance improvements were achieved on smaller number of CPUs which makes the problem more suitable for local clusters that consist of a few multi-core chips (~ 20). The results were very case dependent and different test cases might show better performance data.

Bibliography: Main Report

- [1] E. Achenbach. Vortex shedding from Spheres. *Journal of Fluid Mechanics*, 62:209–221, 1974.
- [2] K. Akselvoll and P. Moin. Large-eddy simulation of turbulent confined coannular jets. *Journal of Fluid Mechanics*, 315:387–411, 1996.
- [3] D. M. Anderson, G. B. McFadden, and A. A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review Fluid Mechanics*, 30:139–165, 1998.
- [4] C. Arcoumanis and J. H. Whitelaw. Fluid mechanics of internal combustion engines - a review. *Proceedings Institution of Mechanical Engineers*, 201:57–74, 1987.
- [5] G. Arfken. Gauss’s Theorem. *Mathematical Methods for Physicists*, 3:57–61, 1985.
- [6] L. Arnold. *Stochastic Differential Equations: Theory and Applications*. Wiley, 1974.
- [7] E. Balaras. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers & Fluids*, 33:375–404, 2004.
- [8] C. Basdevant and R. Sadourny. Numerical solution of incompressible flow: the mask method. Lab di Meteorologie Dynamique, Ecole Normale Supérieure, Paris, 1984.
- [9] G. Batchelor. *The Theory of Homogeneous Turbulence*. Cambridge University Press, 1953.

- [10] M. J. Berger and M. J. Aftosmis. Aspect (and aspect ratios) of Cartesian mesh methods. *Sixteenth International Conference on Numerical Methods in Fluid Dynamics*, pages 1–12, 1998.
- [11] J. Y. Berthou and L. Colombet. Which approach to parallelizing scientific codes—That is the question. *Parallel Computing*, 23:165–180, 1997.
- [12] C. K. Birdsall and A. B. Langdon. *Plasma Physics via Computer Simulation*. McGraw-Hill, 1985.
- [13] M. Boger, D. Veynante, H. Boughanem, and A. Trouvé. Direct numerical simulation analysis of flame surface density concept for large eddy simulation of turbulent premixed combustion. *Symposium (International) on Combustion*, 27:917–925, 1998.
- [14] B. Böhm, O. Stein, A. Kempf, and A. Dreizler. In-nozzle Measurements of a Turbulent Opposed Jet Using PIV. *Flow, Turbulence and Combustion*, 85:73–93, 2010.
- [15] R. Borghi. *Recent advances in aeronautical science*. Pergamon Press, 1984.
- [16] J. U. Brackbill, D. B. Kothe, and H. M. Ruppel. FLIP: A low-dissipation, particle-in-cell method for fluid flow. *Computer Physics Communications*, 48:25–38, 1988.
- [17] M. Briscolini and P. Santangelo. Development of the mask method for incompressible unsteady flows. *Journal of Computational Physics*, 84:57–75, 1989.
- [18] T. D. Butler and P. O’Rourke. A numerical method for two dimensional unsteady reacting flows. *Symposium (International) on Combustion*, 16:1503–1515, 1977.
- [19] W. Cabot and P. Moin. Approximate Wall Boundary Conditions in the Large-Eddy Simulation of High Reynolds Number Flow. *Flow, Turbulence and Combustion*, 63:269–291, 2000.
- [20] R. S. Cant and E. Mastorakos. *An introduction to Turbulent Reactive Flows*. Imperial College Press, 2008.

- [21] F. Charlette, C. Meneveau, and 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*, 131:159–180, 2002.
- [22] J. I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224:757–784, 2007.
- [23] A. J. Chorin. Numerical Solution of the Navier-Stokes Equations. *Mathematics of Computation*, 22:745–762, 1968.
- [24] C. P. Chou, J. Y. Chen, J. Janicka, and E. Mastorakos. Modeling of turbulent opposed-jet mixing flows with model and second-order closure. *International Journal of Heat and Mass Transfer*, 47:1023–1035, 2004.
- [25] R. A. Clark, J. H. Ferziger, and W. C. Reynolds. Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *Journal of Fluid Mechanics*, 91:1–16, 1979.
- [26] D. Clarke, M. Salas, and H. Hassan. Euler calculations for multi-element airfoils using Cartesian grids. *AIAA Journal*, 24:1128–1135, 1986.
- [27] O. Colin, F. Ducros, D. Veynante, and T. Poinso. A thickened flame model for large eddy simulation of turbulent premixed combustion. *Physics of Fluids*, 12:1843–1863, 2000.
- [28] R. Cortez and M. L. Minion. The blob projection method for immersed boundary problems. *Journal of Computational Physics*, 161:428–453, 2000.
- [29] R. Courant, K. Friedrichs, and H. Lewy. On the Partial Difference Equations of Mathematical Physics. *Mathematische Annalen*, 100:32–74, 1928.
- [30] T. J. Craft, L. J. W. Graham, and B. E. Launder. Impinging jet studies for turbulence model assessment - II. An examination of the performance of four turbulence models. *International Journal of Heat and Mass Transfer*, 36:2685–2697, 1993.
- [31] T. J. Craft, B. E. Launder, and K. Suga. Development and application of a cubic eddy-viscosity model of turbulence. *International Journal of Heat and Mass Transfer*, 17:108–115, 1996.

- [32] Z. X. Cui, Y. Liu, and R. M. C. So. Lattice Boltzmann Simulation of Flow past a Sphere up to Reynolds Number of 300. *Computational Mechanics*, pages 5–10, 2004.
- [33] G. Damköhler. Der Einfluss der Turbulenz auf die Flammengeschwindigkeiten in Gasgemischen. *Z. f. Elektroch.*, 46:601–652, 1940.
- [34] D. de Zeeuw and K. G. Powell. An Adaptively Refined Cartesian Mesh Solver for the Euler Equations. *Journal of Computational Physics*, 104:56–68, 1993.
- [35] J. W. Deardorff. A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers. *Journal of Fluid Mechanics*, 41:463–480, 1970.
- [36] J. W. Deardorff. Three-dimensional numerical study of turbulence in an entraining mixed layer. *Boundary-Layer Meteorology*, 7:199–226, 1974.
- [37] E. R. Van Driest. Turbulent Drag Reduction Polymeric Solutions. Technical report, Proceedings of the American Institute of Aeronautics & Astronautics Aerospace Sciences Meeting, 1970.
- [38] M. Düsing, A. Sadiki, and J. Janicka. Towards a classification of models for the numerical simulation of premixed combustion based on a generalized regime diagram. *Combustion Theory and Modelling*, 10:105–132, 2007.
- [39] J. Eckstein, J. Y. Chen, C. P. Chou, and J. Janicka. Modeling of turbulent mixing in opposed jet configuration: One-dimensional Monte Carlo probability density function simulation. *Proceedings of the Combustion Institute*, 28:141–148, 2000.
- [40] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.
- [41] E. A. Ervin and F. H. Harlow. The Particle-in-Cell method for hydrodynamic calculations. Technical report, Los Alamos National Laboratory, 1957.
- [42] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional

- Complex Flow Simulations. *Journal of Computational Physics*, 161:35–60, 2000.
- [43] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag, 2002.
- [44] G. B. Folland. *Fourier Analysis and its application*. Brooks Cole, 1992.
- [45] H. Forkel. *Über die Grobstruktursimulation turbulenter Wasserstoff-Diffusionsflammen*. PhD thesis, Technische Universität Darmstadt, 1999.
- [46] B. Fornberg. Steady viscous flow past a sphere at high Reynolds numbers. *Journal of Fluid Mechanics*, 190:471–489, 1988.
- [47] M. Freitag. *On the Simulation of Premixed Combustion Taking Into Account Variable Mixtures*. PhD thesis, TU Darmstadt, Germany, 2007.
- [48] H. Gebing. *Numerische Simulation and topologisch-physikalische Analyse der instationären, dreidimensionalen, abgelösten Wirbelströmungen an einer Kugel und an Rotationsellipsoiden*. PhD thesis, Georg-August-Universität Göttingen, 1994.
- [49] P. Geipel. *Experimental Research on Turbulent Reacting Flows using Gaseous and Liquid Fuels*. PhD thesis, Imperial College London, 2009.
- [50] P. Geipel. Personal communication, 2012.
- [51] P. Geipel, K. H. H. Goh, and R. P. Lindstedt. Fractal-Generated Turbulence in Opposed Jet Flows. *Flow, Turbulence and Combustion*, 85:397–419, 2009.
- [52] M. Germano. Turbulence: the filtering approach. *Journal of Fluid Mechanics*, 238:325–336, 1991.
- [53] D. Geyer. *1D Raman/Rayleigh experiments in a turbulent opposed-jet*. PhD thesis, TU Darmstadt, 2004.
- [54] D. Geyer, A. Kempf, A. Dreizler, and 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*, 30:681–689, 2005.

- [55] D. Geyer, A. Kempf, A. Dreizler, and J. Janicka. Turbulent opposed-jet flames: A critical benchmark experiment for combustion LES. *Combustion and Flame*, 143:524–548, 2005.
- [56] D. Geyer, S. Omar, A. Ludwig, A. Dreizler, and J. Janicka. A comprehensive characterisation of turbulent counterflow diffusion flames with partial premixing. *VDI-Berlin*, 1750:435–440, 2003.
- [57] R. Ghias, R. Mittal, and H. Dong. A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics*, 225:528–553, 2007.
- [58] A. Gilmanov and F. Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207:457–492, 2005.
- [59] I. Ginzbourg and P. M. Alder. Boundary flow condition analysis for the three-dimensional lattice Boltzmann equation. *Journal of Physics II*, 4:191–214, 1994.
- [60] D. Goldstein, R. Handler, and L. Sirovich. Modeling no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105:354–366, 1993.
- [61] A. D. Gosman. Flow processes in cylinders. *Thermodynamics and gas dynamics of internal combustion engines*, 2:616–772, 1986.
- [62] Z. Han and R. D. Reitz. Turbulence Modeling of Internal Combustion Engines Using RNG k-e Models. *Combustion Science and Technology*, 106:267–295, 1995.
- [63] F. H. Harlow. Hydrodynamic problems involving large fluid distortion. *Journal of the ACM*, 4:137–142, 1957.
- [64] F. H. Harlow. The particle-in-cell computing method for fluid dynamics. *Methods in Computational Physics*, 3:319–343, 1964.
- [65] F. H. Harlow. PIC and its progeny. *Computer Physics Communications*, 48:1–10, 1988.

- [66] F. H. Harlow and J. E. Welsh. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *Physical Fluids*, 8:2182–2189, 1965.
- [67] D. C. Haworth. Large-Eddy Simulation of In-Cylinder Flows. *Oil & Gas Science and Technology*, 54:175–185, 1999.
- [68] D. C. Haworth and K. Jansen. Large-eddy simulation on unstructured deforming meshes. *Computers & Fluids*, 29:493–524, 2000.
- [69] X. He and G. D. Doolen. Lattice Boltzmann method on a curvilinear coordinate system: flow around a circular cylinder. *Journal of Computational Physics*, 134:306–315, 1997.
- [70] F. E. Hernández-Pérez, C. P. T. Groth, and Ö. L. Gülder. LES of a Bunsen Flame using the Thickened Flame Model. *Proceedings of the Combustion Institute - Canadian Section*, 2008.
- [71] C. W. Hirt, A. A. Amsden, and J. L. Cook. An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds. *Journal of Computational Physics*, 135:203–216, 1997.
- [72] M. Hishida, Y. Nagano, and Y. Morimoto. Structure of Pipe Flow Turbulence. Part 1. Characteristics of Velocity Fluctuations and Reynolds Shear Stress. *Japan Society of Mechanical Engineering*, 46:1455–1466, 1980.
- [73] D. Hurst and J. C. Vassilicos. Scaling and decay of fractal-generated turbulence. *Physical Fluids*, 19:035103, 2007.
- [74] G. Iaccarino and R. Verzicco. Immersed boundary technique for turbulent flow simulations. *Applied Mechanical Review*, 56:331–347, 2003.
- [75] A. Jackson and M. S. Campobasso. Shared-memory, distributed-memory, and mixed-mode parallelisation of a CFD simulation code. *Computer Science - Research and Development*, 26:187–195, 2011.
- [76] A. Jackson, F. Reid, J. Hein, A. Soba, and X. Saez. High Performance I/O. In *Euromicro Conference on Parallel, Distributed, and Network-Based Processing*, 2011.

- [77] M. Jaeger and M. Carin. The front-tracking ALE method: application to a model of the freezing of cell suspensions. *Journal of Computational Physics*, 179:704–735, 2002.
- [78] A. Jameson, T. J. Baker, and N. P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. *AIAA Journal*, 86:86–103, 1986.
- [79] T. A. Johnson and V. C. Patel. Flow past a sphere up to a Reynolds number of 300. *Journal of Fluid Mechanics*, 378:19–70, 1998.
- [80] S. A. Jordan. A large-eddy simulation methodology in generalized curvilinear coordinates. *Journal of Computational Physics*, 148:322–340, 1999.
- [81] K. Y. Kang and J. H. Baek. LDV measurements and analysis of tumble formation and decay in a four-valve engine. *Thermal and Fluid Science*, 11:181–189, 1996.
- [82] A. Kempf. *LES of Non-Premixed Turbulent Flames*. PhD thesis, TU Darmstadt, 2003.
- [83] A. Kempf, H. Forkel, J. Y. Chen, A. Sadiki, and J. Janicka. Large-Eddy simulation of a counterflow configuration with and without combustion. *Proceedings of the Combustion Institute*, 28:25–40, 2000.
- [84] A. Kempf, M. Klein, and J. Janicka. Efficient Generation of Initial- and Inflow-Conditions for Transient Turbulent Flows in Arbitrary Geometries. *Flow, Turbulence and Combustion*, 74:67–84, 2003.
- [85] A. Kempf, R. P. Lindstedt, and J. Janicka. Large eddy simulation of a bluff-body stabilized nonpremixed flame. *Combustion and Flame*, 144:170–189, 2006.
- [86] A. M. Kempf, S. Wysocki, and M. Pettit. An Efficient, Parallel Low-Storage Implementation of Kleins Turbulence Generator for LES and DNS. *accepted: Computers and Fluids*, 2012.
- [87] A. R. Kerstein, W. Ashurst, and F. A. Williams. Field equation for interface propagation in an unsteady homogeneous flow field. *Physical Review A*, 37:2728, 1988.

- [88] B. Khalighi, S. H. El Tahry, and M. S. Huebler. Computation and measurement of flow and combustion in a four-valve engine with intake variations. *SAE Paper No. 950287*, 1995.
- [89] H. J. Kim and P. A. Durbin. Observations of the frequency in a sphere wake and of drag increase by acoustic excitation. *Physical Fluids*, 31:3260, 1988.
- [90] M. Klein, A. Sadiki, and J. Janicka. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulation. *Journal of Computational Physics*, 186:652–665, 2003.
- [91] A. N. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Proceedings Royal Society of London*, 434:9–13, 1991.
- [92] L. W. Kostiuk, K. N. C. Bray, and R. K. Cheng. Experimental study of premixed turbulent combustion in opposed streams. Part I - Nonreacting flow fields. *Combustion and Flame*, 92:277–295, 1993.
- [93] L. W. Kostiuk, K. N. C. Bray, and R. K. Cheng. Experimental study of premixed turbulent combustion in opposed streams. Part II - Reacting flow field and extinction. *Combustion and Flame*, 92:396–409, 1993.
- [94] R. H. Kraichnan. The Structure of Isotropic Turbulence at very high Reynolds Numbers. *Institute of Mathematical Sciences, New York*, 5:493–543, 1976.
- [95] P. K. Kundu and I. M. Cohen. *Fluid Mechanics*. Academic Press, 4th edition edition, 2008.
- [96] E. Kurosoy and J. H. Whitelaw. Opposed jets with small separations and their implications for the extinction of opposed flames. *Experimental Fluids*, 31:111–117, 2001.
- [97] M.-C. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160:705–719, 2000.
- [98] P. Lallemand and L.-S. Luo. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184:406–421, 2003.

- [99] A. Leonard. Energy Cascade in Large-Eddy Simulations of Turbulent Fluid Flows. *Advanced Geophysics*, 18:237–248, 1974.
- [100] D. K. Lilly. The representation of small-scale turbulence in numerical simulation experiments. *IBM Scientific Computing Symposium on Environmental Sciences*, pages 195–210, 1967.
- [101] D. K. Lilly. A proposed modification of the Germano subgrid-scale closure model. *Physical Fluids*, 4:633–635, 1992.
- [102] R. P. Lindstedt and D. S. Luff. Velocity fields of lean premixed turbulent opposed jet flames. *Proceedings of the Combustion Institute*, 31:1459–1466, 2007.
- [103] R. P. Lindstedt, D. S. Luff, and J. H. Whitelaw. Velocity and Strain-Rate Characteristics of Opposed Isothermal Flows. *Flow, Turbulence and Combustion*, 74:169–194, 2005.
- [104] R. P. Lindstedt and E. M. Vaos. Second moment modeling of premixed turbulent flames stabilized in impinging jet geometries. *Symposium (Int.) Combustion*, 27:957–962, 1998.
- [105] R. Löhner and P. Parikh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8:1135–1149, 1988.
- [106] R. Löhner and P. Parikh. Interactive generation of unstructured grids for three dimensional problems. *Numerical grid generation in computational fluid mechanics*, pages 687–697, 1988.
- [107] T. Lund, X. Wu, and K. Squires. Generation of turbulent inflow data for spatially developing boundary layer simulations. *Journal of Computational Physics*, 140:223–258, 1998.
- [108] L.-S. Luo. Analytic solutions of linearized lattice Boltzmann equation for simple flows. *Journal of Statistical Physics*, 88:913–926, 1997.
- [109] R. H. Magarvey and R. L. Bishop. Vortices in sphere wakes. *Canadian Journal of Physics*, 43:1649–1656, 1965.

- [110] B. F. Magnusson. On the structure of turbulence and a generalized eddy dissipation concept for chemical reaction in turbulent flow. In *Nineteenth AIAA Meeting, St. Louis*, 1981.
- [111] S. Majumdar, G. Iaccarino, and P. Durbin. RANS solvers with adaptive structured boundary non-conforming grids. *CTR Annual Research Brief, NASA Ames/Stanford University*, pages 353–366, 2001.
- [112] A. Mark and B. G. M. van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227:6660–6680, 2008.
- [113] E. Mastorakos. *Turbulent Combustion in Opposed Jet Flows*. PhD thesis, Imperial College, London, 1993.
- [114] E. Mastorakos, A. M. K. P. Taylor, and J. H. Whitelaw. Conditional scalar dissipation statistics in a turbulent counterflow. *Combustion and Flame*, 91:55–64, 1992.
- [115] D.J. Mavriplis. Adaptive mesh generation for viscous flows using triangulation. *Journal of computational Physics*, 90:271–291, 1990.
- [116] B. Mazzi and J. C. Vassilicos. Fractal-Generated Turbulence. *Journal of Fluid Mechanics*, 502:65–87, 2004.
- [117] D. M. McQueen and C. S. Peskin. A three-dimensional computational method for blood flow in the heart: (II) contractile fibers. *Journal of Computational Physics*, 82:289–297, 1989.
- [118] C. Meneveau and T. Poinso. Stretching and quenching of flamelets in premixed turbulent combustion. *Combustion and Flame*, 86:311–332, 1991.
- [119] S. Menon and W. H. Jou. Simulations of ramjet combustor flow fields Part. I Numerical model, large-scale and mean motions. *Nineteenth Fluid Dynamics, Plasma Dynamics, and Lasers Conference Honolulu, Hawaii, AIAA Paper*, page 87, 1987.
- [120] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32:1598–1605, 1994.

- [121] L. M. Milne-Thomson. *Theoretical Aerodynamics*. Dover Publications, 1966.
- [122] R. Mittal. Planar symmetry in the unsteady wake of a sphere. *AIAA Journal*, 37:388–390, 1999.
- [123] R. Mittal and G. Iaccarino. Immersed Boundary Methods. *Annual Review Fluid Mechanics*, 37:239–261, 2005.
- [124] J. Mohd-Yosuf. Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. *Annual Research Brief, Center for Turbulence Research*, pages 317–328, 1997.
- [125] A. P. Morse, J. H. Whitelaw, and M. Yianneskis. Turbulent flow measurements by Laser Doppler Anemometry in a motored reciprocating engine. Technical report, Imperial College London, 1978.
- [126] I. Nakamura. Steady wake behind a sphere. *Physics of Fluids*, 19:5, 1976.
- [127] R. Natarajan and A. Acrivos. The instability of the steady flow past spheres and disks. *Journal of Fluid Mechanics*, 254:323–344, 1993.
- [128] A. Obieglo, J. Gass, and D. Poulikakos. Comparative study of modeling a hydrogen nonpremixed turbulent flame. *Combustion and Flame*, 122:176–194, 2000.
- [129] C. Olbricht. *Numerische Berechnung technischer Verbrennungssysteme*. PhD thesis, TU Darmstadt, 2009.
- [130] C. Olbricht, O. T. Stein, J. Janicka, J. A. van Oijen, S. Wysocki, and A. M. Kempf. LES of lifted flames in a gas turbine model combustor using top-hat filtered PFGM chemistry. *Fuel*, 96:100–107, 2012.
- [131] C. W. Oseen. Über den Gültigkeitsbereich der Stokesschen Widerstandsformel. *Mathematische und Astronomische Physik*, 9:1–15, 1913.
- [132] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:1–49, 1988.

- [133] A. Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 2002.
- [134] C. S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of Computational Physics*, 10:220–252, 1972.
- [135] C. S. Peskin. The fluid dynamics of the heart valves: Experimental, theoretical and computational methods. *Annual Review Fluid Mechanics*, 14:235–259, 1982.
- [136] C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart: (I) immersed elastic fibres in a viscous incompressible fluid. *Journal of Computational Physics*, 81:372–405, 1989.
- [137] N. Peters. *Turbulent Combustion*. Cambridge University Press, 2000.
- [138] M. Pettit. Optimisation and Parallelisation of a Combustion Large-Eddy Simulation Code for High Performance Computation. Master’s thesis, University of Edinburgh, 2010.
- [139] M. W. A. Pettit, B. Coriton, A. Gomez, and A. M. Kempf. Large-Eddy Simulation and experiments on non-premixed highly turbulent opposed jet flows. *Proceedings of the Combustion Institute*, 33:1391–1399, 2010.
- [140] C. D. Pierce and P. Moin. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *Journal of Fluid Mechanics*, 504:73–97, 2004.
- [141] U. Piomelli and J. Liu. Large-eddy simulation of rotating channel flows using a localized dynamic model. *Physics of Fluids*, 7:839–848, 1994.
- [142] H. Pitsch. Large-Eddy Simulation of Turbulent Combustion. *Annual Review of Fluid Mechanics*, 38:453–482, 2006.
- [143] T. Poinso and D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2001.
- [144] T. Poinso, D. Veynante, and S. Candel. Quenching processes and premixed turbulent combustion diagrams. *Journal of Fluid Mechanics*, 228:561–606, 1991.

- [145] S. B. Pope. PDF methods for turbulent reactive flows. *Progress in Energy and Combustion Science*, 11:119–192, 1985.
- [146] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [147] M. M. Rai. Unsteady Three-Dimensional Navier-Stokes Simulations of Turbine Rotor-Stator Interaction. *AIAA Paper*, 87-2058, 1987.
- [148] M. M. Rai and P. Moin. Direct simulations of turbulent flow using finite-difference schemes. *Journal of Computational Physics*, 96:15–53, 1991.
- [149] M. Reeves, C. P. Garner, J. C. Dent, and N. A. Halliwell. Particle image velocimetry analysis of IC engine in-cylinder flows. *Optics and Laser in Engineering*, 25:415–432, 1996.
- [150] O. Reynolds. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Proceedings of the Royal Society of London*, 35:84–99, 1883.
- [151] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London*, 186:123–164, 1885.
- [152] L. F. Richardson. *Weather prediction by numerical process*. Cambridge University Press, 1922.
- [153] W. J. Rider and D. B. Kothe. A Marker Particle Method for Interface Tracking. *Proceedings of the Sixth International Symposium on Computational Fluid Dynamics*, 163:976–982, 1995.
- [154] J. C. Rolon, D. Vaynante, and J. P. Martin. Counter jet stagnation flow. *Experimental Fluids*, 11:313–324, 1991.
- [155] S. Roux, G. Lartige, T. Poinot, U. Meier, and C. Berat. Studies of mean and unsteady flow in a swirled combustor using experiments, acoustic analysis and large eddy simulations. *Combustion and Flame*, 141:40–54, 2005.
- [156] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer-Verlag, 1998.

- [157] E. M. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *Journal of Computational Physics*, 123:450–465, 1996.
- [158] H. Sakamoto and H. Haniu. A study on vortex shedding from spheres in a uniform flows. *ASME, Transactions, Journal of Fluids Engineering*, 112:386–392, 1990.
- [159] S. M. Salim and S. C. Cheah. Wall y^+ Strategy for Dealing with Wall-bounded Turbulent Flows. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2, 2009.
- [160] K. Sardi, A. Taylor, and J. H. Whitelaw. Conditional scalar dissipation statistics in a turbulent counterflow. *Journal of Fluid Mechanics*, 361:1–24, 1998.
- [161] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flows. *Annual Review Fluid Mechanics*, 31:567–603, 1999.
- [162] M. Schäfer. *Numerik im Maschinenbau*. Springer, 1999.
- [163] U. Schumann. Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli. *Journal of Computational Physics*, 18:376–404, 1975.
- [164] R. E. Seoud and J. C. Vassilicos. Dissipation of decay of fractal-generated turbulence. *Physics of Fluids*, 19:105108, 2007.
- [165] J. A. Sethian. *Theory, algorithms, and applications of level set methods for propagating interfaces*. Cambridge University Press, 1995.
- [166] J. A. Sethian. *Level set methods: evolving interfaces in geometry. Fluid mechanics, computer vision and material sciences*. Cambridge University Press, 1996.
- [167] J. Smagorinsky. General circulation experiments with the primitive equations - the basic experiment. *Monthly Weather Review*, 91, 1963.
- [168] V. Smiljanovski, R. D. Moser, and R. Klein. A capturing-tracking hybrid scheme for deflagration discontinuities. *Combustion Theory and Modelling*, 1997.

- [169] Y. Sommerer, D. Galley, T. Poinso, S. Ducruix, F. Lacas, and D. Veynante. Large eddy simulation and experimental study of flashback and blow-off in a lean partially premixed swirled burner. *Journal of Turbulence*, 5:N37, 2004.
- [170] D. B. Spalding. Development of the eddy-break-up model of turbulent combustion. *Symposium (Int.) Combustion*, 16:1657–1663, 1977.
- [171] J. H. Spurk. *Strömungslehre: Einführung in die Theorie der Strömungen*. Springer-Verlag, 1987.
- [172] G. Stan and D. A. Johnson. Experimental and numerical analysis of turbulent opposed impinging jets. *AIAA Journal*, 39 (10):1901–1908, 2001.
- [173] O. Stein. *Large Eddy Simulation of Combustion in Swirling and Opposed Jet Flows*. PhD thesis, Imperial College London, 2009.
- [174] O. Stein, B. Böhm, A. Dreizler, and A. M. Kempf. Highly-resolved LES and PIV Analysis of Isothermal Turbulent Opposed Jets for Combustion Applications. *Flow, Turbulence and Combustion*, 87:425–447, 2011.
- [175] S. H. El Tahry and D. C. Haworth. Directions in turbulence modeling for in-cylinder flows in reciprocating IC engines. *AIAA Journal of Prop and Power*, 8:1040–1048, 1992.
- [176] S. H. El Tahry and D. C. Haworth. A perspective on the state-of-the-art in IC engine combustion modeling. In *Sixth International Conference on Combustion*, 1996.
- [177] S. Taneda. Experimental investigation of the wake behind a sphere at low Reynolds numbers. *Journal of the Physics Society Japan*, 11:1104–1108, 1956.
- [178] L. Thobois, G. Rymer, T. Souleres, T. Poinso, and B. Van den Heuvel. Large-Eddy Simulation for the Prediction of Aerodynamics in IC Engines. *International Journal of Vehicle Design*, 39:368–382, 2005.
- [179] A. G. Tomboulides. *Direct and large-eddy simulation of wake flows: flow past a sphere*. PhD thesis, Princeton University, 1993.

- [180] A. G. Tomboulides and A. Orszag. Numerical investigation of transitional and weak turbulent flow past a sphere. *Journal of Fluid Mechanics*, 416:45–73, 2000.
- [181] S.R. Turns. *An Introduction to Combustion: Concepts and Applications*. McGraw-Hill, 1996.
- [182] H. S. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna. A sharp Interface Cartesian Grid Method for Simulating Flows with Complex Moving Boundaries. *Journal of Computational Physics*, 174:345–380, 2001.
- [183] H. S. Udaykumar, W. Shyy, and M. M. Rao. Elafint: A mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries. *International Journal for Numerical Methods in Fluids*, 22:691–705, 1996.
- [184] F. Unger. *Numerische Simulation turbulenter Rohrströmung*. PhD thesis, Technische Universität München, 1993.
- [185] J. C. Vassilicos and J. C. R. Hunt. Fractal dimensions and spectra of interfaces with application to turbulence. *Proceedings of the Royal Society of London, Series A: Mathematical and Physical Science*, 435:505–534, 1991.
- [186] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics - The Finite Volume Method*. Pearson Prentice Hall, 2007.
- [187] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth. LES in complex geometries using boundary body forces. *Proceedings of the Center for Turbulence Research Summer Program*, page 171, 1998.
- [188] J. A. Viecelli. A computing method for incompressible flows bounded by moving walls. *Journal of Computational Physics*, 8:119–143, 1971.
- [189] M. Wang. Dynamic wall modeling for LES of complex turbulent flows. *Center for Turbulence Research*, pages 241–250, 2000.
- [190] J. Warnatz, U. Maas, and R. W. Dibble. *Verbrennung*. Springer, 1997.
- [191] J. E. Welsh, F. H. Harlow, J. P. Shannon, and B. J. Daly. The MAC method. Technical report, Los Alamos National Laboratory, 1965.
- [192] J. S. Wu and G. M. Faeth. Sphere wakes in still surroundings at intermediate Reynolds numbers. *AIAA Journal*, 31:1448–1455, 1993.

- [193] S. Wysocki. Performance Measure and Optimisation of a Computer Tomography Code for High Performance Computing. Master's thesis, University of Edinburgh, 2010.
- [194] S. Wysocki, M. Pettit, P. Geipel, and A. Kempf. Fractal-Generated Turbulence in Opposed Jet Flows. *Proceedings Highly Resolved Experimental and Numerical Diagnostics for Turbulent Combustion Workshop*, 2011.
- [195] S. Wysocki, M. Pettit, P. Geipel, and A. Kempf. Fractal-Generated Turbulence in Opposed Jet Flows. *Proceedings European Combustion Meeting*, 2011.
- [196] Z. Z. Xu, S. L. Liu, and W. D. Gong. LDA measurement of turbulent flow field in a new type combustion chamber for developing the EQ6100-1 automobile engines. *Transactions of CSICSE*, 15, 1997.
- [197] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shy. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156:209–240, 1999.
- [198] D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussioletti. A locally refined rectangular grid finite element method: application to computational fluid dynamics and computational physics. *Journal of Computational Physics*, 92:1–66, 1991.
- [199] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.

Bibliography: Appendix MSc

- [1] P. G. Aleiferis, Y. Hardalupas, A. M. K. P. Taylor, K. Ishii, and Y. Urata. Flame chemiluminescence studies of cyclic combustion variations and air-to-fuel ratio of the reacting mixture in a lean-burn stratified-charge spark-ignition engine. *Combustion and Flame*, 136(1-2):72–90, 2004.
- [2] E. Beiting. Fiber-optic fan-beam absorption tomography. *Applied optics*, 31(9):1328–1343, 1992.
- [3] B. Böhm, C. Kittler, A. Nauert, and A. Dreizler. Diagnostics at high repetition rates: new insights into transient combustion phenomena. *Proceedings of the European Combustion Meeting*, pages 5–11, 2007.
- [4] N. Docquier, F. Lacas, and S. Candel. Closed-loop equivalence ratio control of premixed combustors using spectrally resolved chemiluminescence measurements. *Proceedings of the Combustion Institute*, 29(1):139–145, 2002.
- [5] J. Edwards, F. Gouldin, and M. MacDonald. High speed absorption tomography with advanced reconstruction algorithms. *41st Aerospace Sciences Meeting and Exhibit*, 2003.
- [6] G. Elsinga, F. Scarano, B. Wieneke, and B. van Oudheusden. Tomographic particle image velocimetry. *Experimental Fluids*, 41(6):933–947, 2006.
- [7] J. Feng, K. Okamoto, D. Tsuru, H. Madarame, and M. Fumizawa. Visualization of 3D gas density distribution using optical tomography. *Chemical Engineering Journal*, 86(3):243–250, 2002.
- [8] J. Floyd. *Computed Tomography of Chemiluminescence: A 3D Time Resolved Sensor for Turbulent Combustion*. PhD thesis, Imperial College, London, 2009.

- [9] J. Floyd, A.L. Heyes, and A.M. Kempf. Computed Tomography of Chemiluminescence (CTC): Instantaneous measurements of a matrix burner. *Proceedings of the European Combustion Meeting*, 2009.
- [10] B. Gillet, Y. Hardalupas, C. Kavounides, and A. M. K. P. Taylor. Infrared absorption for measurement of hydrocarbon concentration in fuel/air mixtures (methyl-liquid). *Applied Thermal Engineering*, 24(11-12):1633–1653, 2004.
- [11] A. G. Gordon. *The Spectroscopy of Flames*. Chapman and Hall, 1974.
- [12] R. Gordon. A tutorial on art (algebraic reconstruction techniques). *Nuclear Science, IEEE Transactions*, 21(3):78–93, 1974.
- [13] H. Guan and R. Gordon. A projection access order for speedy convergence of ART (algebraic reconstruction technique): a multilevel scheme for computed tomography. *Physics in Medicine and Biology*, 39, 1994.
- [14] Y. Ishino and N. Ohiwa. Three-dimensional computerized tomographic reconstruction of instantaneous distribution of chemiluminescence of a turbulent premixed flame. *JSME International Journal*, 48:34–40, 2005.
- [15] A. C. Kak and M. Slaney. *Principles of computerized tomographic Imaging*. IEEE Press, 1988.
- [16] C. Melvin. Design, development and implementation of a parallel algorithm for computed tomography using algebraic reconstruction technique. Technical report, University of Manitoba, 2006.
- [17] J. Nygren, J. Hult, M. Richter, M. Alden, M. Christensen, A. Hultqvist, and B. Johansson. Three-dimensional laser induced fluorescence of fuel distributions in an HCCI engine. *Proceedings of the Combustion Institute*, 29(1):679–685, 2009.
- [18] C. S. Panoutsos, Y. Hardalupas, and A. M. K. P. Taylor. Numerical evaluation of equivalence ratio measurement using OH^* and CH^* chemiluminescence in premixed and non-premixed methane-air flames. *Combustion and Flame*, 156:273–291, 2009.

- [19] Micheal Pettit. Optimisation and Parallelisation of a Combustion Large-Eddy Simulation Code for High Performance Computation. Master's thesis, University of Edinburgh, 2010.
- [20] D. Verhoeven. Multiplicative algebraic computed tomographic algorithms for the reconstruction of multidirectional interferometric data. *Optical Engineering*, 32(2):410–419, 1993.
- [21] S. Wysocki. Performance Measure and Optimisation of a Computer Tomography Code for High Performance Computing. Master's thesis, University of Edinburgh, 2010.
- [22] S. Wysocki, J. Floyd, and A. M. Kempf. Efficient Parallel Tomographic Reconstruction of Chemiluminescence in Flames. *Proceedings of the European Combustion Meeting*, 2011.