

**★ Computer Prediction of Chemically Reacting Flows in Stirred
Tanks**

by

**Harry John Ziman
M.A., AFIMA**

**Thesis submitted for the degree of
Doctor of Philosophy
of the University of London
and
for the Diploma of Membership
of Imperial College**

**Department of Mechanical Engineering
Imperial College of Science, Technology and Medicine
Exhibition Road, London SW7 2BX
July 1990**

Abstract

This study is concerned with extending and validating computational fluid dynamics techniques for calculating fluid flows, mixing and chemical reactions in stirred tank reactors.

The equations for mass and momentum are expressed in an Eulerian framework and a $k - \epsilon$ turbulence model applied. The formulation of equations describing general chemical reactions is discussed and a new technique based on reaction extent is devised. It is computationally efficient and easy to apply to complex reaction schemes. The influence of turbulence on reaction kinetics is modelled via a local micromixing time.

A considerable portion of the thesis is devoted to finite volume discretisation of the governing transport equations, especially the convection term. Existing methods either suffer from numerical diffusion which degrades the accuracy of the solution or permit unphysical extrema or oscillations to appear. A new non-linear discretisation method is proposed which has low numerical diffusion but does not violate boundedness. Algorithms for solving the resulting non-linear algebraic equations are also devised. Extensive tests demonstrate that the new method is suitable for the calculation of convection dominated flows.

The new discretisation technique is applied to the calculation of fluid flow in a stirred tank. Suitable solution algorithms and stirred tank boundary conditions are described. Comparisons of flow predictions with experimental data show that the new discretisation enables more accurate predictions to be obtained than were previously possible. The velocity field compares favourably with the experimental data but there are serious discrepancies in the turbulence predictions.

Calculated flow patterns are used in the calculation of the impeller power and the mean trajectory of infinitesimal fluid elements.

The additional equations describing mixing and chemical reaction are solved and the results compared with experimental data. Calculations of mixing in stirred tanks show that the technique is not currently capable of good quantitative predictions but does provide useful qualitative information. The results of reaction calculations compare more favourably with experimental data.

Preface

This thesis describes work I carried out at ICI as part of a collaborative project between ICI, Imperial College and Unilever under the auspices of a BRITE project sponsored by the Commission of the European Communities. The project involved the development of computational techniques for single phase Newtonian and non-Newtonian flow and two phase liquid-gas and liquid-solid flows in pipes and stirred tanks, the calculation of mixing and chemical reaction, heat and mass transfer and droplet and bubble breakup and coalescence in these flows. The project also included a complementary experimental program to provide experimental data. The material presented in this study forms only a small part of this project.

During the course of this study, I have referenced work carried out by my colleagues at ICI and not yet published in the open literature. Unfortunately, the relevant company reports are not publicly available so it is inappropriate to give a full reference. I have endeavoured to keep citations of this material to a minimum and they are referred to as private communications. I apologise to any readers who are inconvenienced by this.

This work owes much to my supervisor, Professor David Gosman, for his many suggestions and ideas, constructive criticisms and careful deliberations. Many of the ideas presented herein have developed from seeds that he has sown. I hasten to add that the blemishes are mine alone.

I would also like to thank my many friends and colleagues in ICI, Imperial College and Unilever for their support and encouragement during my studies. In particular I wish to express my gratitude to Kieran Looney, John Middleton and Ian Parker at ICI for their many efforts and continuous interest in this study. I am grateful to Peter Collins at Imperial College for making the results of his calculations available to me.

Finally, I would like to thank ICI for permitting me so much time to pursue this work and for their financial support.

H J Ziman
June 1990

Contents

	Page
Abstract	3
Preface	4
Contents	5
List of Tables	8
List of Figures	10
Notation	12
Abbreviations	20
Coordinate Systems	21
Mathematical Operators and Conventions	22
 Chapter 1: Introduction	 23
1.1 Introductory Remarks	23
1.2 Objectives	25
1.3 Details of the Present Contribution	25
1.4 Thesis Outline	26
 Chapter 2: Formulation of Governing Equations	 28
2.1 Introduction	28
2.2 Instantaneous Transport Equations	28
2.3 Turbulence Modelling Closure	29
2.4 Rationalisation of the Concentration Equations	33
2.5 Interaction of Mixing, Chemical Reaction and Turbulence	41
2.6 Synopsis of Equations	46
2.7 Closure	47
Appendix A.2: Analysis of Near Wall Region	49
 Chapter 3: Discretisation Techniques	 52
3.1 Selection of a Model Equation	52
3.2 The Finite Volume Method	52
3.3 Source Term Discretisation	56
3.4 Diffusion Term Discretisation	57
3.5 Accumulation Term Discretisation	57

3.6 Criteria for Convection Term Discretisation	58
3.7 Two Point Schemes	65
3.8 Extended Molecule Schemes	71
3.9 Flow Oriented Schemes	80
3.10 Flux Correction Techniques (Non-Linear)	88
3.11 Closure and Summary of Findings	93
Appendix A.3: Illustration of the Weaknesses of Taylor Series Analysis	95
Appendix B.3: FVM Discretisations on Non-Uniform Cartesian Meshes	98
Appendix C.3: Unconditionally Positive Contributions to Convection Discretisation Coefficients	101
Appendix D.3: Analysis of Integral Approximations in the FVM	102
 Chapter 4: Proposal of a New Discretisation Technique	 104
4.1 Self Filtered Central Differencing (SFCD)	105
4.2 Consistency	108
4.3 Conservation	110
4.4 Boundedness	110
4.5 Transportiveness	110
4.6 Invariance	110
4.7 Uniqueness	111
4.8 Cost-Effectiveness	111
4.9 Solution Algorithms	111
4.10 Summary	115
Appendix A.4: SFCD Coefficients	116
Appendix 4.B: Proof of Single Cell Uniqueness of SFCD Solution	118
Appendix C.4: A Conservative Solver	121
Appendix D.4: A Cyclic Penta-Diagonal Solver	124
 Chapter 5: Scalar Transport Tests of SFCD	 127
5.1 Introduction	127
5.2 One-Dimensional Convection-Diffusion	127
5.3 Two Dimensional Uniform Convection	132
5.4 Stagnation Flow in Corner	135
5.5 Time Dependent Translation in Solid Body Rotation Flow	139

5.6 Conclusions	142
Chapter 6: Stirred Tank Model	144
6.1 Separation into Two Sub-Problems	144
6.2 Grid Arrangements	144
6.3 Flow Calculation Procedure	145
6.4 Mixing Calculation Procedure	148
6.5 Application to a Stirred Tank	149
6.6 Numerical Parameters	154
6.7 Closure	154
Chapter 7: Stirred Tank Results	156
7.1 Introduction	156
7.2 Stirred Tank Calculations	156
7.3 Computational Results	159
7.4 Conclusions	168
Chapter 8: Calculation of Power Consumption and Flow Trajectories	169
8.1 Calculation of Power Consumption	169
8.2 Calculation of Flow Paths	172
Appendix A.8: Derivation of the Impeller Torque Formula	181
Appendix B.8: Properties of the Alternating Tensor	182
Appendix C.8: Mean Flow Path and Path from Mean Velocity	183
Chapter 9: Mixing and Chemical Reaction Calculations	185
9.1 Mixing Calculations	185
9.2 Reaction Calculations	197
9.3 Conclusions	200
Chapter 10: Conclusions	201
10.1 Summary of Achievements and Findings	201
10.2 Suggestions for Future Work	202
References	204

List of Tables

	Page
Chapter 1: Introduction	
Table 1.1.1: Relative Increase in Computer Time for a Given Error Reduction	24
Chapter 2: Formulation of Governing Equations	
Table 2.3.1: $k - \epsilon$ Model Coefficients	31
Table 2.6.1: Expressions for Transport Equations in Cylindrical Polar Coordinates	47
Chapter 3: Discretisation Techniques	
Table 3.7.1: Blending Factors for Two Point Interpolation	66
Table 3.8.1: Blending Factors for Extended Interpolation	72
Table 3.8.2: Previous Work Concerning QUICK	75
Table 3.9.1: Interpolation Regions for SUD	81
Table 3.9.2: Previous Work Concerning SUD	82
Table 3.9.3: Interpolation Regions for Hassan Scheme	84
Table 3.9.4: Interpolation Regions for VUD	85
Table 3.9.5: Inconsistency in CUPID Scheme	87
Chapter 4: Proposal of a New Discretisation Technique	
Table B.4.1: g_ϕ and g_w	118
Table B.4.2: g_ϕ and $dg_\phi/d\phi$	119
Table B.4.3: g_w and $dg_w/d\phi$	120
Chapter 5: Scalar Transport Tests of SFCD	
Table 5.4.1: Number of Iterations Required for SFCD Calculations	136
Table 5.5.1: Solid Body Rotation Timings	142
Chapter 6: Stirred Tank Model	
Table 6.5.1: ζ_w as a function of z^*	153
Table 6.6.1: Recommended Relaxation Factors in Flow Calculation	154

Chapter 7: Stirred Tank Results

Table 7.2.1: Dimensions of Yianneskis Mixing Vessel	156
Table 7.2.2: MIXFLO Computations Performed by Collins (1990)	157
Table 7.2.3: Comparison of Cost-Effectiveness of UD and SFCD	158

Chapter 8: Calculation of Power Consumption and Flow Trajectories

Table 8.1.1: Power Number Calculations (Method 3)	171
---	-----

Chapter 9: Mixing and Chemical Reaction Calculations

Table 9.1.1: Dimensions of Stirred Tank for Mixing Experiments	186
Table 9.1.2: Addition Point and Probe Locations for Mixing Experiment	187
Table 9.2.1: Dimensions of Stirred Tank for Reaction Experiments	198
Table 9.2.2: Predicted and Experimental Values of X_S	199

List of Figures

	Page
Chapter 2: Formulation of Governing Equations	
Figure 2.4.1: Schematic Representation of the Mixing Problem	39
Figure 2.5.1: The Three Stages of Micromixing	42
Chapter 3: Discretisation Techniques	
Figure 3.2.1: Finite Volume Cell Notation	53
Figure 3.6.1: Variation of ϕ with Position	62
Figure 3.9.1: Interpolation Regions for SUD	80
Figure 3.9.2: Interpolation Regions for Hassan Scheme	84
Figure 3.9.3: Interpolation Regions for VUD	85
Figure 3.9.4: Averaging volume for Kinney and Mahdi Scheme (West Face)	88
Figure A.3.1: Variation of Error Terms with Pe	96
Chapter 4: Proposal of a New Discretisation Technique	
Figure 4.1.1: Variation of R_c	105
Figure 4.1.2: Notation for Boundary Conditions	107
Figure 4.9.1: Compact Computational Molecule	112
Figure 4.9.2: Extended Computational Molecule	114
Chapter 5: Scalar Transport Tests of SFCD	
Figure 5.2.1: Solution to Back Diffusion Problem	128
Figure 5.2.2: Exact Solution to Runchal Test Problem	129
Figure 5.2.3: Errors in Runchal Test Problem Calculations	130
Figure 5.2.4: Sensitivity of the Error to r_{max}	131
Figure 5.2.5: Variation of the r Parameter in SFCD	132
Figure 5.3.1: Uniform Convection Problem	133
Figure 5.3.2: Centre Line Profiles for Unifor Convection Problem	134
Figure 5.3.3: r_{max} Sensitivity of the Centre Line Profile	134
Figure 5.4.1: Stagnation Flow Problem	135
Figure 5.4.2: Predicted Outflow Profiles for Stagnation Flow Problem	136
Figure 5.4.3: Variation of Maximum ϕ Value with r_{max}	138
Figure 5.4.4: Sensitivity in the Number of Iterations to r_{max}	138

Figure 5.5.1: Steadily Rotating Flow in a Cartesian Mesh	139
Figure 5.5.2: Profiles Through Results	140
Figure 5.5.3: Solid Body Rotation Results	141
Figure 5.5.4: Rotation of Slotted Cylinder	143
Chapter 6: Stirred Tank Model	
Figure 6.2.1: Computational Meshes and Storage Locations	145
Figure 6.5.1: Stirred Tank	149
Figure 6.5.2: Computational Mesh	150
Chapter 7: Stirred Tank Results	
Figure 7.3.1: MIXFLO Results (Velocity Vectors in Vertical Plane)	160
Figure 7.3.2: MIXFLO Results (Velocity Vectors in Horizontal Planes)	161
Figure 7.3.3: MIXFLO Results (Axial Velocity)	162
Figure 7.3.4: MIXFLO Results (Radial Velocity)	163
Figure 7.3.5: MIXFLO Results (Radial Velocity)	164
Figure 7.3.6: MIXFLO Results (Circumferential Velocity)	165
Figure 7.3.7: MIXFLO Results (Turbulent Kinetic Energy)	166
Chapter 8: Calculation of Power Consumption and Flow Trajectories	
Figure 8.1.1: Integration Volumes for Power Calculations	170
Figure 8.2.1: Projections of Flow Trajectory ($z_o = 7.3$ cm and $z_o = 8.3$ cm)	178
Figure 8.2.2: Projections of Flow Trajectory ($z_o = 9.3$ cm and $z_o = 10.3$ cm)	179
Figure 8.2.3: Projections of Flow Trajectory ($z_o = 14.3$ cm and $z_o = 26.3$ cm)	180
Figure C.8.1: Local Coordinate System for Flow about O	183
Chapter 9: Mixing and Chemical Reaction Calculations	
Figure 9.1.1: Schematic of Mixing and Reaction Experiments	186
Figure 9.1.3: Computed Probe Responses (Height 1.09m)	190
Figure 9.1.4: Computed Probe Responses (Height 0.55m)	191
Figure 9.1.5: Computed Probe Responses (Height 0.15m)	192
Figure 9.1.6: Flow Trajectory from Tracer Fluid Addition Point	193
Figure 9.1.7: Comparison of Computed and Experimental Probe Responses (90°)	194
Figure 9.1.8: Actual Probe Responses	195
Figure 9.1.9: Comparison of Computed and Experimental Probe Responses (45°)	196

Notation

The centre column is the section number in which the relevant quantity is first introduced. If a subscripted variable is not in this list then see Coordinate Systems. The letters A , B , C , D , E , F , R and S are also used in chapters 2 and 9 to denote chemical species. Concentrations are indicated by writing $[A]$ for the concentration of A etc.

A	3.2	Any discretisation coefficient
A_i, A'_i, A''_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
A_n	3.2	Discretisation coefficient (n specifies the node - see Coordinate Systems)
$A_{i,n}$	6.3	Discretisation coefficient in i 'th momentum equation (n specifies the node - see Coordinate Systems)
$A'_{i,p}$	6.3	Generalised central discretisation coefficient in i 'th momentum equation
$\bar{A}_p$	4.9	Under-relaxed central discretisation coefficient
$\bar{A}_{i,p}$	6.3	Under-relaxed central discretisation coefficient in i 'th momentum equation
A_n^c	3.2	Convection discretisation coefficient (n specifies the node - see Coordinate Systems)
A_n^{cB}	3.10	Convection discretisation coefficient from a bounded scheme (n specifies the node - see Coordinate Systems)
A_n^{cU}	3.10	Convection discretisation coefficient from an unbounded scheme (n specifies the node - see Coordinate Systems)
A_n^D	3.2	Diffusion discretisation coefficient (n specifies the node - see Coordinate Systems)
α_n	4.1	Non-linear SFCD convection discretisation coefficient (n specifies the node - see Coordinate Systems)
B	6.5	Width of baffle
B_i, B'_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
C_i, C'_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
C_k	2.2	Instantaneous concentration of k 'th chemical species
Co	3.5	Courant number
$C_{\epsilon 1}, C_{\epsilon 2}, C_\mu$	2.3	Turbulence model coefficients
$\underline{c}_0$	2.4	Mean concentration of chemical species in the absence of reaction
$\underline{c}_B$	6.5	Bulk fluid concentration vector
c_k	2.3	Mean concentration of k 'th chemical species

c_k'	2.3	Fluctuating component of concentration of k 'th chemical species
c_f	2.4	Chemical composition matrix for mixing streams
$\underline{c}_T$	6.5	Tracer fluid concentration vector
c^{fuel}	2.4	Concentration of fuel in an oxidising flow
c_0^{fuel}	2.4	Concentration of fuel in fuel stream
c^{ox}	2.4	Concentration of oxidant in an oxidising flow
c_0^{ox}	2.4	Concentration of oxidant in oxidant stream
c^{prod}	2.4	Concentration of oxidation products in an oxidising flow
D	6.5	Impeller diameter
D_d	6.5	Diameter of impeller disc
D_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
E	8.1	Total kinetic energy
E^*	2.3	Wall roughness factor in wall function
E_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
E_k	A.3	Error term in Taylor Series analysis
F_i	A.8	Component of body force
F^C	3.2	Convection flux through cell side
F^{CB}	3.10	Convection flux through cell side (bounded scheme)
F^{CU}	3.10	Convection flux through cell side (unbounded scheme)
F^D	3.2	Diffusion flux through cell side
F_i, F'_i, F''_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
f	2.4	Stream fraction or mixture fraction (two streams only)
f_I	2.4	Stream fraction of stream I
f_{II}	2.4	Stream fraction of stream II
$\underline{f}$	2.4	Generalised stream fraction
f'	2.5	Fluctuating stream fraction (two streams only)
G	2.3	Turbulent kinetic energy generation rate
G'_i, G''_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
g	3.6	Coefficient in dependent variable transformation
g_s, g_u	B.4	Functions in SFCD uniqueness investigation
H	B.4	Function in SFCD uniqueness investigation
H'_i, H''_i	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
H_L	6.5	Depth of liquid in the stirred tank
h	3.6	Coefficient in dependent variable transformation

h_e, h_w	A.4	Geometrical terms in SFCD on non-uniform meshes
I	2.3	Identity matrix
k	2.3	Turbulent kinetic energy
k_j	2.5	Rate constant for j 'th reaction
k_{up}	6.5	Impeller discharge level for turbulent kinetic energy
L_B	6.5	Length of impeller blade
l	3.2	Length scale of solution domain
l_B	2.5	Batchelor length scale
l_k	2.5	Kolmogorov length scale
m_{ji}	2.5	Mass action exponent for species j in reaction i
N	8.1	Impeller speed (revolutions per second)
$\underline{N}_B$	6.5	Vector of amount of chemical species in the bulk fluid in the vessel
$\underline{N}_T$	6.5	Vector of amount of chemical species in the tracer fluid in the vessel
n	2.4	Stoichiometric coefficient matrix
n_1, n_2, n_3	2.4	Stoichiometric coefficients
n_{ji}	2.4	Stoichiometric coefficient for species j in reaction i
n^{ox}	2.4	Number of moles of oxidant consumed per mole of fuel burnt
n^{prod}	2.4	Number of moles of oxidation products produced per mole of fuel burnt
n_r	2.4	Number of reactions
n_s	2.4	Number of streams
n_θ	6.5	Number of cells in circumferential direction
$\mathcal{P}$	2.5	Probability density function
P	2.2	Instantaneous pressure
P_0	8.1	Impeller power number
P_I	8.1	Impeller power
Pe	3.2	Peclet number
Pe_Δ	3.2	Grid Peclet number
Pe_Δ^{eff}	3.7	Effective grid Peclet number
p	2.3	Mean pressure (including turbulence pressure)
$\bar{p}$	2.3	Mean pressure (excluding turbulence pressure)
$\underline{R}$	8.2	Spatial coordinates of instantaneous flow path

R_{ϕ} etc	4.1	SFCD ratio (unfiltered)
R	3.10	Ratio in CONDIF
R^{ox}	2.4	Rate of oxidation
R_{max}	3.10	Bounding factor in CONDIF
$\underline{r'}$	8.2	Spatial coordinates of flow path (fluctuating component)
$\underline{r}$	8.2	Spatial coordinates of mean flow path
r	2.6	Radial coordinate in cylindrical coordinate systems
r_0	8.2	Starting radial coordinate in flow path calculation
r_{ϕ} etc	4.1	SFCD ratio (filtered)
r_{max}	4.1	Maximum value of filtered SFCD ratio
S	3.2	Surface of volume V
S_j	A.8	Component of outward normal of surface S
S'	B.4	Source term in SFCD uniqueness investigation
Sc	2.4	Schmidt number
Sc_k	2.4	Schmidt number for k 'th chemical species
Se	3.3	Explicit part of source term discretisation
Se'_i	6.3	Generalised explicit part of source term discretisation in i 'th momentum equation
$\mathcal{S}e$	4.9	Under-relaxed explicit part of source term discretisation
$\mathcal{S}e_i$	6.3	Under-relaxed explicit part of source term discretisation in i th momentum equation
Si	3.3	Implicit part of source term discretisation
Si_i	6.3	Implicit part of source term discretisation in i 'th momentum equation
s, s_{ϕ}	2.6, 3.1	Generalised source term
s_n	A.4	Area of n 'th face in control volume (n specifies the node - see Coordinate Systems)
s_P	3.2	Source term evaluated at P
T_I	6.5	Internal diameter of stirred tank
Tr_i	A.8	Component of impeller torque vector
Tr	8.1	Impeller torque
t	2.2	Time
t_0	8.2	Starting time in flow path calculation
$\underline{U}$	2.2	Instantaneous velocity vector

u	3.1	Mean velocity in x direction
u_*	A.2	Friction velocity
u^*	A.2	Dimensionless velocity in wall region
$\underline{u}'$	2.2	Fluctuating component of instantaneous velocity vector
$\underline{u}$	2.2	Mean velocity vector
u_r, u_θ, u_z	2.6	Cylindrical coordinate component of mean velocity vector
u_i	A.8	Mean velocity component in Cartesian coordinates
u_{res}	2.3	Resultant velocity parallel to the wall
u_z	2.6	Axial velocity component
$u_{z,e}^L$	6.3	Axial velocity used for east face mass flux in axial momentum equation
u_θ	2.6	Circumferential velocity component
V	3.2 (8.1)	Finite volume (also denotes the volume of integration when calculating impeller power)
V_B	6.5	Volume of bulk fluid in the vessel
V_P	3.2	Volume of finite volume cell
V_T	6.5	Volume of tracer fluid in the vessel
v	3.2	Mean velocity in y direction
v_{tip}	6.5	Impeller tip speed
W	6.5	Impeller blade height
X_{n-1}, X_n	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
X_S	9.2	Yield of chemical reaction
x	3.2	Spatial coordinate
x_i	A.8	Spatial coordinate (Cartesian)
y	2.3	Distance from wall in wall functions
Y_{n-1}, Y_n	D.4	Coefficient in PDMA algorithm (appendix D.4 only)
y	3.2	Spatial coordinate
y^*	2.3	Dimensionless distance from the wall
Z	6.5	Impeller clearance
z	2.6	Axial coordinate in cylindrical coordinate systems
z^*	6.5	Dimensionless axial coordinate in impeller
z_0	8.2	Starting axial coordinate in flow path calculation
z_{bot}	6.5	Axial position of bottom of impeller
z_{cl}	6.5	Impeller clearance

z_{top}	6.5	Axial position of top of impeller
α	C.3	Interpolation coefficient in two point convection interpolations
$\alpha(r, \theta, z)$	8.2	Arbitrary function in flow path theory
α_s	3.10	Blending factor in SMART scheme
α_t	D.4	Term in PDMA algorithm (appendix D.4 only)
Γ	3.1	Diffusion constant (physical)
Γ^{false}	3.7	Numerical diffusivity
Γ_k	2.2	Diffusivity of chemical species k (physical)
$\Gamma_{k,eff}$	2.3	Diffusivity of chemical species k (effective)
Γ_{eff}	2.4	Effective diffusivity of chemical species
$\Gamma_\diamond$	2.6	Generalised diffusivity
Δ	3.2	Mesh spacing (uniform mesh)
Δ_i	6.3	Difference operator in i th direction
Δt	3.5	Time step
Δx	1.1	Grid spacing
δ_{ij}	A.8	Kronecker delta (i.e. elements of identity matrix)
ϵ	2.3	Turbulent energy dissipation rate
ϵ_F	4.9	Final residual in solution algorithm
ϵ_I	4.9	Initial residual in solution algorithm
ϵ_{ijk}	A.8	Alternating tensor (defined in appendix B.8)
ϵ_{imp}	6.5	Impeller discharge level of turbulent kinetic energy dissipation rate
ζ_k	6.5	Scaling factor for turbulence intensity at impeller tip
ζ_w	6.5	Slip factor
ζ_ϵ	6.5	Scaling factor for dissipation rate at impeller tip
θ	4.9	Under-relaxation factor
Θ	2.6	Angular coordinate in cylindrical coordinate systems
θ_0	8.2	Starting angular coordinate in flow path calculation
κ	2.3	Von Karman constant
$\lambda, \lambda_1, \lambda_2,$	3.7	Discretisation blending factor
$\lambda_s, \dots$		
μ	2.2	Fluid viscosity

μ_t	2.3	Turbulent viscosity
μ_{eff}	2.3	Effective viscosity
ξ	3.7	Angle between flow and mesh
P	D.3	An arbitrary function of ϕ
ρ	2.2	Fluid density
ρ_ϕ	2.6	Generalised density
σ_{ij}	8.1.1	Stress tensor (including turbulent stresses)
σ_c	2.3	Turbulent Prandtl number for concentration
σ_k	2.3	Turbulent Prandtl number for turbulent kinetic energy
σ_ϵ	2.3	Turbulent Prandtl number for dissipation rate
τ_m	2.5	Overall micromixing time scale
τ_{macro}	2.5	Inertial-convective mixing time scale
τ_{micro}	2.5	Viscous-convective and viscous-diffusive mixing time scale
τ_R	2.5	Reaction time scale
τ_w	2.3	Wall shear stress
Φ	2.4	Strictly conserved scalar in reacting flow
Φ^{fuel}	2.4	Φ value of fuel stream in reacting flow
Φ^{ox}	2.4	Φ value of oxidant stream in reacting flow
ϕ	2.6, 3.1	Transported scalar
ϕ_e^{CD}	3.7	CD interpolation for east face value
ϕ_e^{SFCD}	4.2	SFCD interpolation for east face value
ϕ_e^{UD}	3.7	UD interpolation for east face value
ϕ_e^{exact}	3.7	Interpolated value of ϕ on east face using a differencing scheme
ϕ_e^{scheme}	3.7	Exact value of ϕ on east face
ϕ_P^C	3.10	Value of ϕ due to convection alone
ϕ_{max}, ϕ_{min}	3.10	Bounds on ϕ_P
X_k	2.2	Instantaneous rate of production of chemical species k by chemical reactions
$\bar{X}_k$	2.3	Mean rate of production of chemical species k by chemical reactions
$\bar{X}_k'$	2.3	Fluctuating component of the rate of production of chemical species k by chemical reactions
$\underline{X}$	2.4	Vector of mean rate of production of chemical species by reaction

Ω_j^{inst}	2.4	Instantaneous rate of j 'th reaction
Ω	2.4	Mean rate of chemical reaction
$\underline{\Omega}$	2.4	Vector of mean rate of chemical reaction
ω	2.4	Mean reaction extent
$\underline{\omega}$	2.4	Vector of mean reaction extent

Superscript

n	Time step level or iteration level
T	Transpose (of a matrix or tensor)
*, **, ***,	Updates during iterative solution algorithm
+, ++	

Abbreviations

ADI	Alternating Direction Implicit
BSUD	Bounded Skew Upwind Differencing
CD	Central Differencing
CFD	Computational Fluid Dynamics
CONDIF	COntrolled Numerical Diffusion with Internal Feedback
CPU	Central Processor Unit
CUPID	Corner UPwInDing
EFCA	Explicit Flux Correction Algorithm
FBBM	Flux Blending Bounding Methodology
FCT	Flux Corrected Transport
FRAM	Filter Remedy And Methodology
FVM	Finite Volume Method
HOT	Higher Order Terms
IFCA	Implicit Flux Correction Algorithm
LHS	Left Hand Side (of an equation)
LOADS	Locally Analytic Differencing Scheme
LUD	Linear Upwind Differencing
pdf	Probability Density Function
PDMA	Penta-Diagonal Matrix Algorithm
PISO	Pressure Implicit Split Operator
QUD	see QUICK
QUICK	Quadratic Upstream Interpolation for Convective Kinematics Also known as Quadratic Upstream Differencing (QUD)
RHS	Right Hand Side (of an equation)
SFCD	Self Filtered Central Differencing
SMART	Sharp and Monotonic Algorithm for Realistic Transport
SUD	Skew Upwind Differencing
TOUD	'Third' Order Upwind Differencing Also known as Uniformly Third-Order Polynomial Interpolation Algorithm (UTOPIA)
UD	Upwind Differencing
UPSTREAM	UPwinding in STREAMline direction
UTOPIA	See TOUD
VUD	Vector Upwind Differencing

Coordinate Systems

Cartesian Coordinates

Direction	Velocity component	Subscript for cell face and next two nodes in increasing direction	Subscript for cell face and next two nodes in decreasing direction
x	u	e, E, EE	w, W, WW
y	v	n, N, NN	s, S, SS
z	w	d, D, DD	u, U, UU
x_t	u_t		

Cylindrical Coordinates

Direction	Symbol	Velocity component	Subscript for cell face and next two nodes in increasing direction	Subscript for cell face and next two nodes in decreasing direction
Axial	z	u_z	e, E, EE	w, W, WW
Radial	r	u_r	n, N, NN	s, S, SS
Circumferential	θ	u_θ	d, D, DD	u, U, UU

The position of a discretised quantity can be determined by the subscript applied to it as given in the tables above. The central node in a cell is always labelled with a subscript P .

Sums over all the faces of a computational cell or its neighbouring nodes are denoted as

$$\sum X_m.$$

Mathematical Operators and Conventions

A number of mathematical operators and conventions are used for manipulating vectors, tensors and matrices in this work. This section briefly summarises these topics.

Vectors are used in a general sense in this work. Vectors are denoted by underscoring them (e.g. $\underline{u}$). In most cases a vector quantity has the conventional engineering meaning. That is to say that a vector is a quantity with a magnitude and direction. They are also used to represent a set of quantities when handling chemical reactions and concentrations. For example the set $\underline{c}$ is the set of mean concentrations of chemical species. The i 'th member of this set, c_i , is the concentration of the i 'th chemical species. The dimension of this set is the number of chemical species being considered. $\underline{c}$ should not be regarded as a vector in a physical space, but in a concentration space. 'Vectors' like $\underline{c}$ can be multiplied by matrices in a similar manner to the contraction of tensors and vectors.

The gradient operator, $\underline{\nabla}$, is treated as a vector quantity with components $\partial/\partial x_i$. It should be noted that $\underline{\nabla}$ does not commute with a vector or tensor unless the matrix or tensor is not a function of position.

Summation convention is used throughout this work. That is to say if a subscript is repeated twice in a term then the term is summed over all values of that repeated subscript (e.g. section 1.2 of Arfken, 1970). Summation convention does not apply to the 'compass notation' subscripts defined in Coordinate Systems above.

Operator	Definition
Dot product	$\underline{a} \cdot \underline{b} = a_i b_i$
Tensor product	$\underline{a} \otimes \underline{b} = a_i b_j$
Double contraction	$a : b = a_{ij} b_{ij}$
Transpose	$(a^T)_{ij} = a_{ji}$

Chapter 1: Introduction

1.1 Introductory Remarks

Fluid mechanics is at the heart of the process industry and is very important in many other branches of engineering. Virtually every piece of chemical processing equipment is concerned with the heating, mixing, dispersal, transport or simply storage of one or more fluids. These fluids may be liquids or gases. Indeed, some may be a combination of two fluids in the form of an emulsion and in other situations the fluid may consist of a solid phase suspended in a liquid. The liquids may have simple Newtonian properties or a complex history-dependent viscoelastic behaviour more typical of polymers which exhibit a wide range of behaviours. The flow of these fluids causes a variety of transport phenomena to take place including heat and mass transfer. The successful operation of any item of process equipment depends on understanding the fluid transport processes and the interaction with the flow itself. It is therefore not surprising that the study of fluid mechanics forms a substantial part of engineering research.

The study of fluid mechanics has been developed using experimental and theoretical approaches but the rapid advances in computer science has lead in recent times to the additional tool of computational techniques. The information that can be gathered from experimental studies is often limited due to a number of factors including the time and expense of executing an experimental program, difficulty of extrapolating from laboratory scales to pilot scale apparatus to full size plants, difficulty with working with process fluids due to toxicity or flammability and so on. The outcome of an experimental program is frequently a semi-empirical correlation which is usually poorly understood. The "constants" in the correlations are only established under certain geometrical or operational conditions and are certainly not universal with the consequence that it is often difficult to design a reactor for a new duty.

Theoretical studies of fluid mechanics are highly developed. The difficulty is that the mathematical formulations are complex and impossible to solve analytically for most reactors. This has necessitated the introduction of often sweeping assumptions about the fluid mechanics to facilitate further progress. For instance the calculation of chemical reaction yield may assume that the reactor is well mixed. The advantage of models of this type is that they are reasonably easy to formulate and can be readily solved by analytical or numerical means. The difficulty with them is their limited range of applicability. Simplifying assumptions about the fluid flow and transport processes cannot always be justifiably made.

The advances in computational science have made it feasible to numerically solve the equations governing fluid mechanics and therefore provide a means of direct determination of the fluid flow. This thesis is concerned with the development of these techniques which are known as Computational Fluid Dynamics (CFD). The advantages of a successful computational procedure compared to an experimental program are significant. Computations are relatively cheap to perform. The geometry, scale or operating conditions of the plant can be varied at will. In particular computer modelling techniques provide a direct method of scaling up from laboratory to semi-technical to full scale. Indeed, these calculations can be done as part of a feasibility study at the design stage. The upgrading of existing processing equipment can be done with the minimal interruption of production. There are no problems with handling hazardous materials or difficult measurements. Computations provide unrivalled detailed information that enables considerable insight into the equipment to be gained. The net benefits for the chemical industry is faster and more optimised development of new processes, shorter commissioning times and greater utilisation of existing plants.

The starting point of a CFD technique is the mathematical formulation of the equations governing the flow in a chemical reactor. This formulation will include equations for the transport of momentum, mass, chemical concentration and so on. Modelling approximations are often required to account for processes such as turbulence. The formulation is completed by appropriate boundary conditions.

The mathematical model obtained is a continuum model involving several partial differential equations. These must be discretised to replace these continuum equations with a finite number of algebraic equations which can then be solved computationally. The effectiveness of any resulting CFD programs is strongly influenced by the manner of the discretisation, particularly in three dimensions. This sensitivity can be illustrated in the following manner. Suppose the governing equations are discretised on a three dimensional mesh with spacing Δx . The total number of mesh points varies inversely with $(\Delta x)^3$. Let us optimistically assume that the computer time taken to solve the equations is proportional to the total number of mesh points. Table 1.1.1 compares the increase in computer time taken to reduce the error by factors of 10 and 100 for discretisations that are first and second order accurate. The advantage of second order methods over first order methods is immediately apparent. In practice the computer time taken to achieve a solution increases faster than the total number of mesh points so table 1.1.1 is likely to understate the advantage of second order accurate schemes when reducing numerical errors in the solution.

	Computer time ratio		
	1	10 fold error reduction	100 fold error reduction
First order	1	10^3	10^6
Second order	1	31.62	10^3

Table 1.1.1: Relative Increase in Computer Time for a Given Error Reduction

The discretisation of the governing equations results in a large number of non-linear coupled algebraic equations. An effective procedure for solving these equations is required. This solution algorithm needs to be efficient and reliable. That is to say that the algorithm must be capable of solving the equations for a variety of boundary and initial conditions.

The discretised mathematical model and solution algorithm can now be incorporated into one or more computer programs. The nature of the problems being solved will influence the design of these programs, in particular the user interfaces for the setting up of a calculation and the graphical presentation of the results. The former should include a method for generating computational grids. The latter is necessary to present the large amount of data generated by the CFD calculation in a comprehensible form.

The final step in the development of an engineering tool is the assessment of its performance by computational tests and comparison with experimental data. This will guide future development work and application of the technique to engineering problems. This assessment is an essential pre-requisite to taking engineering decisions based on the results of CFD calculations.

A practical and successful computational technique must have several attributes. The most important is that it is properly verified. The results of the model must be rigorously compared with experimental data to determine the accuracy of the technique and any limitations in the models employed. These tests can also be used to determine the values of parameters in the modelling. The programs should be robust so that the user has a high degree of confidence that the results will be easy to obtain and are reasonably accurate. The program should also be efficient. Computations of this type can generate vast quantities of information (up to 200 million or more items of data) that the results can be presented to the user in a meaningful manner. Graphical presentation is an integral part of any computational procedure.

1.2 Objectives

In the framework of the foregoing, the present study has the following objectives.

- (i) To formulate a mathematical model describing turbulent chemically reacting flow in process equipment.
- (ii) To devise a numerical method for the solution of the proposed equations with attention paid to the discretisation of convection terms.
- (iii) To formulate suitable boundary conditions for a stirred tank reactor.
- (iv) To apply the methodology to the calculation of fluid flow in stirred tanks.
- (v) To use the techniques to investigate mixing and chemical reaction in stirred tanks.

The benefits of achieving these objectives will be to improve existing computational modelling techniques for chemical reactors, particularly stirred tank reactors. The use of a higher order differencing technique will significantly improve the cost-effectiveness of the procedure. The facility to handle complex reaction schemes in a simple, flexible and efficient manner will make it easy to apply the technique to new reaction mechanisms. Previously this has been either inefficient or required significant program development.

Before the technique can be applied to industrial problems a careful assessment of its performance is required. This study will explore the strengths and weaknesses of the proposed model.

1.3 Details of the Present Contribution

This work contributes to many aspects of CFD techniques in general and computational methods for modelling of stirred tanks in particular. The primary contributions are listed below.

- (i) Mathematical formulation of frameworks for mixing and chemical reaction.

Practical CFD techniques for investigation of chemical reaction yields for use in the chemical industry must include efficient and flexible methods for handling a variety of chemical reactions. Careful consideration is therefore given to the mathematical formulation of this aspect of the problem. It is shown

that the solution of transport equations for each chemical species is not the most effective way of tackling this problem. An alternative framework is proposed that decouples the chemical reaction and mixing aspects of the problem providing an efficient and flexible structure.

The problem of the influence of turbulence on chemical reaction rate is considered. A simple approach based on calculating a time scale for mixing in the turbulent eddies is adopted for use in this work.

(ii) The discretisation of the convection term.

The largest single contribution of this work to CFD techniques is concerned with methods for discretising the convection term in the transport equations. Criteria for the selection of a discretisation are discussed and existing techniques reviewed in the light of these tests. They are found to be either inefficient and/or possess undesirable characteristics. A new scheme is proposed and analysed which meet the criteria identified previously. Scalar transport tests demonstrate the power of the new technique. It is further tested by calculations of the complex flow in stirred tanks. Analysis of the results and comparisons with experimental data show clearly the improvement in cost-effectiveness of the new technique and the accuracy of the procedure.

(iii) Computational procedures for stirred tanks.

This thesis describes methods for the calculation of flow patterns, mixing and chemical reaction yield in stirred tanks. This includes an account of the meshing of the tank, boundary conditions and algorithms to handle the non-linearities and couplings in the governing equations.

(iv) Applications of the computation of the flow in stirred tanks.

Two applications of the computation of the flow in stirred tanks are presented. The first is concerned with estimating the power supplied by the impeller. A new method of doing this is presented. This technique is less sensitive to the details of the impeller boundary conditions than those previously proposed. The second application is concerned with the visualisation of the flow fields in the tank. A method of calculating the mean trajectory of infinitesimally small volumes of fluid is devised.

(v) Calculation of mixing and chemical reaction in stirred tanks.

Computations of mixing and chemical reaction in stirred tanks are presented and compared with experimental data. This provides a severe test of the modelling, numerical methods and available computational resources. It is found that the technique provides useful information on mixing processes in stirred tanks and provides a method of estimating the yield of a simple mixing-sensitive chemical reaction scheme.

1.4 Thesis Outline

Chapter 2 is concerned with the formulation of the governing equations for the transport processes. The equations for the flow field itself are reasonably well established and therefore receive relatively little attention. The mathematical framework describing mixing and chemical reaction is examined in greater detail with a view to selecting an efficient and flexible formulation. The new approach based on the concepts of stream fraction and reaction extent is presented. Consideration is also given to the impact of concentration fluctuations induced by turbulence on chemical kinetics.

Chapter 3 reviews approaches to the discretisation of the transport equations with particular emphasis on the treatment of the convection term. A finite volume approach is adopted from the outset. Discretisations for the accumulation and diffusion terms are presented and techniques for handling source terms described. Criteria for the selection of a discretisation method for the convection term are developed and used in a detailed review of the existing literature. No totally satisfactory candidates are found.

Chapter 4 is concerned with the development of the new improved discretisation technique that addresses some of the limitations found in the literature review. The new scheme is analysed and tested against the criteria developed in the previous chapter. Algorithms for solving the resulting non-linear equations are also presented in this chapter.

Scalar transport tests of the new discretisation technique are presented in chapter 5. These tests demonstrate the superiority of the proposed technique over other popular schemes.

Chapter 6 describes the algorithms used to solve the transport equations given in chapter 2 with particular emphasis on a computational procedure for the calculation of the flow, mixing and chemical reactions in stirred tanks. Suitable boundary conditions for this type of equipment are presented.

Flow predictions obtained on meshes of 16,000, 39,000 and 87,000 nodes are analysed and compared with experimental data in chapter 7. The flow patterns in stirred tanks have been extensively investigated in detail by previous workers so attention is primarily paid to assessing the impact of the new discretisation technique on the predictions.

Chapter 8 is concerned with utilising the calculations of the flow field itself. One application (section 8.1) is the calculation of the power delivered by the agitator of a stirred vessel. Various methods of doing this are discussed and a new technique based on the angular momentum balance is selected and tested. Section 8.2 describes a different application involving the calculation of the mean trajectory of infinitesimal elements of the fluid. This is highly beneficial in visualising the complex three dimensional flow field in stirred reactors and provides a simple way of determining some quantitative predictions.

The mixing and reaction methodology is demonstrated in Chapter 9. Calculations of the mixing of a passive tracer quantity are presented and compared to corresponding experiments. The prediction of reaction yield is also undertaken. These problems are found to be severe tests of the modelling, numerical analysis and available computational resources.

Finally, conclusions are drawn in Chapter 10. This summarises the primary achievements of this study and highlights areas that require further attention.

Appendices containing essential details not central to the main arguments are to be found at the end of each chapter. Sections listing the notation, abbreviations and co-ordinate systems used in this work will be found near the front. The former also tabulates the section in which variables are first introduced to assist the reader to find their way around this work. Another brief section at the beginning of the thesis details some of the mathematical conventions and operators used. A list of the references cited will be found at the back.

Chapter 2: Formulation of Governing Equations

2.1 Introduction

The purpose of this chapter is to formulate the transport equations needed to model chemically reacting turbulent flow. We therefore require equations for the transport of momentum, mass and chemical reagents and these equations must be closed by turbulence and turbulent reaction kinetic modelling. The subject of chemically reacting flows is very broad. The objective of this work is particularly concerned with liquid phase reactions which enables a number of simplifications to be made.

(i) Compressibility

The fluid is assumed to be incompressible and constant density. Therefore phenomena such as shock waves and buoyancy are excluded, as are reacting gaseous flows where the density may depend on the local concentrations or temperature. Low concentration gaseous reacting flows and most reacting liquid flows are encompassed.

(ii) Turbulence Levels

The flow is assumed to be highly turbulent everywhere except in the immediate vicinity of the walls. Therefore standard high Reynolds number turbulence models and wall functions can be applied.

(iii) Thermal Effects

The flow conditions are assumed to be isothermal. Therefore it is not necessary to calculate the temperature field to determine the reaction rate. However this restriction can be relatively easily relaxed by incorporating heats of reaction in an enthalpy equation and temperature dependency into the reaction rate.

(iv) Interaction Between Reaction and Flow

The chemical reactions are assumed not to affect the flow field directly or indirectly. In particular physical properties such as viscosity do not depend on the concentration of chemical species.

The remainder of this chapter is organised as follows. The instantaneous transport equations are stated in section 2.2. They are closed by the $k - \epsilon$ turbulence model in section 2.3. Section 2.4 considers various options for the implementation of the concentration transport equations into a computational procedure. A new technique called the reaction extent method is derived. The interaction between turbulence and chemical reaction is discussed in section 2.5. Section 2.6 presents a synopsis of the mathematical framework in cylindrical coordinates. This is used in later chapters for calculations of flows in stirred tanks. Section 2.7 summarises the work presented in this chapter and appendix A.2 presents a brief derivation of the wall functions used.

2.2 Instantaneous Transport Equations

The instantaneous equations governing incompressible flow are well established (e.g. Hirsch, 1988). The continuity or mass conservation equation is

$$\nabla \cdot \underline{U} = 0 \quad (2.2.1)$$

where $\underline{U}$ is the instantaneous velocity. The instantaneous transport equation for momentum (the Navier-Stokes equation) in an incompressible Newtonian fluid is

$$\frac{\partial}{\partial t}(\rho \underline{U}) + \nabla \cdot (\rho \underline{U} \otimes \underline{U}) = -\nabla P + \nabla \cdot \{\mu(\nabla \otimes \underline{U} + (\nabla \otimes \underline{U})^T)\} \quad (2.2.2)$$

where μ is the viscosity and P the instantaneous pressure.

The conservation equation for a chemical species of concentration C_k , is

$$\frac{\partial C_k}{\partial t} + \nabla \cdot (\underline{U} C_k) = \nabla \cdot (\Gamma_k \nabla C_k) + X_k \quad (2.2.3)$$

where X_k is the rate of production of species k by chemical reaction. The formulation of this reaction rate is the subject of section 2.5.

2.3 Turbulence Modelling Closure

Equations of Motion

The instantaneous Navier-Stokes equations are capable of describing the motion of a fluid in both laminar and turbulent flow. However the computational cost of solving them for turbulent flows is prohibitive due to the very small time and length scales that would need to be resolved. Therefore a turbulence model is used to enable the calculation of the mean flows.

The conventional Reynolds averaging approach decomposes the instantaneous pressure and velocity into mean and fluctuating components.

$$\underline{U} = \underline{u} + \underline{u}' \quad (2.3.1)$$

$$P = \bar{P} + p' \quad (2.3.2)$$

The decomposition is defined such that

$$\bar{\underline{U}} = \underline{u}, \quad \bar{\underline{u}'} = 0 \quad (2.3.3)$$

and

$$\bar{P} = \bar{P}, \quad \bar{p'} = 0 \quad (2.3.4)$$

where an overbar denotes an averaged quantity. If the mean flow is steady then the averaging can be defined as either a time average or an ensemble average. These two methods of averaging are equivalent (Landau and Lifshitz, 1980, §1) in this case. Unsteady flows require ensemble averaging.

The time average of a quantity ϕ is defined as

$$\bar{\phi} = \lim_{(\tau \rightarrow \infty)} \frac{1}{\tau} \int_{\tau_0}^{\tau_0 + \tau} \phi(t) dt. \quad (2.3.5)$$

The ensemble average is the average of a quantity over all the different states available to it at any given point in time and space.

$$\bar{\phi} = \frac{1}{N_{states}} \sum_{n=1}^{N_{states}} \phi_n \quad (2.3.6)$$

The majority of experimentally determined means are time averages in steady flows. It is difficult to measure ensemble averages directly. The exception is in periodic flows where the flow repeatedly takes on the same mean state which enables the average (2.3.6) to be formed. These situations include flows in stirred tanks (which are of interest in this study) and other steadily rotating machinery.

The equations given in section 2.2 are unsteady and therefore ensemble averaging will be used unless otherwise stated.

Substitution of (2.3.1-4) into (2.2.1) and (2.2.2), averaging and manipulation produces

$$\underline{\nabla} \cdot \underline{u} = 0 \quad (2.3.7)$$

and

$$\frac{\partial}{\partial t}(\rho \underline{u}) + \underline{\nabla} \cdot (\rho \underline{u} \otimes \underline{u}) = -\underline{\nabla} \bar{p} + \underline{\nabla} \cdot \{ \mu (\underline{\nabla} \otimes \underline{u} + (\underline{\nabla} \otimes \underline{u})^T) \} - \underline{\nabla} \cdot (\rho \underline{u}' \otimes \underline{u}'). \quad (2.3.8)$$

The effect of this decomposition has been to preserve the original form of the flow equations but with the addition of an extra term into the momentum equations, namely $\underline{\nabla} \cdot (\rho \underline{u}' \otimes \underline{u}')$. This is the Reynolds stress term. It involves a correlation between fluctuating velocity components and therefore requires modelling. That is to say that one must either calculate this term directly or replace it with some form of approximation.

A standard $k - \epsilon$ model will be used in this work. More sophisticated models have been proposed but they have not been demonstrated as being better for the calculation of complex engineering flows such as stirred tanks (Bolour-Froushan, 1986).

The Reynolds stresses are modelled using the Boussinesq approximation in which they are assumed to obey a Newtonian type of stress-strain relationship.

$$\rho \underline{u}' \otimes \underline{u}' = -\mu_t (\underline{\nabla} \otimes \underline{u} + (\underline{\nabla} \otimes \underline{u})^T) + \frac{2}{3} \rho k I \quad (2.3.9)$$

Here I is the identity matrix, μ_t the eddy viscosity (to be defined later) and k is the turbulent kinetic energy.

$$k = \frac{1}{2} \overline{\underline{u}' \cdot \underline{u}'} \quad (2.3.10)$$

The momentum equation becomes

$$\frac{\partial}{\partial t}(\rho \underline{u}) + \underline{\nabla} \cdot (\rho \underline{u} \otimes \underline{u}) = -\underline{\nabla} \bar{p} + \underline{\nabla} \cdot \{ \mu_{eff} (\underline{\nabla} \otimes \underline{u} + (\underline{\nabla} \otimes \underline{u})^T) \} \quad (2.3.11)$$

where the effective viscosity μ_{eff} is given by

$$\mu_{eff} = \mu + \mu_t \quad (2.3.12)$$

and the modified pressure $\bar{p}$ by

$$\bar{p} = \bar{p} + \frac{2}{3} \rho k. \quad (2.3.13)$$

An equation for the transport of turbulent kinetic energy can be obtained by manipulation of (2.2.2). The closure of this equation is performed by (2.3.9) and a gradient diffusion assumption. The resulting transport equation is

$$\frac{\partial}{\partial t}(\rho k) + \underline{\nabla} \cdot (\rho \underline{u} k) = \underline{\nabla} \cdot \left(\left(\frac{\mu_t}{\sigma_k} + \mu \right) \underline{\nabla} k \right) + G - \rho \epsilon. \quad (2.3.14)$$

The first term on the RHS represents the diffusion of k by turbulence and σ_k is the turbulent Prandtl number for k . The second term stands for the generation of turbulence by the mean flow field and is given by

$$G = \mu_t (\nabla \otimes \underline{u} + (\nabla \otimes \underline{u})^T) : \nabla \otimes \underline{u} \quad (2.3.15)$$

The third term of (2.3.14) represents the dissipation of turbulent kinetic energy by viscous effects. The dissipation rate, ϵ , is defined as

$$\epsilon = \frac{\mu}{\rho} \overline{(\nabla \otimes \underline{u}') : (\nabla \otimes \underline{u}')}. \quad (2.3.16)$$

An equation for the transport of ϵ can also be derived from the momentum equations. The modelled form for high Reynolds number flows is

$$\frac{\partial}{\partial t}(\rho \epsilon) + \nabla \cdot (\rho \underline{u} \epsilon) = \nabla \cdot \left(\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \nabla \epsilon \right) + C_{\epsilon 1} G \frac{\epsilon}{k} - C_{\epsilon 2} \rho \frac{\epsilon^2}{k} \quad (2.3.17)$$

where σ_ϵ is the turbulent Prandtl number for dissipation and $C_{\epsilon 1}$ and $C_{\epsilon 2}$ are empirical constants.

The eddy viscosity is linked to k and ϵ via

$$\mu_t = C_\mu \rho \frac{k^2}{\epsilon}. \quad (2.3.18)$$

The values of the various empirical coefficients appearing in the turbulence model are the established universal set tabulated in table 2.3.1. These are the values recommended by Launder *et al* (1972) except that σ_ϵ has been modified (appendix A.2).

C_μ	$C_{\epsilon 1}$	$C_{\epsilon 2}$	σ_k	σ_ϵ
0.09	1.44	1.92	1.0	1.22

Table 2.3.1: $k - \epsilon$ Model Coefficients

Near Wall Region

The effects of the boundary layers close to walls are modelled using wall functions (Launder and Spalding, 1974) for the shear stress, turbulence generation and dissipation rates. The near-wall analysis is presented in appendix A.2. The results are as follows.

Calculate the dimensionless distance from the wall (y^*) from

$$y^* = \frac{C_\mu^{\frac{1}{4}} \rho k^{\frac{1}{2}} Y}{\mu} \quad (2.3.19)$$

where Y is the distance from the wall, κ is the von Karman constant (0.4187) and E^* is a wall roughness factor (9.793 for smooth walls). The shear stress is assumed co-linear with the velocity, $\underline{u}$. If y^* is greater than 11.63 then the wall shear stress, τ_w is given by

$$\tau_w = \frac{C_\mu^{1/4} \kappa \rho k^{1/2} u}{\ln\left(\frac{\epsilon \cdot C_\mu^{1/4} \rho k^{1/2} y}{\mu}\right)} \quad (2.3.20)$$

otherwise

$$\tau_w = \mu \frac{u}{Y}. \quad (2.3.21)$$

The turbulence generation term is related to the wall shear stress by

$$G = \tau_w \frac{u}{Y}. \quad (2.3.22)$$

Turbulence generation and dissipation are assumed to be in balance so the dissipation rate is set by

$$\epsilon = \frac{G}{\rho}. \quad (2.3.23)$$

It should be mentioned that these wall functions are based on one-dimensional theory and measurements, not complex three-dimensional flows. This point has been considered by Bolour-Froushan (1986) who concluded that the above expressions are not inappropriate providing the longitudinal wall curvature was not high and that y^+ was less than 500.

Transport Equations for Chemical Species

The species concentration and reaction rate in equation (2.2.3) can be decomposed into mean and fluctuating parts

$$C_k = c_k + c_k' \quad (2.3.24)$$

$$X_k = \chi_k + \chi_k' \quad (2.3.25)$$

such that

$$\overline{C_k} = c_k, \quad \overline{c_k'} = 0 \quad (2.3.26)$$

and

$$\overline{X_k} = \chi_k, \quad \overline{\chi_k'} = 0. \quad (2.3.27)$$

Substitution of equations (2.3.1) and (2.3.24-27) into the averaged form of (2.2.3) results in

$$\frac{\partial c_k}{\partial t} + \nabla \cdot (\underline{u} c_k) = \nabla \cdot (\Gamma_k \nabla c_k) + \chi_k - \nabla \cdot (\overline{u' c_k'}). \quad (2.3.28)$$

This equation is identical to the instantaneous equation except for an additional correlation between velocity and concentration fluctuations. This is modelled by a gradient diffusion law in the spirit of that used for the Reynolds stress (equation (2.3.9)).

$$\overline{u' c_k'} = - \frac{\mu_t}{\rho \sigma_c} \nabla c_k \quad (2.3.29)$$

Hence the transport equation (2.3.28) becomes

$$\frac{\partial c_k}{\partial t} + \nabla \cdot (\underline{u} c_k) = \nabla \cdot (\Gamma_{k,eff} \nabla c_k) + \chi_k \quad (2.3.30)$$

where

$$\Gamma_{k,eff} = \Gamma_k + \frac{\mu_t}{\rho \sigma_c} \quad (2.3.31)$$

The turbulent Prandtl number σ_c is taken as 0.7 following Jones and Whitelaw (1982). Other workers (e.g. Lynch, 1985; Liu and Barkelew, 1986) employed a value of 0.9, but this is more appropriate near a wall than in the bulk flow. The use of gradient diffusion laws for this closure must be treated with caution (e.g. Lumley, 1975; Libby and Bray, 1981). However the $k - \epsilon$ model itself has known deficiencies (Bolour-Froushan, 1986) for certain flows and there is little to be gained in using a more sophisticated closure for the concentration equations than for the momentum equations. Therefore (2.3.29) will be used in this study.

The formulation of χ_k is the subject of section 2.5.

2.4 Rationalisation of the Concentration Equations

The number of equations in the set (2.3.30) is equal to the number of chemical species present. However these concentrations are not independent of each other because they are linked by the processes of mixing and chemical reaction. Therefore the solution of every equation in the set (2.3.30) is not required. The purpose of this section is to examine ways to reduce the number of transport equations needed to determine the concentration of every chemical species.

This section is organised in the following manner. An order of magnitude analysis is performed on the terms in the effective diffusivity relationship (2.3.31) which suggests that the same value of Γ_{eff} can be used for each species. The inefficiency of solving every equation in the set (2.3.30) will be demonstrated by considering an example from the literature. The mixture fraction approach to reducing the number of equations will then be described. This is found to be relatively inflexible so a new method based on the concept of reaction extent is presented to overcome this. Its use is illustrated by applying it to a simple reaction scheme.

Equal Mobility Assumption

The diffusion term in the concentration transport equation (2.3.30) consists of two parts, molecular and turbulent diffusion. The Schmidt number, Sc , is the ratio between the diffusivity of momentum (i.e. μ/ρ) to the molecular diffusivity of chemical species.

$$Sc = \frac{\mu}{\rho \Gamma} \quad (2.4.1)$$

Therefore, from (2.3.31), the effective diffusivity of the k 'th species can be written as

$$\Gamma_{k,eff} = \frac{\mu}{\rho Sc} + \frac{\mu_t}{\rho \sigma_c} \quad (2.4.2)$$

Molecular diffusivity of chemical species in liquids is typically of order $10^{-9} \text{ m}^2\text{s}^{-1}$. The precise value depends on the chemical species and fluid involved and may vary with concentration and temperature. The exact value is unlikely to be known in most chemical engineering situations, but it is unlikely to be of much importance due to its small magnitude. This can be illustrated by comparing the order of magnitude of the two terms in (2.4.2) using water as the process fluid. The viscosity of water is 10^{-3} kg/m s and the density 10^3 kg/m^3 . Therefore the Schmidt number is of order 10^3 . In fully turbulent flow the eddy viscosity is at least an order of magnitude greater than the laminar viscosity. Further, since

σ_c is of order unity and the Schmidt number of order 10^3 , we see that the turbulent contribution to the effective diffusivity is overwhelming. Therefore the diffusion term is insensitive to the precise value of the diffusivity of chemical species.

The turbulent contribution to the effective diffusivity arises from the modelling of a convective process, and it is therefore independent of the chemical species. Since the molecular diffusivity is unimportant it is safe to assume that, at any point in the flow, the same value of effective diffusivity can be applied to every chemical species. This is the equal mobility assumption.

The equal mobility assumption enables the concentration transport equations to be written as

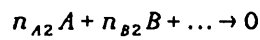
$$\frac{D'c}{D't} = \underline{\chi} \quad (2.4.3)$$

where the linear transport operator $D' / D't$ is defined as

$$\frac{D'}{D't} = \frac{\partial}{\partial t} + \underline{\nabla} \cdot \underline{u} - \underline{\nabla} \cdot \underline{\Gamma} \underline{\nabla}. \quad (2.4.4)$$

The concentrations and rates of reaction have been written in a vector notation (see Mathematical Operators and Conventions at the start of this work).

The chemical equation will be written in the form



...

where the elements of the stoichiometric coefficient matrix n are positive for a reagent in a reaction and negative for a product. Let the mean rate at which the m th reaction proceeds be Ω_m . That is to say that the rate of consumption of A in the m th reaction is $n_{Am}\Omega_m$. The rate of reaction of chemical species ($\underline{\chi}$) is related to the reaction rate by

$$\underline{\chi} = -n \cdot \underline{\Omega}. \quad (2.4.6)$$

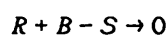
Substitution of (2.4.6) into (2.4.3) results in

$$\frac{D'c}{D't} = -n \cdot \underline{\Omega}. \quad (2.4.7)$$

This is the starting point of the rationalisation of the transport equations. However, it should be noted that the equal mobility assumption may not be appropriate in laminar flows with the result that (2.3.30) may have to be solved instead of (2.4.7) with (2.4.4).

Direct Calculations from Species Equations

The most direct approach to calculating concentrations of chemical species is to employ (2.4.7). Middleton *et al* (1986) used this method investigate yields of the reaction



in a stirred tank. A concentrated solution of B is added rapidly to a solution of A already in the vessel. Four concentration transport equations were solved, one for each species. However, only three equations are actually required to solve this problem as will be demonstrated at the end of this section. Additional computations for this reaction will be presented in chapter 9.

Liu and Barkelew (1986) presented computations for the reaction scheme



A and B are premixed and are introduced into a reaction chamber where the reaction takes place. Liu and Barkelew used six species equations to calculate the concentrations of the six chemical species, despite employing the equal mobility assumption. However, only three equations are actually needed. For instance transport equations could be solved for species D , E and F . The remaining species can be deduced from the stoichiometric relations

$$\begin{aligned} [A] &= [A]_0 - [D] - [E] - [F] \\ [B] &= [B]_0 - [D] - 2[E] - [F] \\ [C] &= [D] + 2[E] \end{aligned} \tag{2.4.10}$$

where $[A]$ denotes the concentration of A etc and $[A]_0$ and $[B]_0$ denote the inlet concentrations of A and B respectively.

This example illustrates that because there are usually many more chemical species present in a reaction scheme than there are reactions the solution of every equation in (2.4.7) is unnecessary and also inefficient. The solution of selected members of (2.4.7) and the use of algebraic relationships like (2.4.10) can overcome this, but this is difficult to implement in a computational procedure designed to tackle the general reaction (2.4.5). Further, when considering (2.4.5), the source term in every member of (2.4.7) involves every reaction rate which adds to the computational workload.

Mixture Fraction

The concept of mixture fraction (Gosman *et al*, 1969; Jones and Whitelaw, 1982) is frequently employed in the calculation of combustions flows. Its use will be illustrated here by means of mixing and reaction between a fuel and oxidant stream. The reaction is



Therefore three concentrations are to be determined. Let us assume that one mole of fuel is oxidised by n^{ox} moles of oxidant and produces n^{prod} moles of oxidation products. If R^{ox} is the rate of oxidation (which is assumed finite so fuel and oxidant can co-exist) we have

$$\frac{D'}{D't} c^{fuel} = -R^{ox} \tag{2.4.12}$$

$$\frac{D'}{D't} c^{ox} = -n^{ox} R^{ox} \tag{2.4.13}$$

$$\frac{D'}{D't} c^{prod} = n^{prod} R^{ox} \tag{2.4.14}$$

where c^{fuel} , c^{ox} and c^{prod} are the concentrations of fuel, oxidant and oxidation products respectively. It is easy to see from (2.4.12) and (2.4.13) that the quantity Φ given by

$$\Phi = c^{fuel} - \frac{c^{ox}}{n^{ox}} \quad (2.4.15)$$

satisfies the equation

$$\frac{D' \Phi}{D' t} = 0. \quad (2.4.16)$$

Φ is a strictly conserved quantity because there are no source terms in its transport equation.

Let Φ^{fuel} and Φ^{ox} denote the value of Φ in the fuel and oxidant streams. Then the mixture fraction f is defined by

$$f = \frac{\Phi^{fuel} - \Phi}{\Phi^{fuel} - \Phi^{ox}}. \quad (2.4.17)$$

This quantity also satisfies the equation

$$\frac{D' f}{D' t} = 0. \quad (2.4.18)$$

The mixture fraction has the value one in the oxidant stream and zero in the fuel stream and takes intermediate values where the two streams have mixed. It represents the degree of mixing between the two streams.

The definition of Φ given by (2.4.15) is not the only definition that leads to (2.4.18). Any strictly conserved scalar quantity can be used. The fuel and product concentrations can be combined to form another candidate.

$$\Phi = c^{fuel} + \frac{c^{prod}}{n^{prod}} \quad (2.4.19)$$

Substitution of this definition of Φ into (2.4.17) again leads (2.4.18). It is not difficult to see that the boundary conditions for f are identical for both definitions, and therefore the value of f is independent of the quantity Φ used to generate the mixture fraction.

Equations linking concentrations with mixture fraction can be obtained by substituting (2.4.15) and (2.4.19) into (2.4.17). These are

$$f = \frac{c_0^{fuel} - \left(c^{fuel} - \frac{c^{ox}}{n^{ox}} \right)}{c_0^{fuel} + \frac{c_0^{ox}}{n^{ox}}} \quad (2.4.20)$$

$$f = \frac{c_0^{fuel} - \left(c^{fuel} + \frac{c^{prod}}{n^{prod}} \right)}{c_0^{fuel}} \quad (2.4.21)$$

where c_0^{fuel} is the concentration of fuel in the fuel stream and c_0^{ox} is the concentration of oxidant in the oxidant stream. Hence a knowledge of the mixture fraction and any concentration enables the remaining concentrations to be deduced. Therefore we need to solve (2.4.18) together with one of (2.4.12-14).

Equations (2.4.20) and (2.4.21) also indicate that it is possible to calculate all three concentrations from the solution to two of the three species equations. However the mixture fraction equation has a simpler structure than the species equations and therefore the solution of this equation is to be preferred.

Looney (1987) applied the mixture fraction concept to more complex reaction schemes including a generalisation of (2.4.8). This required the solution of a mixture fraction equation and two concentration equations. However the complexity of the derivations of the formulae analogous to (2.4.20) and (2.4.21) increases as the complexity of the reaction increases. It is also difficult to apply this formulation to the general reaction scheme (2.4.5). Therefore an alternative approach that overcomes this problem without sacrificing efficiency will be sought.

Reaction Extent Approach

The chemical reactions taking place are defined by the stoichiometric coefficient matrix n . It is therefore desirable that the approach to reducing the number of transport equations is based on the properties of n instead of individual concentrations. This can be done by means of the concept of reaction extent. This is a standard concept in physical chemistry (e.g. Atkins, 1986, section 10.1(a)). It has been used by Georgopoulos and Seinfeld (1986) to simplify the calculation of a single reaction between two streams in the atmosphere. They did not give a formulation for the general chemical equation or for the mixing of more than two streams.

The presentation of the reaction extent method is organised as follows. The concept of the extent of chemical reaction is introduced and used to relate the concentration of chemical species to the concentration in the absence of reaction, $\underline{c}_0$. This enables a transport equation for reaction extent to be derived. Use of the concept of stream fraction provides a means of calculating $\underline{c}_0$. Therefore the solution of the equations for reaction extent and stream fraction provide a means of evaluating $\underline{c}$.

Reaction extent is the amount of chemical reaction that has taken place. It can be used to determine the amount of reagents consumed and products generated in a chemical reaction. For instance consider the reaction



Let us suppose that the extent of this reaction is ω moles. Therefore $n_1 \omega$ moles of A and $n_2 \omega$ moles of B have been consumed while $n_3 \omega$ moles of C have been produced.

ω is related to the rate of reaction Ω by

$$\omega = \int \Omega dt. \quad (2.4.23)$$

The general reaction (2.4.5) has one extent of reaction for each reaction in the chemical equation. This set of reaction extents is denoted by $\underline{\omega}$ and is related to the rate of reaction, $\underline{\Omega}$ by

$$\underline{\omega} = \int \underline{\Omega} dt. \quad (2.4.24)$$

The net amount of each chemical species that has reacted is $n \cdot \underline{\omega}$.

Let $\underline{c}_0$ be the concentration of the chemical species in the absence of any reaction. The actual concentration is $\underline{c}_0$ less the amount that has reacted, $n \cdot \underline{\omega}$.

$$\underline{c} = \underline{c}_0 - n \cdot \underline{\omega} \quad (2.4.25)$$

The transport equation for $\underline{c}_o$ is (2.4.3) without the reaction terms.

$$\frac{D' \underline{c}_o}{D' t} = 0 \quad (2.4.26)$$

Substitution of (2.4.25) into (2.4.7) and subtraction of (2.4.26) leaves

$$\frac{D'}{D' t} (n \cdot \underline{\omega}) = n \cdot \underline{\Omega}. \quad (2.4.27)$$

Therefore if $\underline{\omega}$ satisfies the equation

$$\frac{D' \underline{\omega}}{D' t} = \underline{\Omega} \quad (2.4.28)$$

then (2.4.27) will be satisfied.

The set of equations (2.4.28) contains one equation for each chemical reaction while the set (2.4.27) contains one equation for each chemical species. The number of chemical reactions is usually less than the number of chemical species, especially for simplified reaction schemes. Each equation in (2.4.28) contains a single reaction rate. The effect of the manipulation has been to decouple the reactions from each other.

Equation (2.4.26) is concerned with the mixing of the parent streams and contains one equation for each chemical species. This situation is shown schematically in figure 2.4.1. This number can be easily reduced by introducing the stream fraction, $\underline{f}$. This is defined such that the volume fraction of the fluid at any particular point that originated in the k 'th stream is the k 'th element of $\underline{f}$. This is analogous to the phase fraction in multi-phase flow. Every molecule in the fluid must originate from one of the streams so that

$$\sum_k f_k = 1. \quad (2.4.29)$$

The composition of the various streams is given by the matrix c_r . The k 'th column of c_r is the composition of the k 'th stream. The composition of the fluid in the absence of reaction at any given point is therefore given by the equation

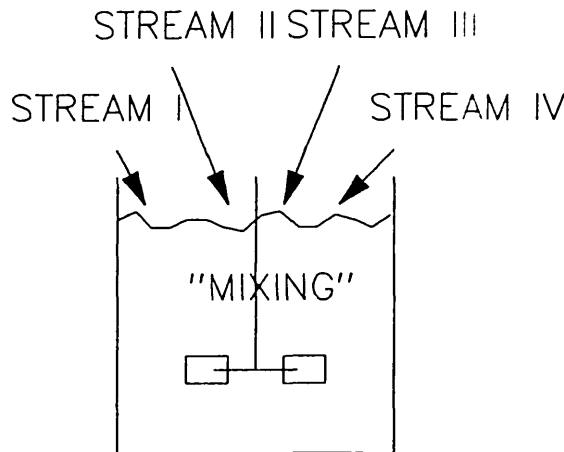


Figure 2.4.1: Schematic Representation of the Mixing Problem

$$c_o = c_f \cdot \underline{f}. \quad (2.4.30)$$

This is substituted into (2.4.26) to produce

$$\frac{D'}{D't}(c_f \cdot \underline{f}) = 0. \quad (2.4.31)$$

This will be satisfied if

$$\frac{D' \underline{f}}{D't} = 0. \quad (2.4.32)$$

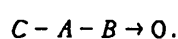
Equation (2.4.32) contains one equation for each of the streams being mixed. This number can be reduced by one by using (2.4.29) to obtain one of the elements of $\underline{f}$ from the others.

The reaction extent approach to reducing the number of equations in (2.4.7) therefore consists of forming the matrices n and c_f . Equation (2.4.28) is used to calculate the extents of reaction and (2.4.29) and (2.4.32) to calculate the stream fraction. The concentrations are evaluated by means of (2.4.25) and (2.4.30). The total number of equations in the set is

$$n_r + n_s - 1 \quad (2.4.33)$$

where n_r is the number of reactions and n_s is the number of streams. The number of equations is the same as proposed by Looney (1987) but this framework is general and can easily be applied to different chemical equations. This is because it only requires the two matrices n and c_f and the kinetic rate Ω to be specified. There is no need to select individual concentration equations or to derive algebraic relationships between various concentrations.

It should be noted that poorly posed problems can introduce extra degrees of freedom into the framework. Consider the simple chemical equation



The rows of n are linearly dependent. The concentration of the chemical species depends on the difference between the reaction extents for the forward and reverse reactions. We can therefore add a constant to both reaction extents without affecting the concentrations. Therefore the linear dependency of the rows of n has introduced one degree of freedom and we are solving for one more reaction extent than necessary. In the case of (2.4.34) the linear dependency can be eliminated by combining both reactions as



and allowing Ω to be positive or negative according to whether the overall reaction rate is in the forward or reverse direction.

Similarly, if the composition of one of the streams can be expressed as a linear combination of the compositions of the other streams (i.e. one of the columns of c_f is a linear combination of the other columns) then a degree of freedom is brought into the problem and an extra equation solved unnecessarily. This can be eliminated by regarding one of the streams as premixed from the other streams.

Application of Methodology to a Simple Reaction

The use of the reaction extent method will be illustrated by applying it to the reaction (2.4.8). The concentration vector is

$$\underline{c} = \begin{pmatrix} [A] \\ [B] \\ [R] \\ [S] \end{pmatrix} \quad (2.4.36)$$

where $[A]$ denotes the concentration of A etc. The stoichiometric matrix corresponding to (2.4.8) and (2.4.36) is

$$n = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ -1 & 1 \\ 0 & -1 \end{pmatrix}. \quad (2.4.37)$$

A small volume of a concentrated solution of B is instantaneously added to a dilute solution of A in a stirred vessel. The solution of B (concentration $[B]_0$) will be called stream I and the solution of A (concentration $[A]_0$) stream II. Therefore the matrix c_f is given by

$$c_f = \begin{pmatrix} 0 & [A]_0 \\ [B]_0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}. \quad (2.4.38)$$

A stream fraction equation will be solved for stream I. The stream fraction of the second stream (f_2) is given by

$$f_2 = 1 - f_1 \quad (2.4.39)$$

where f_1 is the stream fraction of the first stream.

Computations with this reaction will be performed in chapter 9.

2.5 Interaction of Mixing, Chemical Reaction and Turbulence

This section is concerned with the influence of mixing and turbulence on the rate of chemical reaction between two unpremixed streams. Chemical reaction between them will occur where they are brought in contact by mixing processes. Therefore the rate of reaction cannot be isolated from the rate of mixing, particularly in the turbulent eddies. This mixing process at small scales is called micromixing. The mixing due to the large scale motions in the reactor is referred to as macromixing or bulk mixing.

Macromixing is brought about by the large scale mean motions of the fluid in the reactor. This process is therefore incorporated in the equations for stream fraction. The time scale for this process will be denoted as τ_{macro} . Micromixing is due to turbulent fluctuations in concentration. A model for this process and its effect on the rate of chemical reaction is therefore required.

This section is organised as follows. Firstly some criteria for the selection of a model for the interaction of turbulence and chemical reactions will be given and then the process of micromixing is described. Various approaches to the modelling of this process are briefly surveyed. More details can be found in the reviews by Patterson (1981), Jones and Whitelaw (1982) and Baldyga and Bourne (1986). Finally a model for this process is selected.

Criteria for the Formulation of the Reaction Rate

For the purposes of this study, the following criteria for the modelling of the effects of turbulence and chemical reaction will be used.

- (i) The modelling must be a formulation for the local mean reaction rate. That is to say it must not be based on global mixing assumptions or macroscopic properties of the reactor.
- (ii) The model should be applicable to high Schmidt numbers.
- (iii) The resulting model must be computationally tractable when inserted into a CFD model of the whole reactor.
- (iv) The level of approximation in the model should be similar to those made in the other aspects of turbulence modelling.
- (v) The model should be applicable to different reaction schemes and kinetic rates.

The Mixing Process

The process of micromixing can be broken down into three stages (Baldyga and Bourne, 1986) which may take place sequentially or in parallel depending on the type of reactor. These are illustrated in figure 2.5.1.

The first stage is the inertial-convective stage. It is the reduction in the length scale of the segregation of the reacting streams from the macroscopic dimension of the reactor to the Kolmogorov length scale, l_k , given by

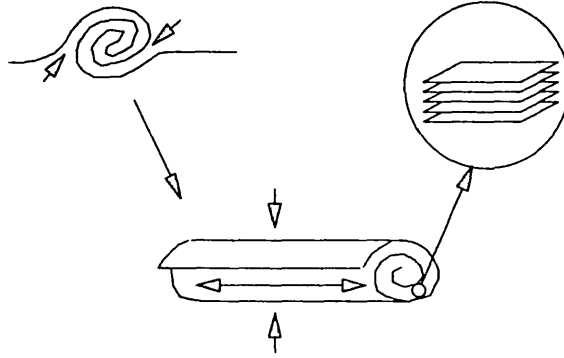


Figure 2.5.1: The Three Stages of Micromixing

$$l_k = \left(\frac{\mu^3}{\rho^3 \epsilon} \right)^{\frac{1}{4}} \quad (2.5.1)$$

This stage consists of the action of turbulent eddies of all sizes in reducing the length scale of the concentration fluctuations. The time scale for this process depends on the turbulence levels in the reactor.

One consequence of the high Schmidt number typical of flows in liquids is that the rate of diffusion over length scales of order l_k is too slow to bring about homogenisation. The stretching of eddies by the turbulence causes them to thin, reducing the length scale further. This is known as the viscous-convective stage of mixing. This phase operates from the Kolmogorov scale to the Batchelor scale, l_B , which is

$$l_B = \left(\frac{\Gamma^2 \mu}{\rho \epsilon} \right)^{\frac{1}{4}} \quad (2.5.2)$$

Viscous-convective mixing is a purely convective phenomena and does not involve physical diffusion. As the length scale is reduced to the Batchelor scale the concentration gradients become much larger and diffusion begins to become significant at the Batchelor scale. It is the dominant mixing mechanism over shorter distances and brings about mixing at the molecular level. The diffusive final step is known as the viscous-diffusive stage and chemical reactions occur during this phase of the mixing process.

The time scale for the viscous-convective and viscous-diffusive processes is denoted as τ_{micro} . It depends on the turbulent dissipation rate and the physical properties of the fluid. It does not depend directly on any macroscopic properties of the vessel.

Various categories of kinetic rate can be identified according to the time scale of the reaction kinetics, τ_R . Fast reactions are defined as those where $\tau_R \ll \tau_{micro}$. The actual kinetic rate of these reactions is rarely important because the rate of mixing is much slower than the rate of reaction. Intermediate reactions are those where the kinetic time scale is of a similar order to the mixing time scale and therefore neither mixing or kinetics control the reaction rate alone. Slow reactions ($\tau_R \gg \tau_{micro}$) are

sufficiently slow that turbulence does not affect their rate because micromixing is more rapid than the chemical reaction. Very slow reactions ($\tau_R \gg \tau_{macro}$) can be treated by elementary methods not requiring CFD techniques.

Instantaneous Reaction Rate

The instantaneous rate of chemical reaction of the i 'th species is assumed to obey the relation

$$\Omega_i^{inst} = k_i \prod_j C_j^{m_{ji}} \quad (2.5.3)$$

where a positive rate corresponds to the consumption of the reagents and generation of the products and vice versa. k_i is the rate constant for the i 'th reaction. The matrix m_{ji} gives the mass action coefficients for the reaction. Frequently they will be the same as the stoichiometric coefficients of the reagents and zero for products and any chemical species not involved in the reaction. The objective of this section is to consider ways of ensemble averaging (2.5.3) in turbulent flows.

Reynolds Decomposition

A possible starting point for the calculation of the mean reaction rate is to employ (2.3.24) to express C_j as the sum of mean and fluctuating concentrations. The resulting mean reaction rate is

$$\Omega_i = k_i \prod_j (\overline{C_j} + c_j')^{m_{ji}} \quad (2.5.4)$$

where

$$\Omega_i = \overline{\Omega_i^{inst}} \quad (2.5.5)$$

First order reactions are those where the reaction rate is proportional to the concentration of a single species.

$$\Omega_i^{inst} = k_i C_j \quad (2.5.6)$$

The use of the decomposition (2.3.24) leads to the expression

$$\Omega_i = k_i \overline{C_j} \quad (2.5.7)$$

for the mean reaction rate which shows that no turbulence closure is needed in this case.

For a second order reaction of the type



the average reaction rate is

$$\begin{aligned} \Omega_i &= k_i \overline{C_A C_B} \\ &= k_i \overline{C_A C_B} + k_i \overline{c_A' c_B'} \end{aligned} \quad (2.5.9)$$

with the consequence that the concentration fluctuation correlation $\overline{c_A' c_B'}$ needs to be modelled. It is this term that encompasses the micromixing process.

Higher order reactions generate increasing numbers of correlations which need modelling.

'Laminar' Approach

It has already been noted that the rate of slow reactions is not affected by micromixing. Therefore the fluctuating concentration correlation is unimportant for these reactions and may be neglected. That is to say that the mean concentration can be used in the mean reaction rate expression (2.5.4). This approach was used by Liu and Barkelew (1986) and Middleton *et al* (1986). The former work does not consider the effects of turbulence on the reaction rate while the latter notes that micromixing is not expected to influence the reaction yield and does not consider it further.

Correlation Modelling

Patterson (1981) has considered the modelling of $\overline{c'_A c'_B}$. Various ways of closing the equations are reviewed. One technique is to use a fast combustion or equilibrium assumption where the reaction between A and B in (2.5.8) is so fast that they cannot both exist at the same point at the same time. This considerably simplifies the problem. Gosman *et al* (1969) have shown that a single mixture fraction can be used to calculate the concentrations of the species in the oxidation reaction (2.4.11) under these circumstances. An additional concentration equation is not required.

Patterson (1981) also discusses higher order models involving transport equations for additional correlations. A transport equation for $\overline{c'^2}$ is derived by manipulating the species equation. This involves triple concentration fluctuation terms and velocity-concentration correlations. Additional approximations are required to close the equations and the system can rapidly become complex. For example up to eight equations are required to describe (2.5.8) when temperature fluctuations are included. The modelling of more complex reactions using this approach would be a formidable problem. Models of this type only partially meet criteria (ii), (iv) and (v).

Jones and Whitelaw (1982) discuss the probability distribution function (pdf) approach to the evaluation of reaction rates in combustions flows. The distribution of mixture fraction values is related to a probability density function (pdf), $\mathcal{P}$. The shape of $\mathcal{P}$ depends on the mean and variance of the mixture fraction. The variance is calculated from its own transport equation. The mean of any quantity ϕ that can be expressed in terms of the mixture fraction is calculated from

$$\bar{\phi} = \int_0^1 \mathcal{P}(f) \phi(f) df. \quad (2.5.10)$$

The pdf is typically a clipped Gaussian, β -function or a series of Dirac delta functions.

The difficulty with this approach for calculating reaction rates is that it is more complex when the fast reaction assumption cannot be invoked. Consider the reaction (2.5.8). Equations (2.4.20) and (2.4.21) show how the knowledge of one concentration and the mixture fraction enables the other concentrations to be deduced. In order to calculate the mean value of the product $C_A C_B$ we need a joint probability function $\mathcal{P}(f, C_A)$ for the distribution of concentration and mixture fraction values and therefore the problem becomes harder.

Global Micromixing Models

Global micromixing models of chemical reactors have been reviewed by Baldyga and Bourne (1986). These use macroscopic characteristics of the vessel such as the mean value of ϵ to develop theories of micromixing. The time required for bulk mixing is usually disregarded and the turbulence

levels in the reactor assumed homogeneous. The models concentrates on the reduction of the segregation ($\overline{c_i^2}$) of chemical species and the chemical reactions themselves. They are therefore unsuitable (criterion (i) above) for the modelling of the local reaction rate.

Mixing Time Scale Modelling

As noted above, the rate of fast reactions are independent of the kinetics themselves because they are mixing controlled. This suggests that the reaction rate can be calculated from a mixing time scale, τ_m , representing the time taken for the reagents to come into contact with each other. Corrsin (1964) used the turbulence spectrum to derive

$$\tau_m = \frac{Sc}{2Sc + 1} \left\{ 4 \frac{k}{\epsilon} + \left(\frac{\mu}{\rho \epsilon} \right)^{\frac{1}{2}} \ln Sc \right\}. \quad (2.5.11)$$

This expression is valid for high Schmidt numbers ($Sc \gg 1$). The first part of (2.5.11) is the inertial-convective time scale and the second part the viscous-convective and viscous-diffusive time scale, τ_{micro} . Pohorecki and Baldyga (1983) have revised the expression for τ_{micro} to

$$\tau_{micro} = 3.08 \left(\frac{\mu}{\rho \epsilon} \right)^{\frac{1}{2}} (\ln Sc - 1.27). \quad (2.5.12)$$

Looney (1987) used (2.5.12) and revised τ_m to

$$\tau_m = \frac{Sc}{2Sc + 1} \left\{ 4 \frac{k}{\epsilon} + 3.08 \left(\frac{\mu}{\rho \epsilon} \right)^{\frac{1}{2}} \ln Sc \right\}. \quad (2.5.13)$$

He then went on to assume that the reaction rate is controlled by the rate at which the species with lowest stoichiometric concentration is being micromixed. Therefore the reaction rate is given by

$$\Omega_i = \frac{1}{\tau_m} \min_j \left(\frac{c_j}{n_{ji}} \right). \quad (2.5.14)$$

Here n_{ji} is the stoichiometric coefficient for species j in the i 'th reaction and the minimum operator operates over the reagents of the reaction and not the products or inert species.

The advantage of this method is its simplicity and generality. That is to say that criteria (i) to (v) are all met to some extent. Its disadvantages are that it is rather crude and may not fully account for the selectivity of complex competing reaction schemes. It is restricted to fast reactions.

Looney (1987) proposed a solution to the latter difficulty by observing that the philosophy of this approach is to determine the reaction rate from the rate of the controlling process. Slower reactions (i.e. $\tau_R > \tau_m$) are controlled by the kinetics and therefore the 'laminar' method is appropriate. The generalisation of (2.5.14) to

$$\Omega_i = \min \left\{ k_i \prod_j c_j^{m_{ji}}, \frac{1}{\tau_m} \min_j \left(\frac{c_j}{n_{ji}} \right) \right\} \quad (2.5.15)$$

is applicable to both fast and slow reactions. Liu and Barkelew (1986) and Middleton *et al* (1986) ignored the interaction between turbulence and reaction rate in their calculations so even this simple model is a step forward for liquid flows.

The calculation of the yield of a chemical reaction depends on all the models applied and the accuracy of the numerical techniques used to solve the resulting equations. When comparing computational predictions with experimental data it is important to carefully assess all these factors before attempting to modify a single part of the procedure to improve the overall performance. It therefore may be necessary to reconsider the modelling of the interaction of turbulence and chemical reaction in the light of detailed comparisons with experimental data if it can be established that the above model is one of the principal limitations to the accuracy of the solution.

2.6 Synopsis of Equations

It is useful to summarise the mathematical formulation developed during this chapter and to express the transport equations in cylindrical polar coordinates. They will be used in this form later in the thesis for calculations of mixing and reaction in stirred tanks.

The transport equations (2.3.7), (2.3.11), (2.3.14), (2.3.17) and (2.4.32) can all be written in the form

$$\begin{aligned} \frac{\partial}{\partial t}(\rho_\phi \phi) + \frac{\partial}{\partial z}(\rho_\phi u_z \phi) + \frac{1}{r} \frac{\partial}{\partial r}(\rho_\phi u_r \phi) + \frac{1}{r} \frac{\partial}{\partial \theta}(\rho_\phi u_\theta \phi) = \\ \frac{\partial}{\partial z} \left(\Gamma_\phi \frac{\partial \phi}{\partial z} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r \Gamma_\phi \frac{\partial \phi}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial \theta} \left(\frac{1}{r} \Gamma_\phi \frac{\partial \phi}{\partial \theta} \right) + s_\phi \end{aligned} \quad (2.6.1)$$

where the generalised density (ρ_ϕ), diffusivity (Γ_ϕ) and source term (s_ϕ) are given in table 2.6.1. It should be noted that the continuity equation ($\phi = 1$) can be simplified to

$$\frac{\partial u_z}{\partial z} + \frac{1}{r} \frac{\partial u_r}{\partial r} + \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} = 0. \quad (2.6.2)$$

The coefficients in the turbulence model are listed in table (2.3.1). σ_c is 0.7.

The effects of wall boundary layers on the shear stress and turbulence generation and dissipation rates is modelled by equations (2.3.19) to (2.3.23).

The concentration of chemical species is calculated from the stream fraction and reaction extent fields by means of equations (2.4.25) and (2.4.30). This calculation involves the stoichiometric coefficients in the chemical equation (2.4.5) and the composition of the various streams being mixed. Advantage is taken of equation (2.4.29) to calculate one of the stream fraction components from the others, thereby reducing the number of transport equations by one.

The rate of chemical reaction is calculated from the rate of mixing brought about by the turbulence and the kinetics of the reaction itself. The expression used is (2.5.15)

Φ	ρ_Φ	Γ_Φ	S_Φ
1	1	0	0
u_z	ρ	μ_{eff}	$-\frac{\partial p}{\partial z} + \frac{\partial}{\partial z} \left(\mu_{eff} \frac{\partial u_z}{\partial z} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r \mu_{eff} \frac{\partial u_r}{\partial z} \right) + \frac{1}{r} \frac{\partial}{\partial \theta} \left(\mu_{eff} \frac{\partial u_\theta}{\partial z} \right)$
u_r	ρ	μ_{eff}	$-\frac{\partial p}{\partial r} + \frac{\partial}{\partial z} \left(\mu_{eff} \frac{\partial u_z}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(r \mu_{eff} \frac{\partial u_r}{\partial r} \right) + \frac{1}{r} \frac{\partial}{\partial \theta} \left(\mu_{eff} \frac{\partial u_\theta}{\partial r} \right)$ $+ \frac{\rho u_\theta^2}{r} - \frac{1}{r} \frac{\partial}{\partial \theta} \left(\mu_{eff} \frac{u_\theta}{r} \right) - 2 \frac{\mu_{eff}}{r^2} \frac{\partial u_\theta}{\partial \theta} - 2 \frac{\mu_{eff} u_r}{r^2}$
u_θ	ρ	μ_{eff}	$-\frac{1}{r} \frac{\partial p}{\partial \theta} + \frac{\partial}{\partial z} \left(\frac{\mu_{eff}}{r} \frac{\partial u_z}{\partial \theta} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(\mu_{eff} \frac{\partial u_r}{\partial \theta} \right) + \frac{1}{r} \frac{\partial}{\partial \theta} \left(\frac{\mu_{eff}}{r} \frac{\partial u_\theta}{\partial \theta} \right)$ $- \rho \frac{u_r u_\theta}{r} - \frac{1}{r} \frac{\partial}{\partial r} (\mu_{eff} u_\theta) + \frac{2}{r^2} \frac{\partial}{\partial \theta} (\mu_{eff} u_r) + \frac{\mu_{eff}}{r} \frac{\partial u_\theta}{\partial r}$ $- \mu_{eff} \frac{u_\theta}{r^2} + \frac{\mu_{eff}}{r^2} \frac{\partial u_r}{\partial \theta}$
k	ρ	$\frac{\mu_t}{\sigma_k} + \mu$	$G - \rho \epsilon$
ϵ	ρ	$\frac{\mu_t}{\sigma_\epsilon}$	$C_{\epsilon 1} G \frac{\epsilon}{k} - C_{\epsilon 2} \rho \frac{\epsilon^2}{k}$
f	1	$\frac{\mu_t}{\sigma_c} + \Gamma$	0
$\underline{\omega}$	1	$\frac{\mu_t}{\sigma_c} + \Gamma$	$\underline{\Omega}$

$$G = \mu_t \left\{ 2 \left(\frac{\partial u_z}{\partial z} \right)^2 + 2 \left(\frac{\partial u_r}{\partial r} \right)^2 + 2 \left(\frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \right)^2 + \left(\frac{\partial u_\theta}{\partial z} + \frac{\partial u_z}{\partial r} \right)^2 + \left(\frac{\partial u_\theta}{\partial z} + \frac{1}{r} \frac{\partial u_z}{\partial \theta} \right)^2 + \left(r \frac{\partial}{\partial r} \left(\frac{u_\theta}{r} \right) + \frac{1}{r} \frac{\partial u_r}{\partial \theta} \right)^2 \right\}$$

Table 2.6.1: Expressions for Transport Equations in Cylindrical Polar Coordinates

2.7 Closure

The purpose of this chapter is to formulate a set of equations to describe chemically reacting flow. The following tasks have been performed to do this.

- (i) The formulation of the instantaneous equations governing the flow and the transport of chemical species.
- (ii) The $k - \epsilon$ turbulence model was applied to average the instantaneous equations. Wall functions were applied to the calculation of shear stresses and turbulence generation and dissipation rates in the vicinity of walls.

- (iii) The equations for chemical species were manipulated to reduce the number of transport equations needed to calculate concentrations. A technique involving equations for the extent of reaction and stream fraction was presented. This provides a flexible mathematical framework for the implementation of different reactions.
- (iv) Consideration has been given to the averaging of the rate of chemical reaction. A simple method based on a local mixing time scale was selected.

Appendix A.2: Analysis of Near Wall Region

This appendix derives the standard wall function expressions and a relationship between some of the turbulence model constants. The flow near the wall is treated as a one-dimensional Couette flow and is regarded as consisting of two layers. Nearest to the wall is a linear layer in which viscous stresses dominate turbulent stresses. This is followed by a logarithmic layer in which turbulent stresses dominate. The model does not have an intermediate buffer region where viscous and turbulent stresses are both significant although, in reality, such a region must exist. The shear stress is constant across the whole boundary layer and is equal to the wall shear stress.

The first stage in the analysis is to cast velocities and distances into dimensionless form. This is done by means of the friction velocity u_* which is obtained from the wall shear stress τ_w

$$u_* = \sqrt{\frac{\tau_w}{\rho}} \quad (\text{A.2.1})$$

This enables the velocity u , which is assumed parallel to the wall and co-linear with the shear stress, and the distance from the wall, Y , to be cast into dimensionless forms by means of

$$u^* = \frac{u}{u_*} \quad (\text{A.2.2})$$

and

$$y^* = \frac{\rho u_*}{\mu} Y \quad (\text{A.2.3})$$

respectively.

Experiments have shown that within the linear sub-layer the Newtonian stress-strain relationship can be applied. In dimensionless variables this can be written as

$$u^* = y^* \quad (\text{A.2.4})$$

The turbulent sub-layer is found to have a logarithmic velocity variation (Landau and Lifshitz, 1959, §42).

$$u^* = \frac{1}{\kappa} \ln(E^* y^*) \quad (\text{A.2.5})$$

Here κ is the von Karman constant (0.4187) and E^* is a wall roughness factor (9.793 for smooth walls). By equating (A.2.4) and (A.2.5) it is found that the transition from linear to logarithmic layers takes place when y^* is 11.63.

It is further assumed that within the boundary layer the rate of production of turbulent kinetic energy is equal to the dissipation rate. That is to say that transport of turbulent kinetic energy perpendicular to the wall is unimportant. The transport equation for turbulent kinetic energy (equation (2.3.14)) therefore reduces to

$$G = \rho \epsilon \quad (\text{A.2.6})$$

The only significant velocity gradients are those normal to the wall so the generation term (equation (2.3.15)) can be taken as

$$G = \mu_t \left(\frac{\partial u}{\partial Y} \right)^2 \quad (\text{A.2.7})$$

in the fully turbulent logarithmic sub-layer. The shear stress in this layer is given by

$$\tau_w = \mu_t \frac{\partial u}{\partial Y}. \quad (\text{A.2.8})$$

The definition of the eddy viscosity (equation (2.3.18)) is used to eliminate ϵ from (A.2.6). (A.2.8) is used to eliminate the velocity gradient from (A.2.7). Equating the resulting two expressions for the generation terms leads to

$$\tau_w = C_\mu^{\frac{1}{2}} \rho k. \quad (\text{A.2.9})$$

From (A.2.1), (A.2.3) and (A.2.5) another expression for τ_w can be derived.

$$\sqrt{\tau_w} = \frac{u \kappa}{\ln(E^+ y^+)} \quad (\text{A.2.10})$$

These two expressions for the turbulent wall shear stress are combined by multiplying the LHS of (A.2.10) by the square root of the LHS of (A.2.9) and the RHS of (A.2.10) by the square root of the RHS of (A.2.9). The result is

$$\tau_w = \frac{C_\mu^{1/4} \kappa \rho k^{1/2} u}{\ln(E^+ y^+)}. \quad (\text{A.2.11})$$

Using (A.2.1), (A.2.3) and (A.2.9) this can be expressed as

$$\tau_w = \frac{C_\mu^{1/4} \kappa \rho k^{1/2} u}{\ln\left(\frac{\epsilon^+ C_\mu^{1/4} \rho k^{1/2} y^+}{\mu}\right)}. \quad (\text{A.2.12})$$

This is the standard expression for the turbulent shear stress wall function recommended by Launder and Spalding (1974).

The shear stress in the linear sub-layer is the usual laminar expression.

$$\tau_w = \mu \frac{u}{Y} \quad (\text{A.2.13})$$

The generation term is obtained by observing that in the logarithmic layer

$$G = \tau_w \frac{\partial u}{\partial Y}. \quad (\text{A.2.14})$$

This expression is also assumed to hold in the linear layer. The velocity gradient in this layer is u/Y so the expression for the generation term becomes simply

$$G = \tau_w \frac{u}{Y}. \quad (\text{A.2.15})$$

These expressions for the generation rate can be used to determine the dissipation rates from (A.2.6). An alternative expression for ϵ in the logarithmic layer can be obtained by combining (A.2.1), (A.2.2) and (A.2.5) to produce

$$\frac{\partial u}{\partial Y} = \sqrt{\frac{\tau_w / \rho}{\kappa Y}}. \quad (\text{A.2.16})$$

(A.2.16) is substituted into (A.2.8) along with (A.2.9) and the turbulent viscosity expression (2.3.18) and rearranged to yield

$$\epsilon = \frac{C_\mu^{3/4} k^{3/2}}{\kappa Y}. \quad (\text{A.2.17})$$

The procedure for calculating the shear stress and dissipation rate near the wall is as follows. Use (A.2.1), (A.2.3) and (A.2.9) to calculate y^+ from Y . If y^+ is greater than 11.63 then Y is in the logarithmic sub-layer, otherwise it is in the linear sub-layer. In the former case (A.2.12) is used to calculate the shear stress and (A.2.17) to calculate ϵ . In the linear sub-layer (A.2.13) is used for the shear stress and ϵ is obtained from (A.2.6) and (A.2.15).

It is now necessary to ensure that the modelling proposed in the logarithmic layer is consistent with the ϵ transport equation, (2.3.17). The flow is regarded as being steady so the time dependent term can be neglected. The gradient in ϵ is assumed to be perpendicular to the wall and therefore perpendicular to the velocity. Hence the second term in (2.3.17) is zero. The simplified equation is

$$\frac{\partial}{\partial Y} \left(\left(\mu + \frac{\mu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial Y} \right) + C_{\epsilon 1} G \frac{\epsilon}{k} = C_{\epsilon 2} \rho \frac{\epsilon^2}{k}. \quad (\text{A.2.18})$$

The laminar viscosity is neglected from (A.2.18) because it is usually small compared to its turbulent counterpart. (2.3.18), (A.2.6) and (A.2.17) are substituted into (A.2.18) to produce

$$\frac{C_\mu \rho}{\sigma_\epsilon} \frac{\partial}{\partial Y} \left(k^{1/2} Y \frac{\partial}{\partial Y} \left(\frac{k^{3/2}}{Y} \right) \right) + \frac{C_{\epsilon 1} C_\mu^{3/2} \rho k^2}{\kappa^2 Y^2} = \frac{C_{\epsilon 2} C_\mu^{3/2} \rho k^2}{\kappa^2 Y^2}. \quad (\text{A.2.19})$$

The derivatives of k are ignored. This can be justified from equation (A.2.9) because the shear stress is constant across the sub-layer. An alternative justification has been given by Landau and Lifshitz (1959, §42). After simplification we are left with

$$\frac{1}{\sigma_\epsilon} + \frac{C_{\epsilon 1} C_\mu^{1/2}}{\kappa^2} = \frac{C_{\epsilon 2} C_\mu^{1/2}}{\kappa^2} \quad (\text{A.2.20})$$

which can be rearranged as

$$\sigma_\epsilon = \frac{\kappa^2}{C_\mu^{1/2} (C_{\epsilon 2} - C_{\epsilon 1})}. \quad (\text{A.2.21})$$

Substituting values from table 2.3.1 we find that σ_ϵ is 1.22. Launder and Spalding recommend a value of 1.3. The former value will be used in this study.

Chapter 3: Discretisation Techniques

This chapter is concerned with methods for discretising the transport equations presented in chapter 2 and in particular the convection term. It is particularly concerned with three dimensional flows where the convection transport dominates the diffusive transport on the scale of a grid cell. The method for solving these equations once formed is given in the appendices to chapter 4.

The transport equations themselves are complex and detailed numerical analysis of each of them in their complete form is extremely difficult. However they all have several features in common and these are used to select a model equation (section 3.1). Section 3.2 introduces the finite volume discretisation method. The discretisation of the source, diffusion and accumulation terms are discussed in sections 3.3 to 3.5 respectively. The hardest term to discretise effectively is the convection term and numerous methods have been proposed in the literature. Section 3.6 identifies criteria for the selection of a suitable method. Sections 3.7 to 3.10 are concerned with reviewing techniques proposed in the literature and testing them against the previously identified criteria. A summary of the main findings of this chapter is presented in 3.11.

3.1 Selection of a Model Equation

Examination of the transport equations given in chapter 2 reveals that they all have a similar structure. Namely they all have terms representing the accumulation, convection, diffusion and source/sink of an intensive scalar quantity. One of the simplest equations of this type is

$$\frac{\partial \phi}{\partial t} + \underline{\nabla} \cdot (\underline{u} \phi) = \underline{\nabla} \cdot (\Gamma \underline{\nabla} \phi) + s. \quad (3.1.1)$$

The velocity, $\underline{u}$, satisfies the continuity equation.

$$\underline{\nabla} \cdot \underline{u} = 0 \quad (3.1.2)$$

Therefore, in one dimensional flow the velocity is constant.

Equation (3.1.1) will be used as a model equation for the examination of discretisation techniques. Although it is much simpler than the transport equations themselves it exhibits all the salient features that make the discretisation problematical. However a scheme that discretises (3.1.1) satisfactorily does not always perform well when solving the more complex Navier-Stokes equations. An example is the flux blending methodology of Peric (1986) which is described in section 3.10. Therefore (3.1.1) is used to analyse discretisation techniques but the review of the performance of discretisation schemes is not restricted to equation (3.1.1).

3.2 The Finite Volume Method

Equation (3.1.1) is a continuum equation which is to be solved by numerical approximation. The objective here is to formulate a finite set of algebraic equations that can be solved numerically such that the solution of these equations is an approximation to the solution of the continuous equation. The larger the number of equations the better the approximation.

This study is restricted to the family of techniques known as the Finite Volume Method (FVM) although some of the closely related finite difference discretisation approaches are inevitably included in the study. A review of the different approaches to discretisation has been given by Lai (1983). The principal advantage of the FVM method is that conservative discretisations arise naturally (see section

3.6 for a discussion on conservation). This is not so with the other discretisation approaches. The work of Peric (1986) and others has also shown that finite volume methods can be applied to arbitrarily shaped domains which were traditionally the territory of finite element methods. A considerable amount of algorithmic knowledge is also available for this family of discretisations.

Finite volume methods discretise (3.1.1) in the following manner. Equation (3.1.1) is integrated over an arbitrary (finite) volume V with surface S and the divergence theorem applied to obtain

$$\int_V \frac{\partial \phi}{\partial t} dV + \int_S \underline{u} \phi \cdot d\underline{S} = \int_S \underline{\Gamma} \nabla \phi \cdot d\underline{S} + \int_V s dV. \quad (3.2.1)$$

The volume of integration is arbitrary. The discretisation of the partial differential equation is achieved by dividing the solution domain into a number of 'finite volumes' and applying (3.2.1) over each volume. This most frequent manner of sub-division is to use a Cartesian mesh although other systems such as cylindrical polar systems can also be used. The choice depends on the geometry of the solution domain. Cartesian meshes will be used in this and the following two chapters during the examination of discretisation techniques. A cylindrical polar mesh will be used from chapter 6 onwards where attention is directed to calculations in stirred tank reactors.

The notation used is illustrated in figure 3.2.1. The finite volume V contains a node labelled P . The nodes W , E etc are in neighbouring finite volumes. The cell faces of V (lines in figure 3.2.1) are defined as being mid-way between the nodes. The point w lies on the cell face on the line joining P with W . Points are labelled on the other cell faces in a similar manner. The labels used in figure 3.2.1 are used to indicate the position at which variables are evaluated. For instance ϕ_e is the value of ϕ at the point e . The generalisation of this notation to three-dimensional systems and cylindrical polar systems is to be found at the beginning of this work.

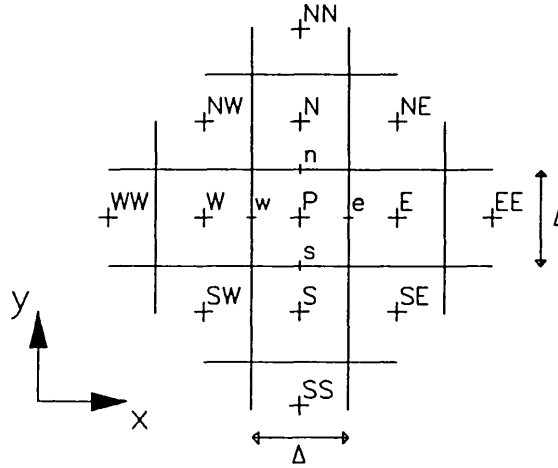


Figure 3.2.1: Finite Volume Cell Notation

For ease of exposition a uniform two-dimensional Cartesian mesh will usually be used in this and subsequent sections. The cell spacing is Δ . Appendix B.3 details the discretisations on non-uniform meshes of the schemes used in this study.

Application of the finite volume V defined in figure 3.2.1 to equation (3.2.1) results in

$$\begin{aligned} \int_{y_e}^{y_n} \int_{x_w}^{x_e} \frac{\partial \phi(x, y, t)}{\partial t} dx dy + \int_{y_e}^{y_n} u(x_e, y) \phi(x_e, y) dy - \int_{y_e}^{y_n} u(x_w, y) \phi(x_w, y) dy + \\ \int_{x_w}^{x_e} v(x, y_n) \phi(x, y_n) dy - \int_{x_w}^{x_e} v(x, y_s) \phi(x, y_s) dy = \\ \int_{y_e}^{y_n} \Gamma(x_e, y) \frac{\partial \phi(x, y)}{\partial x} \Big|_e dy - \int_{y_e}^{y_n} \Gamma(x_w, y) \frac{\partial \phi(x, y)}{\partial x} \Big|_w dy + \int_{x_w}^{x_e} \Gamma(x, y_n) \frac{\partial \phi(x, y)}{\partial y} \Big|_n dx - \\ \int_{x_w}^{x_e} \Gamma(x, y_s) \frac{\partial \phi(x, y)}{\partial y} \Big|_s dx + \int_{y_e}^{y_n} \int_{x_w}^{x_e} s dx dy. \end{aligned} \quad (3.2.2)$$

Discrete approximations are required for the integrals in equation (3.2.2). The 'volume' integrals can be regarded as being the product of the cell average of their integrands multiplied by the cell 'volume'. The cell averages of the integrand are approximated as the value of the integrand at P . This is second order accurate in the mesh spacing (appendix D.3). Hence

$$\int_{y_e}^{y_n} \int_{x_w}^{x_e} \frac{\partial \phi(x, y, t)}{\partial t} dx dy = V_P \frac{d\phi_P}{dt} + V_P O(\Delta^2) \quad (3.2.3)$$

and

$$\int_{y_e}^{y_n} \int_{x_w}^{x_e} s dx dy = V_P s_P + V_P O(\Delta^2) \quad (3.2.4)$$

where V_P is the area of the finite volume which is Δ^2 in this case.

The convection fluxes are similarly interpreted as being proportional to the average flux over the cell face. This is approximated by the flux at the points e , w , n and s . This is also a second order approximation (appendix D.3).

$$\begin{aligned} F_e^c &= \int_{y_e}^{y_n} u(x_e, y) \phi(x_e, y) dy \\ &= \Delta u_e \phi_e + O(\Delta^2) \end{aligned} \quad (3.2.5)$$

$$\begin{aligned} F_w^c &= \int_{y_e}^{y_n} u(x_w, y) \phi(x_w, y) dy \\ &= \Delta u_w \phi_w + O(\Delta^2) \end{aligned} \quad (3.2.6)$$

$$\begin{aligned} F_n^c &= \int_{x_w}^{x_e} v(x, y_n) \phi(x, y_n) dy \\ &= \Delta v_n \phi_n + O(\Delta^2) \end{aligned} \quad (3.2.7)$$

$$\begin{aligned} F_s^c &= \int_{x_w}^{x_e} v(x, y_s) \phi(x, y_s) dy \\ &= \Delta v_s \phi_s + O(\Delta^2) \end{aligned} \quad (3.2.8)$$

The velocities u_e, u_w, v_n and v_s satisfy the discretised continuity equation. Setting ϕ to unity and dropping the source and accumulation terms reduces (3.1.1) to the continuity equation. From (3.2.2) and (3.2.5-8) we obtain the discretised form as

$$\Delta u_e - \Delta u_w + \Delta u_n - \Delta u_s = 0. \quad (3.2.9)$$

The diffusive fluxes are approximated using a similar philosophy.

$$F_e^D = \int_{y_s}^{y_n} \Gamma(x_e, y) \frac{\partial \phi(x, y)}{\partial x} \Big|_e dy \quad (3.2.10)$$

$$= \Delta \Gamma_e \frac{\partial \phi}{\partial x} \Big|_e + O(\Delta^2)$$

$$F_w^D = \int_{y_s}^{y_n} \Gamma(x_w, y) \frac{\partial \phi(x, y)}{\partial x} \Big|_w dy \quad (3.2.11)$$

$$= \Delta \Gamma_w \frac{\partial \phi}{\partial x} \Big|_w + O(\Delta^2)$$

$$F_n^D = \int_{x_w}^{x_e} \Gamma(x, y_n) \frac{\partial \phi(x, y)}{\partial y} \Big|_n dx \quad (3.2.12)$$

$$= \Delta \Gamma_n \frac{\partial \phi}{\partial y} \Big|_n + O(\Delta^2)$$

$$F_s^D = \int_{x_w}^{x_e} \Gamma(x, y_s) \frac{\partial \phi(x, y)}{\partial y} \Big|_s dx \quad (3.2.13)$$

$$= \Delta \Gamma_s \frac{\partial \phi}{\partial y} \Big|_s + O(\Delta^2)$$

Combining (3.2.3-8) with (3.2.10-13) the approximated version of (3.2.2) becomes

$$V_P \frac{d\phi_P}{dt} + F_e^C - F_w^C + F_n^C - F_s^C = F_e^D - F_w^D + F_n^D - F_s^D + V_P s_P. \quad (3.2.14)$$

This is a second order accurate approximation to the original equation regardless of the manner in which the terms in (3.2.14) are actually evaluated. This implies that the use of third or higher order approximations for (3.2.3-8) and (3.2.10-13) will still result in a second order accurate solution.

The values of the transported scalar ϕ are stored at the cell nodes (P, E, W etc). Therefore some form of interpolation is needed to obtain values and gradients on the cell faces. The various FVM discretisations are distinguished according to the method used to do this, particularly for the convection flux. Methods for doing this are the subject of the remainder of this chapter and the following two chapters.

Finite difference methods also store ϕ at the nodes. However the discretisation is performed by directly differencing equation (3.1.1) rather than the integral form (3.2.1).

It will be frequently found that the fluxes are written in a linear or linearised form such that (3.2.14) becomes

$$V_P \frac{d\phi_P}{dt} + (A_P^C + A_P^D)\phi_P = \sum_m (A_m^C + A_m^D)\phi_m + V_P s_P \quad (3.2.15)$$

where the A s are known as discretisation coefficients. The sum is over selected nodes close to (but excluding) P involved in the evaluation of the convection and diffusion fluxes. It is not always necessary to distinguish between convection and diffusion discretisation coefficients in which case we use the equivalent notation

$$V_P \frac{d\phi_P}{dt} + A_P \phi_P = \sum_m A_m \phi_m + V_P s_P \quad (3.2.16)$$

where

$$A_P = A_P^C + A_P^D \quad (3.2.17)$$

and so on.

The Peclet number is frequently used to characterise the relative importance of diffusive and convective transport mechanisms. It is defined as

$$Pe = \frac{u l}{\Gamma} \quad (3.2.18)$$

where u is a typical velocity and l the length scale of the solution domain. The corresponding quantity for a finite volume is the grid Peclet number, Pe_Δ . This is defined as

$$Pe_\Delta = \frac{|u| \Delta}{\Gamma}. \quad (3.2.19)$$

A low value (less than unity) indicates that diffusion is the predominant transport mechanism on the grid scale while the converse applies for large values. It is the high grid Peclet number case that is of particular interest in this study.

Before studying the formulation of the convection term in detail brief accounts of the discretisation of the source, diffusion and accumulation terms are given.

3.3 Source Term Discretisation

The discretisation of the source term can appreciably influence the rate of convergence of the procedure used to solve the discrete equations, particularly when sources and sinks are significant in comparison with the transport terms. There is a very wide variety of source terms and a general analysis of their discretisation and the methods used to cope with them is impractical herein. However the usual approach is to linearise the source term as

$$V_P s_P = Si\phi_P + Se. \quad (3.3.1)$$

The linearisation is usually carried out in such a manner that Si is negative and Se is positive. Equation (3.2.16) is then written as

$$V_P \frac{d\phi_P}{dt} + A_P \phi_P - Si\phi_P = \sum_m A_m \phi_m + Se. \quad (3.2.16)$$

This assists with the diagonal dominance condition for stability (section 3.6).

3.4 Diffusion Term Discretisation

The majority of FVM discretisations use the same method for handling the diffusion term and which is described here. Unless stated otherwise later in the text all the schemes presented herein use the discretisation given below.

Attention is focussed on the east face diffusion flux in particular. The other faces and discretisation coefficients can be discretised by analogy. The standard form of the east diffusion flux has been given previously as

$$F_e^D = \Delta \Gamma_e \left. \frac{\partial \phi}{\partial x} \right|_e + O(\Delta^2). \quad (3.2.10)$$

The role of the discretisation method is to evaluate the partial derivative on the RHS. This is invariably done by assuming that ϕ varies linearly between P and E . This is central differencing. The discretised form is

$$F_e^D = \Delta \Gamma_e \frac{(\phi_e - \phi_P)}{\Delta}. \quad (3.4.1)$$

The discretisation coefficients are given by

$$A_e^D = \Gamma_e \quad (3.4.2)$$

$$A_w^D = \Gamma_w \quad (3.4.3)$$

$$A_P^D = \Gamma_e + \Gamma_w + \Gamma_n + \Gamma_s. \quad (3.4.4)$$

It also should be noted that

$$A_P^D = A_e^D + A_w^D + A_n^D + A_s^D \quad (3.4.5)$$

which applies on uniform and non-uniform meshes. The central differencing assumption of a linear variation between nodes used to obtain the partial derivative is a second order approximation (e.g. Leonard, 1981). However this is not always consistent with the profile assumed in the convection term discretisation.

An alternative approach to the discretisation of the diffusion flux would be to evaluate the gradient on the cell face by differentiating the variation of ϕ between P and E used for the discretisation of the convection term. For instance if the convection term is discretised using central differences (see section 3.7) then the profile is linear. Differentiation and substitution into (3.2.10) results in equation (3.4.1). However this approach is not possible with all convection discretisations. One example is upwind differencing (section 3.7) which assumes a piecewise continuous profile. The derivative would be either zero or infinite, neither of which is correct. Problems will arise with other upwinded schemes where the assumed profile depends on the sign of the normal velocity component at the cell face. If the velocity is zero then there is ambiguity regarding which of the two possible profiles should be used for the diffusion term. The practice of using (3.4.1) can be defended for convection dominated flows because of the relatively low significance of the diffusion terms compared to the convection terms.

3.5 Accumulation Term Discretisation

Combining (3.3.1) with (3.2.16) we have the following system of ordinary differential equations.

$$V_P \frac{d\phi_P}{dt} + (A_P - Si)\phi_P = \sum A_m \phi_m + Se \quad (3.5.1)$$

There is one equation for each finite volume. An important quantity in the discretisation of (3.5.1) is the Courant number, Co . It relates the time step (Δt) to the convection time scales for the mesh.

$$Co = \frac{|u| \Delta t}{\Delta}. \quad (3.5.2)$$

Hirsch (1988) considers methods for the solution of (3.5.1). Most methods require that the Courant number is less than some maximum value. For instance the explicit Euler method

$$\frac{V_p}{\Delta t} (\phi^{n+1} - \phi^n) + (A_p^n - Si^n) \phi_p^n = \sum A_m^n \phi_m^n + Se^n \quad (3.5.3)$$

is stable only if

$$Co \leq 1. \quad (3.5.4)$$

This condition can be very restricting, particularly on fine meshes when Δ is small. The result is that a large number of time steps are required which increases the computational cost. An exception to this is the backward Euler technique which has no Courant number restriction (Hirsch, 1988). The discrete equation is

$$\frac{V_p}{\Delta t} (\phi^{n+1} - \phi^n) + (A_p^n - Si^n) \phi_p^{n+1} = \sum A_m^n \phi_m^{n+1} + Se^n. \quad (3.5.5)$$

This equation is implicit in ϕ_p^{n+1} and therefore requires the solution of a set of linear equations in each time step. Methods for doing this efficiently are given in the appendices to chapter 4. However it is only first order accurate in time so the error introduced into (3.5.1) varies linearly with the size of the time step. Clearly a higher order temporal discretisation method without significant Courant number restrictions would be desirable. However, it has been argued in the introduction that the discretisation of the convection term is the most serious limitation on the numerical solution of (3.1.1) so this study concentrates on the convection term discretisation and equation (3.5.5) will be used for the accumulation term.

3.6 Criteria for Convection Term Discretisation

There are three principal requirements for any discretisation procedure.

- (i) In the limit of an infinitely fine mesh (i.e. $\Delta \rightarrow 0$) the set of discrete equations must reduce to the continuous equation. This is known as consistency.
- (ii) The properties of the solution of a discrete equation set of finite size should be the same as the properties of the solution to the continuous equation. If this criterion is not obeyed then it is possible that the numerical solution will contain physical phenomena that are not actually present or fail to predict features that exist in reality. Examples of such properties are conservation and boundedness. These are discussed below.
- (iii) The use of a discretisation technique should be a computationally efficient way of numerically approximating the solution to equation (3.1.1). In other words an acceptable representation to the solution to equation (3.1.1) should be obtained at an acceptable computational cost. This may entail a low accuracy but very cheap method applied on a fine mesh or alternatively a more expensive but more accurate technique on a coarser mesh.

The first two points are reasonably straightforward to assess. It is usually possible to examine the formulation of a scheme to ascertain if the solution meets these criteria. This is discussed in more detail below. The third point is not easy to test directly and analytical considerations are only indicative of the success or otherwise of computations. The best way to make assessments against this point is by doing trial calculations in circumstances similar to the intended application. Even this is not simple because one is testing not only the performance of the discretisation technique but of the algorithm used to solve the discrete equations. Further, account must be taken not only of the problem being solved but of the accuracy of solution needed.

The remainder of this section expands on the discussion above to explicitly identify criteria for the selection of a discretisation technique for the convection terms and to examine ways to test schemes against these criteria. Although the above guide-lines themselves are applicable to any method of solving equation (3.1.1) (e.g. finite difference, finite volume, finite element, spectral methods and so on), the discussion is restricted to the schemes in the finite volume and occasionally the finite difference categories.

Consistency and Taylor Series Analysis

Consistency requires that in the limit of an infinite number of discrete equations the solution of these equations should be the solution of the continuum equation. If a discretisation technique does not have this property then it is not solving the problem posed but some other equation. In practice this criterion is usually met. However it is often useful to reconstruct the continuous equation from the discrete version to examine the extra discretisation dependent terms that are present when using a finite number of points. This can help us to understand the numerical results from a particular scheme.

Taylor series analysis is usually used for the purpose of examining consistency. By this means discrete terms are related to their continuous counterparts and the governing equation is reconstructed. The converse method can also be applied to derive discrete counterparts to the continuous terms and hence derive discretisation techniques. However indiscriminate use of Taylor series suffers from two weaknesses which are discussed below. However it does indicate the response to grid refinement of a scheme and this can be quite useful.

The first weakness is that the analysis only applies in the limit of an infinite number of points. Taylor series analysis also assumes that the solution is continuous and infinitely differentiable. This will not be the case in the vicinity of shock waves. However this thesis is not concerned with this class of problems. It is customary to only retain the first few terms in a Taylor series analysis and to assume that later terms can be neglected. However when a finite number of points are being considered this is not always valid. Appendix A.3 gives an example where any term in the Taylor series expansion of the error can be made the largest. An example is also given where the first term in a Taylor error series is non-zero but the sum of all the terms is exactly zero.

The other weakness is that the Taylor series alone does not give the numerical properties of the discretisation technique. For instance a pure Taylor series analysis would indicate that the use of central differencing is always to be preferred to the less accurate upwind differencing (section 3.7) and that there is little to choose between upwinding and downwinding practices. Neither of these conclusions is correct.

It should be noted that the use of Taylor series techniques are not always easy to apply to discretisation procedures. In those cases where the discretisation is expressed in the form of an algorithm or procedure rather than as a set of equations it is very difficult to apply Taylor series. These techniques include FCT (Boris and Book, 1973; Book *et al*, 1975; Zalesak, 1979) and FBBM (Peric, 1986). They typically work by blending a low order bounded technique and a high order unbounded technique in a non-linear manner. The amount of blending is controlled by the local solution and is difficult to ascertain in advance. Consistency is then verified by examining the solution algorithm or procedure to ensure that it is consistent with the parent partial differential equation.

In conclusion we can say that Taylor series usually enables us to test the consistency of a discretisation scheme and to indicate likely features of the numerical solution and its response to mesh refinement in the limit $\Delta \rightarrow 0$. However it does not enable one to derive usable schemes of arbitrary accuracy and does not give definitive estimates of errors on finite meshes. Taylor series must be used with care.

Conservation

Equation (3.1.1) is a conservation law for the quantity ϕ . That is to say that the rate at which ϕ enters an arbitrary volume by convection and diffusion is equal to the rate of accumulation of ϕ and the source/sink of ϕ in that volume. One example of this principle is a batch stirred tank in which a reaction is taking place. ϕ could then be the concentration of a chemical species and we can say that the amount of ϕ initially present in the reactor is equal to the amount remaining less the amount that has reacted. We therefore require a similar property in our discretised form of (3.1.1).

Finite volume methods are based on this principle. Equation (3.2.1) is an alternative manner of expressing the conservation principle over a finite volume. The solution of (3.2.1) will therefore be conservative over the finite volume. However it is important to note that the conservation principle does not simply require that ϕ be conservative across a single finite volume, but that ϕ be conservative across adjoining volumes to ensure that the solution is globally conservative. Therefore the flux exiting one cell through any face must be identical to that entering the next cell through the same face. For example identical interpolation must be used for the east face of cell P and the west face of the adjoining cell E .

The solution algorithm employed can also cause conservation errors at intermediate stages of a steady-state calculation or at any time in a time dependent one when applied with a conservative scheme if all the fluxes are not updated simultaneously. Identical treatments must therefore be applied to the evaluation or utilisation of the expressions for F_w^C and F_e^C if conservation under these conditions is required. This point is addressed in the appendices to chapter 4.

A discretisation technique can be tested against this criterion by examining its formulation and the algorithm used to solve it.

Boundedness

It is well known (Forsythe and Wasow, 1960) that the solution to the steady-state convection-diffusion equation when s is zero has no extrema in the solution domain. The maximum and minimum values occur on the boundary of the solution domain. We expect the algebraic equations of our differencing scheme to reflect this property. Similarly we must be confident that when solving problems with accumulation or source terms that any extrema that occur within the solution domain are due to physical phenomena and not numerical effects.

Lack of boundedness in a scheme frequently manifests itself as wiggles in the solution with successive nodes being alternately high and low. One may be prepared to accept this in some circumstances where the magnitude of the oscillation is very small, but there are more serious consequences of lack of boundedness. A physical property such as concentration may be required to be unconditionally non-negative to be physically meaningful. An unbounded scheme may produce negative values. Therefore the solution algorithm must be adapted to be able to handle this possibility while maintaining conservation. This is clearly highly undesirable and convergence difficulties may be expected with this type of approach.

There are a number of criteria that are sufficient to ensure that a discretisation technique is bounded, but none that is both necessary and sufficient. In the absence of source or accumulation terms equation (3.2.16) becomes

$$A_p \phi_p = \sum A_n \phi_n. \quad (3.6.1)$$

This equation is used to discuss means of assessing boundedness.

The boundedness principle requires that ϕ_p is bounded above and below by its neighbours. This can be ensured by requiring

$$A_p = \sum A_n \quad (3.6.2)$$

$$A_n \geq 0 \quad \forall n. \quad (3.6.3)$$

The choice of sign in (3.6.3) is arbitrary and is chosen to be positive for convenience. In essence the above conditions require that ϕ_p is a weighted average of the adjacent values thus ensuring bounds are maintained. This condition is satisfied by the diffusion discretisation given in equations (3.4.2-5).

The first of these conditions is also necessary for mass conservation. This can easily be seen by setting

$$\phi = 1 \quad (3.6.4)$$

in equation (3.1.1) and ignoring the accumulation and source/sink terms. This reduces then to the mass conservation equation (equation (2.2.7)). Under these conditions equation (3.2.16) reduces to (3.6.2).

It must be noted that equations (3.6.2-3) are sufficient conditions, but not necessary. If it is violated then all that can be said is that boundedness is not guaranteed. Furthermore, the extension of this criterion when sources are present is not straightforward although it would be possible to associate boundedness problems with the source discretisation and not the convection terms if latter discretisation meets the above conditions. Finally this condition is not strictly applicable to discretisations using nodes other than adjacent nodes (e.g. discretisations involving *EE*, *WW*, *NN* and *SS*). This is illustrated in figure 3.6.1. Suppose a scheme where the discretisation involved five nodes (two on either side of the primary node) was used to calculate this solution. We see immediately that points A and B are not bounded. However the values at these points could be a weighted mean of those involved in the discretisation and therefore satisfy the above equation. The converse may also be true. A scheme that does not satisfy (3.6.2) may still be bounded. One such example is the linear upwind scheme (LUD) which is described later. However it is generally true to say though that if negative coefficients are present then boundedness problems must be expected unless special precautions have been taken.

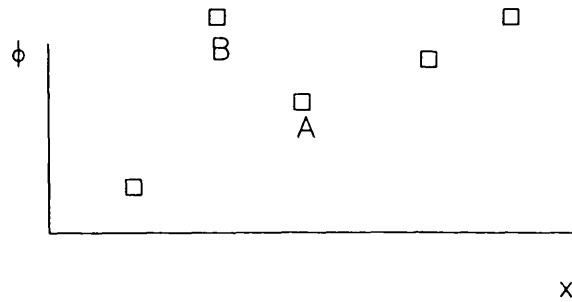


Figure 3.6.1: Variation of ϕ with Position

There is another reason why equation (3.6.2) is not very useful as a boundedness criterion in practice for convection dominated flows. It can be shown (appendix C.3) that the only conservative scheme where the discretisation of the convection flux in equation (3.2.5) contributes in an unconditionally positive manner to the convection discretisation coefficients is upwind differencing. All other schemes have some form of negative contribution to them. Boundedness may then be achieved by counterbalancing them with larger positive contributions from the diffusion term. If the grid Peclet number is low then this may be satisfactory but for higher grid Peclet numbers (which inevitably occur in practice) schemes of this type invariably reduce to upwind differencing (section 3.7). It is possible to use other convection terms to counteract these negative contributions (e.g. BSUD, section 3.9) but the performance is generally disappointing.

The use of equation (3.6.2) is not in testing schemes for boundedness but in indicating when boundedness problems may occur and their likely extent. The larger the ratio of the magnitude of negative discretisation coefficients to positive coefficients the more serious the boundedness problems may be expected to be.

Gaskell and Lau (1988) adopted an alternative boundedness criterion by examining the interpolation used to derive the convection coefficients. Their criterion requires that (i) the interpolated value at a cell face is bounded by the values on either side of that face and (ii) the value immediately upstream of that face is a local extrema then the value of that extrema itself is used as the interpolated value (i.e. upwind differencing is used). Condition (i) is a physical requirement that the interpolated value itself is bounded. Condition (ii) is to ensure that the convection flux exiting the cell with a local maximum is as large as physically possible and not less than the flux entering the cell. The net effect of convection is to try to reduce the maximum value. This ensures that the convection fluxes are not maintaining the maximum and some other term in the governing equation is causing it. Minima are similarly weakened. Thus the local extrema must be due to some other physical effect such as accumulation or a local source.

It is interesting to note that this is an inherently non-linear criterion in contrast to (3.6.2) above. It suggests that the interpolation method itself should depend on the local values of the variables and is not fixed in advance.

Gaskell and Lau argue that this condition is both necessary and sufficient. The sufficiency is apparent from the above remarks that ensure that convection fluxes will attempt to eliminate the local extrema. The necessity of this condition is not clear. The only requirement that can be said to be necessary is that the overall convective outflow is not less than the convective inflow. When convection is dominant the outflow is balanced by the inflow. This is a somewhat less restrictive condition than that claimed to be necessary by Gaskell and Lau.

Other tests for boundedness have been used. For example Flux Corrected Transport (FCT) (Boris and Book, 1973; Book et al, 1975; Zalesak, 1979), Flux Blending and Bounding Methodology (FBBM) (Peric, 1986), Filter Remedy And Methodology (FRAM) (Chapman, 1981) and so on. However these are not criteria for boundedness as such but alterations to the solution algorithm to maintain boundedness. Again these are non-linear methods.

Conditions for boundedness of the solution of (3.6.1) can sometimes be investigated by solving it on a uniform one dimensional mesh with constant velocity and diffusion analytically. The result is a critical grid Peclet number below which the scheme is bounded and above which it is not bounded. The results cannot always be extrapolated to unsteady problems or those of higher spatial dimensions. One example is the LUD scheme discussed in section 3.8.

It is frequently found that unbounded schemes are hard to converge to a solution. This is not particularly surprising as a sufficient conditions for the stability of some iterative solution methods for equations of the form (3.5.5) is equations (3.6.2-3) combined with

$$\frac{\sum |A_n|}{A_p - Si + \frac{v_p}{\Delta t}} < 1 \quad (3.6.5)$$

being true at at least one node (Scarborough, 1958). This condition is sometimes known as diagonal dominance. It holds true regardless of the nodes involved in the local interpolation, unlike criteria for unconditional boundedness discussed earlier in this section. It can be assisted by the discretisation of the source (i.e. $Si < 0$) and accumulation terms (i.e. low Courant numbers) and practices such as under-relaxation (section 4.9).

It is important however to distinguish between boundedness and stability. The former is specific to the properties of the solution obtained by use of a discretisation scheme and the latter is a property of the method being used to solve the equations resulting from a given discretisation.

Transportiveness

Consider a problem involving only convective transport. Information at a point P is convected to that point from upstream. No information from downstream is present under these conditions. We therefore expect our discretisation to have an upstream bias in those regions of the flow where convection is dominant. This property is known as transportiveness and prevents unphysical transfer of information upstream.

It could be argued that the convection term always involves the transfer of information from upstream and never transfer against the current. Therefore only upstream nodes should be involved in the discretisation of the convection term. However it is unreasonable to view this term in isolation from the other terms in the equation and the interpolation technique used to derive the convective flux should take into account not only the convection term but all the other terms. The inclusion of downstream

nodes in the interpolation takes these effects into account and improves the resolution of the scheme. Nevertheless we still expect to see some form of upwind bias in the discretisation, especially where rapid variation in ϕ is concerned.

Invariance

Let us suppose that equation (3.1.1) and its boundary conditions are invariant under the transformation

$$\Phi = g\phi + h \quad (3.6.6)$$

where g and h are constants. This transformation may correspond to a change in units for example. It is trivial to show the condition for the solution of the discretised form of equation (3.1.1) to be invariant under this transformation is that

$$A_r = \sum A_n \quad (3.6.7)$$

and

$$A(\Phi) = A(\phi). \quad (3.6.8)$$

The former condition is the same as equation (3.6.2) and the latter requires that the discretisation coefficients are also invariant under the transformation (3.6.6).

FVM convection discretisations can be divided into two categories. The simpler category is the family of linear schemes where A^c is independent of the transported variable ϕ and the non-linear schemes where A^c is dependent on ϕ . Linear schemes obviously satisfy condition (3.6.8). Non-linear schemes will also satisfy (3.6.8) providing that the ϕ dependence of the coefficients appear through dimensionless terms like

$$\frac{\phi_1 - \phi_2}{\phi_3 - \phi_4} \quad (3.6.9)$$

which are in turn invariant under (3.6.6). Here the subscripts refer to any nodal value of ϕ

Uniqueness

It is desirable that a discretisation technique and associated solution algorithm result in a unique solution to a particular problem regardless of the solution algorithm employed, starting conditions, relaxation factors and so on. Failure to do this introduces an additional degree of uncertainty into the solution and makes it difficult to make comparisons between the solutions of related problems. Linear discretisation techniques do not cause uniqueness problems but non-linear techniques may do so. It is desirable that any non-linear technique only results in one solution regardless of how it is obtained. It is sometimes possible to argue that a non-linear technique will not have a unique solution but it is harder to show that it does have one. The easiest way of assessing this is therefore to perform some numerical tests during the evaluation of candidate discretisation schemes.

Cost-Effectiveness

It is desirable to select a method that will allow one to solve a problem or group of problems with the least effort while still obtaining a satisfactory solution. This includes not only computational cost but the human effort to write or adapt a computer program, test it and then run that program and obtain a converged solution. Several factors need consideration here.

The discretisation technique should be robust and converged solutions can be obtained with the minimum of alteration to adjustable numerical parameters to a wide range of problems. The solution should be accurate but not excessively costly in terms of memory or CPU effort required. The method should be straightforward to program to minimize program development time.

The assessment of the cost-effectiveness of a method is somewhat subjective and does depend on the problem being tackled and the computing system being used. The use of pure computational times to solve a test problem is not a satisfactory approach as it does not take into account the accuracy of the achieved solution nor the human effort used to achieve that solution. Similarly comparisons of the number of nodes required to achieve a given computational accuracy do not include the CPU time taken or human effort. In both cases the use of a different test problem will give different results. However it is possible for one factor to outway all the others. One such example is the poor accuracy of upwind differencing due to numerical diffusion (section 3.7).

Discussion

A number of criteria have been outlined above. Unfortunately it is unlikely to be possible to simultaneously meet all of them, particularly the final cost-effectiveness requirement. The relative importance of these criteria does depend on the problem being tackled and the purpose to which the solution will be put. Consistency is important and is invariably met. For the applications of interest herein conservation, boundedness and cost-effectiveness are also crucial and will be the criteria of principal concern in this study. Transportiveness is only important where convection is the dominant transport mechanism. Invariance and uniqueness are desirable but are only of particular relevance when considering non-linear methods. The purpose of the remainder of this chapter is to examine candidate schemes for convection-dominated problems in the light of the above criteria.

3.7 Two Point Schemes

This section is primarily concerned with discretisation methods that use the two points on either side of the cell face to interpolate for the value on that cell face. For ease of exposition the discussion is restricted to a uniform mesh. The formulae for non-uniform meshes may be found in appendix B.3. Attention is usually focussed on the east face.

It will be found convenient to write the interpolation in the form

$$\phi_e = \frac{1}{2}\lambda(\phi_P + \phi_E) + \frac{1}{2}(1 - \lambda)\left(1 + \frac{u_e}{|u_e|}\right)\phi_P + \frac{1}{2}(1 - \lambda)\left(1 - \frac{u_e}{|u_e|}\right)\phi_E. \quad (3.7.1)$$

Equation (3.7.1) is a blend of interpolation methods for ϕ_e with blending factor λ . The first term is the central differencing (CD) interpolation and the second two terms the upwind differencing (UD) interpolation which depends on the velocity direction. Generally we shall assume u_e is positive for convenience. Many two point schemes can either be directly written in the form (3.7.1) or their equations be cast in a very similar form by Taylor series analysis. Table 3.7.1 lists the schemes discussed herein which fall into this category.

The truncation error^{*} in the interpolation for ϕ_e involved in schemes of this type on uniform meshes is

* Equation (3.7.2) is usually called the discretisation error although it is referred to as the truncation error in this work. Conventionally the truncation error is the error introduced into the parent differential equation by the discretisation.

Scheme	λ
Central Differencing	1
Upwind Differencing	0
HYBRID Differencing	1 $Pe_{\Delta} \leq 2$ 0 $Pe_{\Delta} > 2$
Huh <i>et al</i> (1986)	$\frac{2v}{u + v}$

Table 3.7.1: Blending Factors for Two Point Interpolation

$$\phi_e^{exact} - \phi_e^{scheme} = (1 - \lambda) \frac{\Delta u_e}{2|u_e|} \frac{\partial \phi}{\partial x} - \frac{\Delta^2}{8} \frac{\partial^2 \phi}{\partial x^2} + (1 - \lambda) \frac{\Delta^3}{48|u_e|} \frac{\partial^3 \phi}{\partial x^3} + O(\Delta^4) \quad (3.7.2)$$

where all derivatives are evaluated at e .

Popular members of this family are discussed below.

Central Differencing ($\lambda = 1$)

Central Differencing (Thom, 1933) corresponds to linear interpolation between P and E . It can be seen from (3.7.2) that it is second order accurate which is the highest formal accuracy possible from schemes of this type. However the convection discretisation coefficients (appendix B.3) are negative for any downstream cells and the central coefficient is small compared to the other coefficients. If the grid Peclet number is greater than 2 then the diffusion discretisation coefficients fail to provide overall diagonal dominance and stability and boundedness problems may arise. This manifests itself in the form of oscillations in the solution and/or difficulties in obtaining a converged solution. This has been observed by many authors including Timin and Esmail (1983), Shyy and Correa (1985) and Patel and Markatos (1986). Many engineering calculations including those in this study must be done at grid Peclet numbers exceeding two so this is a very serious drawback. Occasionally non-linearities in the transport equations themselves may provide some smoothing of the CD results at high grid Peclet numbers but care must be taken with the solution algorithm to obtain results. These results can be fairly accurate (Atias *et al*, 1977) which is consistent with the second order nature of the scheme.

The use of CD for high grid Peclet number flows cannot be generally recommended due to the boundedness and stability problems. It is possible to argue that these problems arise because the scheme does not embody the transportiveness concept and therefore conveys downstream information upstream which is unphysical and has a destabilising influence on the system.

Upwind Differencing ($\lambda = 0$)

Upwind differencing overcomes the difficulty encountered in CD by only using the node immediately upstream of the cell face. It was introduced by Courant *et al* (1952).

$$\begin{aligned}\phi_s &= \phi_r & u_s > 0 \\ &= \phi_\varepsilon & \text{otherwise}\end{aligned}\tag{3.7.3}$$

It is therefore always stable and bounded and solutions are readily achievable in practice. Indeed it meets all the criteria outlined herein except the cost-effectiveness test. In multidimensional high Peclet number problems where the flow is not aligned with the mesh then UD exhibits a phenomena known as false or numerical diffusion which seriously degrades the accuracy of the solution. See, for example, Shyy (1983), Timin and Esmail (1983), Hassan and Kim (1985a), Atias *et al* (1977), Smith and Hutton (1982), Leschziner (1980). The cause of this problem can be traced to its truncation error. The error in the east flux from (3.7.2) and (3.2.5) is

$$-\frac{1}{2}\Delta^2|u_s|\frac{\partial\phi}{\partial x} + O(\Delta^2).\tag{3.7.4}$$

Comparing this with (3.2.14) and (3.2.10) we see that this truncation error is equivalent to a diffusion flux with a diffusivity of

$$\Gamma_s^{false} = \frac{1}{2}\Delta|u_s|.\tag{3.7.5}$$

The effective grid Peclet number is therefore given by

$$\begin{aligned}Pe_{\Delta s}^{eff} &= \frac{|u_s|\Delta}{\Gamma_s + \frac{1}{2}\Delta|u_s|} \\ &= \frac{Pe_{\Delta s}}{1 + \frac{1}{2}Pe_{\Delta s}}\end{aligned}\tag{3.7.6}$$

from (3.2.19). We therefore see that for grid Peclet numbers in excess of two that the numerical diffusion exceeds the physical diffusion and that the effective grid Peclet number cannot exceed 2, regardless of the amount of physical diffusion present. The solution to this problem is to reduce Δ so that the physical grid Peclet number defined in (3.2.19) is substantially less than 2. However this is frequently not possible in two or three dimensional problems with current computing hardware and if one was going to operate in this regime then CD could be used safely and with greater accuracy. Hence the low cost-effectiveness of UD.

The impact of numerical diffusion on the accuracy obtained in practice does vary according to the problem being tackled and the computational mesh being used. Numerical diffusion will only noticeably effect the accuracy of the calculation when the convection terms are dominant in the calculation and the numerical diffusion flux is significant compared to the convection flux. The former condition requires that the source terms should not dominate the solution of the equation. The latter requires that there should be a significant gradient in ϕ along a mesh direction in which the flow velocity is significant.

There are two situations in which this condition occurs. The first is when source terms are present. These cause significant variations in ϕ along the mesh directions and hence numerical diffusion degrades the accuracy of the solution (Leonard, 1979; Raithby, 1976). This can occur in one, two or three dimensions.

The other situation is in two or three dimensional flows where the velocity is not aligned with the mesh. The cross-stream gradients are usually greater than the gradients in the flow direction. These gradients have significant components along the mesh directions where the velocities are also significant and hence numerical diffusion will rapidly erode these gradients. If the flow is aligned with the mesh then these cross-stream components are negligible where the velocity component along a mesh line is strong and significant only where the velocity is weak so the numerical diffusion flux (3.7.4) is always low.

Attempts have been made to evaluate the effective diffusion constant which erodes cross-stream gradients as a function of the angle between the flow and the mesh, ξ . Leschziner (1979) and Lai (1983) have shown from Taylor series analysis that this diffusion constant can be expressed as

$$\Gamma^{eff} = \frac{1}{2\sqrt{2}} \Delta |u| \sin(2\xi) \sin\left(\frac{\pi}{4} + \xi\right). \quad (3.7.7)$$

Lai (1983) and Leschziner (1980) review other similar formulae obtained by other workers. They all have a maximum at $\pi/4$ and give a numerical diffusivity of about $0.35|u|\Delta$ at that angle.

The effect of numerical diffusion in the latter case can be very dramatic and is easily spotted in a numerical solution although quantification of its impact may be slightly harder (McGuirk *et al*, 1981). This has lead some authors (e.g. Patel *et al*, 1985a) to argue that numerical diffusion is a purely multidimensional phenomena caused by the failure of UD to take into account the flow direction. It also has a significant, though less visible, effect on the accuracy of flows with sources or sinks and is therefore not a purely multidimensional phenomena or one associated with flow direction alone. However the improvement in accuracy obtained from flow oriented schemes (section 3.8) would suggest that the poor accuracy that occurs when the flow is not aligned with the mesh has the most serious effect on the solution.

The consequences of high numerical diffusion can mask changes in modelling. The errors introduced into the solution on practical computational meshes by numerical diffusion can mask the effects of changing turbulence models which makes it difficult to draw firm conclusions on these models (Hackman *et al*, 1984; Leonard *et al*, 1978; Bolour-Froushan, 1986). The relatively slow response to mesh refinement of UD and other first order schemes also contributes to this problem.

Patel *et al* (1987) have concluded that UD is the most appropriate scheme in some circumstances. However their test was very specific and the flow was generally aligned with the mesh so minimal numerical diffusion would result. Additionally the mesh was rather coarse for the complexity of the flow they were computing. Numerous other computational tests in the literature do not support this view and UD cannot be generally recommended.

It should be noted that the above analysis of numerical diffusion relies on the assumption that the first term in the error series is dominant. This may not always be the case as demonstrated in appendix C.3. However the conclusions are generally true.

HYBRID Differencing

HYBRID differencing (Spalding, 1972) was proposed to use the more accurate CD where possible (i.e. $Pe_\Delta \leq 2$) and use the unconditionally bounded UD at other times. The resulting scheme is sometimes more accurate than UD and is always unconditionally bounded. However it does not overcome the difficulties with UD at high grid Peclet numbers and it offers only a slight advantage over UD for convection dominated problems. Tests comparing HYBRID with other schemes have been carried out by many authors including Shyy and Correa (1985), Hah (1983), Patel and Markatos (1986), Patel *et al*

(1987), Joseph (1983), Claus *et al* (1985) and Aggarwal (1987). Some versions of HYBRID do not include the physical diffusion terms if the grid Peclet number exceeds two because there is already too much numerical diffusion introduced.

Exponential Differencing

This scheme, originally introduced by Allen and Southwell (1952) and Spalding (1972), seeks to employ the solution to a simplified version of the convection-diffusion equation to perform the interpolation. Several variants have been proposed. The simplest one is based on the equation

$$u \cdot \frac{\partial \phi}{\partial x} - \Gamma \cdot \frac{\partial^2 \phi}{\partial x^2} = 0 \quad (3.7.8)$$

subject to

$$\begin{aligned} \phi &= \phi_P & \text{at } x &= x_P \\ \phi &= \phi_E & \text{at } x &= x_E. \end{aligned} \quad (3.7.9)$$

The solution is

$$\phi(x) = \phi_P + (\phi_E - \phi_P) \frac{e^{\frac{u}{2\Gamma}(x-x_P)} - 1}{e^{\frac{u}{2\Gamma}(x_E-x_P)} - 1} \quad (3.7.10)$$

and this expression is used for the interpolation. The diffusion flux is obtained by differentiating it rather than the usual discretisation by (3.4.1).

The advantage of this method is that it is exact for one dimensional sourceless steady transport. For more complex situations it is attempting to use the local solution of the equation for the calculation of the fluxes. It is not difficult to see that for small grid Peclet numbers it is equivalent to CD and for high values then it is the same as UD. The scheme does have unconditionally positive coefficients and is always bounded. However it suffers from numerical diffusion when the flow is not aligned with the mesh in two or three dimensional flows at high grid Peclet numbers. This is not surprising in view of the similarity with UD under these conditions.

Patankar (1980) proposed a power law approximation to the exponentials to reduce computational cost which has been tested by Huang *et al* (1983), Beier *et al* (1983), Patel and Markatos (1987) and Patel *et al* (1987). The former three works report appreciable numerical diffusion. The latter used a test problem where significant numerical diffusion would not be expected.

Other authors have attempted to extend the concept of using a locally analytic solution to build multidimensional discretisation schemes to overcome the difficulties when the flow is not aligned with the mesh. Wong and Raithby (1979) proposed the Locally Analytic Differencing Scheme (LOADS). In two dimensions the scheme is based on the analytic solution of the two equations

$$\frac{\partial}{\partial x}(u\phi) - \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) = -\frac{\partial}{\partial y}(v\phi) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right) + s \quad (3.7.11)$$

$$\frac{\partial}{\partial y}(v\phi) - \frac{\partial}{\partial y}\left(\Gamma \frac{\partial \phi}{\partial y}\right) = -\frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial x}\left(\Gamma \frac{\partial \phi}{\partial x}\right) + s \quad (3.7.12)$$

where, when performing the east-west discretisation, (3.7.11) is used with the RHS explicitly evaluated from the current solution. (3.7.12) is used for the north-south interpolation with the RHS again an explicit term. Huang *et al* (1983) replaced the exponentials with a power law approximation and applied this scheme to a number of problems. They found it generally performed well in comparison with the simple power law variant of the exponential scheme for scalar transport. However they found that it was not unconditionally bounded and were unable to obtain a solution for laminar flow problems. Prakash (1984) also applied LOADS to scalar transport tests and found it to be superior to the exponential scheme. However boundedness was not maintained.

Stubley *et al* (1980) proposed a method based on the exact solution of the locally linearised two-dimensional convection-diffusion equation. This was solved over the rectangle with corners at *NW*, *NE*, *SE* and *SW* with either a linear or quadratic profile assumed along the boundary. The resulting scheme involves the summation of infinite series and is non-conservative and expensive. The extension to three dimensional flows would require a significant amount of work. Comparative computations with this scheme are not reported.

Chen *et al* (1981) independently derived a similar scheme based on the same principals. They successfully calculated the flow in a cavity with a driven lid and did not report any boundedness, stability or numerical diffusion problems. Again the scheme is very complex and a three dimensional generalisation is daunting. However Chen and Chen (undated) have reported doing this successfully. However the complexity and associated costs of the scheme and lack of the conservative property makes it look relatively unattractive for the applications considered herein.

Raithby (1976) has also proposed an interpolation method based on the local solution around a single cell face. This is considerably simpler than the full cell approximations described above but is not bounded.

Diffusion Compensated Differencing Schemes

A number of differencing schemes have been proposed that seek to add diffusion to CD to stabilise it or remove diffusion from UD to improve its accuracy. The equivalence of these two approaches can be seen by comparing the truncation errors associated with CD and UD.

Owen (1984) proposed a scheme that evaluated the numerical diffusivity constant and subtracted it from the physical diffusivity to apply a reduced diffusion term to improve the accuracy. The resulting effective diffusion constant may become negative. For steady state-state problems it is easy to show that this scheme is equivalent to CD and oscillations accordingly appear in the solution. To overcome these problems Owen suggests that the diffusivity correction should only be made if the effective diffusion constant would remain positive. This is equivalent to requiring that the grid Peclet number is less than 2 (i.e. HYBRID differencing). A bounded scheme could be obtained by using the maximum of zero and the reduced diffusivity but again this would be very similar to HYBRID and suffer from its deficiencies.

Huh and Golay (1983) and Huh *et al* (1986) employed similar ideas in multidimensional flow to eliminate numerical diffusion. They evaluated the cross stream numerical diffusivities and subtracted them from the physical diffusion. This turns out to be a blend of UD and CD with the blending factor depending on the angle of the flow with the mesh lines. Increasing amounts of CD are used as the flow becomes increasingly skewed to the mesh. This caused boundedness problems to occur.

Denis and Hudson (1978) bounded CD by increasing the physical diffusivity according to the expression

$$\Gamma_{\bullet}^{eff} = \Gamma_{\bullet} + \frac{\Delta^2 u^2}{8\Gamma_{\bullet}}. \quad (3.7.13)$$

This can be rearranged as

$$\Gamma_{\bullet}^{eff} = \Gamma_{\bullet} \left(1 + \frac{1}{8} Pe_{\Delta\bullet}^2 \right) \quad (3.7.14)$$

from which we can express the effective grid Peclet number as

$$Pe_{\Delta\bullet}^{eff} = \frac{Pe_{\Delta\bullet}}{1 + \frac{1}{8} Pe_{\Delta\bullet}^2}. \quad (3.7.15)$$

Comparison with equation (3.7.6) reveals that this method is only less diffusive than UD for grid Peclet numbers less than four and is considerably more diffusive as the grid Peclet number increases. Indeed in the limit of zero diffusivity the effective grid Peclet number becomes infinite corresponding to a purely diffusive problem without convective transport. Gupta (1983) has illustrated this computationally. This is an example of where Taylor series analysis may break down because this second order scheme is less accurate than a first order scheme for a considerable range of cell Peclet numbers. Despite this clear failing of the method at high grid Peclet numbers Curtis (1987) advocates its use under these conditions.

Timin and Esmail (1983) applied blends of CD and UD to the Navier-Stokes equations in the calculation of the flow in a driven cavity. They used λ values of 0.2, 0.9 and 1.0. They found CD to be fairly accurate for the Reynolds numbers used although oscillations appeared on coarse meshes. They found that predominantly upwind schemes (i.e. low λ) caused significant changes to the solution. Small amounts of upwinding did not change the flow fields greatly but did affect the pressure field.

Discussion

Two point interpolation methods suffer from two problems at high grid Peclet numbers. Upwind schemes are inaccurate, particularly in multidimensional flows, although they are generally robust and easy to solve. It is expensive and inefficient to obtain accurate solutions with them. Centred schemes are more accurate than upwind schemes at low grid Peclet numbers but are unbounded at higher values. Further, they are difficult to converge which makes them impractical as well. Schemes which are a blend of UD and CD suffer from both of these problems to a greater or lesser extent.

The simplest members of the exponential family are upwind in nature at high grid Peclet numbers. More complex multidimensional formulations are difficult to work with and suffer from boundedness problems and therefore cannot be recommended.

3.8 Extended Molecule Schemes

This section is concerned with discretisation techniques that are formally second order accurate or above and that employ the additional nodes EE , WW and so on. On uniform meshes schemes these schemes can be interpreted as CD with a curvature correction term. The interpolation is

$$\phi_{\bullet} = \frac{1}{2}(\phi_P + \phi_E) - \frac{1}{2}\lambda \left(1 + \frac{u_{\bullet}}{|u_{\bullet}|} \right) (\phi_E - 2\phi_P + \phi_W) - \frac{1}{2}\lambda \left(1 - \frac{u_{\bullet}}{|u_{\bullet}|} \right) (\phi_{EE} - 2\phi_E + \phi_P) \quad (3.8.1)$$

where the value of λ distinguishes between different schemes. The formulae for CD, LUD and QUICK on non-uniform meshes and their discretisation coefficients are given in appendix B.3. The principal members of this scheme are listed in table 3.8.1. If the grid Peclet number is below the critical grid Peclet number then boundedness problems will not arise (Shyy, 1983; Too and Sparrow, 1987).

Scheme	λ	Critical Grid Peclet Number
Central Differencing	0	2
Linear Upwind Differencing (LUD)	1/2	∞
Quadratic Upwind Differencing (QUICK)	1/8	8/3
'Third' Order Upwind Differencing	1/6	3

Table 3.8.1: Blending Factors for Extended Interpolation

The LUD critical grid Peclet number applies only to one-dimensional steady-state problems only.

The truncation error in the interpolation for ϕ_e involved in schemes of this type on uniform meshes is

$$\phi_e^{exact} - \phi_e^{scheme} = -\left(\frac{1}{8} - \lambda\right) \Delta^2 \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{2} \lambda \Delta^3 \frac{\partial^3 \phi}{\partial x^3} + O(\Delta^4) \quad (3.8.2)$$

where all derivatives are evaluated at e. However one needs to be careful when applying Taylor series analysis to these extended schemes. The truncation error introduced depends on whether the scheme is regarded as a finite volume interpolation or a finite difference approximation. This can be illustrated in the following manner. Let $u = 1$ for ease of exposition and construct the finite difference approximation

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta} (\phi_e^{exact} - \phi_w^{exact}) - \frac{1}{24} \Delta^2 \frac{\partial^3 \phi}{\partial x^3} + O(\Delta^3) \quad (3.8.3)$$

where all derivatives are evaluated at P. Substitution of equation (3.8.2) and its equivalent for the west face into (3.8.3) gives

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta} (\phi_e^{scheme} - \phi_w^{scheme}) - \left(\frac{1}{6} - \lambda\right) \Delta^2 \frac{\partial^3 \phi(x_P)}{\partial x^3} + O(\Delta^3). \quad (3.8.4)$$

A Taylor series expansion has been used in the derivation of (3.8.4) to express the derivatives at e and w in terms of derivatives evaluated at P. Equation (3.8.2) is the truncation error introduced by the convection discretisation into the integral form of the transport equation (equation (3.2.2)). Equation (3.8.4) is the truncation error introduced by the convection discretisation into the differential form of the transport equation (equation (3.1.1)) from the discretisation of the convection term.

From table 3.8.1 and equations (3.8.2) and (3.8.4) the following conclusions can be drawn. CD and LUD are second order accurate schemes. The 'third' order scheme is only a second order interpolation although in a finite difference context it is third order accurate. The QUICK scheme has a third order accurate interpolation but is only second order in a finite difference context. However in multidimensional situations it has already been argued in section 3.2 that the assumption of constant ϕ across a cell face leading to equations (3.2.5-8) is at most second order accurate so the evaluation of a flux based on (3.8.1) is at most second order accurate.

The Taylor series analysis shows that there is not a diffusion like truncation error for schemes in this family. Therefore the problems of low accuracy when the flow is inclined to the mesh is less serious than with UD. That is not to say that there will be no loss of accuracy. Calculations presented in chapter 5 will illustrate this.

The coefficient λ controls the amount of third order dissipation which is introduced in an attempt to stabilise the CD scheme. Increasing this factor increases the stability although a fully stable method will not be achieved.

Central differencing has been discussed in section 3.7.

Linear Upwind Differencing ($\lambda = 1/2$)

This discretisation technique was originally introduced in a finite difference context by Price *et al* (1966). When expressed in finite volume form the scheme corresponds to extrapolating for ϕ_e from the two nodes immediately upstream.

$$\begin{aligned}\phi_e &= \phi_P + \frac{1}{2}(\phi_P - \phi_W) & u_e &\geq 0 \\ &= \phi_E + \frac{1}{2}(\phi_E - \phi_{EE}) & u_e &< 0\end{aligned}\tag{3.8.5}$$

In one dimension LUD is unconditionally bounded despite having negative coefficients but it is not bounded for unsteady problems (Shyy, 1984; Gupta and Manohar, 1978) or multidimensional problems.

Atias *et al* (1977) applied LUD to flow in a driven cavity and impinging jet flow. They found that the accuracy was better than UD and frequently nearly as high as CD. The computational cost was comparable with both CD and UD. They noted that LUD was least accurate on coarse meshes at high Reynolds numbers. This could be due to boundedness problems although no mention of this in connection with LUD is given. However it does appear that it is easier to obtain a converged solution with LUD than CD (Hodge *et al*, 1979).

Gupta and Manohar (1978) analysed LUD in one dimension in a finite difference implementation. They concluded that it was not likely to be significantly better than UD in these situations. Shyy (1983, 1984), Shyy and Correa (1985) and Castro and Jones (1987) performed a number of scalar transport and laminar flow tests with LUD and other popular schemes. They concluded that LUD was generally the best scheme although boundedness problems were reported.

Wilkes and Thompson (1983) found LUD to be considerably more accurate than UD. For two dimensional problems their implementation of LUD required about 60% more computer time than UD. Vanka (1987) compared three different ways of implementing LUD in the calculation of flow in a driven cavity. Generally the convergence rate was slower than that of HYBRID, especially for higher Reynolds numbers. The computational cost of an LUD solution could be up to seven times that of HYBRID. Overshoots were also observed at high Reynolds numbers. In contrast to Wilkes and Thompson Vanka does not recommend use of LUD. However it is possible that the difficulties encountered are due not solely to the discretisation but to the multi-grid solution algorithm used to solve the continuity and momentum equations.

QUICK ($\lambda = 1/8$)

This scheme, due to Leonard (1979), is called Quadratic Upstream Interpolation for Convective Kinematics and is also known as Quadratic Upwind Differencing (QUD). On a uniform mesh it corresponds to fitting a parabola to the two mesh points upstream of the cell face and one cell downstream. The form given in the appendix to Leonard's 1979 paper proposing QUICK does not correspond to fitting a parabola on non-uniform meshes. It is possible to adapt QUICK such that it does use a parabola on all meshes (Syed and Chiapetta, 1985).

QUICK has been extensively tested in the literature where many comparisons with other schemes are described. The advocate findings of these studies are presented in table 3.8.2. A number of different implementations of QUICK have been tried. Part of the interpolation is treated implicitly and therefore contributes to the convection discretisation coefficients (A^c) and the remainder is evaluated explicitly and is treated as a source term via S_e which is updated at the beginning of each iteration. This is illustrated in the table by separating the interpolation for a positive velocity on the east face into implicit and explicit parts. Full details can be found in the original references.

Blank entries in table 3.8.2 indicate an aspect of QUICK that was not discussed in the original reference. The following conclusions concerning QUICK can be drawn from the work cited in table 3.8.2.

- (i) QUICK is more accurate than two point interpolations. However it is important not to judge accuracy from scalar transport tests alone, especially in one dimension.
- (ii) The accuracy of QUICK responds rapidly to refinement of the mesh.
- (iii) Boundedness problems are usually manifest in scalar transport problems although difficulties not as apparent when calculating flows themselves. This is because violations in boundedness in the momentum equations are difficult to spot if not accompanied by wiggles. Therefore it is possible that bounds are violated in more calculations than recorded in the table.
- (iv) Many solution algorithms show signs of instability with QUICK. This results in difficulties in obtaining converged solutions and significantly increased computational costs.
- (v) Many different implementations of QUICK have been tried (Han *et al*, 1981; Pollard and Siu, 1982; Hah, 1983 and 1984; Huang *et al*, 1983; Patel and Markatos, 1986; Patel *et al*, 1987) and the stability and convergability is sensitive to the implementation. Some of these implementations may not be strictly conservative, especially in a time dependent context.
- (iv) QUICK has not been adequately compared to LUD. There is evidence (Shyy, 1983; Shyy and Correa, 1985) to suggest that LUD may be preferable for general purpose flows.

Consideration to the formal extension of QUICK to two dimensions has been given by Leonard *et al* (1978). The required correction was found to be small and was therefore ignored.

Author(s)	Implementation in a positive velocity field for east face		Nature of Tests	Boundedness Problems	Convergence Difficulties/ Computational Cost	Response to Mesh Refinement	Principal Conclusions
	Implicitly solved	Explicitly evaluated					
Leonard et al (1978)			Scalar transport at angle to mesh. Annular jet.	Yes			QUICK approach recommended over HYBRID.
Leschziner (1980)			Scalar transport in various flows. Backward facing step and pipe expansion.	Yes		Good	QUICK and SUD much better than HYBRID.
Han et al (1981)	various	various	Stagnation flow. Flow in driven cavity. Flow in sudden pipe expansion.		Care with implementation needed. Convergence slow and sometimes not achieved.	Good	QUICK more accurate than HYBRID. Convergence not always easy but QUICK cost-effective.
Leonard (1981)			One dimensional scalar transport with sources.	Only one case observed.			QUICK approach recommended.
Leschziner and Rodi (1981)			Turbulent jet flows.			Good	SUD and QUICK preferable to HYBRID.
Pollard and Siu (1982)	$-\frac{1}{8}\phi_v, -\frac{7}{8}\phi_r, -\frac{3}{4}\phi_z$ $-\frac{1}{8}\phi_v, -\frac{3}{4}\phi_r, -\frac{3}{4}\phi_z$	$-\frac{1}{8}\phi_r, -\frac{9}{8}\phi_z$ $-\frac{9}{8}\phi_z$	Laminar flow between plates, in driven cavity, sudden pipe expansion.		Convergence slow. Cost about four times UD.		QUICK better than UD. Stability implementation dependent.

Author(s)	Implementation in uniform positive velocity field for east face		Nature of Tests	Boundedness Problems	Convergence Difficulties/ Computational Cost	Response to Mesh Refinement	Principal Conclusions
	Implicitly solved	Explicitly evaluated					
Smith and Hutton (1982)			Scalar transport.	Yes			QUICK less diffusive than UD or LUD. Results comparable with other higher order schemes.
Beier et al (1983)			Scalar transport.				QUICK and CD recommended on grounds of accuracy.
Hah (1983)	$\frac{3}{4}\phi_r + \frac{1}{4}\phi_s$	$\frac{1}{8}(\phi_r - \phi_s)$	Conical diffuser.				SUD or QUICK preferable to HYBRID.
Huang et al (1983)	$\frac{5}{8}\phi_r + \frac{3}{8}\phi_s$	$\frac{1}{8}(\phi_r - \phi_s)$	Scalar transport in uniform velocity and solid body rotation. Cavity, corner and jet flows.	Observed where gradients high.		Good (limited tests)	Accuracy good despite occasional problems with boundedness. QUICK preferred to SUD.
Shyy (1983)			Scalar transport.	Yes			LUD performed better than QUICK or SUD. Oscillations in QUICK smaller than CD.
Hah (1984)	$\frac{3}{4}\phi_r + \frac{1}{4}\phi_s$	$\frac{1}{8}(\phi_r - \phi_s)$	Flow over turbine blade.	Upwinding used when oscillations appear.			SUD or QUICK preferable to HYBRID.
Claus et al (1985)			Scalar transport, laminar flows.		Cost 3-5 times UD.	Good	QUICK found to be accurate. FBBM methodology (Peric, 1986) applied to SUD may be as accurate but cheaper under some circumstances.

Author(s)	Implementation in uniform positive velocity field for east face		Nature of Tests	Boundedness Problems	Convergence Difficulties/ Computational Cost	Response to Mesh Refinement	Principal Conclusions
	Implicitly solved	Explicitly evaluated					
Patel et al (1985a)			One dimensional scalar transport.	Yes	Convergence slower than for UD or CD.		QUICK and two finite difference schemes recommended for accurate solutions. No two dimensional results.
Shyy and Correa (1985)			Scalar transport, laminar and turbulent flows.	Boundedness violated in scalar transport.			LUD satisfactory for complex flows.
Syed and Chiapetta (1985)			Laminar and turbulent flows.		Unstable on some occasions.	Good	QUICK usually found to be accurate where convergence achieved. FBBM methodology (Peric, 1986) more stable but not quite as accurate.
Syed et al (1985)			Scalar transport.	Yes	Expensive, new solvers better?	Good	Bounded version of QUICK promising but improved solvers needed.
Patel and Markatos (1986)	$-\frac{1}{8}\phi_v, -\frac{7}{8}\phi, -\frac{3}{4}\phi,$ $-\frac{1}{8}\phi_v, -\frac{3}{4}\phi, -\frac{3}{4}\phi,$ $-\frac{1}{8}\phi_v, -\frac{3}{4}\phi, -\frac{3}{8}\phi,$	$-\frac{1}{8}\phi, -\frac{9}{8}\phi,$ $-\frac{9}{8}\phi,$	Sudden pipe expansion. Flow in driven cavity.		Stability implementation dependent. Cost 1.5-8 times UD.	Good	QUICK accurate but not cheap. Stability can be a problem.

Author(s)	Implementation in uniform positive velocity field for east face		Nature of Tests	Boundedness Problems	Convergence Difficulties/ Computational Cost	Response to Mesh Refinement	Principal Conclusions
	Implicitly solved	Explicitly evaluated					
Aggarwal (1987)	Time dependent		Flame propagation.				QUICK the best scheme studied.
Castro and Jones (1987)			Forward step in 2-D channel. Confined flow over flat plate.	Yes	More expensive than other techniques.		TOUD preferred to QUICK on grounds of robustness and accuracy.
Patel et al (1987)	$-\frac{1}{8}\phi_v - \frac{7}{8}\phi_r - \frac{3}{4}\phi_s$ $-\frac{1}{8}\phi_v - \frac{3}{4}\phi_r - \frac{3}{4}\phi_s$ $-\frac{1}{8}\phi_v - \frac{3}{4}\phi_r - \frac{3}{8}\phi_s$	$-\frac{1}{8}\phi_r - \frac{9}{8}\phi_s$ $-\frac{9}{8}\phi_s$	Supersonic jet.		Unable to converge.		QUICK unstable.
Bradley et al (1988)			Scalar transport in uniform flow at angle to mesh with and without sources.				QUICK more accurate than CD or UD.

Table 3.8.2: Previous Work Concerning QUICK

A blank entry in the table indicates that no information is reported on this aspect of QUICK. TOUD is discussed below and the SUD scheme is discussed in the next section.

'Third' Order Upwind Differencing (TOUD) ($\lambda = 1/6$)

The TOUD scheme has been proposed independently by many authors (Leonard, 1981; Argawal, 1981; Bradley *et al*, 1988) in a finite difference manner. Tsui (1987) indicated that TOUD may be applied in a finite volume sense but did not pursue this further. The finite difference version of this scheme is

$$\begin{aligned}\frac{\partial \phi}{\partial x} &= \frac{1}{6}(2\phi_E + 3\phi_P - 6\phi_W + \phi_{WW}) & u > 0 \\ \frac{\partial \phi}{\partial x} &= -\frac{1}{6}(\phi_{EE} - 6\phi_E + 3\phi_P + 2\phi_W) & u < 0\end{aligned}\quad (3.8.6)$$

where the derivative is at P . This is equivalent to using (3.8.1) with λ as one sixth to evaluate the cell face values (ϕ_e^{TOUD} and ϕ_w^{TOUD}) and taking

$$\frac{\partial \phi}{\partial x} = \frac{1}{\Delta}(\phi_e^{TOUD} - \phi_w^{TOUD}). \quad (3.8.7)$$

Written in the form (3.8.6) TOUD is not a conservative scheme but if applied in the manner of (3.8.1) it would be. In fact there is little difference between QUICK and the conservative implementation of TOUD; the only difference of note being that the latter has a slightly higher critical grid Peclet number.

On non-uniform meshes TOUD is consistent with using a cubic polynomial to evaluate $\partial \phi / \partial x$ at P .

Agarwal (1981) reported that TOUD was stable and accurate for the calculation of a number of laminar flows. No comparisons were made with other discretisations. Leonard (1981) named this scheme UTOPIA (Uniformly Third-Order Polynomial Interpolation Algorithm) and claimed a high degree of accuracy and stability. However all these calculations were one dimensional and therefore these findings cannot be extended to more general flows. Patel *et al* (1985a) and Syed *et al* (1985) also used this method on a simple one dimensional problem but they found appreciable oscillations in the solution for high grid Peclet number and therefore did not recommend the scheme. Castro and Jones (1987) found the method to be accurate and more stable than QUICK in their two dimensional flow calculations. Bradley *et al* (1988) considered scalar transport in two dimensions on very coarse meshes using this method. They concluded that this scheme is more accurate than QUICK.

Discussion

The family of extended schemes contains many second or higher order schemes that can be related to each other on uniform meshes by a blending factor, λ . All the members of this family suffer from lack of boundedness and the lower the value of λ the more severe this problem. Stability of CD and QUICK is also very poor. The lack of boundedness with these schemes seems to be most serious when considering scalar transport. Boundedness problems are less apparent in the solution of the Navier-Stokes equations for QUICK and LUD. The same may be expected of TOUD. However QUICK can be expensive to converge and on other occasions convergence may not be achieved at all. The most robust member of the scheme is probably the LUD scheme although QUICK is the most popular.

This family of schemes is not suitable for solving the scalar transport equations involved in reacting flows where it is important that both the continuity and boundedness are strictly maintained.

3.9 Flow Oriented Schemes

Flow oriented schemes attempt to overcome the cross-stream numerical diffusion associated with UD by upwinding in the flow direction instead of the mesh lines. They are therefore not higher order schemes in a formal Taylor series sense, but simply first order methods. Their improved performance comes from the reduction in numerical diffusion which allows higher effective grid Peclet numbers to be achieved. Second or higher order flow oriented schemes have not been reported in the literature.

A number of schemes based on this principle have been proposed and they are discussed below. Other less popular schemes have already been discussed in the section on exponential schemes.

Skew Upwind Differencing (SUD)

SUD was proposed by Raithby (1976). It embodies the flow oriented upwind by interpolating for ϕ_e according to figure 3.9.1 and table 3.9.1. The interpolation region is the section in figure 3.9.1 that the flow is leaving. The interpolation is not strictly oriented with the flow in regions 2, 3, 6 and 7 which may introduce a little numerical diffusion. The resulting coefficient matrix involves the additional corner nodes (*SE* etc) complicating the solution algorithm and making the scheme cumbersome in three dimensions. However these corner coefficients are always positive unlike the principal coefficients (*E* etc) which may become negative. Boundedness problems arise as a result.

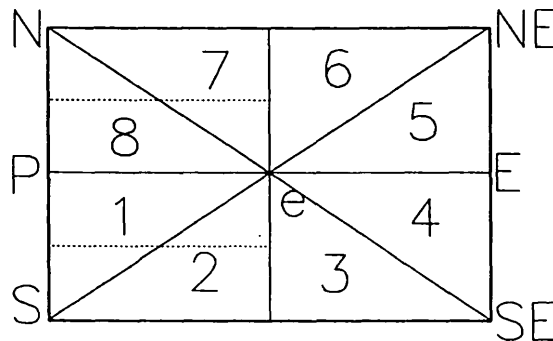


Figure 3.9.1: Interpolation Regions for SUD

The effectiveness of SUD is reduced on non-uniformed meshes. Suppose the computational cell in figure 3.9.1 is stretched in the x direction. The angle subtended by triangle 1 at the point e decreases and the angle subtended by triangle 2 increases. Therefore UD will be used for a greater proportion of flow angles. Similarly if the cell is compressed in the x direction more UD is introduced into the discretisations of the north and south fluxes.

SUD has been extensively tested in the literature. Table 3.9.2 summarises the principal findings concerning SUD and the following conclusions can be drawn.

Region	Interpolated Value
1	Linear interpolation between ϕ_P and ϕ_S
2	ϕ_S
3	ϕ_{SE}
4	Linear interpolation between ϕ_E and ϕ_{SE}
5	Linear interpolation between ϕ_E and ϕ_{NE}
6	ϕ_{NE}
7	ϕ_N
8	Linear interpolation between ϕ_P and ϕ_N

Table 3.9.1: Interpolation Regions for SUD

- (i) SUD is more accurate than HYBRID or upwind differencing. In particular numerical diffusion is reduced. QUICK and LUD are sometimes recommended in preference to SUD.
- (ii) Boundedness problems particularly manifest themselves in scalar transport tests.
- (iii) Converged solutions can usually be obtained.
- (iv) Source terms degrade the accuracy of SUD

It is interesting to note that three dimensional applications of SUD have not been reported in the literature. A three dimensional version of SUD would be feasible but rather complex.

Author(s)	Nature of Tests	Boundedness Problems	Principal Conclusions
Raithby (1976)	Scalar transport in uniform and solid body rotating velocity fields	Yes	Numerical diffusion significantly reduced compared to UD. Stable.
Smith and Hutton (1982)	Scalar transport		Some numerical diffusion observed, but much less than UD. VUD (below) more accurate
Hah (1983)	Conical diffuser		SUD or QUICK preferable to HYBRID.
Huang <i>et al</i> (1983)	Scalar transport in uniform velocity and solid body rotation. Cavity, corner and jet flows.	Yes	Stable and generally more accurate than UD. Stable. More susceptible to boundedness problems than QUICK. QUICK preferred to SUD.
Lai (1983)	Scalar transport and flow in cavity.	Yes	SUD not accurate when source terms significant.
Shyy (1983)	Scalar transport	Yes	SUD reduces numerical diffusion but LUD and QUICK are to be preferred.
Hackman <i>et al</i> (1984) ¹	Laminar and turbulent flow over backward facing steps. UD used for k and ϵ equations.		SUD more accurate than UD for these flows. When UD is used it is possible to mistake poor numerical accuracy for deficiencies in turbulence modelling.
Shyy and Correa (1985)	Scalar transport	Yes	No specific conclusions concerning SUD.
Claus <i>et al</i> (1985)	Scalar transport, laminar flows		SUD more expensive to converge than HYBRID. Bounded versions of SUD (discussed later) preferable.
Patel <i>et al</i> (1987)	Supersonic flow aligned with mesh		Unable to obtain converged solution.

¹ CD was used for $Pe_{\Delta} \leq 1$, SUD for $Pe_{\Delta} \geq 2$ and a blend of SUD and CD otherwise.

Table 3.9.2: Previous Work Concerning SUD

Bounded Skew Upwind Differencing (BSUD)

BSUD is a blend of UD, CD and SUD formulated by Lai (1983).

$$\phi_p^{BSUD} = \lambda_1 \phi_p^{CD} + (1 - \lambda_1)(\lambda_2 \phi_p^{SUD} + (1 - \lambda_2)\phi_p^{UD}) \quad (3.9.1)$$

where

$$0 \leq \lambda_1 \leq 1$$

$$0 \leq \lambda_2 \leq 1. \quad (3.9.2)$$

Lai showed that it was always possible to choose λ_1 and λ_2 such that the coefficients of the blended scheme were positive and the scheme therefore bounded. This could be done in a conservative manner by associating the blending factors with cell faces. Clearly it is best to maximise the blending factors to use the maximum amount of the higher accuracy CD and SUD schemes to reduce numerical diffusion. This was done by using a modification of the Simplex optimisation algorithm. It is important to note that the blending factors only depend on the local flow field and diffusivity, not the value of ϕ itself. Therefore BSUD is a linear method.

Applications of BSUD have been reported by Lai (1983), Gosman and Lai (1982), Syed *et al* (1985) and Claus *et al* (1985). The scheme has been demonstrated to be stable and bounded solutions are readily obtained. However care must be taken when significant sources are present (Lai, 1983; Gosman and Lai, 1982). Lai (1983) proposed corrections to the scheme to take into account the variation in ϕ along a stream-line due to source terms. Numerical diffusion is reduced but is still significant, particularly for flows modestly inclined to the mesh (about 26° on a uniform mesh). BSUD is an improvement on UD, but only a modest one. The restriction to positive coefficients appears to be too severe and therefore more UD is used than is strictly necessary to maintain bounds.

Hassan *et al* (1983a) also found that by limiting the amount of the SUD introduced unconditionally positive coefficients could be obtained. The SUD interpolation ranges given in figure 3.9.1 are changed to those in figure 3.9.2 and the interpolation rules given by table 3.9.3. These are essentially the same as for SUD except that interpolation is carried out over a smaller range of flow angles. The flow direction applied in figure 3.9.2 is not calculated using u_n and v_n as it is in SUD. Instead it is determined from u_n and v_n , if both are positive, u_n and v_n if the former is positive and the latter negative and so on. It is not clear how the velocity direction is to be calculated if v_n and v_s have different signs and u_n is positive.

Detailed comparisons with other schemes have not been given but one would expect a similar performance to BSUD. Applications of this scheme have been reported by Hassan *et al* (1983b and 1984) and Hassan and Kim (1985a,b). These were thermally stratified flows in two and three dimensions. The skew scheme was only applied to the temperature calculation, not the flow calculation. Significant improvements over UD were reported. Stability problems can be envisaged due to the discontinuous interpolation between regions 2 and 3 and between 6 and 7.

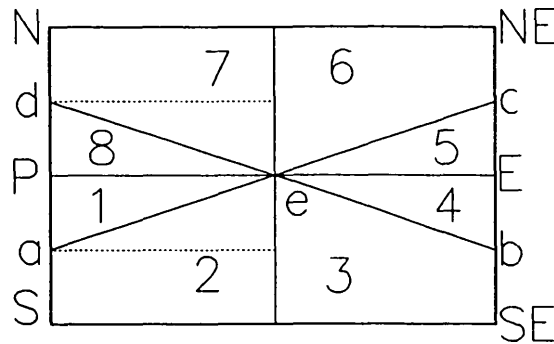


Figure 3.9.2: Interpolation Regions for Hassan Scheme

Region	Interpolated Value
1	Linear interpolation between ϕ_P and ϕ_a
2	ϕ_a
3	ϕ_b
4	Linear interpolation between ϕ_E and ϕ_b
5	Linear interpolation between ϕ_E and ϕ_c
6	ϕ_c
7	ϕ_d
8	Linear interpolation between ϕ_P and ϕ_d

Table 3.9.3: Interpolation Regions for Hassan Scheme

Vector Upwind Differencing (VUD)

Lillington (1981) introduced VUD although consideration to this scheme was given by Raithby (1976). Leschziner (1980) also reported using a scheme equivalent to VUD. The scheme interpolates for ϕ_e in the manner shown in figure 3.9.3 and table 3.9.4. The selected region is the one the flow at e is coming from. The VUD interpolation is always strictly aligned with the flow, in contrast with SUD where it is not. One consequence of this is that VUD has no cross-stream numerical diffusion.

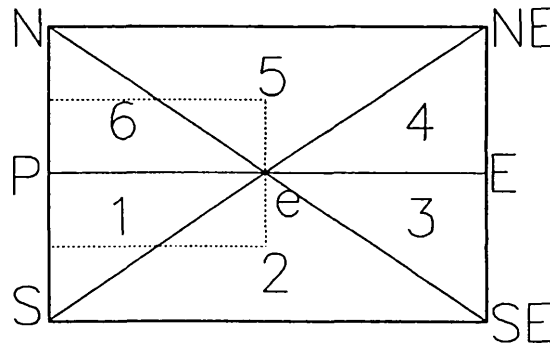


Figure 3.9.3: Interpolation Regions for VUD

Region	Linear Interpolation Between Nodes
1	<i>P</i> and <i>S</i>
2	<i>S</i> and <i>SE</i>
3	<i>SE</i> and <i>E</i>
4	<i>E</i> and <i>NE</i>
5	<i>NE</i> and <i>N</i>
6	<i>N</i> and <i>P</i>

Table 3.9.4: Interpolation Regions for VUD

Leonard (1980) reports boundedness problems with VUD although the accuracy of its solutions is generally better than that of UD. Lillington (1981) found that VUD was slightly more accurate than SUD and reports a little difficulty with boundedness and convergence in some situations. Raithby (1976) found considerable stability problems with VUD. Sharif and Busnaina (1988) report that VUD removed cross-stream numerical diffusion but found considerable boundedness problems, particularly when calculating scalar transport. Smith and Hutton (1982) found VUD was preferable to SUD for a scalar transport problem.

Lillington (1981) recognised that the first order upwinding principle is in error when significant source terms are present. This is due to streamwise variations in ϕ from the upstream point of interpolation to the face centre. A correction to the interpolation was therefore proposed which integrated the source term along the stream-line from the point where the VUD value is calculated to the point of application on the cell face. This did improve further the accuracy of the method. Other workers with this scheme have not included this correction.

No three-dimensional applications of VUD have been reported.

UPSTREAM and CUPID

Patel *et al* (1985b) have proposed a flow oriented scheme that is claimed to be bounded and conservative which they called UPSTREAM (UPwind in STREAMline direction). The description of the scheme is incomplete and a revised version called CUPID (Corner UPwInDing) was later presented (Patel *et al*, 1988). The scheme is based on calculating the flow direction in the corner of the computational volume to determine the interpolation to be used. Interpolation is only used for the inflow fluxes and ϕ_f is never used. Outflow fluxes are not evaluated. The resulting discretisation coefficients are always positive.

The CUPID scheme is illustrated in the following manner. Suppose the flow angle in the SW corner of the control volume is ξ . Inflow will only occur at this corner if $0^\circ < \xi < 90^\circ$. If $0^\circ \leq \xi < 45^\circ$ then

$$\phi_w = \frac{\xi}{45^\circ} \phi_{sw} + \left(1 - \frac{\xi}{45^\circ}\right) \phi_v \quad (3.9.3)$$

$$\phi_s = \phi_s \quad (3.9.4)$$

else if $\xi = 45^\circ$ then

$$\phi_w = \phi_{sw} \quad (3.9.5)$$

$$\phi_s = \phi_{sw} \quad (3.9.6)$$

else if $45^\circ < \xi \leq 90^\circ$ then

$$\phi_w = \phi_v \quad (3.9.7)$$

$$\phi_s = \left(2 - \frac{\xi}{45^\circ}\right) \phi_{sw} + \left(\frac{\xi}{45^\circ} - 1\right) \phi_s. \quad (3.9.8)$$

ϕ_{sw} is not involved in any other interpolation under any other circumstances. The central coefficient is calculated from

$$A_P^c = \sum A_n^c. \quad (3.9.9)$$

The following points can be made concerning CUPID

- (i) The resulting coefficients are unconditionally positive and hence the scheme is bounded.
- (ii) CUPID is not formulated as a flux balance and is therefore not conservative. The proposers effectively concede this point by applying an explicit correction to re-establish conservation. Details of this correction have not been given.
- (iii) The CUPID interpolation (unlike UPSTREAM) appears to be a discontinuous function of the flow angle. This may cause stability problems when applying the scheme to the momentum equations.
- (iv) The presentation of the scheme given by Patel *et al* (1988) is inconsistent. This inconsistency is illustrated in table 3.9.5 for the coefficient A_{sw}^c . Similar inconsistencies are present for other coefficients.

- (v) The use of corner angles is a totally unnecessary complication to the scheme. Velocity components at corners are not usually available and must be interpolated. Further, the determination of the angle from the velocity direction requires the evaluation of an arctan function which would contribute to the computational cost. A practical simpler alternative that operates on the same principle would be to use VUD or SUD interpolation for the inflow fluxes instead of the CUPID interpolation. Equation (3.9.9) could then be used to determine the central coefficient. This scheme would be easier to implement and would have unconditionally positive coefficients and therefore it would be bounded. A correction step would still be required to maintain conservation.

The above five points leads one to conclude that CUPID is not a useful scheme.

Angle	A_{SW}^c	
	Patel <i>et al</i> (1988)	Equations (3.9.3-8)
$0^\circ \leq \xi < 45^\circ$	$\Delta \frac{\xi}{45^\circ} (u_w + v_s)$	$\Delta \frac{\xi}{45^\circ} u_w$
$\xi = 45^\circ$	$\Delta (u_w + v_s)$	$\Delta (u_w + u_s)$
$45^\circ < \xi \leq 90^\circ$	$\Delta \left(2 - \frac{\xi}{45^\circ} \right) (u_w + v_s)$	$\Delta \left(2 - \frac{\xi}{45^\circ} \right) u_w$
Otherwise	0	0

Table 3.9.5: Inconsistency in CUPID Scheme

Kinney and Mahdi

Kinney and Mahdi (1988) have proposed a flow oriented scheme based on averaging over upstream volumes. The scheme is complex so only the salient features are outlined herein.

- (i) All variables are expressed locally as quadratic functions of position. The coefficients of the quadratic are chosen so that it passes through the eight points surrounding P and P itself.
- (ii) Volumetric flow rates through cell faces are found by integrating the velocity over the cell face.
- (iii) Diffusive fluxes are evaluated by integrating the normal gradients across the cell face.
- (iv) Convective fluxes are evaluated as the mass flow rate from (ii) multiplied by a characteristic value of ϕ
- (v) The characteristic value of ϕ is obtained by averaging over an upstream area (figure 3.9.4) defined by the cell face and the velocity directions in the corners of the cell. The extent of this area is a function of the local grid Peclet number.

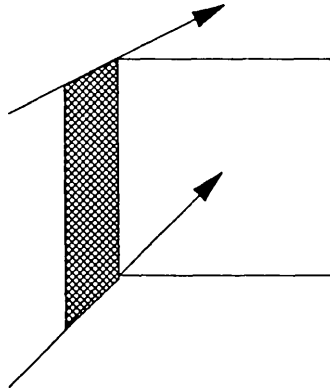


Figure 3.9.4: Averaging Volume for Kinney and Mahdi Scheme (West Face)

The resulting scheme is rather complex and cumbersome to implement. The proposers present two scalar transport calculations. These do not show evidence of numerical diffusion or boundedness problems but do not constitute a particularly rigorous test of the scheme. No theoretical analysis of the boundedness of this scheme is presented and it is not compared with any other scheme.

A three dimensional version of this scheme could be derived but it would be very involved.

Discussion

Flow oriented schemes have been shown to reduce numerical diffusion at the expense of boundedness difficulties. Further they are first order schemes and therefore will not respond rapidly to mesh refinement. Accuracy is reduced in the presence of source terms unless special precautions are taken. They involve off-diagonal nodes which increases the complexity of the scheme, especially in three dimensions. Indeed few three-dimensional applications have been reported.

The most popular flow oriented scheme is SUD although there is some evidence that VUD may be more accurate but less stable. Bounding techniques have been applied to SUD but an unacceptable loss of accuracy is incurred. However it is still more accurate than that of UD.

There is little evidence that these flow oriented methods offer much improvement over LUD and QUICK, especially in three dimensional flows.

3.10 Flux Correction Techniques (Non-Linear)

This section is concerned with non-linear techniques for the discretisation of the convection term. They have the advantage that the solution itself can be inspected and action taken to minimise the numerical diffusion introduced or to eliminate boundedness difficulties. However linear differential equations are discretised to non-linear algebraic equations. Complexities such as invariance and uniqueness are also involved (section 3.6). Several non-linear methods have been proposed and they are discussed below.

Explicit Correction

Sharif and Busnaina (1988) propose a simple correction procedure that can be used with a higher order scheme to maintain bounds. The use of this method with VUD was illustrated. The procedure is as follows.

- (i) Solve the higher order scheme.
- (ii) For each point define

$$\phi_{\max} = \max(\phi_E, \phi_W, \phi_N, \phi_S) \quad (3.10.1)$$

$$\phi_{\min} = \min(\phi_E, \phi_W, \phi_N, \phi_S) \quad (3.10.2)$$

and correct the value at that point by

$$\phi_P = \max\{\phi_{\min}, \min(\phi_{\max}, \phi_P)\}. \quad (3.10.3)$$

- (iii) Repeat step (ii) if necessary.

This approach although very simple and effective is highly non-linear and is not conservative. Computations other than for scalar transport have not been presented. Care would also be needed to avoid erroneous clipping of physical extrema where source terms are present. It would be difficult to show that explicitly corrected techniques satisfy the consistency condition (section 3.6).

CONDIF

Runchal (1987) has proposed a modification to central differencing that maintains boundedness. The scheme was proposed in a finite difference context but is presented here in a finite volume form. The CD interpolations are

$$u_e \phi_e = \frac{1}{2} u_e (\phi_P + \phi_E) \quad (3.10.4)$$

$$u_w \phi_w = \frac{1}{2} u_w (\phi_P + \phi_W) \quad (3.10.5)$$

which can be rewritten as

$$u_e \phi_e = \frac{1}{2} (u_e + |u_e|) \left\{ \phi_P + \frac{1}{2} R (\phi_P - \phi_W) \right\} + \frac{1}{2} (u_e - |u_e|) \left\{ \phi_E - \frac{1}{2} R (\phi_P - \phi_W) \right\} \quad (3.10.6)$$

$$u_w \phi_w = \frac{1}{2} (u_w + |u_w|) \left\{ \phi_W + \frac{1}{2} R (\phi_E - \phi_P) \right\} + \frac{1}{2} (u_w - |u_w|) \left\{ \phi_P - \frac{1}{2} R (\phi_E - \phi_P) \right\} \quad (3.10.7)$$

where

$$R = \frac{\phi_E - \phi_P}{\phi_P - \phi_W}. \quad (3.10.8)$$

The coefficients can be written as

$$A_e^c = -\frac{1}{2} (u_e - |u_e|) + \frac{1}{4R} (u_w + |u_w|) - \frac{1}{4R} (u_w - |u_w|) \quad (3.10.9)$$

$$A_w^c = \frac{1}{2} (u_w - |u_w|) + \frac{1}{4} R (u_e + |u_e|) - \frac{1}{4} R (u_e - |u_e|) \quad (3.10.10)$$

$$A_P^c = \frac{1}{2} (u_e + |u_e|) \left(1 + \frac{1}{2} R \right) - \frac{1}{2} (u_w - |u_w|) \left(1 + \frac{1}{2} R \right) - \frac{1}{4} R (u_e - |u_e|) - \frac{1}{4R} (u_w + |u_w|). \quad (3.10.11)$$

This is still CD but written in a different way. CONDIF is formulated by restricting R by replacing (3.10.8) with

$$R = \max \left\{ \frac{1}{R_{\max}}, \min \left(\frac{\phi_E - \phi_P}{\phi_P - \phi_W}, R_{\max} \right) \right\} \quad (3.10.12)$$

where $R_{\max}$ is a positive constant typically in the range 4 to 10. It can be shown from (3.10.6) and (3.10.7) that this is equivalent to using a blend between CD and UD. Inspection of (3.10.9-11) reveals that the restriction of R to positive values results in unconditionally positive coefficients and hence a bounded scheme.

Scalar transport tests have demonstrated that CONDIF is bounded and more accurate than HYBRID. However flow calculations have not been presented and algorithms for solving the non-linear system have not been considered. This is important because of the potentially high values of the non-linear terms involving R compared to the linear terms.

The disadvantage of this scheme is that it is not conservative. This can be seen from (3.10.6), (3.10.7) and (3.10.12). The expression for the west face value in cell E is not the same as the east face value in cell P . This is necessary condition for a conservative scheme.

SMART

This scheme has been proposed by Gaskell and Lau (1988) who called it Sharp and Monotonic Algorithm for Realistic Transport for convection. For a positive east face velocity we have

$$\phi_e = 3\phi_P - 2\phi_W \quad 0 \leq \frac{\phi_P - \phi_W}{\phi_E - \phi_W} < \frac{1}{6} \quad (3.10.13)$$

$$\phi_e = \phi_E - \phi_P + \phi_W \quad \frac{1}{6} \leq \frac{\phi_P - \phi_W}{\phi_E - \phi_W} < \frac{5}{6} \quad (3.10.14)$$

$$\phi_e = \frac{3}{8}\phi_E + \frac{3}{4}\phi_P - \frac{1}{8}\phi_W \quad \frac{5}{6} \leq \frac{\phi_P - \phi_W}{\phi_E - \phi_W} < 1 \quad (3.10.15)$$

$$\phi_e = \phi_P \quad \text{otherwise.} \quad (3.10.16)$$

The scheme is not actually implemented in the above form but via an equation similar to (3.8.1) with a non-linear λ value in the range $0 \leq \lambda \leq 0.5$. It can be shown that this scheme is second order accurate and satisfies the boundedness condition of Gaskell and Lau given in section 3.6. Scalar test problems suggest that the accuracy is comparable with that of QUICK but without the boundedness difficulties. This is not surprising in view of the fact that the scheme is equivalent to QUICK except where the gradient changes very rapidly.

This scheme was published too late to be included in the testing phase of this study but it is recommended that future workers should test this scheme, especially for the Navier-Stokes equations. In particular it is important to assess the stability of the scheme in view of the convergence problems encountered with low λ (i.e. CD and QUICK) in the solution of (3.8.1).

FCT

Flux Corrected Transport is an algorithm developed by Boris and Book (1973), Book *et al* (1975) and Zalesak (1979). It consists of a non-linear blend of a higher order scheme and a low order scheme (usually UD). The blend is chosen to maximise the amount of the higher order scheme while maintaining boundedness. The algorithm operates in an explicit non-linear time-stepping method.

- (i) Calculate explicitly a provisional solution for the next time step from the bounded lower order scheme.
- (ii) Calculate an anti-diffusive flux as the difference between the unbounded higher order flux and the diffusive lower order flux.
- (iii) Reduce the anti-diffusive flux such that when it is applied to the provisional solution boundedness is not violated. The method for doing this is based on the provisional solution calculated at step (i) and is therefore non-linear.
- (iv) Apply the reduced anti-diffusive flux to the provisional solution.

The critical stage in FCT methods is the evaluation of the reduced flux in step (iii), particularly for multidimensional flow (Zalesak, 1979). The method is relatively complex and an explicit time marching method is needed to solve steady-state problems which makes it relatively expensive.

FRAM

Filter Remedy And Methodology was proposed by Chapman (1981). It is similar to FCT except that the higher order scheme is applied first and then a diffusive flux introduced to bound it. The steps are as follows.

- (i) Calculate explicitly a provisional solution for the next time step from the unbounded higher order scheme.
- (ii) Use a local Lagrangian solution to obtain bounds for the provisional solution.
- (iii) Introduce a local artificial diffusion flux where necessary to bring the provisional unbounded solution into bounds.

FRAM is also an explicit time marching algorithm. It is slightly simpler than FCT and performs at least as well.

FBBM

The Flux Blending Bounding Methodology was introduced by Peric (1986) and can be used to blend any higher order scheme with a lower order scheme to obtain a bounded solution. The algorithm proceeds as follows.

- (i) Calculate candidate convection coefficients based on fluxes expressed as

$$F^C = F^{CB} + \lambda (F^{CU} - F^{CB}). \quad (3.10.17)$$

F^{CB} is the flux evaluated with a bounded scheme (typically UD) and F^{CU} is the flux evaluated with a higher order unbounded scheme. The FBBM seeks to maximise the blending factor λ while achieving boundedness. At the beginning of the calculation the blending factors are set to unity. Each subsequent iteration uses the value obtained on the previous iteration as the starting point (Syed and Chiapetta, 1985). Denote the coefficients obtained from the flux in (3.10.17) as A^{CU} . The coefficient of the bounded scheme are written as A^{CB} .

- (ii) Calculate a provisional solution (ϕ^*) using the above fluxes.
- (iii) Establish bounds on ϕ . The procedure uses nodes that are upstream of the point being considered. That is to say that in two dimensions and if both velocity components are positive then

$$\phi_{\max} = \max(\phi_w^*, \phi_s^*, \phi_{sw}^*) \quad (3.10.18)$$

$$\phi_{\min} = \min(\phi_w^*, \phi_s^*, \phi_{sw}^*). \quad (3.10.19)$$

(iv) Define that value of ϕ_r that is due to convection alone, ϕ_r^c .

$$\phi_r^c = \frac{\sum A_m^{CU} \phi_m^*}{A_r^{CU}} \quad (3.10.20)$$

(v) Test the boundedness of ϕ_r^c . Because it is due to convection alone it is bounded by the upstream values. Therefore the bounds are given by (3.10.18-19) rather than (3.10.1) and (3.10.2) as used by Sharif and Busnaina (1988). If bounds are violated then recalculate the blending factor λ . This is done in steps (vi) to (ix).

(vi) Define a new set of coefficients as

$$A^C = A^{CB} + \lambda_r (A^{CU} - A^{CB}) \quad (3.10.21)$$

to obtain ϕ^{**} , a new estimate for ϕ_r^c .

$$\phi_r^{**} = \frac{\sum \{A_m^{CB} + \lambda_r (A_m^{CU} - A_m^{CB})\} \phi_m^*}{A_r^{CB} + \lambda_r (A_r^{CU} - A_r^{CB})} \quad (3.10.22)$$

(vii) The problem is to maximise λ_r while ensuring that boundedness is maintained. Application of the boundedness criteria (3.10.18-19) to (3.10.22) yields the following inequality.

$$\phi_{\min} \leq \frac{\sum \{A_m^{CB} + \lambda_r (A_m^{CU} - A_m^{CB})\} \phi_m^*}{A_r^{CB} + \lambda_r (A_r^{CU} - A_r^{CB})} \leq \phi_{\max} \quad (3.10.23)$$

This inequality can always be satisfied for $0 \leq \lambda_r \leq 1$ because the lower limit corresponds to a bounded scheme which automatically satisfies (3.10.23). Determine λ_r from the extremes of (3.10.23).

$$\lambda_r = \frac{\phi_{\max} A_r^{CB} - \sum A_m^{CB} \phi_m^*}{\sum (A_m^{CU} - A_m^{CB}) \phi_m^* - \phi_{\max} (A_r^{CU} - A_r^{CB})} \quad \phi_r^{**} > \phi_{\max} \quad (3.10.24)$$

$$\lambda_r = \frac{\phi_{\min} A_r^{CB} - \sum A_m^{CB} \phi_m^*}{\sum (A_m^{CU} - A_m^{CB}) \phi_m^* - \phi_{\min} (A_r^{CU} - A_r^{CB})} \quad \phi_r^{**} < \phi_{\min} \quad (3.10.25)$$

Note that λ_r is unity if bounds are not violated.

(viii) The blending factor λ_r is interpreted as controlling the blend of the flux defined in (3.10.17) with that of the bounded scheme. Blending factors should be associated with cell faces, not nodes so the new blend to replace (3.10.17) should be based on the minimum value of the two new blending factors on either side of the cell face. For instance the new east face flux is therefore

$$F_e^C = F_e^B + \min(\lambda_e, \lambda_r) \lambda_e (F_e^U - F_e^B) \quad (3.10.26)$$

The new blending factor (λ_e^*) is therefore given by

$$\lambda_e^* = \min(\lambda_e, \lambda_r) \lambda_e. \quad (3.10.27)$$

- (ix) If the solution ϕ^* was not bounded go to step (i) and repeat the cycle. This may not be necessary if unbounded solutions are acceptable in the intermediate stages of a large calculation.

The following comments can be made on this algorithm.

- (i) It is a complex bounding method that is not guaranteed to converge in a finite number of steps.
- (ii) The solutions obtained from this algorithm will depend on the initial conditions, under-relaxation factors and so on. The blending factors are determined by the history of the calculation rather than the solution of a defined equation.
- (iii) The blending factors are not allowed to increase, only decrease. Therefore the amount of numerical diffusion introduced increases as the calculation proceeds. This could have serious implications for the accuracy of time dependent calculations. This restriction could be lifted by modifying step (vii).

Applications of the scheme have been reported by Syed *et al* (1985), Syed and Chiapetta (1985) and Claus *et al* (1985). Some authors use the full discrete equation instead of just the convection coefficients in the calculation of the blending factors from (3.10.20). The former two works used both QUICK and SUD as the higher order scheme. The former was found difficult and expensive to converge although the solutions were satisfactory and it was concluded that improvements to the linear solver are necessary. Syed and Chiapetta used SUD as the higher order scheme in two and three dimensional flow calculations. They found that this scheme did have some stability problems but it was considerably more stable than the QUICK version. The bounded schemes were more accurate than HYBRID. Claus *et al* found that the SUD version was not very accurate for flow in a driven cavity, a 6400 point solution having about the same accuracy as a 400 point solution of QUICK.

The QUICK version of this scheme appears to be too unstable to be considered for general use but the SUD version could be used. However it is very complex in three dimensions and has only first order accuracy and slow response to mesh refinement. This would be more accurate than UD but have the slow response to grid refinement and low accuracy when source terms are present due to the nature of SUD. Uniqueness would not be ensured.

Discussion

Non-linear methods have been demonstrated to significantly improve the accuracy of solutions to convection-diffusion equations while maintaining boundedness. With the possible exception of SMART none of the techniques presented above are ideally suited for use in three dimensional steady-state and time dependent calculations. The SMART scheme needs further assessment and was published too late to be included in the testing phase of the current work.

3.11 Closure and Summary of Findings

This chapter has addressed the means for discretising the partial differential equations obtained in chapter 2. The finite volume approach has been adopted and a backward Euler used method for the temporal discretisation. A framework for the discretised source term has been given and the diffusion term has been discretised. Considerable attention has been given to the selection of a technique for the convection term. Criteria for the selection of a discretisation method were discussed. Four categories of techniques were identified and members of these categories examine for high grid Peclet number flows.

Two point interpolations used the two points on either side of the cell face to interpolate for the value on the cell face. Upwind differencing has very poor accuracy but otherwise meets the identified criteria. It is a frequently used method. Central differencing is more accurate at low grid Peclet numbers but suffers badly from boundedness and stability problems at higher values. Bounded schemes have properties very similar to upwind differencing at high grid Peclet numbers and therefore offers little improvement.

Extended schemes involve an additional node on the upstream side of the cell face. They are more accurate and suffer from boundedness and stability problems to a lesser extent than central differencing. QUICK is the most popular of these schemes but LUD appears to be more robust.

Flow oriented methods attempt to use the first order upwind philosophy but in a manner that is aligned with the flow. The truncation error is generally smaller than the upwind truncation error. They generally suffer from boundedness problems and respond slowly to grid refinement. These methods are complex in three dimensions which diminishes their appeal.

Non-linear techniques have not been extensively explored for elliptic flow problems. They show considerable potential but a suitable method was not identified. This is a promising area for the development of new schemes. One such scheme is presented in the next chapter.

The most promising of the existing techniques for complex three dimensional flows are LUD and QUICK. Boundedness difficulties in the form of oscillations are not often observed in laminar flow calculations suggesting that the non-linearities in the equations and couplings between them smooth the solution. Additionally it is harder to observe when quantities have violated bounds. Scalar transport is very susceptible to boundedness problems. The application of LUD and QUICK to the stream fraction equation is therefore not satisfactory. Difficulties may also be expected with the reaction extent, k and ϵ equations where boundedness is particularly important. The only schemes that can currently be applied to these equations is upwind differencing and its derivatives (section 3.7) which have low accuracy.

A new scheme is presented in the next chapter which attempts to redress this problem.

Appendix A.3: Illustration of the Weaknesses of Taylor Series Analysis

This appendix illustrates some of the difficulties that may occur with the application of Taylor series analysis. It gives an example where any term in the Taylor series expansion of the error can be made the largest. An example is also given where each term of the Taylor series becomes vanishingly small but the sum of all the terms is finite. This is an extension of an example given previously by Lai (1983).

Consider the simple equation

$$\frac{d^2\phi}{dx^2} = Pe \frac{d\phi}{dx} \quad (\text{A.3.1})$$

where the Peclet number (Pe) is positive. The boundary conditions are that

$$\begin{aligned} \phi(0) &= \phi_0 \\ \phi(2) &= \phi_2. \end{aligned} \quad (\text{A.3.2})$$

We consider the properties of the exact solution, upwind and central differencing scheme at the point $x = 1$. It is easy to show that the exact solution at this point is

$$\phi(1) = \frac{(\phi_2 - \phi_0)e^{Pe} + \phi_0 e^{2Pe} - \phi_2}{e^{2Pe} - 1} \quad (\text{A.3.3})$$

and the first derivative at this point is

$$\left. \frac{d\phi}{dx} \right|_{x=1} = \frac{(\phi_2 - \phi_0)}{e^{Pe} - e^{-Pe}} Pe. \quad (\text{A.3.4})$$

Repeated differentiation of (A.3.1) and substitution into (A.3.1) gives a recurrence relation for higher derivatives.

$$\frac{d^k\phi}{dx^k} = Pe^{k-1} \frac{d\phi}{dx} \quad (\text{A.3.5})$$

Therefore the higher derivatives at the point $x = 1$ are given by

$$\left. \frac{d^k\phi}{dx^k} \right|_{x=1} = \frac{(\phi_2 - \phi_0)}{e^{Pe} - e^{-Pe}} Pe^k. \quad (\text{A.3.6})$$

We now use upwind differencing to evaluate $\phi(1)$ for use in the convection term discretisation. The upwind value is ϕ_0 and this is upwind approximation to $\phi(1)$.

Expand ϕ as a Taylor series about $x = 1$ to obtain

$$\phi(1) = \phi_0 - \sum_k \frac{(-1)^k}{k!} \frac{d^k\phi}{dx^k}. \quad (\text{A.3.7})$$

The sum in (A.3.7) is the error introduced by using upwind differencing. Utilisation of (A.3.6) gives

$$\phi(1) = \phi_0 - \sum_k E_k \quad (\text{A.3.8})$$

where

$$E_k = \frac{(\phi_2 - \phi_0)(-Pe)^k}{e^{Pe} - e^{-Pe} k!}. \quad (\text{A.3.9})$$

The variation of the first few error terms with Pe is shown in figure A.3.1. It can be seen that any term in the Taylor error series can be made the largest by adjusting Pe . Further higher order terms are not necessarily negligible compared to the first term. This is particularly true for high Pe . Therefore conventional Taylor series error analysis which includes only the first error term may be misleading when applied to this type of problem.

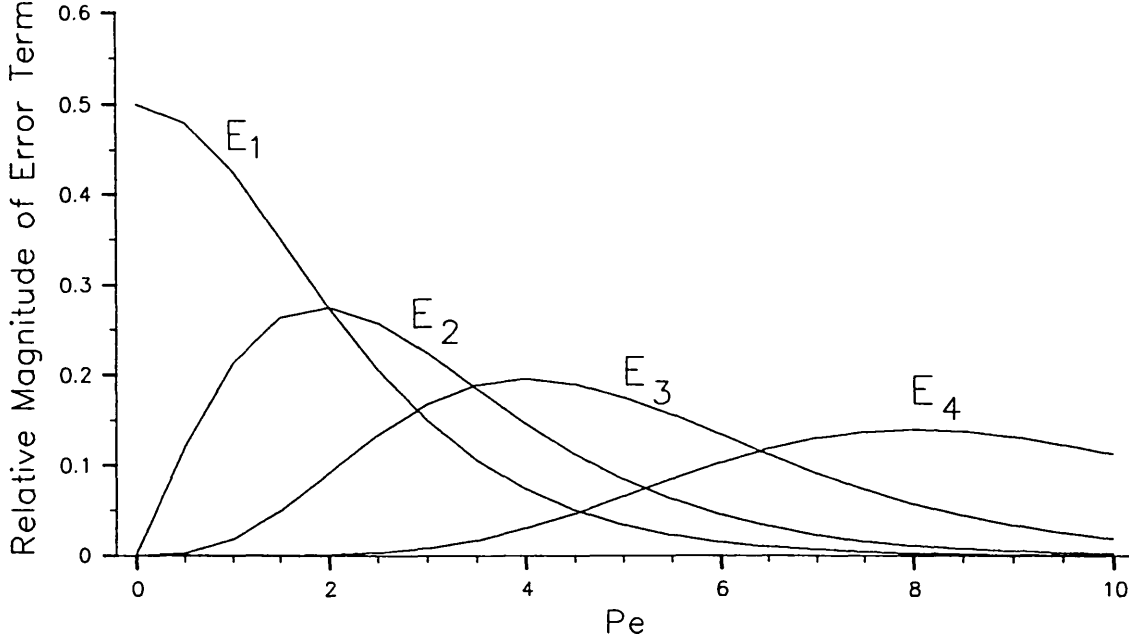


Figure A.3.1: Variation of Error Terms with Pe

In the limit $Pe \rightarrow \infty$ each of the terms of the Taylor series tends to zero. However it is not always safe to conclude that the error term itself is zero under these circumstances. This is readily demonstrated by applying central differencing to (A.3.1). The central differencing result for $\phi(1)$ in the convection term is the average of ϕ_0 and ϕ_2 . We use the Taylor series result

$$\phi_2 = \phi(1) + \sum_k \frac{1}{k!} \frac{d^k \phi}{dx^k} \bigg|_{x=1} \quad (A.3.10)$$

in conjunction with (A.3.7-9) to derive

$$\phi(1) = \frac{1}{2}(\phi_0 + \phi_2) + \sum_k E_{2k}. \quad (A.3.11)$$

Again in the limit $Pe \rightarrow \infty$ each of the terms of the Taylor series tends to zero but their sum does not. Indeed it is easy to sum the series to show that the sum is finite and is $(\phi_2 - \phi_0)/2$. The first term in the error series does not supply a reasonable estimate of the error in the interpolated flux.

The root cause of these difficulties with Taylor series analysis is that it is being applied with finite mesh spacings, or more precisely finite grid Peclet numbers. However the analysis is only reliable when these quantities are vanishingly small. This cannot usually be achieved for convection dominated transport with the current generation of computers, especially in three dimensions.

Appendix B.3: FVM Discretisations on Non-Uniform Cartesian Meshes

This appendix gives the interpolations and discretisations of some of the popular FVM techniques. They are given for equation (4.2.1) in two dimensions on a non-uniform Cartesian mesh. The east face interpolation and the east, west and central (P) discretisation coefficients are given. The north and south coefficients can be obtained by analogy. The generalisation to three dimensions is usually straightforward except for the flow oriented schemes.

Diffusion Discretisation

$$\frac{\partial \phi_e}{\partial x} = \frac{\phi_E - \phi_P}{x_E - x_P} \quad (\text{B.3.1})$$

$$A_E^D = (y_n - y_s) \frac{\Gamma_e}{x_E - x_P} \quad (\text{B.3.2})$$

$$A_W^D = (y_n - y_s) \frac{\Gamma_w}{x_P - x_W} \quad (\text{B.3.3})$$

$$A_P^D = A_E^D + A_W^D + A_N^D + A_S^D \quad (\text{B.3.4})$$

Central Differencing

$$\phi_e = \frac{x_E - x_e}{x_E - x_P} \phi_P + \frac{x_e - x_P}{x_E - x_P} \phi_E \quad (\text{B.3.5})$$

$$A_E^C = -(y_n - y_s) \frac{x_e - x_P}{x_E - x_P} u_e \quad (\text{B.3.6})$$

$$A_W^C = (y_n - y_s) \frac{x_P - x_w}{x_P - x_W} u_w \quad (\text{B.3.7})$$

$$A_P^C = (y_n - y_s) \frac{x_E - x_e}{x_E - x_P} u_e - (y_n - y_s) \frac{x_w - x_W}{x_P - x_W} u_w + \text{north-south terms} \quad (\text{B.3.8})$$

If the discretised flow field is conservative then

$$A_P^C = A_E^C + A_W^C + A_N^C + A_S^C. \quad (\text{B.3.9})$$

Upwind Differencing

$$\phi_e = \frac{1}{2} \left(1 + \frac{u_e}{|u_e|} \right) \phi_P + \frac{1}{2} \left(1 - \frac{u_e}{|u_e|} \right) \phi_E \quad (\text{B.3.10})$$

$$A_E^C = -\frac{1}{2} (u_e - |u_e|) (y_n - y_s) \quad (\text{B.3.11})$$

$$A_W^C = \frac{1}{2} (u_w + |u_w|) (y_n - y_s) \quad (\text{B.3.12})$$

$$A_P^C = \frac{1}{2} (u_e + |u_e|) (y_n - y_s) - \frac{1}{2} (u_w - |u_w|) (y_n - y_s) + \text{north-south terms} \quad (\text{B.3.13})$$

If the discretised flow field is conservative then

$$A_P^C = A_E^C + A_W^C + A_N^C + A_S^C. \quad (\text{B.3.14})$$

Linear Upwind Differencing

$$\phi_s = \frac{1}{2} \left(1 + \frac{u_s}{|u_s|} \right) \left(\phi_P + \frac{x_s - x_P}{x_P - x_W} (\phi_P - \phi_W) \right) + \frac{1}{2} \left(1 - \frac{u_s}{|u_s|} \right) \left(\phi_E + \frac{x_E - x_s}{x_{EE} - x_E} (\phi_E - \phi_{EE}) \right) \quad (\text{B.3.15})$$

$$A_{EE}^C = \frac{1}{2} (u_s - |u_s|) (y_n - y_s) \frac{x_E - x_s}{x_{EE} - x_E} \quad (\text{B.3.16})$$

$$A_E^C = -\frac{1}{2} (u_s - |u_s|) (y_n - y_s) \frac{x_{EE} - x_s}{x_{EE} - x_E} - \frac{1}{2} (u_w - |u_w|) (y_n - y_s) \frac{x_P - x_w}{x_E - x_P} \quad (\text{B.3.17})$$

$$A_W^C = \frac{1}{2} (u_w + |u_w|) (y_n - y_s) \frac{x_w - x_{WW}}{x_W - x_{WW}} + \frac{1}{2} (u_s + |u_s|) (y_n - y_s) \frac{x_s - x_P}{x_P - x_W} \quad (\text{B.3.18})$$

$$A_{WW}^C = -\frac{1}{2} (u_w + |u_w|) (y_n - y_s) \frac{x_w - x_{WW}}{x_W - x_{WW}} \quad (\text{B.3.19})$$

$$A_P^C = \frac{1}{2} (u_s + |u_s|) (y_n - y_s) \frac{x_s - x_W}{x_P - x_W} - \frac{1}{2} (u_w - |u_w|) (y_n - y_s) \frac{x_E - x_w}{x_E - x_P} + \text{north-south terms} \quad (\text{B.3.20})$$

If the discretised flow field is conservative then

$$A_P^C = A_E^C + A_W^C + A_N^C + A_S^C. \quad (\text{B.3.19})$$

QUICK

The formulation for non-uniform meshes presented here follows the algorithm given in the appendix of Leonard's original paper (1979). On a uniform mesh this is equivalent to fitting a parabola through the two points upstream of the cell face and one point downstream. It is possible to re-formulate the scheme such that it is always equivalent to the use of a parabola regardless of the mesh spacing (Syed and Chiapetta, 1985).

$$\phi_s = \frac{1}{2} (\phi_P + \phi_E) - \frac{1}{8} \left(1 + \frac{u_s}{|u_s|} \right) \frac{(x_E - x_P)^2}{x_E - x_W} \left(\frac{\phi_E - \phi_P}{x_E - x_P} - \frac{\phi_P - \phi_W}{x_P - x_W} \right) - \quad (\text{B.3.20})$$

$$\frac{1}{8} \left(1 - \frac{u_s}{|u_s|} \right) \frac{(x_E - x_P)^2}{x_{EE} - x_P} \left(\frac{\phi_{EE} - \phi_E}{x_{EE} - x_E} - \frac{\phi_E - \phi_P}{x_E - x_P} \right)$$

$$A_{EE}^C = \frac{1}{8} (u_s - |u_s|) (y_n - y_s) \frac{(x_E - x_P)^2}{(x_{EE} - x_E)(x_{EE} - x_E)} \quad (\text{B.3.21})$$

$$A_E^C = -\frac{1}{2} (y_n - y_s) u_s + \frac{1}{8} (u_s + |u_s|) (y_n - y_s) \frac{x_E - x_P}{x_E - x_W} - \quad (\text{B.3.22})$$

$$\frac{1}{8} (u_s - |u_s|) (y_n - y_s) \frac{x_E - x_P}{x_{EE} - x_E} - \frac{1}{8} (u_w + |u_w|) (y_n - y_s) \frac{(x_P - x_W)^2}{(x_E - x_W)(x_E - x_P)}$$

$$A_W^C = \frac{1}{2} (y_n - y_s) u_w + \frac{1}{8} (u_s + |u_s|) (y_n - y_s) \frac{(x_E - x_P)^2}{(x_E - x_W)(x_P - x_W)} + \quad (\text{B.3.23})$$

$$\frac{1}{8} (u_w + |u_w|) (y_n - y_s) \frac{x_P - x_W}{x_W - x_{WW}} - \frac{1}{8} (u_w - |u_w|) (y_n - y_s) \frac{x_P - x_W}{x_E - x_W}$$

$$A_{WW}^C = -\frac{1}{8} (u_w + |u_w|) (y_n - y_s) \frac{(x_P - x_W)^2}{(x_W - x_{WW})(x_P - x_{WW})} \quad (\text{B.3.24})$$

$$\begin{aligned}
 A_P^c = & \frac{1}{2}(y_n - y_s)u_e - \frac{1}{2}(y_n - y_s)u_w + \frac{1}{8}(u_e + |u_{ee}|)(y_n - y_s)\frac{x_E - x_P}{x_P - x_W} - \\
 & \frac{1}{8}(u_e - |u_e|)(y_n - y_s)\frac{x_E - x_P}{x_E - x_P} + \frac{1}{8}(u_w + |u_w|)(y_n - y_s)\frac{x_P - x_W}{x_P - x_W} - \\
 & \frac{1}{8}(u_w - |u_w|)(y_n - y_s)\frac{x_P - x_W}{x_E - x_P} + \text{north-south terms}
 \end{aligned} \tag{B.3.25}$$

If the discretised flow field is conservative then

$$A_P^c = A_E^c + A_W^c + A_N^c + A_S^c. \tag{B.3.26}$$

Appendix C.3: Unconditionally Positive Contributions to Convection Discretisation Coefficients

This appendix is a one-dimensional analysis of the construction of the convection discretisation coefficients of a generalised two-point interpolation for a conservative discretisation method. In particular conditions for unconditional positive coefficients are sought. It will be proved that only upwind differencing has this property. The argument can easily be extended to other interpolation formulae involving additional nodes with identical results.

Consider the discretisation of

$$\frac{\partial}{\partial x}(u\phi) = 0. \quad (\text{C.3.1})$$

Let

$$\phi_e = \alpha_e \phi_P + (1 - \alpha_e) \phi_E \quad (\text{C.3.2})$$

$$\phi_w = \alpha_w \phi_W + (1 - \alpha_w) \phi_P. \quad (\text{C.3.3})$$

Assembling the coefficients analogous to those in equation (3.2.16) we have

$$A_E^C = -(1 - \alpha_e) u_e \quad (\text{C.3.4})$$

$$A_W^C = \alpha_w u_w \quad (\text{C.3.5})$$

$$A_P^C = \alpha_e u_e - (1 - \alpha_w) u_w \quad (\text{C.3.6})$$

and the final equation is

$$A_P^C \phi_P = A_E^C \phi_E + A_W^C \phi_W. \quad (\text{C.3.7})$$

The conservation condition requires that an identical interpolation be used for ϕ_e when discretising cell E . This is automatically satisfied by equations (C.3.2-3). The boundedness condition requires that

$$A_P^C = A_E^C + A_W^C \quad (\text{C.3.8})$$

$$A_E^C, A_W^C \geq 0. \quad (\text{C.3.9})$$

The former condition is always met but the latter can only be achieved with upwind differencing where $\alpha_e = 1$ if u_e is positive and zero otherwise. No other scheme will succeed in contributing in an unconditionally positive manner to the convection discretisation coefficients in one dimension.

Appendix D.3: Analysis of Integral Approximations in the FVM

This appendix is concerned with the accuracy of the approximations to the integrals in the FVM. There are two types of these integrals; the integration over a whole finite volume and the integration over a side of a finite volume. These will be examined in turn.

The former integral can be written as

$$\int_{y_s}^{y_n} \int_{x_w}^{x_e} P(x, y) dx dy \quad (D.3.1)$$

where P is an arbitrary function of position. Integrals of this type occur for the accumulation term (3.2.3) and the source term (3.2.4). We expand P as a Taylor series

$$\begin{aligned} P(x, y) = & P_r + (x - x_r) \left. \frac{\partial P}{\partial x} \right|_r + (y - y_r) \left. \frac{\partial P}{\partial y} \right|_r + \frac{1}{2} (x - x_r)^2 \left. \frac{\partial^2 P}{\partial x^2} \right|_r + \\ & \frac{1}{2} (y - y_r)^2 \left. \frac{\partial^2 P}{\partial y^2} \right|_r + (x - x_r)(y - y_r) \left. \frac{\partial^2 P}{\partial x \partial y} \right|_r + \text{HOT} \end{aligned} \quad (D.3.2)$$

where HOT stands for the higher order terms. (D.3.2) is substituted into (D.3.1) and the integral evaluated thus

$$\begin{aligned} \int_{y_s}^{y_n} \int_{x_w}^{x_e} P(x, y) dx dy = & (x_e - x_w)(y_n - y_s) P_r + \frac{1}{2} ((x_e - x_r)^2 - (x_w - x_r)^2) (y_n - y_s) \left. \frac{\partial P}{\partial x} \right|_r + \\ & \frac{1}{2} (x_e - x_w) ((y_n - y_r)^2 - (y_s - y_r)^2) \left. \frac{\partial P}{\partial y} \right|_r + \frac{1}{6} ((x_e - x_r)^3 - (x_w - x_r)^3) (y_n - y_s) \left. \frac{\partial^2 P}{\partial x^2} \right|_r + \\ & \frac{1}{6} (x_e - x_w) ((y_n - y_r)^3 - (y_s - y_r)^3) \left. \frac{\partial^2 P}{\partial y^2} \right|_r + \\ & \frac{1}{4} ((x_e - x_r)^2 - (x_w - x_r)^2) ((y_n - y_r)^2 - (y_s - y_r)^2) \left. \frac{\partial^2 P}{\partial x \partial y} \right|_r + \text{HOT}. \end{aligned} \quad (D.3.3)$$

This can be rewritten as

$$\begin{aligned} \int_{y_s}^{y_n} \int_{x_w}^{x_e} P(x, y) dx dy = & (x_e - x_w)(y_n - y_s) \times \left\{ P_r + \right. \\ & \frac{1}{2} (x_e - 2x_r + x_w) \left. \frac{\partial P}{\partial x} \right|_r + \frac{1}{2} (y_n - 2y_r + y_s) \left. \frac{\partial P}{\partial y} \right|_r + \\ & \frac{1}{6} ((x_e - x_r)^2 + (x_e - x_r)(x_w - x_r) + (x_w - x_r)^2) \left. \frac{\partial^2 P}{\partial x^2} \right|_r + \\ & \frac{1}{6} ((y_n - y_r)^2 + (y_n - y_r)(y_s - y_r) + (y_s - y_r)^2) \left. \frac{\partial^2 P}{\partial y^2} \right|_r + \\ & \left. \frac{1}{4} (x_e - 2x_r + x_w)(y_n - 2y_r + y_s) \left. \frac{\partial^2 P}{\partial x \partial y} \right|_r \right\} + \text{HOT}. \end{aligned} \quad (D.3.4)$$

(D.3.3) shows that the approximation

$$\int_{y_s}^{y_n} \int_{x_w}^{x_e} P(x, y) dx dy = (x_e - x_w)(y_n - y_s) P_r \quad (D.3.5)$$

is second order accurate if

$$\begin{aligned}x_u + x_s &= 2x_r \\ y_s + y_n &= 2y_r\end{aligned}\tag{D.3.6}$$

and first order accurate otherwise. Therefore maximum accuracy is achieved on uniform grids where P is at the mid-point of the computational cell. The error introduced by non-uniform meshes is proportional to the expansion factor. This is $x_s + x_u - 2x_r$ for the x direction and $y_n + y_s - 2y_r$ for the y direction. It is therefore desirable to keep changes in mesh spacing smooth.

Attention is now focussed on the integrals like (3.2.5) which are over cell faces. The integral

$$\int_{y_s}^{y_n} P(x_s, y) dy\tag{D.3.7}$$

will be analysed. Substitution of the Taylor series expansion

$$P(x_s, y) = P_s + (y - y_s) \frac{\partial P}{\partial y} \Big|_s + \frac{1}{2} (y - y_s)^2 \frac{\partial^2 P}{\partial y^2} \Big|_s + \text{HOT}\tag{D.3.8}$$

into (D.3.7) and integration yields

$$\begin{aligned}\int_{y_s}^{y_n} P(x_s, y) dy &= (y_n - y_s) P_s + \frac{1}{2} ((y_n - y_s)^2 - (y_s - y_s)^2) \frac{\partial P}{\partial y} \Big|_s + \\ &\quad \frac{1}{6} ((y_n - y_s)^3 - (y_s - y_s)^3) \frac{\partial^2 P}{\partial y^2} \Big|_s + \text{HOT}.\end{aligned}\tag{D.3.9}$$

This can be rewritten as

$$\begin{aligned}\int_{y_s}^{y_n} P(x_s, y) dy &= (y_n - y_s) \times \left\{ P_s + \frac{1}{2} (y_n - 2y_s + y_s) \frac{\partial P}{\partial y} \Big|_s + \right. \\ &\quad \left. \frac{1}{6} ((y_n - y_s)^2 + (y_n - y_s)(y_s - y_s) + (y_s - y_s)^2) \frac{\partial^2 P}{\partial y^2} \Big|_s \right\} + \text{HOT}.\end{aligned}\tag{D.3.10}$$

The conclusions to be drawn from (D.3.10) are identical to those from (D.3.6). This analysis can easily be extended to three dimensions with identical results.

Chapter 4: Proposal for a New Discretisation Technique

A new discretisation scheme is proposed in this chapter. It employs some of the ideas behind the CONDIF scheme (Runchal, 1987) but is a conservative formulation and boundedness is achieved due to a different mechanism. It will be shown that the new scheme, Self Filtered Central Differencing (SFCD), satisfies many of the criteria presented in the previous chapter and should be subjected to computational tests to assess it on the remaining points. This is done in chapters 5 and 7.

This chapter is organised as follows. Section 4.1 contains the initial proposal for the new scheme including boundary condition treatments. Sections 4.2 to 4.8 analyse SFCD in the light of the criteria discussed in chapter 3. Section 4.9 discusses algorithms that can be used to solve the non-linear discrete equations and finally this chapter is summarised in section 4.10.

4.1 Self Filtered Central Differencing (SFCD)

As recorded in the previous chapter, Runchal (1987) proposed a new method of bounding the central differencing scheme (CONDIF). Unfortunately this is a non-conservative finite difference scheme and so unsuitable for the present application. The ideas leading to CONDIF have been adapted to formulate a conservative bounded scheme, SFCD.

In one dimension on a uniform mesh SFCD can be derived as follows. Consider the steady-state equation

$$\frac{\partial(u\phi)}{\partial x} = \frac{\partial}{\partial x} \left(\Gamma \frac{\partial \phi}{\partial x} \right) + s. \quad (4.1.1)$$

The diffusion terms are discretised in the usual fashion using central differences (section 3.4). This does not cause any difficulties. The interpolation for the convection terms is more problematical. Using central differencing we have

$$\phi_o = \frac{1}{2}(\phi_E + \phi_P) \quad (4.1.2)$$

which can be rewritten as

$$\phi_o = \phi_P + \frac{1}{2}R_o(\phi_P - \phi_W) \quad u_o \geq 0 \quad (4.1.3)$$

$$\phi_o = \phi_E - \frac{1}{2}R_o(\phi_{EE} - \phi_E) \quad u_o < 0 \quad (4.1.4)$$

where

$$R_o = \frac{\phi_E - \phi_P}{\phi_P - \phi_W} \quad u_o \geq 0 \quad (4.1.5)$$

$$R_o = \frac{\phi_E - \phi_P}{\phi_{EE} - \phi_E} \quad u_o < 0. \quad (4.1.6)$$

We now consider the problem of boundedness. Violation of boundedness is manifested as unphysical extrema. At these turning points the ratio

$$\frac{\phi_E - \phi_P}{\phi_P - \phi_W} \quad (4.1.7)$$

becomes negative (figure 4.1.1). At all other points where boundedness is maintained it is positive. We use this observation as the basis of a filter for the central differencing scheme. We replace (4.1.3) to (4.1.6) with

$$\phi_e = \phi_P + \frac{1}{2} r_e (\phi_P - \phi_W) \quad u_e \geq 0 \quad (4.1.8)$$

$$\phi_e = \phi_E - \frac{1}{2} r_e (\phi_{EE} - \phi_E) \quad u_e < 0 \quad (4.1.9)$$

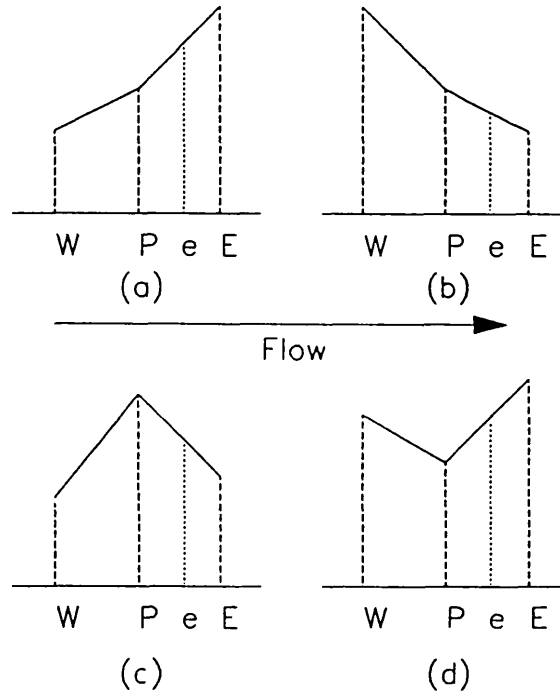


Figure 4.1.1: Variation of R_e .

R_e is positive where the profile is monotonic (cases (a) and (b)). R_e is negative at local extrema (cases (c) and (d))

where

$$r_e = \max \left\{ 0, \min \left(\frac{\phi_E - \phi_P}{\phi_P - \phi_W}, r_{\max} \right) \right\} \quad u_e \geq 0 \quad (4.1.10)$$

$$r_e = \max \left\{ 0, \min \left(\frac{\phi_E - \phi_P}{\phi_{EE} - \phi_E}, r_{\max} \right) \right\} \quad u_e < 0. \quad (4.1.11)$$

This ensures that ratio r_e is non-negative. Whenever a turning point is encountered the filter sets r_e to zero which has the effect of switching over to upwind differencing and thereby ensuring that boundedness is maintained (section 4.4). The $r_{\max}$ upper limit to r_e prevents it from becoming too large and degrading the performance of the linear solver. A value of unity is usually adequate for this parameter. The upper bound is also a way of implementing transportiveness. Both of these points are discussed later. At other times the ratio R_e is used and the scheme is equivalent to central differencing.

It is also worth remarking upon the similarity between SFCD and linear upwind differencing (LUD). If r_e (and hence r_w) is set to unity then SFCD reduces to LUD. This introduces an alternative interpretation of SFCD, namely that of a non-linear blend between UD and LUD. The r_e 's are blending factors restricted between zero and $r_{\max}$.

The discretised version of equation (4.1.1) above can now be written as

$$(A_P + \alpha_P)\phi_P = \alpha_{EE}\phi_{EE} + (A_E + \alpha_E)\phi_E + (A_W + \alpha_W)\phi_W + \alpha_{WW}\phi_{WW} + Si\phi_P + Se \quad (4.1.12)$$

where $Si\phi_P$ and Se are the discretised source terms. The A coefficients are the usual upwind and diffusion coefficients and the α terms are the non-linear corrections introduced by SFCD. The coefficients are given by

$$A_E = -\frac{1}{2}(u_e - |u_e|) + \frac{\Gamma_e}{\Delta} \quad (4.1.13)$$

$$A_W = \frac{1}{2}(u_w + |u_w|) + \frac{\Gamma_w}{\Delta} \quad (4.1.14)$$

$$A_P = \frac{1}{2}(u_e + |u_e|) - \frac{1}{2}(u_w - |u_w|) + \frac{\Gamma_e}{\Delta} + \frac{\Gamma_w}{\Delta} \quad (4.1.15)$$

$$\alpha_{EE} = \frac{1}{4}(u_e - |u_e|)r_e \quad (4.1.16)$$

$$\alpha_E = -\frac{1}{4}(u_e - |u_e|)r_e - \frac{1}{4}(u_w - |u_w|)r_w \quad (4.1.17)$$

$$\alpha_W = \frac{1}{4}(u_w + |u_w|)r_w + \frac{1}{4}(u_e + |u_e|)r_e \quad (4.1.18)$$

$$\alpha_{WW} = -\frac{1}{4}(u_w + |u_w|)r_w \quad (4.1.19)$$

$$\alpha_P = \frac{1}{4}(u_e + |u_e|)r_e - \frac{1}{4}(u_w - |u_w|)r_w \quad (4.1.20)$$

The coefficients on non-uniform meshes can be found in appendix A.4. If the flow is conservative so that

$$u_e = u_w, \quad (4.1.21)$$

then it is easy to show that

$$A_P = A_W + A_E \quad (4.1.22)$$

and

$$\alpha_P = \alpha_{WW} + \alpha_W + \alpha_E + \alpha_{EE}. \quad (4.1.23)$$

The r_{α} factors above are defined in a corresponding manner to r_{β} previously. This ensures that the scheme is conservative. Note that the A 's are independent of the transported scalar and are unconditionally positive. The α 's are not. These latter terms are non-linear and the outermost coefficients are always non-positive. However SFCD contributes other positive terms to the principal coefficients. The scheme is bounded because of the non-linear filter introduced which prevents extrema from appearing in the converged solution. This is discussed in section 4.4. The lack of diagonal dominance may degrade the performance of the linear solver used to solve the above equations.

The implementation of the scheme in two or three dimensions is simple. The extra coefficients can be obtained by analogy with (4.1.13) to (4.1.23). In higher dimensions SFCD is applied to each dimension independently. The coefficients on two dimensional non-uniform meshes can be found in appendix A.4. These are obtained by using the general version of CD (appendix B.3) in place of (4.1.2)

It is important not to associate the problems associated with central differencing to self-filtered central differencing. The filter introduced by (4.1.10) and (4.1.11) results in a set of algebraic equations with quite different properties to those of central differencing despite their equivalence under some circumstances. It is the numerical properties of the SFCD equations that are important, not those of the CD equations. These properties are explored in the following sections. Before this is done the method of implementing boundary conditions with SFCD is introduced.

Zero Gradient Boundary Conditions

The task here is to calculate convection fluxes for faces α and β in figure 4.1.2. The boundary condition on face α is that the derivative normal to the wall is zero.

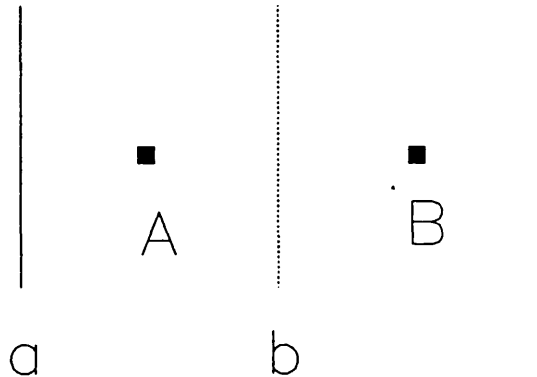


Figure 4.1.2: Notation for Boundary Conditions
The boundary condition is applied to face α .

If the flow at β is towards the boundary then β is treated in the normal manner. If the velocity is in the other direction then we use a Taylor series to expand ϕ about α using the value at A and zero derivative condition to derive

$$\phi_{\beta} = \phi_A + O(\Delta^2). \quad (4.1.24)$$

Hence the use of upwind differencing is second order accurate in this case.

The coefficients in directions other than normal to the wall are not affected.

Fixed Value Boundary Conditions

The boundary condition on the face α is that ϕ takes a specified value, ϕ_α , but its gradient is not defined. Because the value on α is known the flux through α can easily be calculated regardless of the direction of the flow. If the flow through b is directed towards the boundary then ϕ_b is calculated in the usual manner. If the flow is in the other direction then SFCD concepts are employed in the following manner.

$$\phi_b^{CD} = \frac{1}{2}(\phi_A + \phi_B) \quad (4.1.25)$$

so that

$$\phi_b^{SFCD} = \phi_A + \frac{1}{2}r_b(\phi_A - \phi_\alpha) \quad (4.1.26)$$

where

$$r_b = \max\left\{0, \min\left(\frac{\phi_B - \phi_A}{\phi_A - \phi_\alpha}, r_{\max}\right)\right\}. \quad (4.1.27)$$

4.2 Consistency

Taylor Series are used to investigate the consistency of the SFCD interpolation. For the purpose of this analysis the velocity is assumed to be positive so that the interpolation on the east face is given by

$$\phi_e = \phi_P + \frac{1}{2}r_e(\phi_P - \phi_W) \quad (4.2.1)$$

$$r_e = \max(0, \min(R_e, r_{\max})) \quad (4.2.2)$$

$$R_e = \frac{\phi_E - \phi_P}{\phi_P - \phi_W}. \quad (4.2.3)$$

It is apparent that three cases may arise. Namely (i) $r_e = 0$, (ii) $r_e = R_e$ and (iii) $r_e = r_{\max}$. Each of these cases is considered in turn.

Case (i): $r_e = 0$

This is upwind differencing. The truncation error is

$$\phi(x_e) - \phi_e = \frac{1}{2}\Delta \frac{\partial \phi}{\partial x} + \frac{1}{8}\Delta^2 \frac{\partial^2 \phi}{\partial x^2} + O(\Delta^3) \quad (4.2.4)$$

where $\phi(x_e)$ is the exact value of ϕ at x_e . The partial derivatives are always evaluated at x_P in this section. This is the usual first order truncation error for UD.

We now consider the circumstances under which case (i) occurs. That is to say when R_e is negative. From Taylor series we have

$$\phi_P - \phi_W = \Delta \frac{\partial \phi}{\partial x} - \frac{1}{2}\Delta^2 \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{6}\Delta^3 \frac{\partial^3 \phi}{\partial x^3} - O(\Delta^4) \quad (4.2.5)$$

$$\phi_e - \phi_p = \Delta \frac{\partial \phi}{\partial x} + \frac{1}{2} \Delta^2 \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{6} \Delta^3 \frac{\partial^3 \phi}{\partial x^3} + O(\Delta^4) \quad (4.2.6)$$

Because R_e is negative we deduce that in the limit when Taylor series analysis is applicable (i.e. $\Delta \rightarrow 0$),

$$\left| \frac{\partial \phi}{\partial x} \right| < \left| \frac{1}{2} \Delta \frac{\partial^2 \phi}{\partial x^2} \right|. \quad (4.2.7)$$

Combining this with the earlier expression for the truncation error we see that it is bounded by the magnitude of

$$\frac{5}{8} \Delta^2 \frac{\partial^2 \phi}{\partial x^2}. \quad (4.2.8)$$

Therefore the scheme is in fact second order at extrema despite the use of upwind differencing. This is because the first derivative is $O(\Delta)$ and hence the truncation error is $O(\Delta^2)$. Therefore, in multidimensional flow, the numerical diffusion introduced under these circumstances is second order and therefore will not cause serious problems.

Case (ii): $r_e = R_e$

This case is central differencing. The truncation error is

$$\phi(x_e) - \phi_e = -\frac{1}{8} \Delta^2 \frac{\partial^2 \phi}{\partial x^2} + O(\Delta^3) \quad (4.2.9)$$

where the derivative is evaluated at x_p . This is a second order term.

Case (iii): $r_e = r_{\max}$

The SFCD interpolation for this case can be written as

$$\phi_e = \phi_p + \frac{1}{2} r_{\max} (\phi_p - \phi_w). \quad (4.2.10)$$

Substitution of (4.2.5) into this expression yields

$$\phi_e = \phi_p + \frac{1}{2} r_{\max} \left(\Delta \frac{\partial \phi}{\partial x} - \frac{1}{2} \Delta^2 \frac{\partial^2 \phi}{\partial x^2} \right) + O(\Delta^3). \quad (4.2.11)$$

Now, from Taylor series analysis we have

$$\phi(x_e) - \phi_p = \frac{1}{2} \Delta \frac{\partial \phi}{\partial x} + \frac{1}{8} \Delta^2 \frac{\partial^2 \phi}{\partial x^2} + O(\Delta^2). \quad (4.2.12)$$

Substitution of this into (4.2.11) yields the following expression for the truncation error.

$$\phi(x_e) - \phi_e = \frac{1}{2} \Delta (1 - r_{\max}) \frac{\partial \phi}{\partial x} + \frac{1}{8} \Delta^2 (1 - 2r_{\max}) \frac{\partial^2 \phi}{\partial x^2} + O(\Delta^3) \quad (4.2.13)$$

We see that if $r_{\max}$ is unity then the truncation error is second order and the scheme is equivalent to the second order LUD scheme. Otherwise the scheme is first order and some numerical diffusion may occur. However, providing $r_{\max} \geq 1$, mesh refinement will reduce the number of places where case (iii) applies so the truncation error will be reduced.

The conclusion from this section is that SFCD is consistent with the continuum equation and is usually second order accurate. In particular problems with excess numerical diffusion are not expected.

4.3 Conservation

The formulation of SFCD given in section 4.1 is conservative. This is because the discretised equations are obtained by balancing the fluxes over a finite volume as described in section 3.2 so the scheme is conservative over that volume. Further, the flux through each cell face is evaluated in an identical manner for the two cells on either side of the cell face. Therefore the scheme is globally conservative, unlike CONDIF.

It is possible that conservation errors could be introduced by the algorithm used to handle the non-linearities in SFCD. That is to say that the flux in one cell is updated in a slightly different manner to the flux in the next cell and therefore introducing a conservation error. The solution algorithms presented in section 4.9 ensure that this problem does not arise.

4.4 Boundedness

Runchal's CONDIF scheme is unconditionally bounded because it always uses positive coefficients. This is done by switching from CD to UD where extrema occur. In contrast SFCD also switches from CD to UD at extrema but does not have positive coefficients and boundedness is achieved by a different mechanism. This can be explained using the approach of Gaskell and Lau (1987) which was described in section 3.6.

The first criterion for boundedness is that the interpolation used to determine the values on the cell faces is bounded thus

$$\min(\phi_E, \phi_F) \leq \phi_e \leq \max(\phi_E, \phi_F). \quad (4.4.1)$$

SFCD satisfies this condition but this alone is insufficient to ensure boundedness. The second requirement is that upwind differencing is used for fluxes leaving cells containing local extrema. This is also satisfied by SFCD so the scheme is bounded.

4.5 Transportiveness

SFCD is consistent with the transportiveness criterion. Again it is convenient to consider three separate situations.

Case (i) ($r_e = 0$) is upwind differencing and only uses upstream nodes. It is therefore transportive.

Case (ii) ($r_e = R_e$) is equivalent to CD. There is some spatial variation in ϕ . Therefore the use of the downstream node ensures these effects are properly taken into account by the interpolation, as discussed in section 3.6.

Case (iii) ($r_e = r_{max}$) isolates a point from any downstream information when there is a rapid change in the solution. The interpolation is independent of the precise value of the downstream node and therefore information is not erroneously transmitted upstream and hence the transportive criterion is upheld.

4.6 Invariance

The formulation of SFCD is consistent with the requirement for invariance given in section 3.6.

4.7 Uniqueness

SFCD is a non-linear discretisation technique so the issue of uniqueness must be addressed. A proof that SFCD is unique when considering the solution over a single cell is given in appendix B.4. This does not prove that SFCD will be bounded for more general problems. However computational tests of uniqueness will be presented in the following chapter which show that unique solutions are obtained with SFCD.

4.8 Cost-Effectiveness

Cost-effectiveness will be assessed in following chapters where computational results with SFCD are reported. However the analysis presented in section 4.2 suggests that numerical diffusion will be greatly reduced with SFCD and therefore if efficient solution algorithms are found SFCD may be cost-effective. Evidence to support this tentative conclusion will be presented in chapters 5 and 7.

4.9 Solution Algorithms

This section is concerned with methods for the solution of the non-linear discrete equations of SFCD of the form

$$\left(\frac{V_f}{\Delta t} + A_f - Si\right)\phi_f^{n+1} = \frac{V_f}{\Delta t}\phi_f^n + \sum A_m\phi_m^{n+1} + \sum \alpha_m^{n+1}\phi_m^{n+1} + Se. \quad (4.9.1)$$

This is a generalisation of equation (4.1.12). It should be noted that the sum $\sum \alpha_m\phi_m$ includes the P term, unlike $\sum A_m\phi_m$ which excludes it. The temporal discretisation has been performed using the backward Euler scheme as described in section 3.5. The non-linear α coefficients are evaluated implicitly to ensure that the convection flux terms are all based on ϕ at the time step $n + 1$. This will ensure that the SFCD bounding mechanism operates and that time step restrictions are not introduced by the scheme.

An efficient algorithm for the solution of (4.9.1) will keep the computational workload, complexity and storage requirements to a minimum while ensuring that conservation and boundedness are maintained. Due to the complexity and non-linearity of (4.9.1) the only practical way to obtain this solution is by iterative techniques. Two algorithms are described which differ in the manner in which the SFCD terms are treated. The first one (EFCA) treats them in an explicit manner and can easily be incorporated into existing CFD programs which use UD. The second attempts to use a more implicit treatment by linearising the SFCD terms. Steady-state algorithms are obtained by setting Δt to infinity and replacing time step number with iteration number. Complex steady-state CFD calculations usually require some under-relaxation to improve the stability and convergence rate. This is also described.

Explicit Flux Correction Algorithm (EFCA)

This technique uses a compact computational molecule (five points in two dimensions, figure 4.9.1) to implicitly solve the discrete equations. The procedure for advancing from time level n to $n + 1$ is as follows.

1. Calculate a new set of the UD coefficients (A_s) and source terms (Se, Si).
2. Calculate the starting residual, ϵ_s .

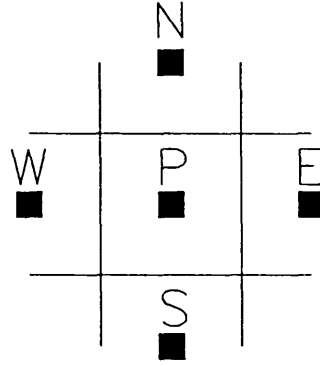


Figure 4.9.1: Compact Computational Molecule

$$\epsilon_I = \sum_{nodes} \left\{ \left(A_P + \frac{V_P}{\Delta t} - Si \right) \phi_P - \sum A_m \phi_m - \sum \alpha_m \phi_m - Se - \frac{V_P}{\Delta t} \phi_P^{n-1} \right\}. \quad (4.9.2)$$

All quantities are evaluated at time level n unless otherwise stated. This residual is a measure of how well the time dependent equation was solved during the previous time step. If the time dependent terms are omitted in this calculation and the residual suitably normalised then this is a measure of the degree of convergence of the steady state equation.

3. The provisional value for ϕ^{n+1} is denoted as ϕ^* . The algorithm improves upon this solution to obtain a new estimate, ϕ^{**} . This field is tested by calculating the new residual and further updates of the solution may be carried out. The steps in this process are detailed below.
4. Use ϕ^n as the first estimate of the solution at the next time step, ϕ^* .
5. Calculate an new estimate of the solution, ϕ^{**} , by solving the implicit equation

$$\left(A_P + \frac{V_P}{\Delta t} - Si \right) \phi_P^{**} = \frac{V_P}{\Delta t} \phi_P^n + \sum A_m \phi_m^{**} + \sum \alpha_m \phi_m^n + Se \quad (4.9.3)$$

for ϕ^{**} . Note that the non-linear SFCD terms are treated explicitly. If an iterative method is used to solve (4.9.3) then it is only necessary to carry out one iteration. The use of several iterations in the solution of (4.9.3) is inefficient because the non-linear SFCD terms will not have been updated and therefore ϕ^{**} would still only be an approximation to the solution of (4.9.1). The additional cost of obtaining ϕ^{**} will not be recouped from the reduction in the number of iterations in the EFCA.

Methods for the solution of (4.9.3) are described in appendices C.4 and D.4.

6. Calculate the new residual, ϵ_F .

$$\epsilon_F = \sum_{nodes} \left\{ \left(A_P + \frac{V_P}{\Delta t} - Si \right) \phi_P^{**} - \sum A_m \phi_m^{**} - \sum \alpha_m \phi_m^{**} - Se - \frac{V_P}{\Delta t} \phi_P^n \right\}. \quad (4.9.4)$$

Note that the SFCD terms have been calculated using the latest estimate of ϕ^{n+1} .

7. If the residual reduction (ϵ_f / ϵ_i) is acceptable use ϕ^{**} as ϕ^{n+1} . Otherwise go to step 5.

Computational tests have shown that if the residuals are reduced by a factor of 0.01 then the accuracy of the whole time-dependent calculation is controlled by the Courant number (section 3.5) and the grid refinement and not by the solution of (4.9.1). The most cost-effective value of residual reduction for steady-state calculations is typically in the range 0.05 to 0.2.

The following comments can be made concerning this algorithm.

- (i) This algorithm has low storage requirements and is reasonably efficient and easy to program. In particular it is straightforward to incorporate into an existing CFD program that uses upwind differencing.
- (ii) The non-linear SFCD equations are solved implicitly. This ensures that the new solution is bounded.
- (iii) The algorithm can be made conservative by careful choice of the solver used in step 5 regardless of the number of iterations used. This is the subject of appendix C.4.
- (iv) Experience suggests that the non-linearities introduced by the need to update the SFCD terms do not cause many additional iterations in the solver loop. The modest additional cost of SFCD is principally caused by the repeated evaluation of the non-linear terms.

The algorithm can be modified by including an explicit corrector step to improve the handling of the non-linear the SFCD terms. This is done by introducing an additional step, 5a, after step 5 in the above algorithm and replacing $**$ with $***$ in steps 6 and 7.

- 5a. Update the current solution explicitly from the 'split' form

$$\left(A_p + \frac{V_p}{\Delta t} - Si \right) \phi_p^{***} = \frac{V_p}{\Delta t} \phi_p^* + \sum A_m \phi_m^{**} + \sum \alpha_m^{**} \phi_m^{**} + Se. \quad (4.9.5)$$

This is most easily done by subtracting (4.9.3) from (4.9.5) to leave

$$\phi_p^{***} = \frac{\sum \alpha_m^{**} \phi_m^{**} - \alpha_m^* \phi_m^*}{A_p + \frac{V_p}{\Delta t} - Si} + \phi_p^{**}. \quad (4.9.6)$$

This modification was highly successful for one dimensional calculations where step 5 can be done exactly and hence the non-linearity of SFCD controlled the number of iterations in the solver loop. For multidimensional calculations where step 5 is not performed exactly then the performance gain was slight. The expense of performing step 5 exactly in these cases is not recovered by the use of this explicit corrector step.

Implicit Flux Correction Algorithm (IFCA)

This technique uses a compact computational molecule (nine points in two dimensions, figure 4.9.2) to solve the discrete equations in a more implicit manner. The procedure for advancing from time level n to $n + 1$ is identical to that in the EFCA with the exception of step 5. Here one solves the equation

$$\left(A_p + \frac{V_p}{\Delta t} - Si \right) \phi_p^{**} = \frac{V_p}{\Delta t} \phi_p^a + \sum A_m \phi_m^{**} + \sum \alpha_m^* \phi_m^{**} + Se \quad (4.9.7)$$

for ϕ^{**} instead of equation (4.9.3). The non-linear SFCD terms have been factorised into explicit and implicit parts. The motivation for this is to increase the rate of convergence of the algorithm by using a more implicit treatment of the non-linear terms. However, the use of equation (4.9.7) instead of (4.9.3) introduces additional points into the computational molecule and a more sophisticated solver is required.

Methods for solving (4.9.7) are described in appendices C.4 and D.4.

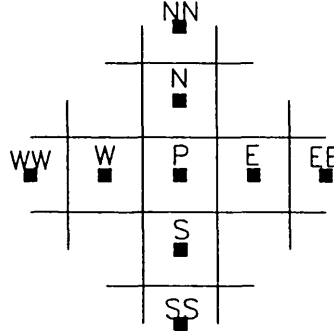


Figure 4.9.2: Extended Computational Molecule

An explicit correction technique can again be introduced into the algorithm. If equation (4.9.5) is subtracted from (4.9.7) then the resulting expression for ϕ^{***} is found to be

$$\phi_p^{***} = \frac{\sum (\alpha_m^{**} - \alpha_m^*) \phi_m^{**}}{A_p + \frac{V_p}{\Delta t} - Si} + \phi_p^{**} \quad (4.9.8)$$

The IFCA is more complex than the EFCA algorithm due to the use of the extended computational molecule. The incorporation of the IFCA into an existing UD CFD program is not as straightforward as for the simpler EFCA. In three dimensions storage requirements increase by six arrays compared to the IFCA. The advantage of the IFCA is that the more implicit treatment of the non-linear SFCD terms may be more cost-effective than the explicit treatment used in the EFCA. This is investigated in the next chapter.

Under-Relaxation for Steady-State Problems

Steady-state problems usually require under-relaxation to improve stability and maximise the convergence rate. The under-relaxed discrete equations are written as

$$A_p \phi_p^{a+1} = \Theta \left(\sum A_m \phi_m^{a+1} + \sum \alpha_m^{a+1} \phi_m^{a+1} + Si \phi_p^{a+1} + Se \right) + (1 - \Theta) A_p \phi_p^a \quad (4.9.9)$$

where Θ is the under-relaxation parameter and superscripts denote the iteration number instead of the time step. This can be rearranged as

$$(\lambda_p - Si) \phi_p^{a+1} = \sum A_m \phi_m^{a+1} + \alpha_m^{a+1} \phi_m^{a+1} + Se \quad (4.9.10)$$

where

$$\lambda_r = \frac{A_r}{\theta} \quad (4.9.11)$$

and

$$S_e = S_e + \frac{1 - \theta}{\theta} A_r \phi_r^n. \quad (4.9.12)$$

Equation (4.9.10) is in the same form as the steady-state version of (4.9.1) and can therefore be solved by either the EFCA or IFCA algorithms.

4.10 Summary

A new discretisation technique for conservation laws has been presented and analysed in the light of the criteria developed in the previous chapter. The new method, which is a variation of one proposed by Runchal (1987), is a non-linear conservative higher order discretisation technique. The higher order feature of the scheme substantially reduces the amount of numerical diffusion present in schemes such as upwind differencing while the non-linear aspect of its formulation ensures that the boundedness criteria is not violated. The scheme has also been shown to satisfy the transportive and invariance criteria. A single cell analysis suggests that the non-linear nature of the scheme will not introduce uniqueness problems. Computational results presented in the next chapter will be used to support this hypothesis.

Two solution algorithms designed to efficiently solve the non-linear equations have been devised. These take advantage of the iterative nature of most linear solvers to solve the non-linear equations introduced by SFCD. The EFCA is based on the compact computational molecule already employed by computational procedures based on upwind differencing with the consequence that existing programs can easily be adapted to use SFCD. The other technique, IFCA, uses an extended computational molecule and a more implicit treatment of the SFCD terms. This is a more complex algorithm with significantly increased storage requirements. Variations of these algorithms involving explicit correction steps have also been developed.

SFCD meets many of the criteria developed in chapter 3. Computational tests and comparisons with existing techniques are required to complete the appraisal of SFCD. In particular it is necessary to consider the cost-effectiveness, uniqueness and choice of solution algorithm. These points are addressed in chapters 5 and 7.

Appendix A.4: SFCD Coefficients

This appendix gives the convection coefficients for SFCD on a non-uniform three dimensional Cartesian mesh. The term being discretised is

$$\nabla \cdot (\underline{u} \phi) \quad (\text{A.4.1})$$

The Cartesian coefficients for the x direction only are given. The other directions can be obtained by analogy. The A_x^c coefficient involves terms from these other directions which are denoted as 'north-south' and 'up-down'.

The convection fluxes are

$$F_s^c = (y_n - y_s)(z_d - z_u)u_s \quad (\text{A.4.2})$$

$$F_u^c = (y_n - y_s)(z_d - z_u)u_u \quad (\text{A.4.3})$$

It is convenient to introduce the following geometrical factors.

$$h_s = \frac{x_e - x_f}{x_e - x_f} \quad (\text{A.4.4})$$

$$h_u = \frac{x_w - x_v}{x_f - x_v} \quad (\text{A.4.5})$$

The basic upwind convection coefficients are given by (A.4.6-8).

$$A_e^c = -\frac{1}{2}(F_s^c - |F_s^c|) \quad (\text{A.4.6})$$

$$A_v^c = \frac{1}{2}(F_u^c + |F_u^c|) \quad (\text{A.4.7})$$

$$A_f^c = -\frac{1}{2}(F_u^c - |F_u^c|) + \frac{1}{2}(F_s^c + |F_s^c|) + \text{north-south terms} + \text{up-down terms} \quad (\text{A.4.8})$$

(A.4.8) can be replaced with

$$A_f^c = A_v^c + A_e^c + A_s^c + A_n^c + A_u^c + A_d^c. \quad (\text{A.4.9})$$

This can easily be verified by subtracting (A.4.9) from (A.4.8) and employing the mass conservation condition

$$F_s^c - F_u^c + F_n^c - F_s^c + F_d^c - F_u^c = 0 \quad (\text{A.4.10})$$

The SFCD correction terms are formulated as follows.

$$\alpha_{ee} = -A_e^c(1 - h_s) \min \left\{ r_{\max}, \max \left(0, \frac{\phi_e - \phi_f}{\phi_{ee} - \phi_e} \right) \right\} \quad (\text{A.4.11})$$

$$\alpha_e = -\alpha_{ee} - \frac{1}{2}(F_u^c - |F_u^c|)(1 - h_u) \min \left\{ r_{\max}, \max \left(0, \frac{\phi_f - \phi_v}{\phi_e - \phi_f} \right) \right\} \quad (\text{A.4.12})$$

$$\alpha_v = -\alpha_{vv} + \frac{1}{2}(F_s^c + |F_s^c|)h_s \min \left\{ r_{\max}, \max \left(0, \frac{\phi_e - \phi_f}{\phi_f - \phi_v} \right) \right\} \quad (\text{A.4.13})$$

$$\alpha_{vv} = -A_v^c h_u \min \left\{ r_{\max}, \max \left(0, \frac{\phi_f - \phi_v}{\phi_v - \phi_{vv}} \right) \right\} \quad (\text{A.4.14})$$

$$\alpha_P = -\frac{1}{2}(F_u^c - |F_u^c|)(1 - h_u) \min \left\{ r_{\max}, \max \left(0, \frac{\phi_P - \phi_U}{\phi_E - \phi_P} \right) \right\} + \frac{1}{2}(F_E^c + |F_E^c|)h_E \min \left\{ r_{\max}, \max \left(0, \frac{\phi_E - \phi_P}{\phi_P - \phi_U} \right) \right\} +$$
(A.4.15)

north-south terms + up-down terms

It can be seen from (A.4.11-15) that

$$\alpha_P = \alpha_{EE} + \alpha_E + \alpha_U + \alpha_{UU} + \alpha_{NN} + \alpha_N + \alpha_S + \alpha_{SS} + \alpha_{DD} + \alpha_D + \alpha_U + \alpha_{UU}.$$
(A.4.16)

Appendix B.4: Proof of Single Cell Uniqueness of SFCD Solution

One potential problem with a non-linear differencing scheme is the possible existence of multiple solutions to the discrete equations. This may cause stability problems as the algorithm to solve the equations switches between solutions and also different results to the same problem may be obtained using different algorithms, initial conditions, relaxation factors etc. A unique solution is highly desirable. This appendix performs a unit cell analysis of SFCD to investigate its properties in this respect.

Consider the simple convection-diffusion equation

$$\frac{\partial \phi}{\partial x} = \Gamma \frac{\partial^2 \phi}{\partial x^2} + s \quad (\text{B.4.1})$$

where the source term, s is taken as a constant. The discrete analogue is

$$\phi_e - \phi_w = \Gamma(\phi_E - 2\phi_P + \phi_W) + Se \quad (\text{B.4.2})$$

where, without loss of generality, the mesh interval is taken as unity. The 'velocity' in equation (B.4.1) is +1 so the SFCD scheme is applied in the form

$$\phi_e = (1 - g_e)\phi_P + g_e\phi_E \quad (\text{B.4.3})$$

$$\phi_w = (1 - g_w)\phi_P + g_w\phi_W. \quad (\text{B.4.4})$$

The terms g_e and g_w are tabulated in tables B.4.1 and B.4.2 respectively.

g_e	R_e
0	$R_e < 0$
$\frac{1}{2}$	$0 \leq R_e \leq r_{\max}$
$\frac{1}{2}r_{\max} \frac{\phi_P - \phi_W}{\phi_E - \phi_P}$	$r_{\max} < R_e$

g_w	R_w
0	$R_w < 0$
$\frac{1}{2}$	$0 \leq R_w \leq r_{\max}$
$\frac{1}{2}r_{\max} \frac{\phi_W - \phi_{WW}}{\phi_P - \phi_W}$	$r_{\max} < R_w$

Table B.4.1: g_e and g_w

R_e is given by (B.4.5) and R_w by (B.4.6).

$$R_e = \frac{\phi_E - \phi_P}{\phi_P - \phi_W} \quad (\text{B.4.5})$$

$$R_w = \frac{\phi_P - \phi_W}{\phi_W - \phi_{WW}} \quad (\text{B.4.6})$$

We apply (B.4.2-6) to a unit cell analysis. This is done by taking ϕ_{WW} , ϕ_W and ϕ_E as fixed and determining ϕ_P .

Define the function $H(\phi)$ as

$$H(\phi) = (\phi_E - \phi)\phi_e + \phi - (\phi - \phi_W)g_w + 2\Gamma\phi - S' \quad (\text{B.4.7})$$

where the g 's are obtained from table B.4.1 by replacing ϕ_r with ϕ in (B.4.5) and (B.4.6). S' contains all the terms independent of ϕ and is given by

$$S' = Se + \phi_v + \Gamma(\phi_f + \phi_v) \quad (\text{B.4.8})$$

and may take any value.

The zeros of H are the values of ϕ_r that satisfy (B.4.2). H is a continuous and single-valued function although the g 's are not. Further in the limit ϕ tends to minus infinity, H tends to minus infinity. Similarly in the limit ϕ tends to plus infinity, H tends to plus infinity. Therefore there always exists at least one zero. Further, if the derivative of H with respect to ϕ is always positive, then there is only one distinct zero.

$$\frac{dH}{d\phi} = (\phi_f - \phi) \frac{dg_s}{d\phi} - g_s + 1 - (\phi - \phi_v) \frac{dg_u}{d\phi} - g_u + 2\Gamma \quad (\text{B.4.9})$$

(The derivative does not exist at two points due to the discontinuities in the g 's. However because H is continuous at these points extra zeros cannot be produced and therefore these discontinuities do not concern us.)

One of the cases in table B.4.2 will arise in conjunction with one from table B.4.3.

g_s	$\frac{dg_s}{d\phi}$	R_s	Case
0	0	$R_s < 0$	(i)
$\frac{1}{2}$	0	$0 \leq R_s \leq r_{\max}$	(ii)
$\frac{1}{2}r_{\max} \frac{\phi - \phi_v}{\phi_f - \phi}$	$\frac{1}{2}r_{\max} \frac{\phi_f - \phi_v}{(\phi_f - \phi)^2}$	$r_{\max} < R_s$	(iii)

Table B.4.2: g_s and $dg_s/d\phi$

R_s is given by (B.4.5) where ϕ_r is replaced by ϕ .

Cases (i) and (iv) obviously do not effect the value of the derivative $dH/d\phi$. It is easy to see that (vi) also has no effect. (ii) and (iii) will each contribute $-1/2$ to the derivative which will still remain positive because Γ is positive. The overall effect of (iii) is a contribution of $+1/2r_{\max}$. We therefore conclude that the derivative is always positive and that SFCD has a single unique distinct solution for this problem.

The above proof is easy to extend for the case a non-uniform velocity or two or three spatial dimensions. Real computations involve more than one computational cell and the extension of the above theory to this case has not yet been devised or a counter-example found. However, if the unit cell analysis did not have a unique solution then one could definitely prove that SFCD did not have a unique solution for general problems. The above proof corroborates the numerical studies (to be presented in chapter 5) which have not indicated uniqueness problems. There is therefore significant circumstantial evidence that the SFCD scheme has a unique solutions although a complete proof has not been given.

g_w	$\frac{dg_w}{d\phi}$	R_w	Case
0	0	$R_w < 0$	(iv)
$\frac{1}{2}$	0	$0 \leq R_w \leq r_{\max}$	(v)
$\frac{1}{2} r_{\max} \frac{\phi_v - \phi_{vv}}{\phi - \phi_v}$	$-\frac{1}{2} r_{\max} \frac{\phi_v - \phi_{vv}}{(\phi - \phi_v)^2}$	$r_{\max} < R_w$	(vi)

Table B.4.3: g_w and $dg_w/d\phi$
 R_w is given by (B.4.6) where ϕ_r is replaced by ϕ .

Appendix C.4: A Conservative Solver

Algorithms for updating the solution of the implicit set of equations (4.9.3) and (4.9.7) are given in this appendix. This is one cycle of an iterative solution algorithm, the complete solution is obtained using the procedures given in section 4.9. A general equation corresponding to three dimensional time dependent problems using a conservative discretisation with an extended computational molecule is given. Simpler problems can be solved by eliminating superfluous terms from the algorithm given below.

The equation to be solved is taken as

$$\begin{aligned} \left(A_P + Si + \frac{V_P}{\Delta t} \right) \phi_P^{n+1} = & A_{EE} \phi_{EE}^{n+1} + A_E \phi_E^{n+1} + A_V \phi_V^{n+1} + A_{VV} \phi_{VV}^{n+1} + \\ & A_{NN} \phi_{NN}^{n+1} + A_N \phi_N^{n+1} + A_S \phi_S^{n+1} + A_{SS} \phi_{SS}^{n+1} + \\ & A_{DD} \phi_{DD}^{n+1} + A_D \phi_D^{n+1} + A_U \phi_U^{n+1} + A_{UU} \phi_{UU}^{n+1} + \frac{V_P}{\Delta t} \phi_P^n + Se. \end{aligned} \quad (C.4.1)$$

Here DD , D , U and UU are the nodes in the third coordinate direction (Downstream and Upstream) for three dimensional flows. Equations (4.9.3) and (4.9.7) can be written in this form. Writing ϕ^* as the best estimate for ϕ^{n+1} , then a method for improving this estimate to ϕ^{**} is described.

Equation (C.4.1) is solved by treating it as a series of one-dimensional sub-problems each of which can be solved exactly by a Penta-Diagonal Matrix Algorithm (PDMA) (appendix D.4). The sequence of sub-problems is

$$\begin{aligned} \left(A_P + Si + \frac{V_P}{\Delta t} \right) \phi_P^* = & A_{EE} \phi_{EE}^* + A_E \phi_E^* + A_V \phi_V^* + A_{VV} \phi_{VV}^* + \\ & A_{NN} \phi_{NN}^* + A_N \phi_N^* + A_S \phi_S^* + A_{SS} \phi_{SS}^* + \\ & A_{DD} \phi_{DD}^* + A_D \phi_D^* + A_U \phi_U^* + A_{UU} \phi_{UU}^* + \frac{V_P}{\Delta t} \phi_P^n + Se \end{aligned} \quad (C.4.2)$$

for ϕ^* . Then solve

$$\begin{aligned} \left(A_P + Si + \frac{V_P}{\Delta t} \right) \phi_P^{**} = & A_{EE} \phi_{EE}^{**} + A_E \phi_E^{**} + A_V \phi_V^{**} + A_{VV} \phi_{VV}^{**} + \\ & A_{NN} \phi_{NN}^{**} + A_N \phi_N^{**} + A_S \phi_S^{**} + A_{SS} \phi_{SS}^{**} + \\ & A_{DD} \phi_{DD}^{**} + A_D \phi_D^{**} + A_U \phi_U^{**} + A_{UU} \phi_{UU}^{**} + \frac{V_P}{\Delta t} \phi_P^n + Se \end{aligned} \quad (C.4.3)$$

for ϕ^{**} . Finally solve

$$\begin{aligned} \left(A_P + Si + \frac{V_P}{\Delta t} \right) \phi_P^{***} = & A_{EE} \phi_{EE}^{***} + A_E \phi_E^{***} + A_V \phi_V^{***} + A_{VV} \phi_{VV}^{***} + \\ & + A_{NN} \phi_{NN}^{***} + A_N \phi_N^{***} + A_S \phi_S^{***} + A_{SS} \phi_{SS}^{***} + \\ & A_{DD} \phi_{DD}^{***} + A_D \phi_D^{***} + A_U \phi_U^{***} + A_{UU} \phi_{UU}^{***} + \frac{V_P}{\Delta t} \phi_P^n + Se \end{aligned} \quad (C.4.4)$$

for ϕ^{***} .

We now consider transport of conserved variables such as stream fraction in confined flows (e.g. stirred vessel). Consider the three dimensional temporally discretised version of equation (3.2.14). Sum this equation over all the nodes in the computational domain. The internal fluxes cancel and there are no boundary fluxes due to the confined nature of the flow. The remaining terms are

$$\sum_{nodes} V_p \phi_p^{n+1} = \Delta t \sum_{nodes} (Si \phi_p^{n+1} + Se). \quad (C.4.5)$$

Equation (C.4.5) states that the net change in ϕ during a time step is balanced by the sources and sinks within the solution domain. We therefore desire that the solution to (C.4.1) satisfies (C.4.5). This can clearly be done if (C.4.5) is solved exactly. However this is prohibitively expensive and a finite residual is accepted (section 4.9).

The algorithm (C.4.2-4) is not consistent with (C.4.5). This is because the A_p coefficient includes contributions from all the fluxes, not solely those that are being implicitly solved. Hence the fluxes F^c and F^D are not simultaneously updated. Therefore these fluxes do not cancel out exactly and a conservation error is introduced.

The small conservation error is not important for the momentum, k and ϵ transport equations. It is very important for the transport equations describing chemical reactions, especially in confined reacting flows. If a conservation error occurs then chemical reagents will either be incorrectly introduced into the flow or removed from it. At the end of a calculation it will not be possible to perform a mass balance and the reaction yields will be suspect. This effect will be particularly serious for long reaction times when the conservation errors can accumulate. We therefore conclude that (C.4.2-4) are inappropriate for stream fraction and reaction extent calculations.

This objection can be overcome by the following modification to the ADI method which was stimulated by a suggestion made by Lynch (1985). Rewrite (C.4.1) as

$$\begin{aligned} \left(A_{px} + A_{py} + A_{pz} + Si + \frac{V_p}{\Delta t} \right) \phi_p^{n+1} = & A_{EE} \phi_{EE}^{n+1} + A_E \phi_E^{n+1} + A_V \phi_V^{n+1} + A_{VV} \phi_{VV}^{n+1} \\ & + A_{NN} \phi_{NN}^{n+1} + A_N \phi_N^{n+1} + A_S \phi_S^{n+1} + A_{SS} \phi_{SS}^{n+1} + \\ & A_{DD} \phi_{DD}^{n+1} + A_D \phi_D^{n+1} + A_U \phi_U^{n+1} + A_{UU} \phi_{UU}^{n+1} + \frac{V_p}{\Delta t} \phi_p^n + Se \end{aligned} \quad (C.4.7)$$

where

$$A_p = A_{px} + A_{py} + A_{pz}. \quad (C.4.8)$$

A_{px} is the part of A_p that arises from the discretisation of the east and west convection and diffusion fluxes. Similarly for A_{py} and A_{pz} in the other two directions. The solution of (C.4.7) is again broken down into three sub-problems. Firstly solve

$$\begin{aligned} \left(A_{px} + Si + \frac{V_p}{\Delta t} \right) \phi_p^* + (A_{py} + A_{pz}) \phi_p^* = & A_{EE} \phi_{EE}^* + A_E \phi_E^* + A_V \phi_V^* + A_{VV} \phi_{VV}^* + \\ & A_{NN} \phi_{NN}^* + A_N \phi_N^* + A_S \phi_S^* + A_{SS} \phi_{SS}^* + \\ & A_{DD} \phi_{DD}^* + A_D \phi_D^* + A_U \phi_U^* + A_{UU} \phi_{UU}^* + \frac{V_p}{\Delta t} \phi_p^n + Se \end{aligned} \quad (C.4.9)$$

for ϕ^* . This can be done to machine precision by the method given in the next appendix. The next sub-problem is

$$\begin{aligned} \left(A_{py} + Si + \frac{V_p}{\Delta t} \right) \phi_p^{**} + A_{px} \phi_p^* + A_{pz} \phi_p^* = A_{EE} \phi_{EE}^* + A_E \phi_E^* + A_W \phi_W^* + A_{WW} \phi_{WW}^* + \\ A_{NN} \phi_{NN}^{**} + A_N \phi_N^{**} + A_S \phi_S^{**} + A_{SS} \phi_{SS}^{**} + \\ A_{DD} \phi_{DD}^* + A_D \phi_D^* + A_U \phi_U^* + A_{UU} \phi_{UU}^* + \frac{V_p}{\Delta t} \phi_p^* + Se \end{aligned} \quad (C.4.10)$$

which is most easily done by subtracting (C.4.9) to leave

$$\begin{aligned} \left(Si + \frac{V_p}{\Delta t} \right) (\phi_p^{**} - \phi_p^*) + A_{py} (\phi_p^{**} - \phi_p^*) = A_{NN} (\phi_{NN}^{**} - \phi_{NN}^*) + A_N (\phi_N^{**} - \phi_N^*) + \\ A_S (\phi_S^{**} - \phi_S^*) + A_{SS} (\phi_{SS}^{**} - \phi_{SS}^*) \end{aligned} \quad (C.4.11)$$

which again can be solved to machine precision. Finally solve

$$\begin{aligned} \left(A_{pz} + Si + \frac{V_p}{\Delta t} \right) \phi_p^{**} + A_{px} \phi_p^* + A_{py} \phi_p^{**} = A_{EE} \phi_{EE}^* + A_E \phi_E^* + A_W \phi_W^* + A_{WW} \phi_{WW}^* + \\ A_{NN} \phi_{NN}^{**} + A_N \phi_N^{**} + A_S \phi_S^{**} + A_{SS} \phi_{SS}^{**} + \\ A_{DD} \phi_{DD}^{**} + A_D \phi_D^{**} + A_U \phi_U^{**} + A_{UU} \phi_{UU}^{**} + \frac{V_p}{\Delta t} \phi_p^* + Se \end{aligned} \quad (C.4.12)$$

by subtracting (C.4.10) from it to leave

$$\begin{aligned} \left(Si + \frac{V_p}{\Delta t} \right) (\phi_p^{**} - \phi_p^*) + A_{pz} (\phi_p^{**} - \phi_p^*) = A_{DD} (\phi_{DD}^{**} - \phi_{DD}^*) + A_D (\phi_D^{**} - \phi_D^*) + \\ A_U (\phi_U^{**} - \phi_U^*) + A_{UU} (\phi_{UU}^{**} - \phi_{UU}^*) \end{aligned} \quad (C.4.13)$$

which can also be easily solved.

Note that all east-west fluxes are updated simultaneously in the solution of (C.4.9) to (C.4.13). Similarly for the other two coordinate directions. Therefore if the internal fluxes are summed exact cancellation will take place and therefore (C.4.5) is satisfied at any point regardless of magnitude of the error in the solution of (C.4.1).

Appendix D.4: A Cyclic Penta-Diagonal Solver

This appendix describes an algorithm for the solution of a penta-diagonal matrix with cyclic boundary conditions. These frequently arise when periodic or cyclic boundary conditions are used to relate the values of ϕ on one edge of the solution domain to values on the opposite edge.

The solver described herein, which is based on the well known Thomas tri-diagonal matrix algorithm described by many authors (e.g. Bolour-Froushan, 1986), is a direct solver requiring $O(n)$ operations where n is the number of equations in the system. The system of equations to be solved is written as

$$A_i \phi_i = B_i \phi_{i+2} + C_i \phi_{i+1} + D_i \phi_{i-1} + E_i \phi_{i-2} + F_i \quad i = 1, \dots, n \quad (D.4.1)$$

where subscripts are calculated using modulo n arithmetic. This system has a penta-diagonal matrix with three elements in the top right and bottom left corners due to the cyclic boundary conditions. The algorithm can be significantly simplified for systems with fewer diagonals or without the cyclic boundary conditions.

The algorithm starts by eliminating the lower diagonals from the i th equation. It is not possible to do this entirely because of the cyclic nature of the problem. We therefore elect to keep terms in ϕ_n and ϕ_{n-1} . The resulting family of equations is

$$A'_i \phi_i = B'_i \phi_{i+2} + C'_i \phi_{i+1} + G'_i \phi_n + H'_i \phi_{n-1} + F'_i. \quad (D.4.2)$$

It is obtained by substituting the first equation into the second to eliminate ϕ_1 from it, the first and the new second equation into the third to eliminate ϕ_1 and ϕ_2 and so on. The new coefficients are given below.

$$A'_1 = A_1 \quad (D.4.3)$$

$$B'_1 = B_1 \quad (D.4.4)$$

$$C'_1 = C_1 \quad (D.4.5)$$

$$F'_1 = F_1 \quad (D.4.6)$$

$$G'_1 = D_1 \quad (D.4.7)$$

$$H'_1 = E_1 \quad (D.4.8)$$

$$A'_2 = A_2 - C_1 \frac{D_2}{A_1} \quad (D.4.9)$$

$$B'_2 = B_2 \quad (D.4.10)$$

$$C'_2 = C_2 + B_1 \frac{D_2}{A_1} \quad (D.4.11)$$

$$F'_2 = F_2 + F_1 \frac{D_2}{A_1} \quad (D.4.12)$$

$$G'_2 = E_2 + D_1 \frac{D_2}{A_1} \quad (D.4.13)$$

$$H'_2 = E_1 \frac{D_2}{A_1} \quad (D.4.14)$$

$$A'_i = A_i - B'_{i-2} \frac{E_i}{A'_{i-2}} - \alpha_i C'_{i-1} \quad (D.4.15)$$

$$B'_i = B_i \quad (D.4.16)$$

$$C'_i = C_i + \alpha_i B'_{i-1} \quad (D.4.17)$$

$$F'_i = F_i + F'_{i-2} \frac{E_i}{A'_{i-2}} - \alpha_i F'_{i-1} \quad (D.4.18)$$

$$G'_i = G'_{i-2} \frac{E_i}{A'_{i-2}} - \alpha_i G'_{i-1} \quad (D.4.19)$$

$$H'_i = H'_{i-2} \frac{E_i}{A'_{i-2}} - \alpha_i H'_{i-1} \quad (D.4.20)$$

$$\alpha_i = \frac{D_i}{A'_{i-1}} + \frac{E_i C'_{i-2}}{A'_{i-1} A'_{i-2}} \quad (D.4.21)$$

Equations (D.4.15-21) only apply for $i > 2$

The algorithm next eliminates the upper diagonals from the equations again retaining terms involving ϕ_n and ϕ_{n-1} . The $n-2$ th equation is already in the correct format.

$$A'_{n-2} \phi_{n-2} = (B'_{n-2} + G'_{n-2}) \phi_n + (C'_{n-2} + H'_{n-2}) \phi_{n-1} + F'_{n-2} \quad (D.4.22)$$

This is substituted into the $n-3$ th equation to eliminate ϕ_{n-2} from it and so on down to the first equation. The equations are now

$$A''_i \phi_i = G''_i \phi_n + H''_i \phi_{n-1} + F''_i \quad (D.4.23)$$

for $i < n-1$. The new coefficients are given below.

$$A''_{n-2} = A'_{n-2} \quad (D.4.24)$$

$$F''_{n-2} = F'_{n-2} \quad (D.4.25)$$

$$G''_{n-2} = G'_{n-2} + B'_{n-2} \quad (D.4.26)$$

$$H''_{n-2} = H'_{n-2} + C'_{n-2} \quad (D.4.27)$$

$$A''_{n-3} = A'_{n-3} \quad (D.4.28)$$

$$F''_{n-3} = F'_{n-3} + F''_{n-2} \frac{C'_{n-3}}{A'_{n-2}} \quad (D.4.29)$$

$$G''_{n-3} = G'_{n-3} + G''_{n-2} \frac{C'_{n-3}}{A'_{n-2}} \quad (D.4.30)$$

$$H''_{n-3} = H'_{n-3} + H''_{n-2} \frac{C'_{n-3}}{A'_{n-2}} + B'_{n-3} \quad (D.4.31)$$

$$A''_i = A'_i \quad (D.4.32)$$

$$F''_i = F'_i + F''_{i+1} \frac{C'_{i+1}}{A'_{i+1}} + F''_{i+2} \frac{B'_{i+1}}{A'_{i+2}} \quad (D.4.33)$$

$$G''_i = G'_i + G''_{i+1} \frac{C'_{i+1}}{A'_{i+1}} + G''_{i+2} \frac{B'_{i+1}}{A'_{i+2}} \quad (D.4.34)$$

$$H''_i = H'_i + H''_{i+1} \frac{C'_i}{A''_{i+1}} + H''_{i+2} \frac{B'_i}{A''_{i+2}} \quad (D.4.35)$$

Equations (D.4.32-5) apply for $i < n-3$

Once ϕ_n and ϕ_{n-1} are known (D.4.23) can be used to solve the system. These quantities can be determined from the first two and last two equations of the set.

$$A''_2 \phi_2 = G''_2 \phi_n + H''_2 \phi_{n-1} + F''_2 \quad (D.4.36)$$

$$A''_1 \phi_1 = G''_1 \phi_n + H''_1 \phi_{n-1} + F''_1 \quad (D.4.37)$$

$$A'_n \phi_n = B'_n \phi_2 + C'_n \phi_1 + G'_n \phi_n + H'_n \phi_{n-1} + F'_n \quad (D.4.38)$$

$$A'_{n-1} \phi_{n-1} = B'_{n-1} \phi_1 + C'_{n-1} \phi_n + G'_{n-1} \phi_n + H'_{n-1} \phi_{n-1} + F'_{n-1} \quad (D.4.39)$$

Combining these four equations we obtain

$$\phi_n = \frac{X_n F''_{n-1} - X_{n-1} F''_n}{X_{n-1} Y_n - X_n Y_{n-1}} \quad (D.4.40)$$

$$\phi_{n-1} = \frac{Y_{n-1} F''_n - Y_n F''_{n-1}}{X_{n-1} Y_n - X_n Y_{n-1}} \quad (D.4.41)$$

where

$$X_{n-1} = H'_{n-1} + H''_1 \frac{B'_{n-1}}{A''_1} - A'_{n-1} \quad (D.4.42)$$

$$Y_{n-1} = G'_{n-1} + G''_1 \frac{B'_{n-1}}{A''_1} + C'_{n-1} \quad (D.4.43)$$

$$F''_{n-1} = F'_{n-1} + F''_1 \frac{B'_{n-1}}{A''_1} \quad (D.4.44)$$

$$X_n = H'_n + H''_2 \frac{B'_n}{A''_2} + H''_1 \frac{C'_n}{A''_1} \quad (D.4.45)$$

$$Y_n = G'_n + G''_2 \frac{B'_n}{A''_2} + G''_1 \frac{C'_n}{A''_1} - A'_n \quad (D.4.46)$$

$$F''_n = F'_n + F''_2 \frac{B'_n}{A''_2} + F''_1 \frac{C'_n}{A''_1} \quad (D.4.47)$$

The algorithm can be summarised as follows. Use equations (D.4.3-21) to eliminate the lower diagonals from the second to n th equation while retaining terms involving ϕ_n and ϕ_{n-1} . Use equations (D.4.24-35) to eliminate the upper diagonals from the $n-2$ th equation to first equation again retaining terms involving ϕ_n and ϕ_{n-1} . Determine ϕ_n and ϕ_{n-1} from equations (D.4.40-47) and determine the remaining values from (D.4.23).

Chapter 5: Scalar Transport Tests of SFCD

5.1 Introduction

This chapter is concerned with scalar transport tests of SFCD and comparison with other schemes from chapter 3. It is also concerned with assessing those aspects of SFCD that could not be analysed in the previous chapter. In particular, cost-effectiveness and uniqueness are considered.

Scalar transport tests are particularly suited to the investigation of boundedness difficulties and potential uniqueness problems associated with a non-linear discretisation. The effects of numerical diffusion can also be readily demonstrated. They are not as well-suited to the investigation of the cost-effectiveness of a scheme when applied to the full set of equations governing laminar or turbulent flow for reasons discussed in chapter 3. SFCD flow calculations are presented in chapter 7.

Results from a number of tests are presented. Section 5.2 describes some simple one-dimensional convection-diffusion problems frequently used to test discretisations these are used to investigate the accuracy, boundedness and $r_{\max}$ sensitivity of the calculations. Simple two-dimensional scalar transport problems are presented in sections 5.3 and 5.4. These are used to look at numerical diffusion and different solution algorithms. The final problems (section 5.5) are time dependent scalar transport tests which are sensitive to numerical diffusion and are designed to test the scheme and algorithms in more complex situations. Finally, conclusions are drawn in section 5.6.

5.2 One Dimensional Convection-Diffusion

The purpose of these test problems is to verify that the scheme is bounded and acceptably accurate for high Peclet number convection-diffusion equations. Because the problems are one-dimensional, cross-stream numerical diffusion does not appear. Two problems are considered. The first is a simple back-diffusion case and the second involves a source term. In both cases the equation to be solved is

$$\frac{\partial \phi}{\partial x} - \frac{1}{Pe} \frac{\partial^2 \phi}{\partial x^2} = s \quad (5.2.1)$$

where the Peclet number, Pe , is always positive.

The first problem has boundary conditions

$$\phi(0) = 0 \quad \text{and} \quad \phi(1) = 1. \quad (5.2.2)$$

and no source term (i.e. $s = 0$). The analytical solution is

$$\phi(x) = \frac{e^{Pe x} - 1}{e^{Pe} - 1}. \quad (5.2.3)$$

This corresponds to flow from left to right with some back diffusion from the exit boundary which produces a rapid change in ϕ close to this edge. When the grid Peclet number, $Pe_\Delta (= \Delta Pe)$, is high this sharp change makes it a numerically demanding problem.

The equation is solved using twenty intervals and UD, CD, LUD, QUICK and SFCD (with $r_{\max} = 5$) are applied. The latter scheme is solved using the IFCA without a correction step. The results are shown in figure 5.2.1 for three values of Pe_Δ . In the case when $Pe_\Delta = 2$ the SFCD solution cannot be distinguished from the exact solution in the figure. The results clearly demonstrate the superior accuracy of SFCD over

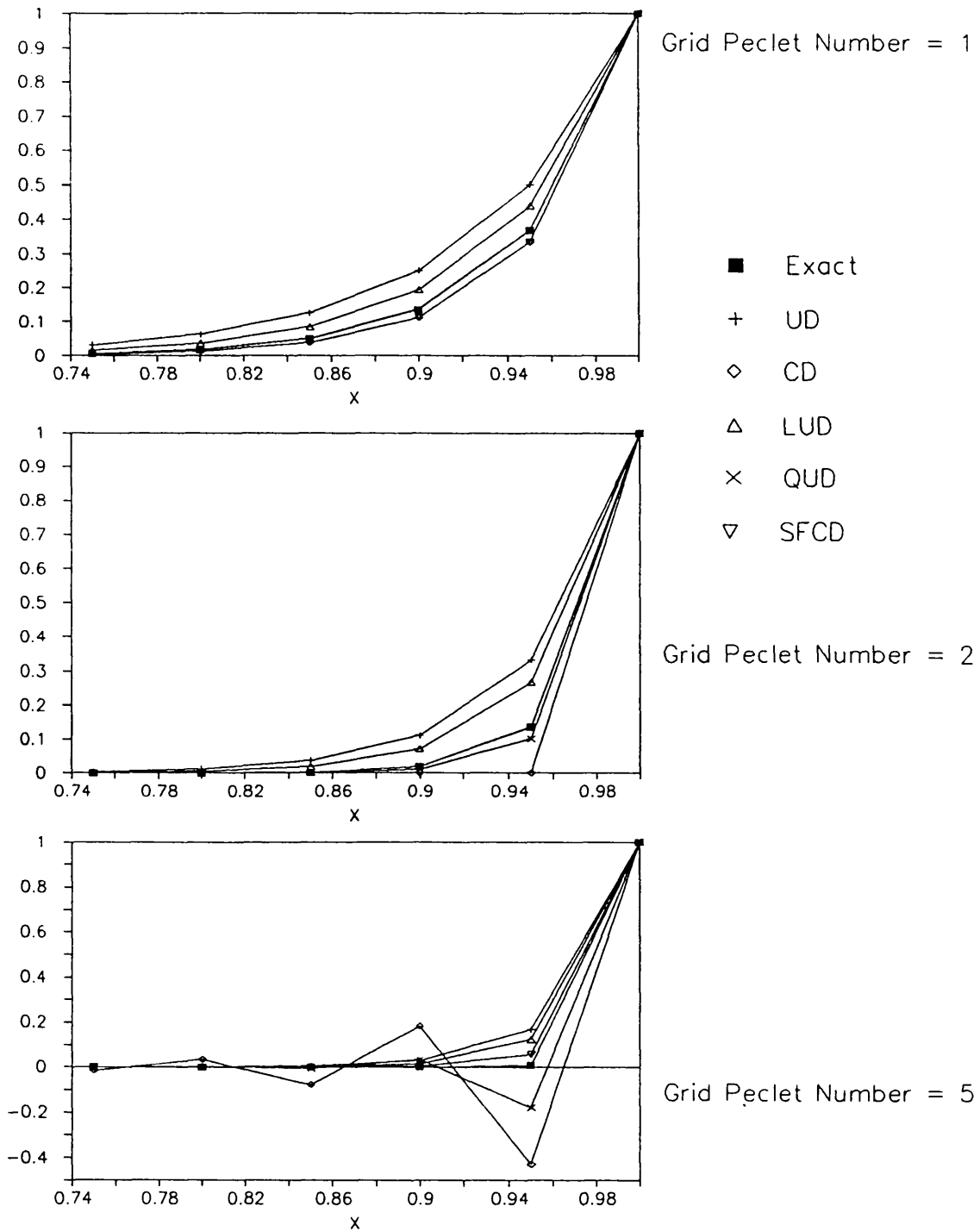


Figure 5.2.1: Solution to Back Diffusion Problem

Grid Peclet Number = 1 (top), 2 (middle) and 5 (bottom). In the middle figure the SFCD results cannot be distinguished from the exact solution.

UD, even in the absence of cross-stream numerical diffusion. It is also evident that the bounding is successfully achieved when both CD and QUICK are producing oscillations. The LUD scheme is bounded in one dimension (chapter 3) but is not as accurate as SFCD.

The second problem, used by Runchal (1986), has a source term which is given by

$$s = 1 + 2x + \pi \cos \pi x + \frac{\pi^2 \sin \pi x - 2}{Pe} \quad (5.2.4)$$

and boundary conditions

$$\phi(0) = 1 + e^{-Pe} \quad \text{and} \quad \phi(1) = 4. \quad (5.2.5)$$

The exact solution to this problem is

$$\phi(x) = 1 + x + x^2 + e^{-Pe(1-x)} + \sin \pi x \quad (5.2.6)$$

and is shown in figure 5.2.2 for Peclet numbers of 10 and 100. The former is a monotonically increasing function while the latter is not. It rises to a value of 3.03 at $x = 0.816$, then drops slightly to 3.01 at $x = 0.938$ before rising sharply as x approaches one.

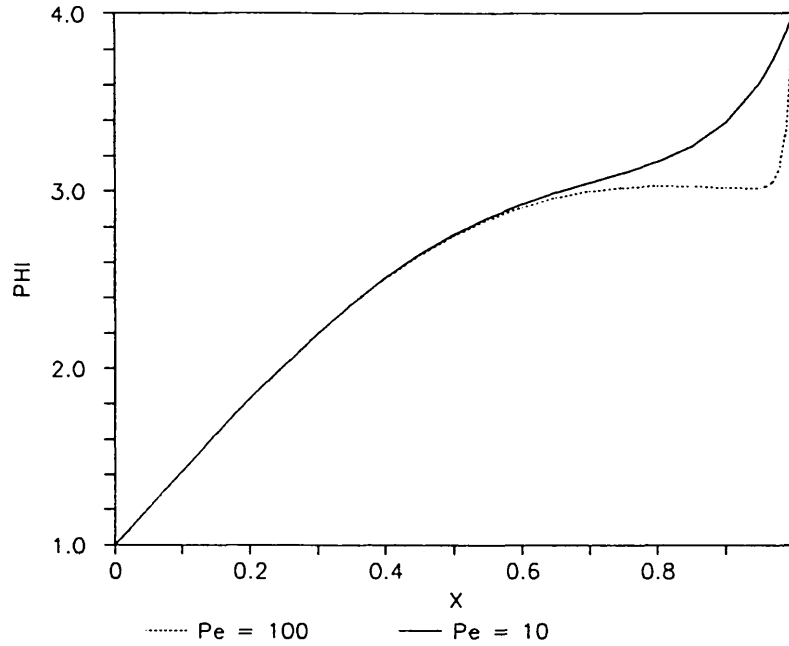


Figure 5.2.2: Exact Solution of Runchal Test Problem

Calculations for Peclet numbers of 10 and 100 were performed on a number of meshes using UD, LUD and SFCD ($r_{\max} = 1, 2$ and 5). The error in the calculation was evaluated as

$$\Delta \sum_{\text{nodes}} |\phi^{\text{calc}} - \phi^{\text{exact}}| \quad (5.2.7)$$

where ϕ^{calc} is the calculated solution and ϕ^{exact} the exact solution. The results are shown in figure 5.2.3 as a function of the number of grid points. The results for the higher two $r_{\max}$ values are virtually identical.

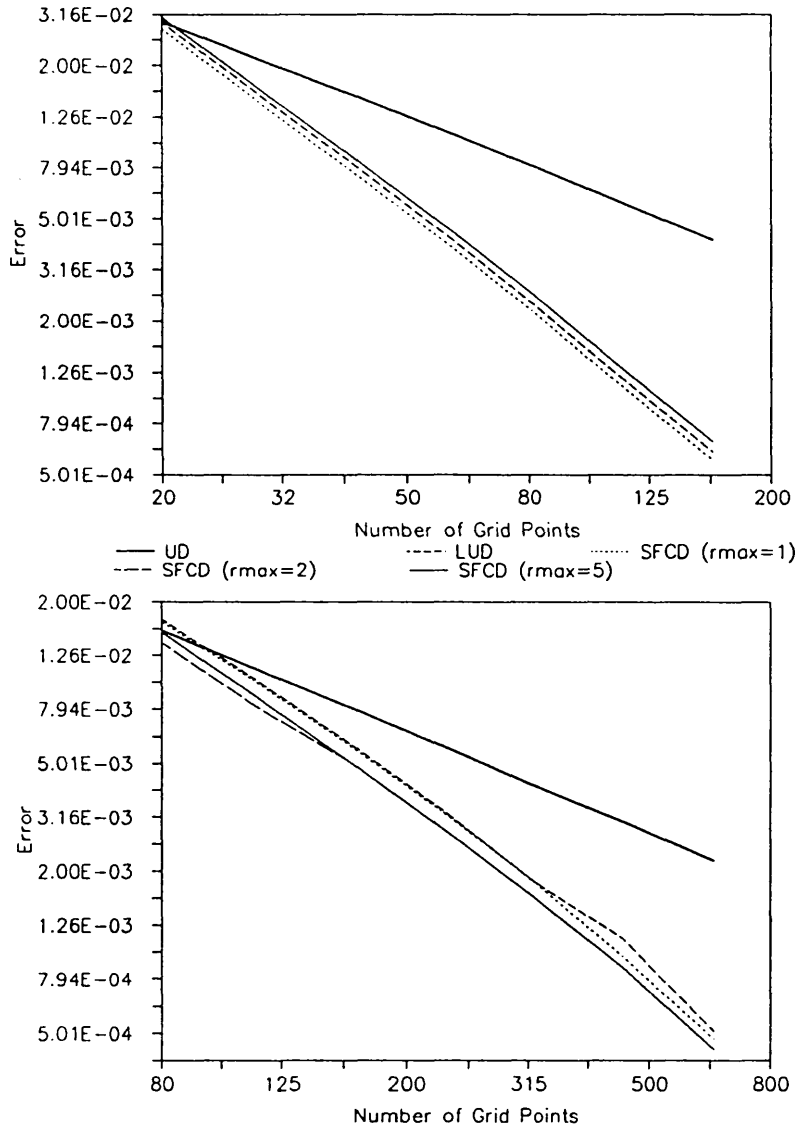
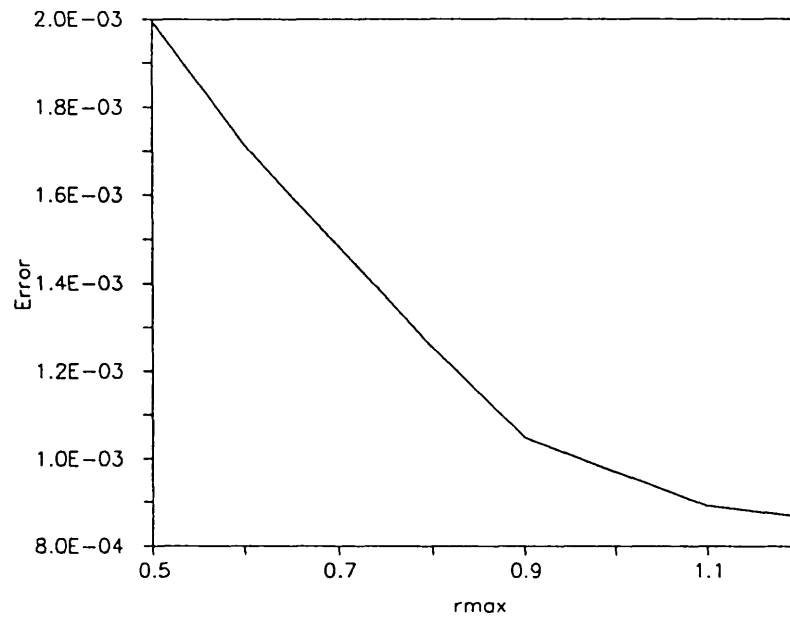


Figure 5.2.3: Errors in the Runchal Test Problem Calculations
Grid Peclet Number = 10 (top) and 100 (bottom). The results for $r_{\max}$ values of 2 and 5 are virtually identical.

The poor response of UD to grid refinement is immediately apparent. This is to be expected in view of its first order behaviour and the presence of source terms as discussed in chapter 3. The LUD and SFCD schemes give similar results and are more accurate than UD in accord with their higher order nature. The SFCD results are consistent with a second order scheme, although full second order behaviour is only observed on the finer grids in the higher Peclet number calculations.

The sensitivity of the error in the solution to $r_{\max}$ was investigated by various calculations on a grid containing 452 points for a Peclet number of 100. The results are shown in figure 5.2.4. There is appreciable sensitivity for values below 1.0 but very little change thereafter. The loss of accuracy for $r_{\max}$ values of less than one is caused by the blending of UD with the higher order scheme.



**Figure 5.2.4: Sensitivity of the Error to $r_{\max}$
452 point calculation, Peclet number of 100.**

The variation of the r parameter in SFCD is shown in figure 5.2.5 for several grids and a Peclet number of 100 and an $r_{\max}$ value of 5. It can be seen that r becomes increasingly close to unity over most of the solution domain as the grid is refined. This is to be expected from the definition of r which can be seen by substituting the Taylor expansions for $\phi_E - \phi_P$ and $\phi_P - \phi_N$ into the definition of r (i.e. equations (4.2.5), (4.2.6) and (4.1.10)) and considering the limit $\Delta \rightarrow 0$. This is another reason why the error in the solution is insensitive to $r_{\max}$ providing it is greater than unity.

In the region where x is larger than 0.8, the behaviour of r is complex. This is caused by the aforementioned maximum and minimum that occur in this region. In both cases r is zero at the extremum and takes a high value immediately afterwards. This demonstrates the way the self-filtering aspect of SFCD operates to ensure that boundedness is maintained.

The two test problems presented in this section demonstrate that SFCD is bounded and exhibits higher order behaviour. No difficulties were encountered in solving the non-linear discrete equation.

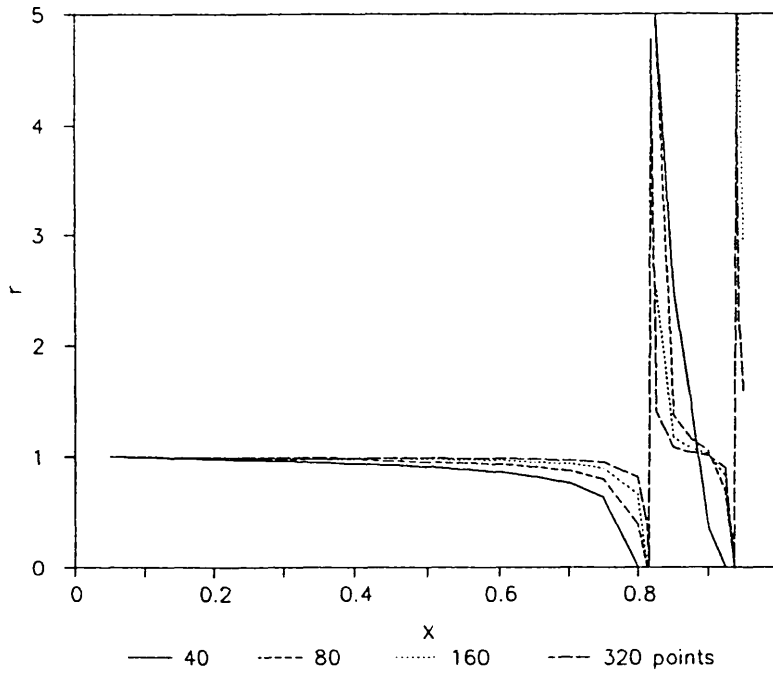


Figure 5.2.5: Variation of the r Parameter in SFCD
Peclet number of 100 and $r_{\max}$ is 5.

5.3 Two Dimensional Uniform Convection

This problem consists of pure convection in two dimensions with a uniform flow field inclined at forty-five degrees to the mesh as shown schematically in figure 5.3.1. The boundary conditions are that ϕ is unity along the y axis and zero along the x axis. The transport equations is

$$\frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y} = 0 \quad (5.3.1)$$

and inlet conditions are

$$\phi(0, y) = 0 \quad \text{and} \quad \phi(x, 0) = 1. \quad (5.3.2)$$

Zero gradient conditions are impose on the exit sides. The exact solution is a step profile.

$$\phi(x, y) = 1 \quad x < y \quad (5.3.3)$$

$$\phi(x, y) = 0 \quad x > y \quad (5.3.4)$$

The objective of this problem is to demonstrate that SFCD can substantially reduce the erosion of the step due to numerical diffusion. This problem is a very demanding one for numerical methods not specially designed to maintain sharp fronts, which is the case for all those examined here.

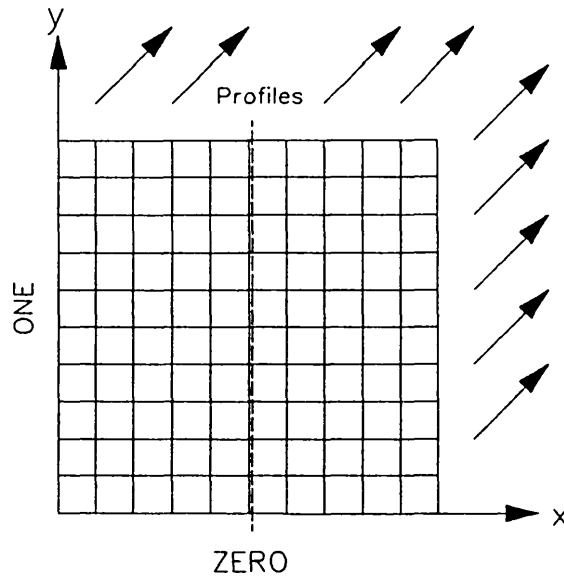


Figure 5.3.1: Uniform Convection Problem

The profile along $x = 0.5$ is shown in figure 5.3.2 for the four schemes UD, LUD, QUICK and SFCD (with $r_{\max} = 1$). The SFCD solutions are obtained by using the IFCA. The erosion of the UD profile due to numerical diffusion is immediately apparent; so are the over- and undershoots due to LUD. The SFCD results are clearly more accurate than UD while maintaining proper bounds. A CD solution to this problem cannot be obtained because the A_p term is zero. The other higher order schemes (LUD and QUICK) have boundedness problems.

The dependence of the centre-line profile on $r_{\max}$ was investigated (figure 5.3.3). There is appreciable sensitivity for $r_{\max}$ below 1, but above that value it is slight. This is consistent with the results presented in one dimension in the previous section.

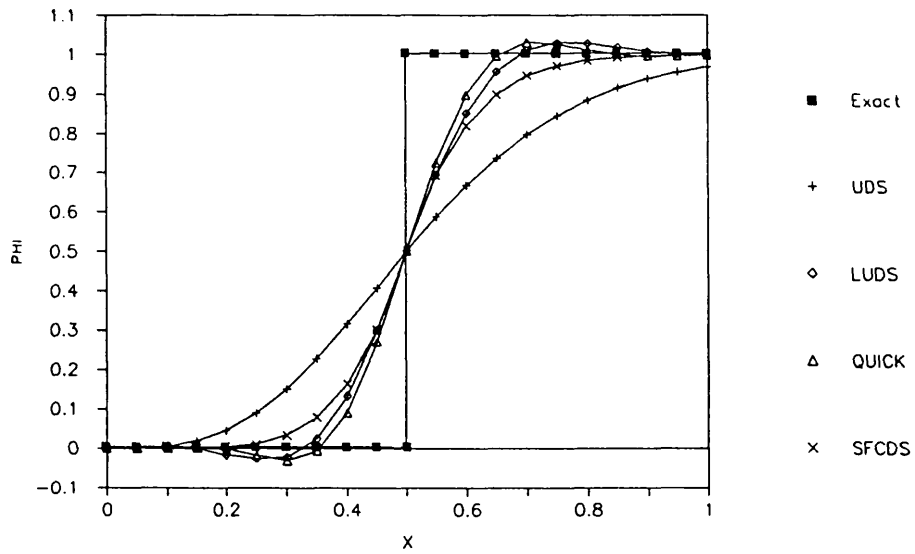


Figure 5.3.2: Centre Line Profiles for Uniform Convection Problem
The value of $r_{\max}$ used is unity.

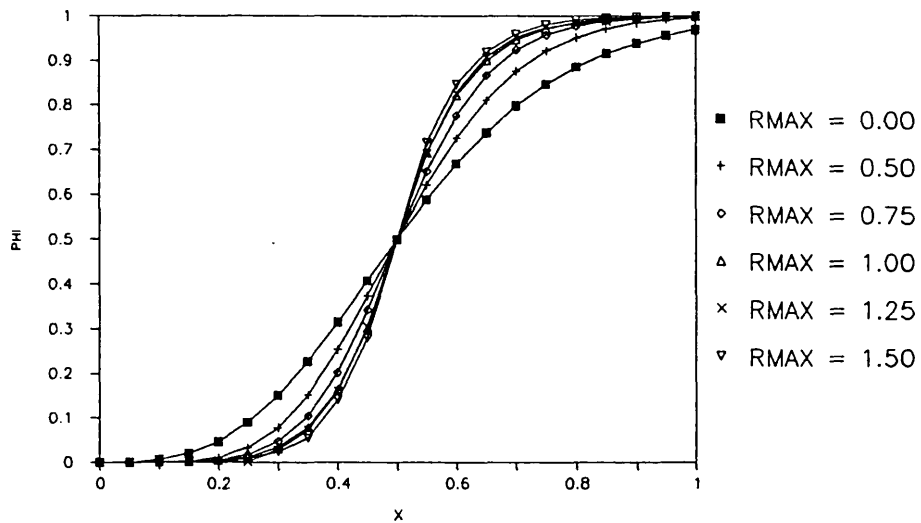


Figure 5.3.3: $r_{\max}$ Sensitivity of SFCDS Centre Line Profile for Uniform Convection Problem

5.4 Stagnation Flow in Corner

This problem consists of stagnation flow where fluid turns a corner (figure 5.4.1). It is used to investigate the performance of SFCD and the different solution algorithms, the uniqueness of the solution and its dependence on r_{max} .

The velocity field in this case is

$$u = x \quad (5.4.1)$$

$$v = (1 - y) \quad (5.4.2)$$

and the equation

$$\frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi) = 0 \quad (5.4.3)$$

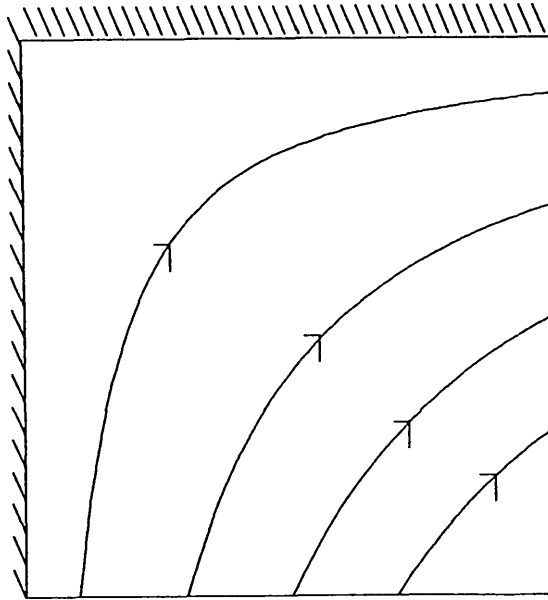


Figure 5.4.1: Stagnation Flow Problem

is solved in the domain

$$0 \leq x \leq 1 \quad \text{and} \quad 0 \leq y \leq 1. \quad (5.4.4)$$

A zero gradient boundary condition on the outlet boundary at $x = 1$ and the inlet condition is a Gaussian given by

$$\phi(x, 0) = e^{-\left(\frac{0.5-x}{0.3}\right)^2}. \quad (5.4.5)$$

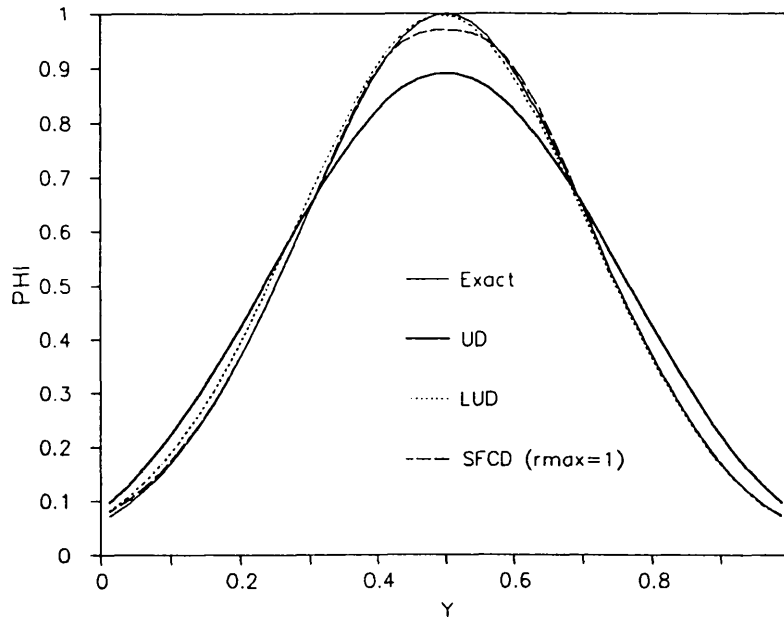


Figure 5.4.2: Predicted Outflow Profiles for Stagnation Flow Problem

Calculations were performed on a uniform mesh with forty nodes on each side with UD, LUD and SFCD ($r_{\max} = 1$). The outlet face profiles are shown in figure 5.4.2. The low accuracy of UD compared with the other two schemes is immediately apparent. The SFCD scheme results are considerably more accurate and the LUD scheme more accurate still and, in this case, does not violate the boundedness criterion. The SFCD calculations were performed by the EFCA and IFCA algorithms both with and without a corrector step. The number of iterations required to solve the equations to four figure accuracy are listed in table 5.1.1. The EFCA calculation without a corrector required an under-relaxation factor of 0.75 to converge, but otherwise no difficulties were experienced in obtaining solutions.

Algorithm	Corrector	Number of Iterations
EFCA	No	64
EFCA	Yes	56
IFCA	No	22
IFCA	Yes	3

Table 5.4.1: Number of Iterations Required for SFCD Calculations

The most efficient algorithm was the IFCA with the corrector step while the EFCA algorithm without a corrector showed some signs of instability. However, these findings cannot be generalised to other situations for the following reason. The north and east face coefficients (i.e. A_E , α_E etc.) are zero for this problem with the consequence that the linearised equations in step 5 of the algorithms (i.e. equations (4.9.3) and (4.9.7)) can be solved directly thereby reducing the total number of iterations required. Usually, these coefficients are not zero and iterations are required for the solution of equations (4.9.3) and (4.9.7), regardless of the non-linearities introduced by SFCD.

The need for some under-relaxation in the EFCA is also not a serious problem since it is required in most engineering calculations to improve stability and optimise the convergence rate. Indeed, the calculations described in chapter 7 were performed with the EFCA and no convergence difficulties or serious cost-penalties incurred. Therefore table 5.4.1 considerably overstates the likely benefits to be obtained from the use of the IFCA instead of the simpler EFCA. The cost of using the EFCA to solve the SFCD equations for a complex flow calculation is compared with the cost of UD in section 7.2.

The results of these outlet profile calculations obtained from the various algorithms used to solve the SFCD equations (with $r_{\max} = 1$) were compared to investigate any dependence on the algorithm used. This was done by calculating the sum

$$\Delta \sum_{\text{outlet}} |\phi^{IFCA} - \phi^{\text{other}}|. \quad (5.4.6)$$

Here ϕ^{IFCA} is the solution calculated using the IFCA without a corrector and ϕ^{other} is the solution obtained by one of the other methods and the sum is over the nodes on the outflow face. This expression is the average difference in the value of ϕ between the two calculations and was found to be in the range 3.0×10^{-5} to 10^{-4} . Recalling that the calculations were all performed to four figure accuracy, it can be concluded that the SFCD results are independent of the algorithm used to solve the equations.

The uniqueness of the solution was further investigated by comparing calculations using initial conditions in which ϕ was randomly distributed between zero and one with an initial ϕ value of zero everywhere. The average difference in ϕ defined in a similar way to (5.4.6), was no higher than 10^{-4} which is of the same order as the accuracy of the calculation. This suggests that the non-linear SFCD equations have a unique solution regardless of the way in which they are solved.

The sensitivity of the solution on $r_{\max}$ was investigated. The variation of the maximum ϕ value on the outflow face with $r_{\max}$ is shown in figure 5.4.3. There is significant variation when $r_{\max}$ is below unity and less sensitivity when $r_{\max}$ is higher than unity. The accuracy improves as $r_{\max}$ is increased. However the number of iterations required in the IFCA without a corrector increases rapidly as $r_{\max}$ increases (figure 5.4.4).

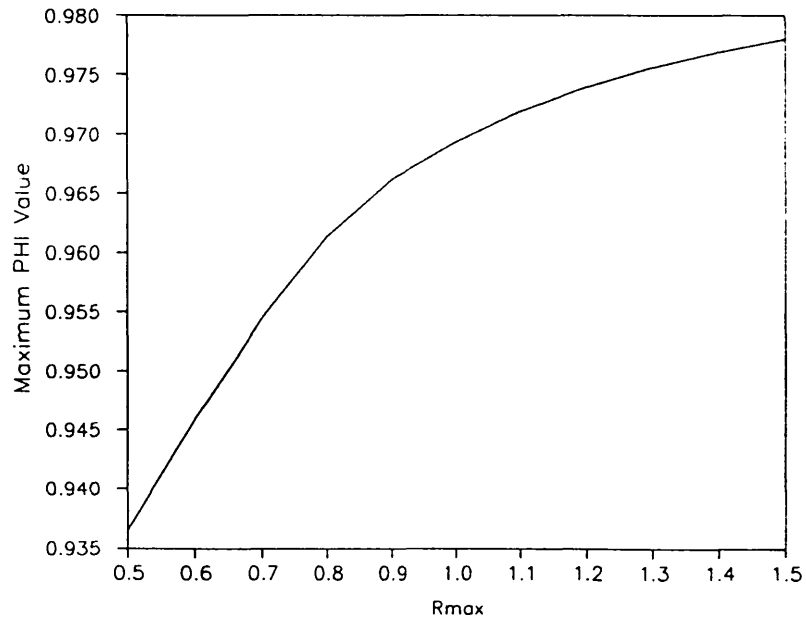


Figure 5.4.3: Variation of Maximum ϕ Value with $r_{\max}$

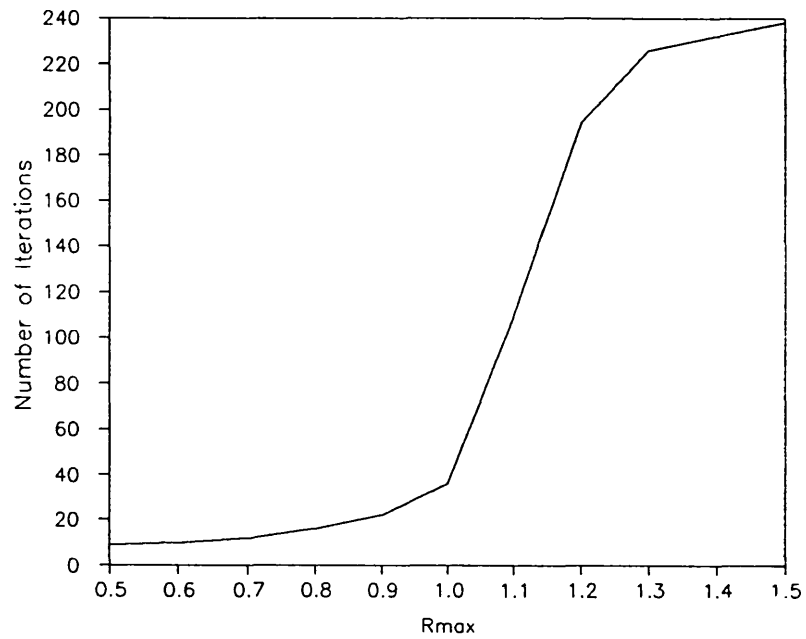


Figure 5.4.4: Sensitivity of the Number of Iterations to $r_{\max}$

5.5 Time Dependent Translation in Solid Body Rotation Flow

This problem is concerned with the time dependent convection of an initial ϕ distribution, to be specified below, in a steadily rotating flow field for which

$$u = y \quad (5.5.1)$$

$$v = -x. \quad (5.5.2)$$

The transport equation is

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi) = 0 \quad (5.5.3)$$

A Cartesian mesh is used so the velocity is not generally aligned with the grid as illustrated in figure 5.5.1. The grid used consisted of one hundred points in each direction and the Courant number was 0.5 unless otherwise specified. This corresponds to a rotation of 0.01 of a radian each time step. The initial distribution was rotated through ninety degrees which is equivalent to 157 steps. The boundary conditions applied on the edge of the mesh are as follows. On an outflow boundary the gradient of ϕ normal to the boundary is zero. A zero ϕ value is applied on an inflow boundary.

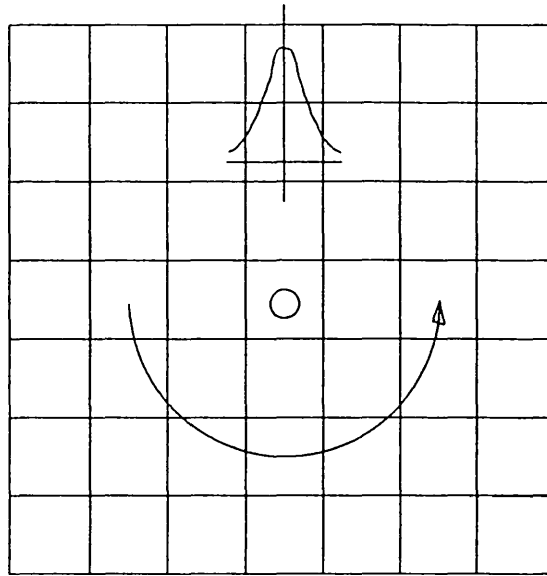


Figure 5.5.1: Steadily Rotating Flow in a Cartesian Mesh

The first initial distribution used is a Gaussian centred half way between the centre and one edge with a standard deviation of ten cells. Calculations were performed with using UD, LUD and SFCD with $r_{\max}$ values of one and five. The SFCD results were obtained using the IFCA.

Calculated profiles through the peak shown in figure 5.5.2 and the results are shown in three dimensional projections in figure 5.5.3. An additional SFCD computation with a Courant number of 0.25 and $r_{\max}$ unity is also shown in the latter figure. The erosion of the peak due to the numerical diffusion in UD is very apparent. Less apparent in this figure is the boundedness violation in the LUD result. The

initially much larger negative values generated during the calculation have decayed to a small magnitude after the 157 time steps so they cannot be observed in the figures. LUD also violated the upper bounds at early stages in the calculation. The SFCD results show less degradation of the profile than the UD results although the peak is not predicted quite as well as with LUD. There is little difference between the SFCD results with $r_{\max}$ as unity and five. Indeed, the reduction in the Courant number from 0.5 to 0.25 leads to almost as large a change in the results (figure 5.5.2) as the change in $r_{\max}$ and hence the $r_{\max}$ value is not the only factor limiting the accuracy of the solution.

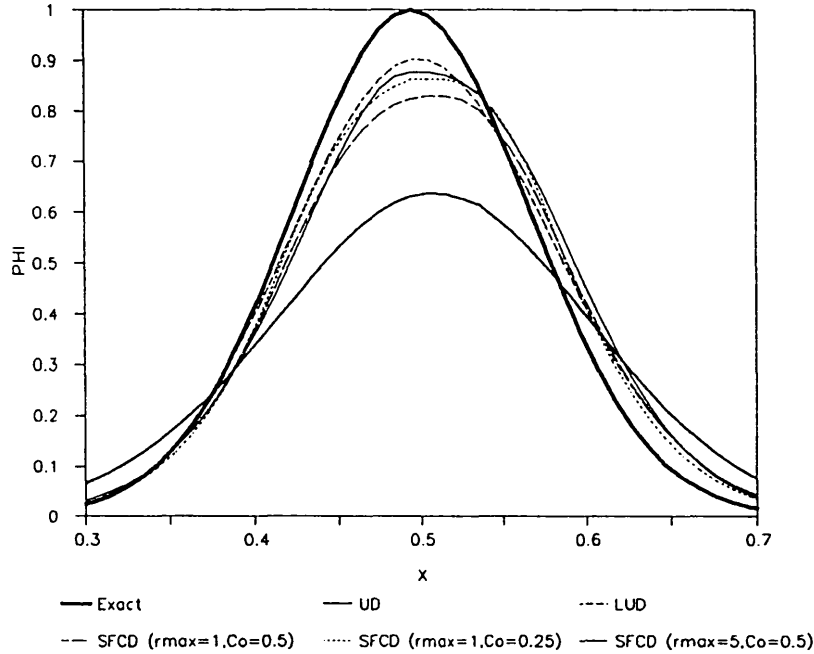


Figure 5.5.2: Profiles Through Results

The computer time required for the calculations with a Courant number of 0.5 are shown in table 5.5.1. This demonstrates that for complex multidimensional problems SFCD is not significantly more expensive than more classical linear schemes because the solution algorithms devised in this study are efficient and effective. However, increasing the value of $r_{\max}$ does increase the computational cost significantly due to the reduced diagonal dominance causing more iterations in the IFCA during each time step.

The conclusions from this problem are that SFCD significantly reduces numerical diffusion in time dependent problems without causing computational problems or violating bounds. There is little to be gained by increasing $r_{\max}$ above unity because the modest gain in accuracy achieved is at the expense of considerably increased computational cost.

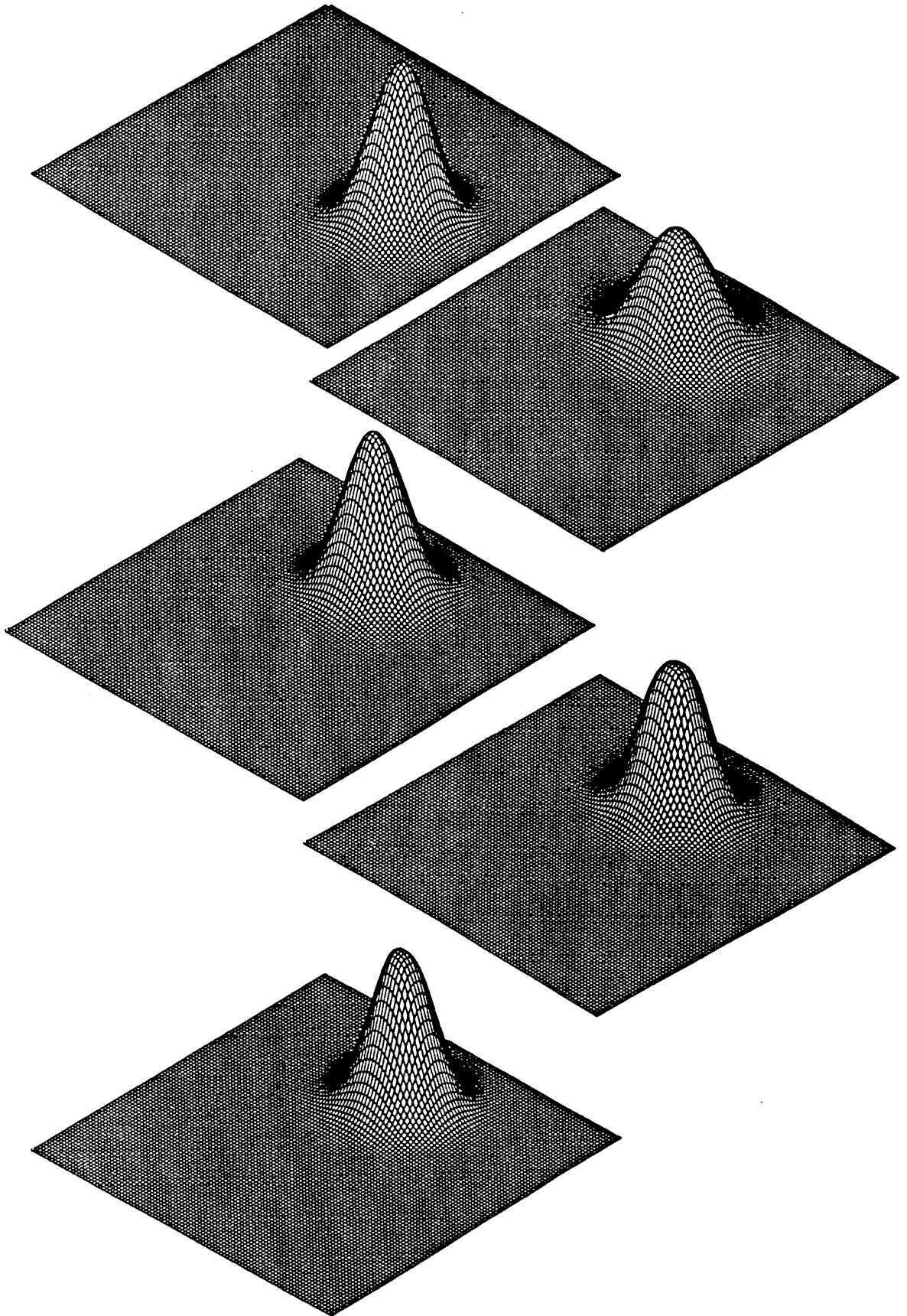


Figure 5.5.3: Solid Body Rotation Results
Top to Bottom: Original Gaussian Profile, UD solution, LUD solution, SFCD solution ($r_{\max} = 1$),
SFCD solution ($r_{\max} = 5$).

Scheme	Time (s)
UD	1014
LUD	1116
SFCD ($r_{\max} = 1$), IFCA	1452
SFCD ($r_{\max} = 5$), IFCA	1823

Table 5.5.1: Solid Body Rotation Timings on a Gould PN9080

A more severe problem is now considered which dramatically shows the difference in performance of the various schemes under consideration. The problem differs from the one above in having the initial ϕ distribution in the form of a cylinder with a slot cut in it (figure 5.5.4). It is important to emphasize that this is very demanding for schemes not designed to capture fronts and is not typical of the problems for which a higher order differencing scheme is sought. The results are shown in figure 5.5.4. The improvement over UD is immediately apparent. Note the negative values in the LUD solution which can be seen at the foot of the 'neck' joining the two halves of the cylinder.

5.6 Conclusions

The scalar transport calculations presented in this chapter demonstrate that SFCD is bounded and more accurate than UD. In particular, there is a significant reduction in numerical diffusion.

The sensitivity of the solution to the $r_{\max}$ parameter was investigated. The accuracy of the calculation increases as $r_{\max}$ rises towards unity, but there is only a slight improvement beyond this point. Additionally, there is a significant cost penalty associated with $r_{\max}$ values greater than unity. It is therefore concluded that the optimum value for this parameter is 1.0.

The effectiveness of the various solution algorithms was investigated. All the tested algorithms successfully solved the equations although under-relaxation was occasionally needed for the EFCA. The IFCA with a corrector step appears to be the most efficient solution method. However, it has not been demonstrated that this complex technique is the best one to use in complex flow calculations.

The dependence of the results on the initial conditions and solution algorithm was investigated and no evidence of uniqueness problems found.

The analysis presented in the previous chapter and the calculations described in this one show that SFCD satisfies the criteria for solving scalar transport problems. However, as noted in chapter 3, this is insufficient to ensure cost-effectiveness in flow calculations due to possible stability problems with the solution algorithm. This point is considered in chapter 7 where SFCD is tested in a three-dimensional model of a stirred tank.

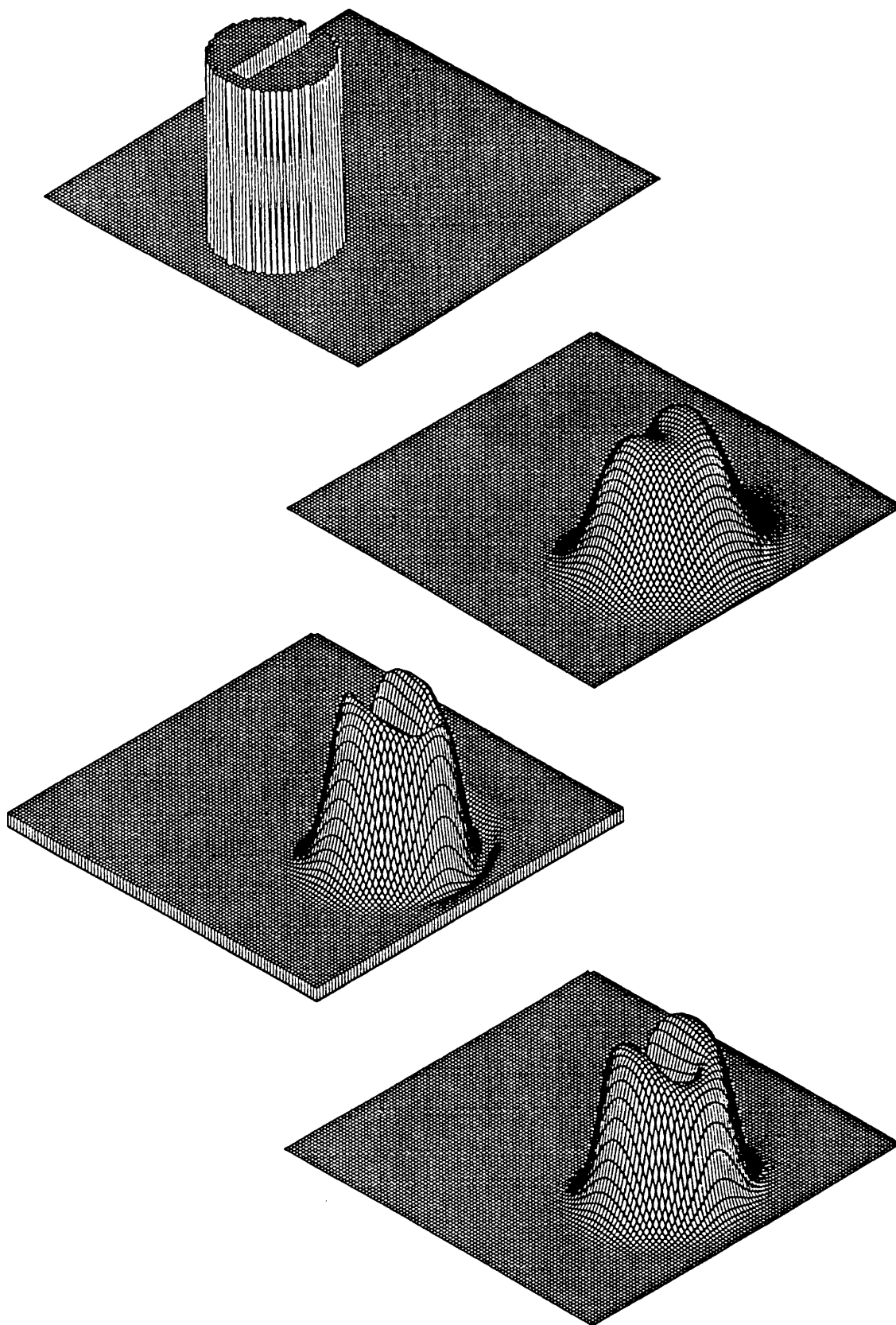


Figure 5.5.4: Rotation of Slotted Cylinder

Top to Bottom: Original Profile, UD solution, LUD solution, SFCD solution ($r_{\max} = 1$).

Chapter 6: Stirred Tank Model

Chapters 3, 4 and 5 of this study have been concerned with methods for discretising scalar transport laws and the solution of the resulting discrete equations. This chapter is concerned with the application of these discretisation techniques to the equations formulated in chapter 2 to perform calculations in stirred tank reactors. The realisation of this goal requires that (i) algorithms to handle non-linearities and couplings between the equations be assembled and (ii) suitable initial and boundary conditions for stirred tank reactors are applied. Sections 6.1 to 6.4 of this chapter are concerned with the first task while section 6.5 addresses the second.

Section 6.1 breaks the problem into two parts, the calculation of the flow field and the calculation of mixing and chemical reactions. The former calculation is performed first to generate the flow fields. The latter calculation uses these flow fields to solve the mixing and chemical reaction problem. Section 6.2 describes the computational mesh and variable storage locations used in the stirred tank calculation. Section 6.3 describes the procedure for the solution of the discretised equations governing the fluid flow on this mesh. Section 6.4 is concerned with methods for solving the mixing and chemical reaction problem. Suitable initial and boundary conditions are described in section 6.5 and section 6.6 recommends values of relaxation factors and other numerical parameters in the algorithm. Finally, section 6.7 summarises the progress made in this chapter.

6.1 Separation into Two Sub-Problems

The assumption made in chapter 2 that the mixing and chemical reaction does not have an effect on the flow results in a one-way coupling between the equations describing the flow and the mixing and chemical reaction; the mixing and reaction equations depend on the flow but not vice versa. It is therefore possible to calculate the flow field and then solve the mixing and reaction equations. Extension to the coupled situation can be done and is straightforward.

The mean flow in a stirred tank is steady and so the steady-state versions of the momentum and turbulence equations will be solved. However, the mixing and chemical reaction problem is unsteady so the time dependent form of the stream fraction and reaction extent equations will be used.

Two separate computational procedures have been developed. The first, MIXFLO, calculates the steady flow. The second, MIXREACT, uses the flow fields calculated by MIXFLO to tackle the mixing and chemical reaction problem.*

6.2 Grid Arrangements

Figure 6.2.1 illustrates the computational mesh used for the discretisation of the equations obtained in chapter 2. The notation for the mesh has been given in chapter 3 and is also summarised at the beginning of this work. All quantities except velocity components are stored at the scalar node near the centre of the computational cell. The velocity components are stored on the faces of the cells. These faces are defined as being midway between scalar nodes.

The three momentum equations are discretised on different meshes as indicated in figure 6.2.1. The momentum cell is based around the velocity storage location and has two scalar nodes on its edges. This is known as the staggered mesh arrangement. It is employed here for the following reasons.

* Many people including Gosman, Issa, Bolour-Froushan, Politis, Looney, Collins and Lekakou have contributed to the development of MIXFLO. MIXREACT has evolved from programs written by Lynch and extended by Looney.

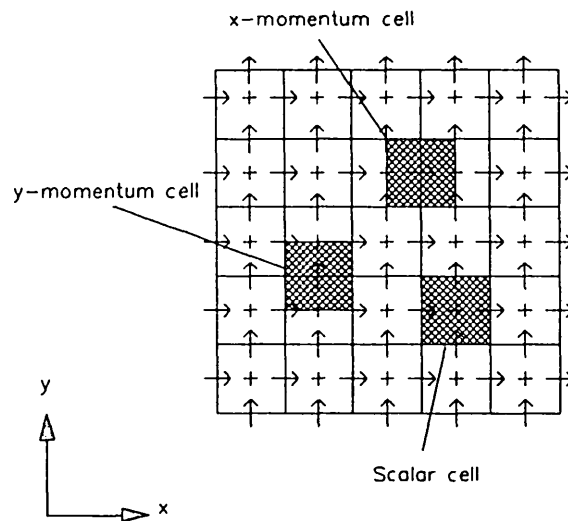


Figure 6.2.1: Computational Meshes and Storage Locations

- (i) The pressure gradient terms in the momentum equations can be evaluated directly from the pressure nodes on the control volume boundary. On non-staggered meshes alternate pressure nodes are decoupled which leads to unrealistic predictions of the pressure field (e.g. Patankar, 1980)
- (ii) Velocities are directly available for the evaluation of fluxes in the continuity and scalar transport equations. The solution of the continuity equation on staggered meshes results in a decoupling of alternate velocity nodes.

Rhie and Chow (1982) have derived a solution method that overcomes the decoupling problem in points (i) and (ii) on non-staggered meshes. However an existing staggered mesh program described by Bolour-Froushan (1986) and others was used as the starting point for this study and the existing staggered mesh practice has therefore been retained.

It is frequently found that a quantity is required at a location other than the one where it is stored. For instance the turbulent viscosity is required on cell faces but it is stored at the scalar nodes. With the exception of the discretisation of the convection terms, linear interpolation is used to estimate the value of variables at other locations.

6.3 Flow Calculation Procedure

Outline of the Procedure

The flow equations are highly non-linear and coupled so the solution is obtained by using an iterative procedure. This is done by means of the Pressure Implicit Split Operator (PISO) algorithm of Issa (1983) which is outlined below.

- (i) Solve linearised versions of the three momentum equations in turn.
- (ii) Adjust the pressure and velocities to satisfy the continuity equation. This procedure is described further on in this section.

- (iii) Solve linearised forms of the turbulent kinetic energy equation and the dissipation equation and update the turbulent viscosity.
- (iv) If satisfactory convergence has not been achieved then go to step (i).

All the transport equations are linearised before solving them by writing them in the form (3.1.1). The discretisation techniques derived in chapters 3 and 4 are applied. The coefficients in the discretised equations are always evaluated from the current estimate of the variable fields. Under-relaxation is employed on the momentum and turbulence equations but not on the continuity equation.

Implementation of SFCD

SFCD is implemented using the EFCA. Preliminary computational experiments on a two dimensional Navier-Stokes solver suggest that the inclusion of a corrector step does not significantly reduce the number of iterations required in the solver. However care must be taken with the implementation of SFCD in the momentum equations due to the non-linear nature of the convection term as will now be described. For example consider the 'east face' discretisation of

$$\frac{\partial}{\partial z}(\rho u_z u_z). \quad (6.3.1)$$

The east face axial velocity, $u_{z,e}$, is the velocity used to calculate the flux through the face and is also the transported quantity itself. In common with many other workers (e.g. Lai, 1983) these two roles are separated. The velocity used to calculate the mass flux through the cell face ($u_{z,e}^L$) is obtained by linear interpolation between E and P . Therefore, on a uniform mesh,

$$u_{z,e}^L = \frac{1}{2}(u_{z,P} + u_{z,E}). \quad (6.3.2)$$

The velocity that is used as the transported quantity is obtained using SFCD in the usual manner. The momentum flux through the east face arising from (6.3.1) is thus approximated as

$$\begin{aligned} & \frac{1}{2} \rho s_e (u_{z,e}^L + |u_{z,e}^L|) u_{z,P} + \frac{1}{2} \rho s_e (u_{z,e}^L - |u_{z,e}^L|) u_{z,E} + \\ & \frac{1}{4} \rho s_e (u_{z,e}^L + |u_{z,e}^L|) \max \left\{ 0, \min \left(\frac{u_{z,E} - u_{z,P}}{u_{z,P} - u_{z,W}}, r_{\max} \right) \right\} (u_{z,P} - u_{z,W}) - \\ & \frac{1}{4} \rho s_e (u_{z,e}^L - |u_{z,e}^L|) \max \left\{ 0, \min \left(\frac{u_{z,E} - u_{z,P}}{u_{z,EE} - u_{z,E}}, r_{\max} \right) \right\} (u_{z,EE} - u_{z,E}) \end{aligned} \quad (6.3.3)$$

where s_e is the area of the east face. Note that either terms I and III or terms II and IV contribute to the flux depending on the sign of $u_{z,e}^L$.

The discretisation coefficients due to terms I and II are calculated at the beginning of the iteration (section 4.9). Terms III and IV are calculated during each cycle of the EFCA. It is therefore tempting to use the latest values of u_z to calculate $u_{z,e}^L$ for use in terms III and IV. However this can cause stability problems if the sign of $u_{z,e}^L$ changes during these cycles. Under these conditions terms I and IV or terms II and III will be combined together. Experience has shown that these combinations are highly unstable and rapid divergence will ensue.

This problem can be overcome by ensuring that all the $u_{z,e}^L \pm |u_{z,e}^L|$ factors in equation (6.3.3) are updated simultaneously. This can either be done as part of the EFCA iterations or by calculating them at the beginning of the EFCA procedure and not updating them with the SFCD r factors. The latter approach has been adopted in this study.

Solution of Momentum and Continuity Equations

The three velocity components are obliged to satisfy the three momentum equations and the continuity equation. However pressure does not have its own equation. It is inferred from the requirement that the three velocities must satisfy these four equations. The PISO algorithm (Issa, 1983) proceeds as follows.

The discretised i 'th momentum equation is written as

$$A'_{i,p} u_{i,p} = \sum A_{i,m} u_i - \Delta_i p + Se'_i \quad (6.3.4)$$

where

$$A'_{i,p} = \hat{\lambda}_{i,p} - Si_i \quad (6.3.5)$$

and

$$Se'_i = \sum \alpha_m u_{i,m} + Se_{i,p} \quad (6.3.6)$$

If SFCD is not being used then the first term on the RHS of (6.3.6) is omitted. The difference operator Δ_i is defined by

$$\Delta_x \phi = s_e \phi_e - s_w \phi_w \quad (6.3.7)$$

$$\Delta_y \phi = s_n \phi_n - s_s \phi_s \quad (6.3.8)$$

$$\Delta_z \phi = s_d \phi_d - s_u \phi_u \quad (6.3.9)$$

s_m is the area of the m 'th face of the finite volume. This operator is the discretised form of the gradient operator, ∇ .

The PISO algorithm advances the velocity from the n 'th iteration level to the $n + 1$ 'th in the following manner. Evaluate the coefficients using the velocities from the n 'th iteration level. Use the EFCA to solve

$$A'_{i,p} u_{i,p}^* = \sum A_{i,m} u_i^* - \Delta_i p^n + Se'^*_i \quad (6.3.10)$$

for u_i^* for each of the velocity components. (The *superscript in the explicit part of the source term refers only to the SFCD terms re-evaluated in the EFCA, as described previously, and not to the other parts of the source term.)

The algorithm now seeks an updated velocity, u_i^{**} , and pressure, p^* , such that the continuity equation (discretised on the scalar mesh) is satisfied simultaneously with the momentum equations. These updated quantities satisfy

$$\sum_i \Delta_i u_i^{**} = 0 \quad (6.3.11)$$

and

$$A'_{i,p} u_{i,p}^{**} = \sum A_{i,m} u_i^* - \Delta_i p^* + Se'^*_i \quad (6.3.12)$$

The new estimates of the variables are found by subtracting (6.3.12) from (6.3.10) and rearranging to give

$$u_i^{**} = -\frac{1}{A'_{i,p}} \Delta_i (p^* - p^n) + u_i^* \quad (6.3.13)$$

Substitution into (6.3.11) yields a difference equation for p^* .

$$\sum_i \Delta_i \left(\frac{1}{A'_{i,p}} \Delta_i (p^* - p^n) \right) = \sum_i \Delta_i u_i^* \quad (6.3.14)$$

If this equation is written out in full it is found to be of the form

$$\begin{aligned} A_p(p_p^* - p_p^n) &= A_e(p_e^* - p_e^n) + A_w(p_w^* - p_w^n) + A_n(p_n^* - p_n^n) + \\ &A_s(p_s^* - p_s^n) + A_d(p_d^* - p_d^n) + A_v(p_v^* - p_v^n) + \sum_i \Delta_i u_i^* \end{aligned} \quad (6.3.15)$$

which can be solved by the method described in appendix D.4. The velocity update is then obtained from (6.3.13).

The pressure correction $p^* - p^n$ is dominated by the continuity error in the first velocity update. To improve the pressure estimate the velocity and pressure are corrected again. Seek u_i^{n+1} and p^{n+1} such that

$$\sum_i \Delta_i u_i^{n+1} = 0 \quad (6.3.16)$$

and

$$A'_{i,p} u_i^{n+1} = \sum A_{i,m} u_i^{**} - \Delta_i p^{n+1} + S e_i^* \quad (6.3.17)$$

are satisfied simultaneously. Subtract (6.3.12) from (6.3.17) to leave

$$u_i^{n+1} = \sum \frac{A_{i,m}}{A'_{i,p}} (u_i^{**} - u_i^*) - \frac{1}{A'_{i,p}} \Delta_i (p^{n+1} - p^*) + u_i^* \quad (6.3.18)$$

(6.3.18) is substituted into (6.3.16) to leave an implicit equation for the new pressure.

$$\sum_i \Delta_i \left(\frac{1}{A'_{i,p}} \Delta_i (p^{n+1} - p^*) \right) = \sum_i \Delta_i \left(\sum_m \frac{A_{m,i}}{A'_{i,p}} (u_i^{**} - u_i^*) \right) \quad (6.3.19)$$

This is of the same form as (6.3.14) and can be solved by an identical technique. Equation (6.3.18) is then used to obtain the velocity.

The pressure only appears in the discrete equations as a pressure difference and therefore any constant can be added to the pressure field without changing the flow. This degree of freedom is eliminated by adding a constant to the pressure at the end of the PISO algorithm such that the pressure at a predetermined point takes on a predetermined value.

6.4 Mixing Calculation Procedure

The mixing and reaction calculation is based around the EFCA. A strictly conservative linear solver is used as described in appendix C.4. During each time step the stream fraction and reaction extent equations are solved in turn. The source terms representing chemical reaction are evaluated explicitly and are always based on data from the old time level to ensure consistency in their evaluation. This treatment has been found to be stable and bounded providing the Courant number is below five. This empirical restriction is not a serious restriction since temporal resolution is lost at higher Courant numbers.

The coefficients for the stream fraction and reaction extent equations are identical. Therefore these need only be evaluated once at the beginning of the calculation.

6.5 Application to a Stirred Tank

Suitable boundary conditions for a stirred tank reactor are described in this section. A typical stirred tank is shown in figure 6.5.1. The flow is driven by a Rushton turbine impeller. The baffles increase the effectiveness of the mixing and maintain an approximately flat free surface. The effect of the impeller shaft is not sufficient to warrant its modelling.

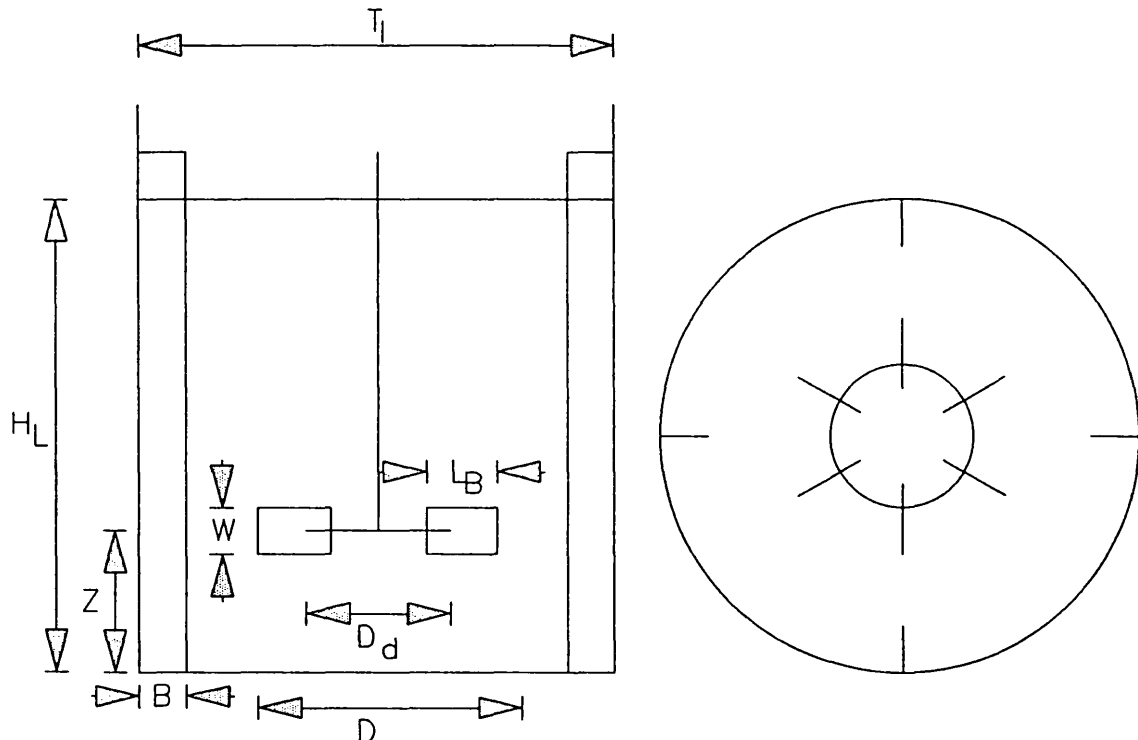


Figure 6.5.1: Stirred Tank

Coordinate System and Mesh Arrangement

A cylindrical polar mesh is used to handle the stirred tank geometry. Most stirred tanks have rotational symmetry. If a tank has four equally spaced baffles it is only necessary to consider a ninety degree sector of the tank to calculate the flow field. This results in a considerable computational saving. However mixing problems usually involve the addition of one fluid to another at a single point in the vessel which breaks the symmetry. Therefore the full tank must be considered for this part of the calculation.

Figure 6.5.2 shows the positioning of the baffle and impeller relative to the mesh.

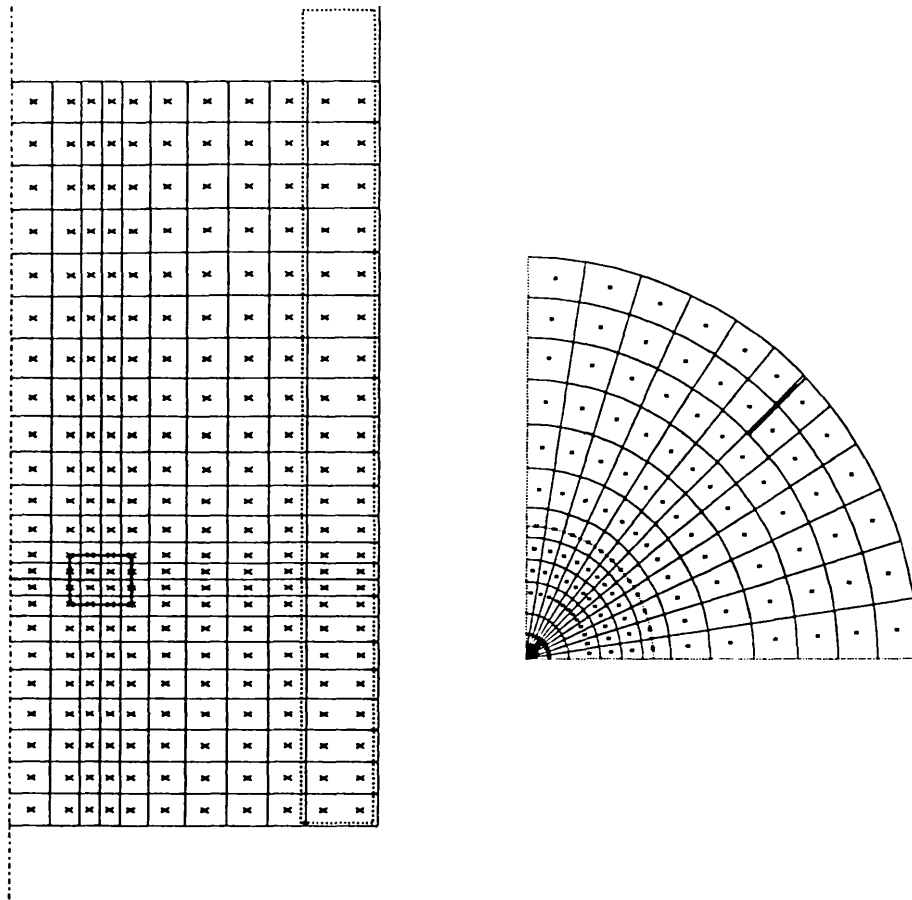


Figure 6.5.2: Computational Mesh

The figure shows the position of the baffle and impeller relative to the mesh. This mesh is used for use in the flow calculation. It is extended to 360° for the mixing and reaction calculation.

Wall Conditions

The near wall modelling presented in section 2.3 is used in the boundary conditions for the momentum, k and ϵ equations. That is to say that the shear stress parallel to the wall, generation term and dissipation rate at the node nearest to the wall are calculated using equations (2.3.19-23). The 'diffusive' fluxes of the turbulent kinetic energy, stream fraction and reaction extent at the wall are zero. This is equivalent to a zero normal gradient condition. The velocity normal to the wall is zero.

Axis

The radial velocity (v_r) and circumferential velocity (v_ϕ) are set to zero on the axis. All other variables have a zero normal gradient condition imposed.

Free Surface

The free surface of the tank is assumed to be flat. This is justifiable because significant troughing is prevented by the baffling system (Uhl and Gray, 1966). The boundary conditions are taken as zero axial velocity (u_z) on the surface and zero tangential shear stress in the other two momentum equations. The remaining variables have a zero normal gradient condition imposed. This boundary condition is not valid for inadequately baffled vessels where troughing or other significant surface features can be expected. Nouri (1988) investigated the effects of a lid on a stirred vessel. It was found that the changed top surface condition had very little effect on the flow. This suggests that the modelling of the free surface is not critical to the successful computation of flows in fully baffled tanks.

Cyclic Boundaries

Cyclic boundary conditions arise from the requirement that the conditions prevailing on the downstream plane of the computational domain are identical to those on the upstream plane. It is therefore not a physical boundary condition but a manifestation of the rotational symmetry of the problem. As discussed above, the typical angular size of the MIXFLO computational domain is 90° and 360° for the MIXREACT domain. Therefore the orders of symmetry for these two calculations are four and one respectively. However the treatment in both cases is identical. Let us suppose there are n_ϕ cells in the circumferential dimension and a term requires a value of ϕ in cell $i + i'$. This value of ϕ is found in cell $(i + i' - 1) + 1$ where the sum in brackets is evaluated in modulo n_ϕ arithmetic (e.g. Stewart, 1981).

Baffle

The baffle is treated as being an infinitely thin wall placed along the scalar control volume sides (figure 6.5.2). Wall boundary conditions are applied on each side of this wall in the usual manner.

Impeller

The impeller drives the whole flow and therefore its modelling is important to the realism of calculations of the flow in a stirred vessel. This work is confined to Rushton turbines. These consist of a horizontal disc to which several uniformly spaced vertical blades are fastened. The impeller disc lies on the side of scalar control volumes (figure 6.5.2) while the volume swept out by the impeller blades is bounded by scalar nodes.

Previous work by Bolour Froushan (1986) excluded the impeller from the computational domain and imposed appropriate boundary conditions on the surface of the impeller to model its effects. This is unsatisfactory for the stream fraction and reaction extent equations where the reaction rate must be calculated everywhere. Lynch and Parker (1983) proposed an alternative treatment which involves calculating the flow within the impeller region in an approximate fashion. This overcomes the aforementioned objection. The principles of this model will be described below but some general points regarding the modelling of the impeller are first discussed.

It is well known (e.g. Yianneskis *et al*, 1987) that vortices trail from behind the outer corners of the blades of the Rushton turbine. This vortex structure rotates with the impeller and occupies a significant fraction of the volume of the tank. The flow in a stirred vessel is therefore not only turbulent but includes periodic flow structures. A steady-state method is being used to solve the flow equations in this project

so a way of averaging this periodic motion must be sought. This is done by modelling the time averaged effects of the impeller blades in the volume swept out by them. The periodic flow structures are included in the fluctuations employed in the turbulence model despite the fact that they are not true turbulence. Yianneskis *et al* (1987) have shown experimentally that a time averaged measurement of k can be up to four times that of the true turbulence intensity.

Measurements of the flow in a stirred tank made by Yianneskis (1982) show that the time averaged velocity components and turbulent kinetic energy do not vary greatly around the circumference of the impeller but do vary significantly with the vertical distance above or below the impeller disc. The impeller model will reflect this.

The flow leaving a Rushton turbine in a standard vessel configuration is from the outer edge of the impeller. The flow enters the impeller from the top, bottom and inner faces. Lynch and Parker (1983) have therefore proposed the following model for the region swept out by the impeller blades. The circumferential velocity (v_θ) is related to the velocity of the impeller blade at that radius by a slip factor, ζ_u .

$$v_\theta(z, r, \theta) = 2\pi r N \zeta_u(z) \quad (6.5.1)$$

Here N is the rotational speed in revolutions per second.

The other two momentum equations are regarded as being unaffected by the impeller. The turbulence equations are also left unchanged within the impeller volume. However the turbulent kinetic energy and dissipation rate in the flow leaving the impeller tip is specified in the following manner. The turbulent kinetic energy level k_{tip} is related to the speed of the impeller tip v_{tip} by

$$k_{tip} = \zeta_k(z) v_{tip}^2. \quad (6.5.2)$$

The dissipation rate, ϵ_{tip} , is related to the turbulence length scale, which is here taken to be proportional to the impeller diameter D , by

$$\epsilon_{tip} = 2\zeta_\epsilon \frac{k_{tip}^{3/2}}{D} \quad (6.5.3)$$

The factors ζ_u , ζ_k and ζ_ϵ have been optimised by Politis (1989a) for the stirred tank investigated by Yianneskis (1982) and described in the next chapter. The two constants ζ_k and ζ_ϵ are postulated as being independent of the impeller geometry and rotational speed. They take the values 0.15 and 3.3 respectively. The function ζ_u is postulated to be a function of the dimensionless axial co-ordinate z^* within the impeller. This is defined as

$$z^* = 2 \frac{z - z_{cl}}{z_{top} - z_{bot}} \quad (6.5.4)$$

where z_{cl} is the clearance of the impeller. This is the height of the centre of the impeller disc above the bottom of the tank. z_{top} and z_{bot} are the height of the top and bottom of the impeller blades above the bottom of the tank. Therefore z^* varies from -1 to +1 from the bottom to the top of the impeller. The function adopted in this study is tabulated in table 6.5.1.

No special boundary conditions are required in the stream fraction or the reaction extent equations because the species equations are solved everywhere in the volume swept out by the impeller blades.

The impeller disc is treated simply as a rotating wall for both the flow calculation and the mixing/reaction calculation.

z^*	-1.00	-0.58	-0.10	0.10	0.59	1.00
ξ_w	0.39	0.65	0.90	0.90	0.65	0.39

Table 6.5.1: ξ_w as a Function of z^*

Initial Conditions for Mixing Calculation

The assumption that the mean flow in the vessel is steady and independent of the mixing has implications for the initial conditions for the mixing processes in the vessel. In particular it requires that the volume of fluid in the vessel is constant during the mixing process and that any chemical reagents are added isokinetically. In reality this cannot be fully achieved for batch reactors. Consideration of these limitations is required when applying these computational procedures.

This study concentrates on the mixing of a small amount of one fluid (the tracer) into a tank containing a second miscible fluid (the bulk fluid). Under these circumstances a single stream fraction equation is required to describe the mixing process, as described in chapter 2. The bulk fluid in the vessel is arbitrarily assigned a stream fraction value of zero. The added tracer fluid is assigned the stream fraction value of unity. All the tracer is assumed to be added instantaneously and isokinetically at time zero at a specified place in the vessel. It is further assumed to occupy the whole of one or more computational cells. Therefore the initial stream fraction in these cells is unity, and zero elsewhere. The concentrations in these two fluids are such that the correct amounts of the chemical species are present.

Let V_b be the volume of the bulk fluid in the tank, then

$$V_b \underline{c}_b = \underline{N}_b \quad (6.5.5)$$

where $\underline{c}_b$ is the concentration vector for this fluid and $\underline{N}_b$ is the vector representing the amount of the chemical species in the bulk fluid, which is known in advance. The volume V_b is evaluated by summing the volumes of the cells with zero stream fraction initially. That is to say that V_b is the volume of the tank less the volume of the cells in which the tracer is introduced. Hence $\underline{c}_b$ can be obtained from (6.5.5).

Analogously for the tracer system we have

$$V_t \underline{c}_t = \underline{N}_t \quad (6.5.6)$$

from which $\underline{c}_t$ can be determined. V_t is the volume of the computational cells containing tracer at the beginning of the calculation. Note that $\underline{c}_b$ is the first column in the matrix c , introduced in section 2.3, $\underline{c}_t$ is the second column of c .

No reaction has taken place at time zero so the initial condition for reaction extent is zero everywhere.

6.6 Numerical Parameters

Flow Calculation

The rate of convergence of a flow calculation depends significantly on the numerical parameters used. Appropriate values can minimise both the number of solver sweeps in the solution of a single transport equation and the number of outer iterations used. Numerous calculations have suggested the values listed in table 6.6.1 as guide-lines for the relaxation factors in the flow equations. The same value is applied to the momentum and turbulence transport equations.

θ	Number of Mesh Cells
0.80	Under 2000 cells
0.75	2000 to 8000 cells
0.70	Over 8000 cells

Table 6.6.1: Recommended Relaxation Factors in Flow Calculation

The tolerance used in step 7 of the EFCA (see section 4.9) is 0.2 for the momentum equations and 0.1 for the pressure equations. The solution is said to be converged when the normalised residuals falls below 0.001 for every equation. The normalised residual is the residual defined in equation (4.9.2) normalised by a factor based on the flow through the impeller. This is the momentum flux for the momentum equations, the volumetric throughput for the continuity equation and the appropriate turbulence discharges for the k and ϵ equations (Bolour-Froushan, 1986). This criterion approximately corresponds to three significant figures of accuracy in the solution of the discretised equation.

Mixing Calculation

The Courant number used to determine the time step in the mixing calculation is 2.0. This is based upon the cell adjoining the zero degree plane on the upper corner of the impeller. The reason for this choice is that the flow will be strong and the cell small due to the grid refinement near the impeller. The Courant number in most of the solution domain will be less than this value and only in a few places will it be higher.

The tolerance in the linear solver step 7 (section 4.9) is 0.01. The calculation terminates when mixing either a specified time has been reached or a maximum number of time steps has been executed.

6.7 Closure

This section has described how the mathematical formulation presented in chapter 2 and the numerical methods analysed in chapters 3 to 5 can be applied to generate a technique for the calculation of flows in stirred tanks. The following tasks have been performed.

- (i) The solution of the equations has been divided into two parts. The first is the solution of the equations governing the flow. This solution is used in the solution of the stream fraction and reaction extent equations.

- (ii) A staggered mesh arrangement has been adopted for the discretisation of the transport equations.
- (iii) A method for simultaneously solving the momentum and continuity equations for the three velocity components and pressure has been presented.
- (iv) A procedure for the solution of the stream fraction and reaction extent equations has been described.
- (v) Initial and boundary conditions appropriate for calculations of flows in stirred tanks have been presented.

The remainder of the thesis is concerned with assessing the performance of these computational procedures and illustrating how they may be applied to tackle engineering problems.

Chapter 7: Stirred Tank Results

7.1 Introduction

The objective of this chapter is to assess the impact of the use of SFCD on fluid flow calculations rather than the simple scalar transport problems discussed in chapter 5. This is done by using the MIXFLO program for stirred tanks described in the previous chapter. A secondary objective is to comment on the accuracy of the MIXFLO predictions. Computations performed by Peter Collins at Imperial College using SFCD and UD are presented on computational meshes of 16,000, 39,000 and 87,000 grid points. Detailed comparisons with experimental data have been reported by Collins (1990) so only a limited assessment will be presented in this work. Attention is focussed on the response to grid refinement of the two discretisation schemes to investigate the benefits of the use of the higher order method.

This chapter is organised as follows. Section 7.2 describes the stirred tank geometry used in this chapter and discusses the calculations performed. The flow patterns themselves, the response to grid refinement and discretisation scheme and comparison with experimental data is presented in section 7.3. Finally, section 7.4 combines the findings of the previous two sections and comments on the impact of SFCD in this type of calculation.

7.2 Stirred Tank Calculations

The stirred tank geometry selected for the tests of SFCD is listed in table 7.2.1. The impeller diameter and clearance are one third of the diameter of the tank. Yianneskis (1982) and Hockey (1990) performed detailed measurements in this geometry using water as the process fluid and an impeller speed of 300rpm. Comparisons with this data will be presented in the next section.

Diameter (T_d)	0.294
Height (H_d)	0.294
Clearance (Z)	0.096
Impeller Diameter (D)	0.098
Disc Diameter (D_d)	0.074
Blade Length (L_b)	0.025
Blade Height (w)	0.020
Baffle Width (B)	0.029

Table 7.2.1: Dimensions of Yianneskis Mixing Vessel
The dimensions refer to figure 6.5.1.

Collins (1990) performed MIXFLO computations on four computational meshes with UD and SFCD ($r_{max} = 1$) as detailed in table 7.2.2. The total number of iterations and absolute amounts of CPU time used in each computation was not reported. Within the time constraints available, he was not able to fully converge the finest calculations and a comparison of computational cost was not given for these runs. An analysis of the iteration-to-iteration change in the results suggest that the uncertainty in the velocity results was in the region of 0.1% and 0.4% for the pressure and turbulent kinetic energy fields. Therefore it is safe to use the results of this computation for the purposes of comparison and assessment of SFCD.

It was found that the convergence rate and CPU time required for each iteration on the 87,000 point grid was very similar to those for an UD calculation performed on the same grid. The conclusion was that SFCD was not responsible for this convergence problem which must be attributed to some other aspect of the solution algorithm.

Number of Grid Points				Normalised CPU Time		CPU Time per Grid Point		CPU Time Ratio SFCD/UD
Axial	Radial	Circumferential	Total	UD	SFCD	UD	SFCD	
-	-	-	2,240	0.015	0.057	0.101	0.407	3.74
31	27	19	15,903	1.00	2.07	1.00	2.07	2.07
44	34	26	38,896	10.9	15.2	4.45	6.21	1.40
62	39	36	87,048	-	-	-	-	-

Table 7.2.2: MIXFLO Computations Performed by Collins (1990)
CPU times are normalised by the 15,903 point UD computation.

Collins (1990) notes that the use of SFCD in the k and ϵ equations resulted in negative values at a small number of grid points early in the calculation. This problem rapidly corrected itself as the calculation proceeded. He also found negligible differences between the solutions on a 16,000 point grid when SFCD was employed only for the momentum equations and when it was employed for the k and ϵ equations as well. This was attributed to the turbulence equations being dominated by their source terms and therefore the solution was only slightly changed by the different discretisation of the convection terms. Because of this observation, and to avoid any computational difficulties, Collins used UD for the turbulence equations for the purposes of his study. (It should be noted that, as will be shown in the next section, the turbulence fields resulting from the use of SFCD for the momentum equations differ markedly from their UD counterparts. The reason for this is that SFCD resolves the velocity gradients to a greater extent than the diffusive UD scheme with the consequence that the turbulence generation rate, equation (2.3.15) is higher.)

No study was undertaken to determine the cause of these boundedness problems with the turbulence equations. It is likely that the reason they were observed in the early stages of the solution was that the non-linear SFCD equations were not solved to a sufficiently high tolerance. The algorithms presented in section 4.9 do not ensure that boundedness is maintained if only an approximate solution to the discrete equations is obtained.

Two observations can be immediately made from table 7.2.2. The first is that SFCD is more expensive than UD by a factor between 1.40 and 3.74. However the extra cost is reduced as the grid is refined. The second observation is that the CPU cost per grid point increases rapidly as the number of grid points is increased, regardless of the discretisation scheme used. For instance, the cost per grid point increases by a factor of 4.4 for UD and 3.0 when going from 16,000 points to 39,000 points. This corresponds to an increase of only 35% in the number of grid points in each linear dimension. This increase in computational cost is due to an increased number of iterations being required to reach convergence.

Table 7.2.3 examines the cost of reducing the error UD and SFCD assuming that they exhibit first and second order behaviour respectively. The row for the 2,240 point grid should be viewed with caution because it is unlikely that the grid is sufficiently refined for the first and second order nature of the scheme have manifest themselves. The linear grid point density is defined as the cube root of the number of grid points which is inversely proportional to the characteristic grid spacing, Δ . The grid density in table 7.2.3 has been normalised by the value for the 16,000 point grid. The error in the solution is proportional to Δ for a first order scheme and Δ^2 for a second order scheme. Therefore a normalised linear grid point density of 1.35 implies that the error is reduced to 0.74 ($= 1.35^{-1}$) of its value on the 16,000 point grid for UD. The corresponding figure for SFCD is 0.55 ($= 1.35^{-2}$).

Table 7.2.3 compares the increase in CPU times with the reduction in the error for UD and SFCD. It can be seen that UD requires an eleven-fold increase in computing time to reduce the error by 26% while SFCD achieves a 45% reduction for only a seven-fold increase in computing time. It is expected that this difference would be even more pronounced if data for the 87,000 point grid was available. We therefore conclude that SFCD is a more cost-effective way of reducing the error than the use of UD. However, the current solution algorithms, regardless of the discretisation scheme, become increasingly inefficient as the grid is refined. This is a serious problem that is beyond the scope of the current study.

Grid	Normalised Linear Grid Point Density	UD		SFCD	
		Relative Error	Relative CPU Time	Relative Error	Relative CPU Time
2,240	0.52	1.92	0.015	3.69	0.028
15,903	1.00	1.00	1.0	1.00	1.0
38,896	1.35	0.74	10.9	0.55	7.3
87,048	1.76	0.57	-	0.32	-

Table 7.2.3: Comparison of Cost-Effectiveness of UD and SFCD

CPU times and linear grid point densities are normalised by the corresponding 15,903 point computation.

The above analysis has been concerned with theoretical reductions in the error when using UD and SFCD and does not conclusively prove the superiority of SFCD over UD. It is also important to consider the magnitude of the error when using the two discretisation schemes. This is done in the next section.

7.3 Computational Results

This section is concerned with the results of the MIXFLO computations performed by Collins (1990) and discussed in greater detail by him. First the flow patterns in stirred tanks will be described, then results using the various discretisation schemes and grids will be compared to assess the performance of SFCD in complex engineering calculations and finally MIXFLO itself will be assessed by comparing the result with experimental data.

Flow Pattern

The flow pattern in this vessel has been described many times (e.g. Yianneskis, 1982; Bolour-Froushan, 1986; Politis, 1989b; Collins, 1990) so only a limited description is presented here. Vector plots from the 39,000 point SFCD results are presented in figures 7.3.1 and 7.3.2.

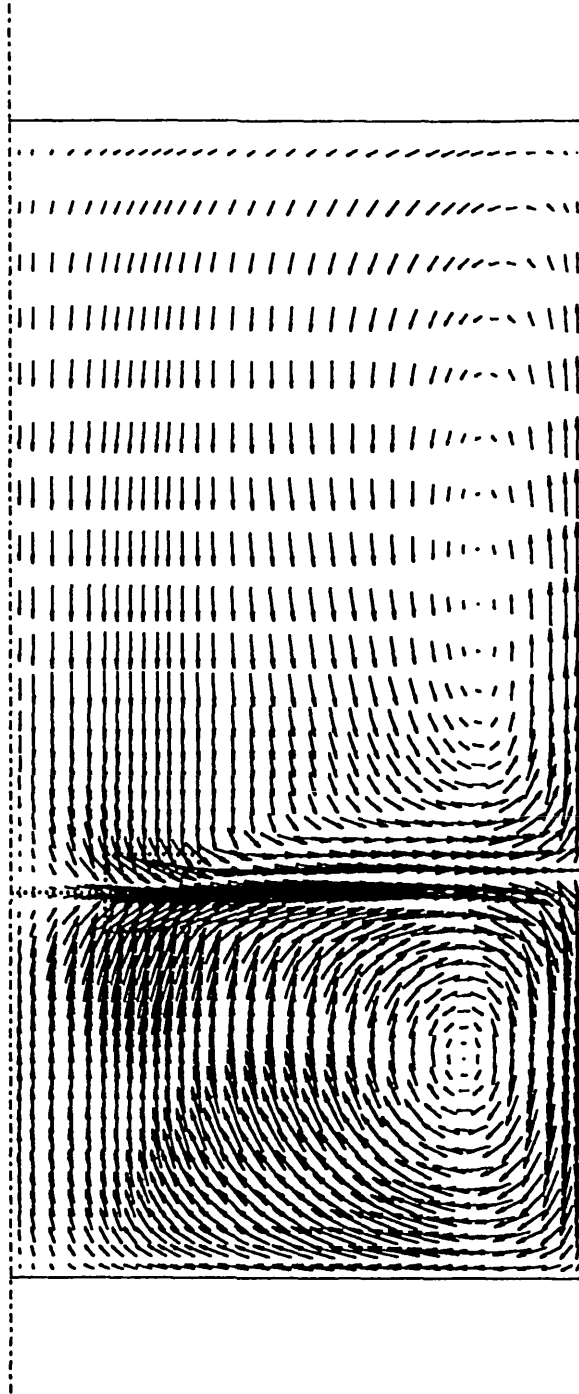
The most striking feature of the flow in this stirred vessel is the two ring vortices that exist above and below the impeller jet. The impeller forms a radial jet of fluid between these vortices. This jet, which has a slight upward inclination, is deflected by the wall and the baffles into the upwards and downwards directions. The fluid flows around the ring vortex and back into the impeller region again. The velocities in the lower ring vortex are generally stronger than in the upper vortex due to its confinement into the lower third of the vessel. Smaller recirculations are to be found behind the baffles and near the tank axis at the bottom of the vessel (figure 7.3.2).

The turbulence fields are highest in the region of the impeller due in part to the high turbulence levels set as part of the impeller boundary condition (section 6.5). They decay rapidly in the impeller jet and are lower still in the bulk of the vessel. A detailed discussion of turbulence predictions in stirred tank reactors is given by Bolour-Froushan (1986), although his results are acknowledged by him to be contaminated by numerical diffusion.

Computational Results

Discrepancies between experimental and computational results in CFD can arise from three sources. These are numerical truncation errors (i.e. insufficient grid refinement), modelling (e.g. turbulence model, boundary conditions etc.) and inexact solution of the discretised equations. The latter error can be assessed by monitoring the solution as the calculation proceeds and can be controlled by means of the convergence criterion. It has already been noted that this is less than 0.4% for the calculations presented herein and therefore this source of error can be eliminated from further consideration. The numerical errors can be assessed by performing computations on successively finer grids until such time as the solution does not change upon grid refinement. Modelling errors can be examined once numerical errors have been eliminated by this refinement process.

Results from 16,000 and 39,000 point calculations with both discretisation schemes as well as the 87,000 point SFCD computation are presented in figures 7.3.3 to 7.3.7. Experimental data from Yianneskis (1982) and Hockey (1990) are also shown. It should be noted that the finest grid contains less than twice the number of points in each direction of the 16,000 point grid. This is less than is desirable for a conclusive assessment of numerical errors, but the escalating computational costs prohibited further refinement.



Velocity vectors in the XY Plane
Scale : $\longrightarrow$ $0.87E+00$ m/s

Figure 7.3.1: MIXFLO Results (Vertical Plane at 2°)

Results obtained from 39,000 point SFCD Computation. Results on other planes show the same flow structures.

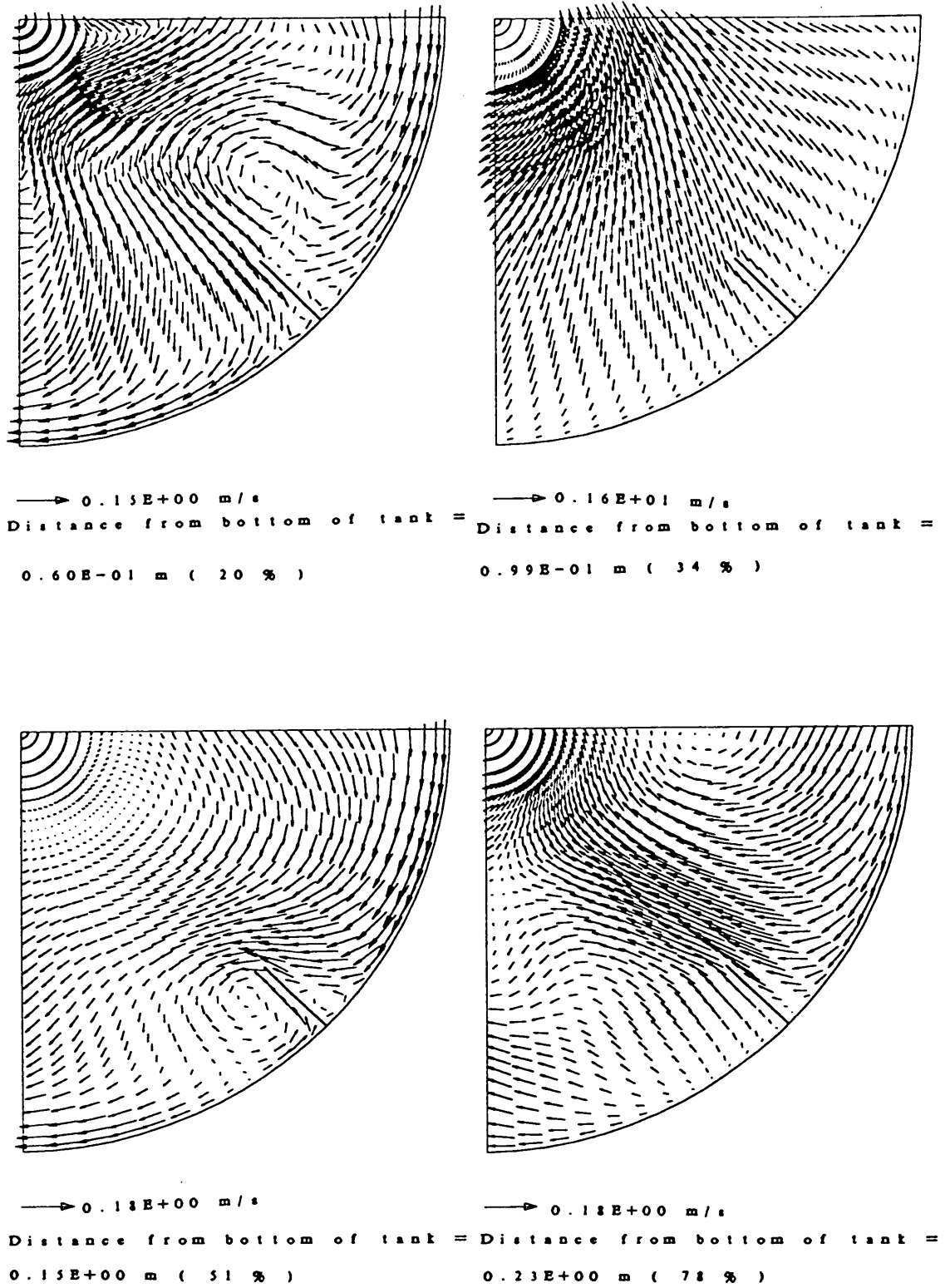


Figure 7.3.2: MIXFLO Results (Horizontal Planes)
 Results obtained from 39,000 point SFCD Computation.

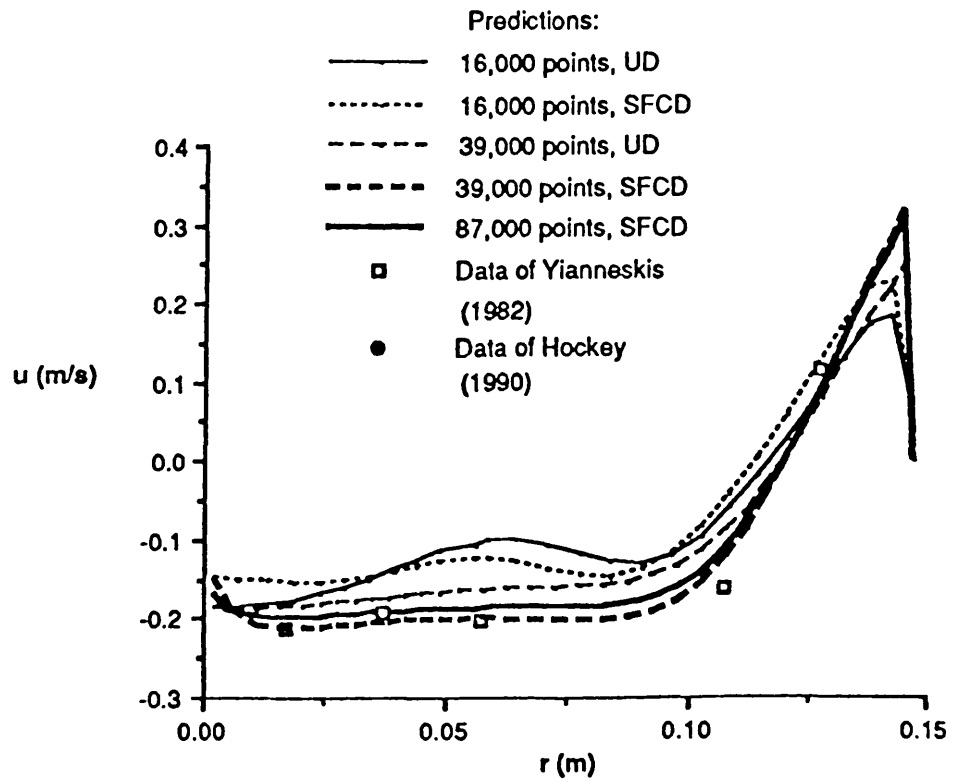
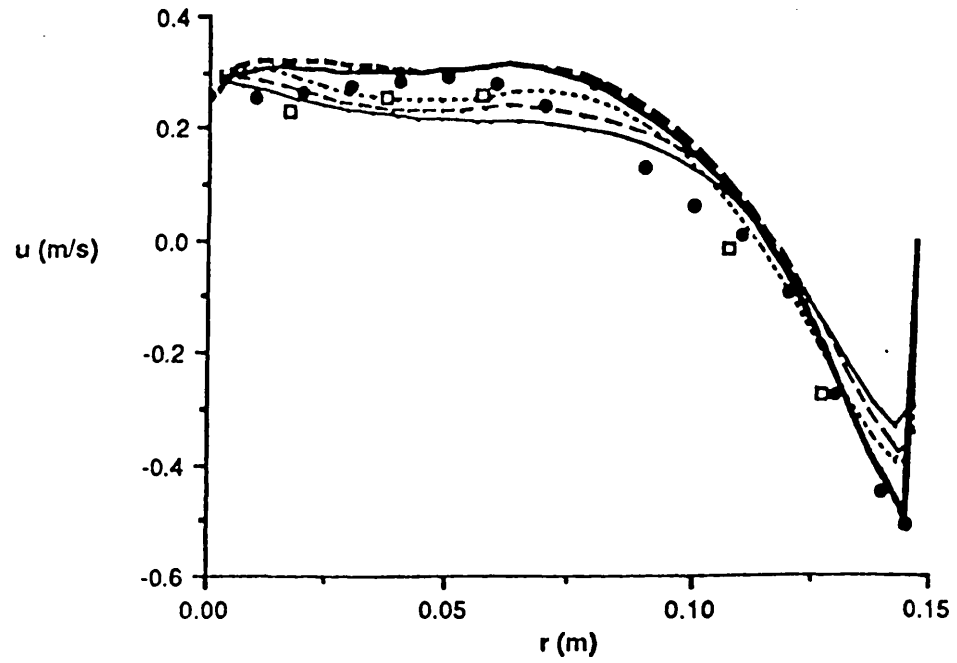


Figure 7.3.3: MIXFLO Results

Axial velocity (u) as a function of radius at heights of 0.05m (top) and 0.20m on the plane midway between the baffles.

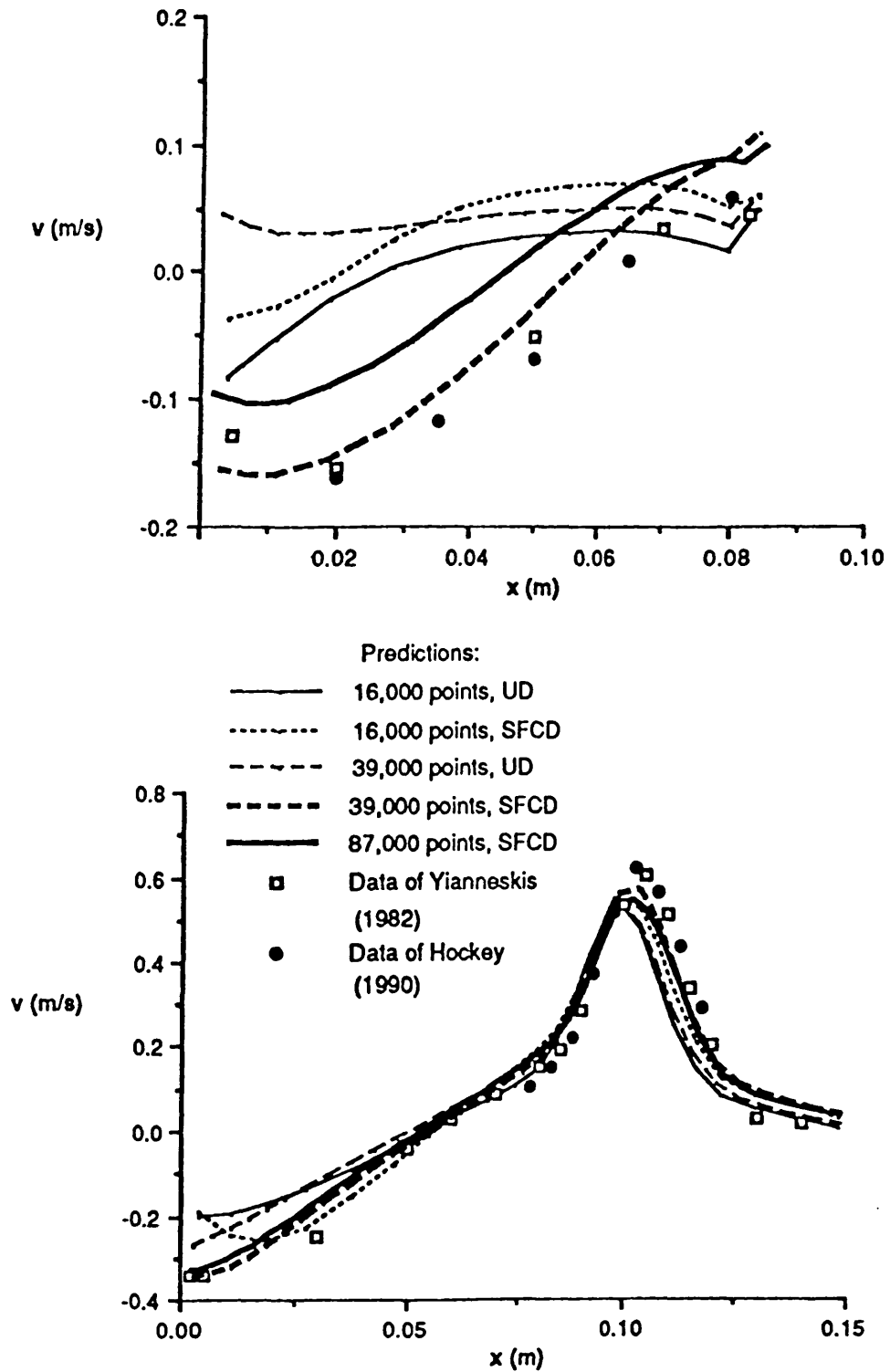


Figure 7.3.4: MIXFLO Results
Radial velocity (v) as a function of height (x) at radii of 0.05m (top) and 0.11m on the plane midway between the baffles.

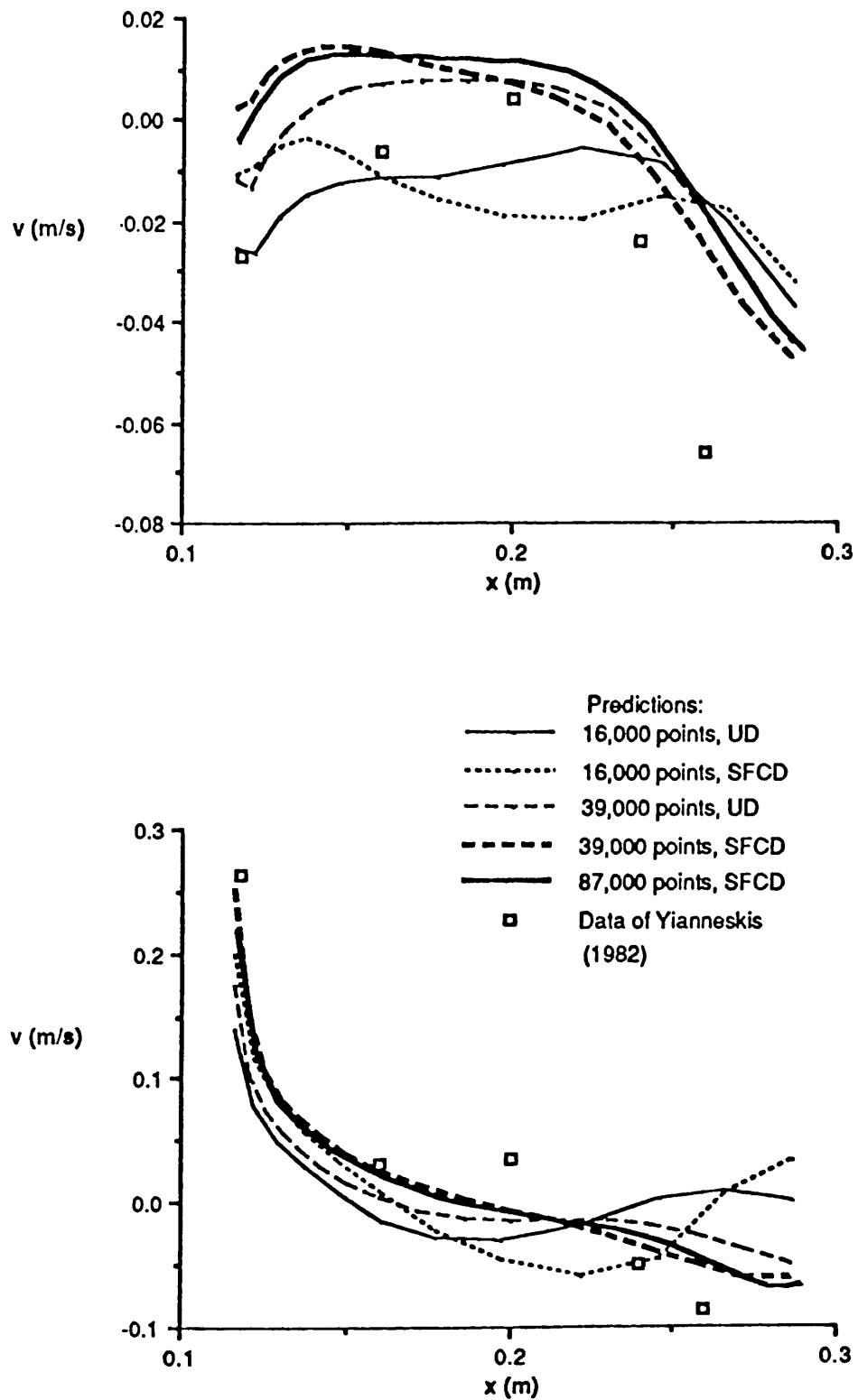


Figure 7.3.5: MIXFLO Results

Radial velocity (v) as a function of height (x) at radii of 0.057m (top) and 0.107m on the plane midway between the baffles.

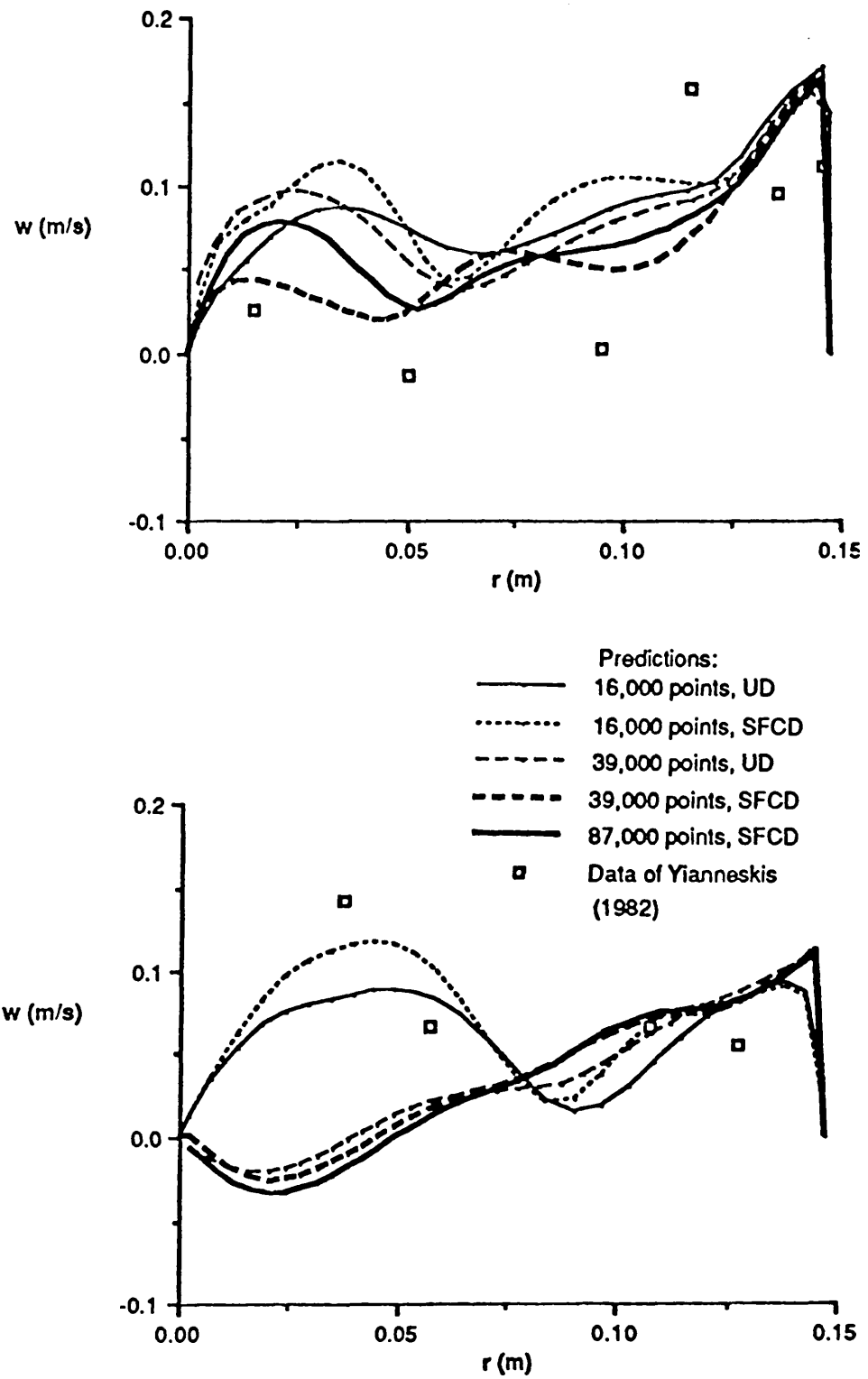


Figure 7.3.6: MIXFLO Results
Circumferential velocity (w) as a function of radius at heights of 0.07m (top) and 0.20m on the plane midway between the baffles.

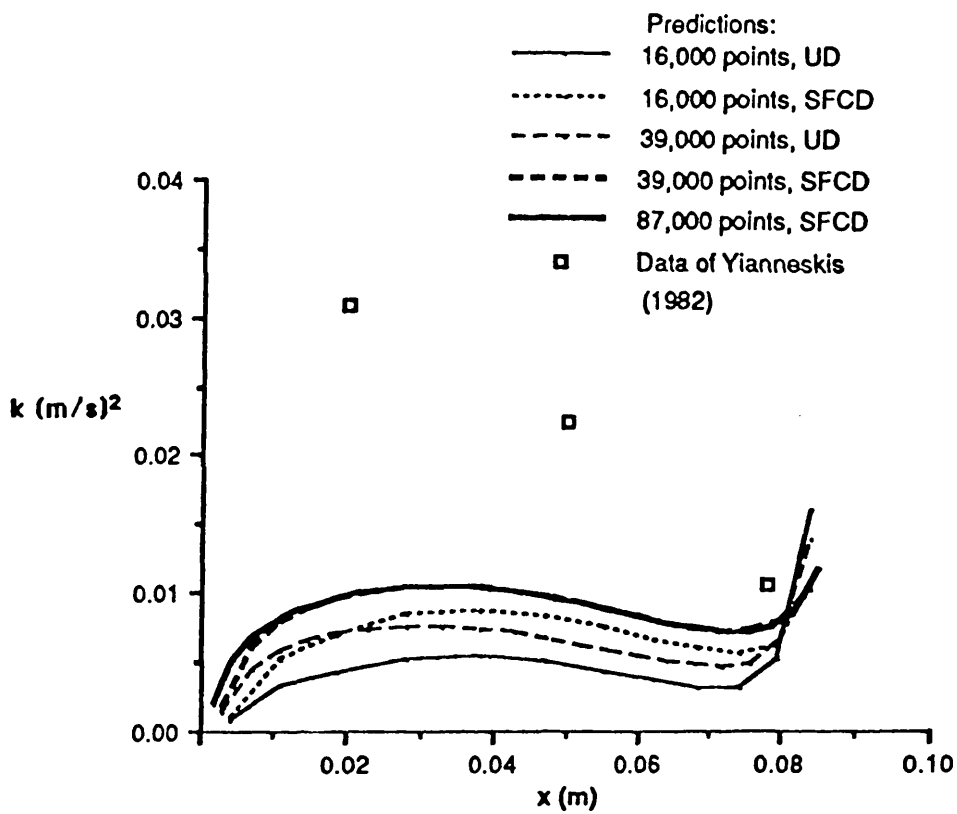
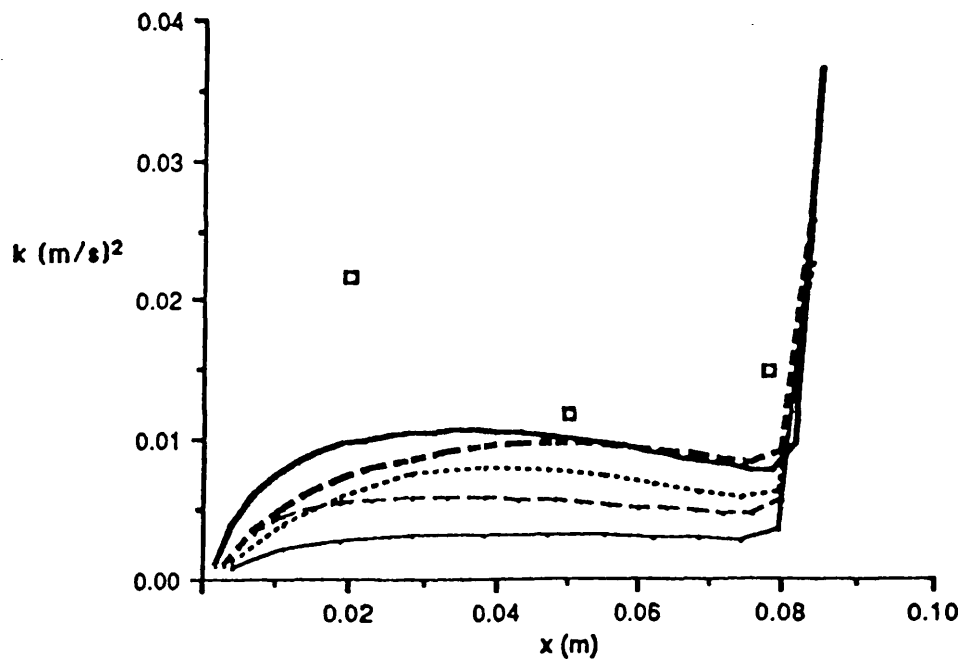


Figure 7.3.7: MIXFLO Results

Turbulent kinetic energy as a function of height (x) at radii of 0.037m (top) and 0.214m on the plane midway between the baffles.

The variations in axial velocity with discretisation scheme and mesh are not large (figure 7.3.3). However it is noticeable that the SFCD results on the 39,000 and 87,000 points grids are very similar suggesting that grid independence is close to being achieved. The performance of the 16,000 point SFCD computation is generally similar to the 39,000 point UD calculation which took five times as long to compute. The agreement with experimental data on the finest mesh is satisfactory, particularly in the upper recirculation zone (lower graph in figure 7.3.3). The maximum downward velocity near the wall in the lower recirculation zone is accurately predicted, but there is a discrepancy at a radius of about 0.1m.

The radial velocity below the impeller in the lower recirculation zone (upper graph in figure 7.3.4) shows a wide variation in values. It is clear that grid independency has not been achieved. However the magnitude of the velocity is somewhat smaller here than in other parts of the vessel and the magnitude of the difference between experimental and computational data is no greater than that elsewhere.

The other diagram in figure 7.3.4 encompasses part of the impeller jet. Again the SFCD results on the finest two meshes are in good agreement. Also, the 16,000 point SFCD computation is closer to the grid independent results than the 39,000 point UD case. The agreement with experimental data is reasonable although the strength of the impeller jet is under-predicted.

Figure 7.3.5 shows the radial velocity in the upper recirculation zone. The finest two SFCD results are again similar and the 39,000 point UD results are better than the 16,000 point SFCD predictions. There is some discrepancy between predictions and the experimental data although the magnitude of the discrepancy is not large.

The circumferential velocity results are shown in figure 7.3.6. Some degree of grid independency is apparent in the upper recirculation zone (lower graph) but not in the lower zone. In the former case, the results on the coarse grids provide better agreement with the experimental data than those in the finer meshes. In the lower recirculation zone (upper graph) the velocity is generally over-predicted.

Turbulence kinetic energy comparisons are presented for the lower recirculation zone in figure 7.3.7. The results are much lower than the experimental data, although the SFCD scheme produces significantly higher predictions than UD. As already mentioned, this is due to better resolution of the velocity gradients. The SFCD calculations on the finest meshes show signs of grid independency.

The Impact of SFCD

The use of SFCD has had a considerable impact on the MIXFLO predictions. The accuracy has improved on all grids and there are clear indications that grid independency is being approached. The 16,000 point SFCD computation and 39,000 point UD calculation have similar degrees of accuracy but the former only required about one fifth of the amount of CPU time. It is therefore not unreasonable to claim that SFCD is five times more cost-effective than UD at these accuracies. The analysis presented in the previous section suggests that this figure will increase as the accuracy requirements increase.

It is important to note that full grid independency requires that all the variable fields do not vary as the mesh is refined. If one variable is changing as the number of grid points increases, then it is possible that this may affect others even if they exhibit grid independence. In view of the differing computational results shown in figures 7.3.4 and 7.3.6, calculations on even finer grids will be required before grid independent solutions are obtained.

Assessment of MIXFLO

The flow patterns predicted by MIXFLO are in broad agreement with the experimental data. The axial and radial velocities are generally well predicted but the circumferential velocity less so. It is not altogether surprising that the agreement is generally better in those parts of the vessel where the flow is strong and less good where the flow is weak. The turbulent kinetic energy comparisons are poor. The grid refinement tests suggest that the problem lies with the modelling employed in the calculation and not in numerical errors. The most likely weaknesses are the turbulence model and the impeller model. The former was investigated by Bolour-Froushan (1986) who considered modifications to the $k - \epsilon$ model. This work was hampered by difficulty in distinguishing between numerical and modelling errors. The subject of impeller modelling and the effect of the trailing vortices on the mean flow and turbulence characteristics has not yet been subjected to detailed scrutiny. A study of this kind is now required to improve the accuracy of the predictions.

It is interesting to note that despite the poor turbulence predictions the mean velocities generally compare favourably with experimental data. This could be due to a number of causes.

- (i) Although the relative errors in the Reynolds stresses are large, they are small compared to the size of other terms in the momentum equations. Hence, the flow is insensitive to the errors in the turbulence predictions.
- (ii) Errors in the turbulence model are balanced by weaknesses in the Bousinesq approximation so the Reynolds stresses are more accurately predicted than the turbulence itself.
- (iii) The turbulence model 'constants' applied in this study correctly predict the stresses in those regions of the flow where they are important.
- (iv) The impeller model employed is satisfactory for the momentum equations but not for the turbulence equations. In particular the trailing vortices, which are not true turbulence, may be inadequately modelled.

Some of these points can be investigated by comparing the magnitude of the various terms in the momentum equations to determine those parts of the flow where the Reynolds stresses are significant.

7.4 Conclusions

The implementation of SFCD via the EFCA as described in chapter 6 does not cause any algorithmic difficulties and the methodology can be successfully applied to solve the Navier-Stokes equations. There is an increase in CPU time which is more than offset by the gain in accuracy. The performance of SFCD superior to that of UD and the cost-effectiveness of the technique has been improved by a factor of at least five. The use of SFCD can be recommended for general CFD problems.

The solution algorithm becomes increasingly inefficient as the grid is refined, regardless of the discretisation scheme employed. This makes grid independent solutions (which will require 100,000 or more points) very expensive to obtain, even with the improved discretisation technique.

MIXFLO is capable of predicting the velocities in a stirred tank, but the turbulence predictions are poor. The accuracy of the predictions is limited by the physical modelling and not the discretisation methods applied.

Chapter 8: Calculation of Power Consumption and Flow Trajectories

The objective of this chapter is to demonstrate how the flow fields calculated by MIXFLO can be used in the prediction of certain other quantities of interest that depend on the flow field. Two quantities of practical interest are selected. The first is the power supplied by the agitator to the fluid in the tank (section 8.1) and the second is the mean fluid flow paths (section 8.2).

8.1 Calculation of Power Consumption

There are two approaches to calculating the power dissipated in a stirred tank from the flow field calculated by MIXFLO. The first is to calculate the power directly from some form of energy balance and the second is to infer the power from the torque on the impeller shaft, as will be done here. The former approach has been used by Bolour-Froushan (1986) but the latter has not been previously applied in a computational procedure.

The section is organised as follows. First the concept of a power number is introduced. This is followed by a description of Bolour-Froushan's energy based method. The new method based on torque is described next and finally results presented and discussed.

Power Number

It is useful to introduce the concept of the dimensionless power number, P_o when discussing power consumption in stirred tanks. It is defined in terms of the input power, P_i , rotational speed, N , and impeller diameter, D , by

$$P_o = \frac{P_i}{\rho N^3 D^5}. \quad (8.1.1)$$

It is found experimentally (e.g. Uhl and Gray, 1966) that at high Reynolds numbers the power number is independent of viscosity or any other physical property of the fluid or operating conditions. This is the regime which is relevant to this study.

Calculation of Power from an Energy Balance

Bolour-Froushan (1986) developed a formula for the power dissipated by the impeller in a stirred vessel by deriving a transport equation for the total kinetic energy defined as

$$E = \frac{1}{2} \underline{u} \cdot \underline{u} + k \quad (8.1.2)$$

from the momentum and k equations. The resulting power formula, in tensor notation, is

$$P_i = \int_S \left(\rho u_i E - \sigma_{ij} u_i - \mu_i \frac{\partial k}{\partial x_j} \right) dS_j + \int_V \rho \epsilon dV \quad (8.1.3)$$

where V is an arbitrary volume with surface S that encloses the impeller and

$$\sigma_{ij} = -p \delta_{ij} + \mu_{eff} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \rho k. \quad (8.1.4)$$

There are two obvious choices for V . The first is to consider the volume immediately surrounding the impeller itself. This is denoted as method 1 (figure 8.1.1). Here one has to evaluate every term in equation (8.1.3). The second, method 2, is to take V to be the whole vessel. The surface integral is then zero due to the boundary conditions so the impeller formula becomes simply

$$P_I = \int_V \rho \epsilon dV. \quad (8.1.5)$$

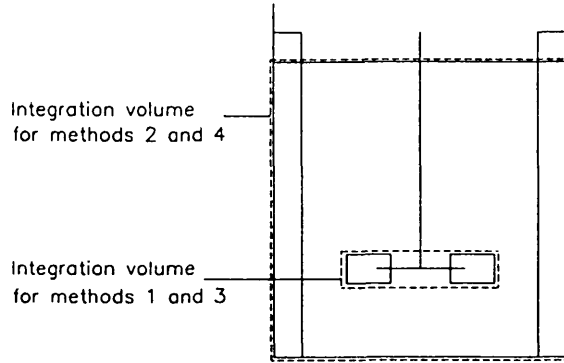


Figure 8.1.1: Integration Volumes for Power Calculations

Method 1 is dominated by the term describing the convective transport of kinetic energy (Boulour-Froushan, 1986). This varies as the third power of velocity. The velocity scale is related to the impeller tip speed by ζ_ω (section 6.5) and therefore the predicted power will vary as the cube of ζ_ω . Hence a small change in ζ_ω can cause an appreciable change in the predicted power.

Trial calculations with method 2 have shown the predicted power to be very sensitive to the ζ_ϵ value used to determine the ϵ boundary condition for the impeller. This is probably the least well quantified parameter in the impeller model and it is therefore undesirable to base predictions upon it. A second drawback of method 2 arises in systems with more than one agitator. It is impossible to calculate the power of the individual impellers, only the total power.

Calculation of Power from an Angular Momentum Balance

This new method of calculating power in a stirred vessel is based on the observation that the power and torque (Tr) are related by

$$P_I = 2\pi N Tr. \quad (8.1.6)$$

In appendix A.8 the impeller torque is shown to be given by

$$Tr = \int_S (\rho u_r u_\theta - \sigma_{\theta r}) r dS_I. \quad (8.1.7)$$

Again the same two volumes of integration are available; integration over a surface surrounding the impeller (method 3) and consideration of the complete volume of the tank (method 4).

We first consider method 3. If we take the integration volume as the cylinder enclosing the impeller then the only non-zero contribution from the stress tensor is due to the shear stresses $\sigma_{\theta r}$ and $\sigma_{r\theta}$. Also the power supplied by each impeller in a vessel agitated by two or more impellers can be found.

Method 4 is also easy to apply. The convective part is zero on the bounding surfaces. The stress tensor contributions arise from the pressure drop across the baffles and the shear stress on the wall. However one cannot calculate the power of individual impellers in a multi-impeller system. Therefore method 3 is preferred to method 4.

We see from (8.1.7) that the torque varies as the square of the velocity. Therefore one would expect the predicted power to vary as the square of ζ_w and test calculations have confirmed this.

Selection of Power Calculation Method

If the calculation of the flow in the stirred vessel was entirely consistent then all approaches to the calculation of the impeller power number should produce the same result. However the boundary conditions used to model the impeller involve the specification of velocity, turbulent kinetic energy and dissipation levels. It is possible that the specified levels are not consistent with the actual forces applied by the blades of the impeller. Therefore, as a result of this inconsistency, the various methods of calculating power may produce different results. The magnitude of the difference in the power numbers predicted are a measure of the discrepancy.

A practical method for predicting power should display the least sensitivity to the impeller boundary conditions, especially ζ_ϵ which is the hardest to accurately characterise. Additionally the method should be capable of calculating the power delivered by each impeller in a multi-agitator system. This second criterion eliminates methods 2 and 4 from the selection. Method 1 has a cubic sensitivity to ζ_w and has some direct dependency on ζ_ϵ from the final term in (8.1.3). The power number from method 3 only varies quadratically with ζ_w . Therefore this approach will be used for power calculations in this work.

Results

Power numbers were calculated for the stirred tank configuration used in the previous chapter. Results for five different computational meshes are presented in table 8.1.1 for computations with both SFCD and UD. Yianneskis (1982) measured the power number to be 4.8.

Grid Size (points)	Power Number	
	UD	SFCD
3,036	6.24	6.74
7,200	4.99	5.48
11,664	4.86	5.03
15,903	4.67	4.75
38,896	4.81	5.05

Table 8.1.1: Power Number Calculations (Method 3)

The results in table 8.1.1 agree well with the experimental value, particularly on the finest mesh. However it is clear that a fully grid independent results has not yet been obtained. This is consistent with the findings of the previous chapter.

Bolour-Froushan (1986) obtained a power number of 4.35 from method 1. The convection fluxes in the power formula (8.1.3) were obtained by integrating the experimental results of Yianneskis. The pressure and shear stress terms were evaluated from the MIXFLO results and only accounted for 15% of the observed power. The energy dissipated in the impeller region was not included in the calculation so the result is likely to be an underestimate.

The only power number based entirely on computational results presented by Bolour-Froushan (1986) was using method 2 (equation (8.1.5)). Upwind differencing and a grid containing 25920 cells was used. The calculated power number was 3.7 which is substantially below the observed value. The cause of this discrepancy was attributed errors in the value of ζ_ϵ in the impeller boundary condition or the near wall approximations used to calculate ϵ at the boundary nodes.

Conclusions

It may be concluded that the method adopted in this study for the calculation of power numbers will produce satisfactory results providing the computational mesh is sufficiently fine. The predicted power numbers are less sensitive to the impeller boundary conditions than the methods employed in previous studies. This is advantageous when applying the computational procedure to other vessel geometries where ζ_w and ζ_ϵ are less well characterised.

The success of the calculation of a power number is dependent on the accuracy of the calculation of the flow fields by MIXFLO and in particular the impeller boundary conditions. Incorrect boundary conditions will result in a poor estimate of the power number regardless of the method used to calculate it.

8.2 Calculation of Flow Paths

The objective of this work is to provide a method for the calculation of the mean path taken by fluid elements in a stirred tank from the velocity field. There are several reasons why this information is needed. The first is visualisation of the complex flow patterns that arise in stirred vessels. This will provide a pictorial representation of the way the various flow structures interact in three dimensions and time. This simplistic picture can also be used as a qualitative tool to investigate mixing by looking at the way paths starting close to each other diverge and move into other recirculation regions of the tank. In many cases the actual flow path itself contains valuable information. For instance it may be of interest to know how frequently a microbe following the flow in a fermentation vessel enters a region of high oxygen concentration.

This section is organised as follows. The influence of turbulence on the flow paths is briefly discussed. Then the algorithm used to calculate the fluid flow path is introduced and finally example calculations of flow paths are presented.

The Effect of Turbulence on the Mean Flow Paths

The equation of a flow path is

$$\frac{d}{dt} \underline{R}(t) = \underline{U}(\underline{R}(t)) \quad (8.2.1)$$

where $\underline{U}$ is the instantaneous velocity and $\underline{R}(t)$ is the trajectory. This is not a practical equation for the calculation of flow paths because only the mean velocity, not the instantaneous velocity $\underline{U}$, is available

from the flow calculation. Moreover, for flow visualisation one is more interested in the mean flow path than the instantaneous flow paths. We therefore derive an equation of motion for the mean flow path, $\underline{r}(t)$. First we decompose $\underline{U}$ into mean ($\underline{u}$) and fluctuating ($\underline{u}'$) parts in the usual manner.

$$\underline{U}(t) = \underline{u}(t) + \underline{u}'(t) \quad (8.2.2)$$

We define $\underline{r}$ as the solution of

$$\frac{d}{dt}\underline{r}(t) = \underline{u}(\underline{r}(t)) \quad (8.2.3)$$

and the flow path fluctuations $\underline{r}'$ by

$$\underline{r}'(t) = \underline{R}(t) - \underline{r}(t). \quad (8.2.4)$$

Substitution of these three equations into (8.2.1) gives

$$\frac{d}{dt}\underline{r}(t) + \frac{d}{dt}\underline{r}'(t) = \underline{u}(\underline{r} + \underline{r}') + \underline{u}'(\underline{r} + \underline{r}'). \quad (8.2.5)$$

Expansion of this as a Taylor series about $\underline{r}$ and subtraction of (8.2.3) leaves

$$\frac{d}{dt}\underline{r}'(t) = \underline{u}' + \underline{r}' \cdot (\underline{\nabla} \otimes \underline{u}) + \underline{r}' \cdot (\underline{\nabla} \otimes \underline{u}') + O(\underline{r}'^2) \quad (8.2.6)$$

where all velocities and derivatives are evaluated at $\underline{r}$. If we ensemble average this equation. The first and second terms on the RHS are zero by definition. Further, temporal differentiation commutes with ensemble averaging so we are left with

$$\frac{d}{dt}\overline{\underline{r}'} = \overline{\underline{r}' \cdot (\underline{\nabla} \otimes \underline{u}')} + O(\overline{\underline{r}'^2}). \quad (8.2.7)$$

It is easy to see from (8.2.7) that if the initial disturbance in the flow path is zero then $\underline{r}'$ will be zero for all time. Thus ensemble averaging (8.2.4) gives

$$\overline{\underline{R}}(t) = \underline{r}(t). \quad (8.2.8)$$

The effect of turbulence is to provide a source of disturbances in the flow paths. Depending on the behaviour of the RHS of (8.2.7), these disturbances may grow so that $\overline{\underline{r}'} \neq 0$ and (8.2.8) cannot be obtained. Therefore the solution to (8.2.3) is only an approximation to the mean path. A demonstration of this is given in appendix C.8.

There is another limitation concerning the concept of the mean path that arises when recirculating flows are considered. The magnitude of the departure between mean and instantaneous paths of fluid elements (i.e. $\overline{\underline{r}' \cdot \underline{r}'}^{1/2}$) will grow in time. However the magnitude of $\underline{r}$ is limited by the size of the recirculation zone. Eventually a time will be reached when these two quantities are comparable and the instantaneous positions of the fluid elements will be distributed around the whole recirculation zone. The concept of a mean position, and therefore of a mean path, at later times becomes meaningless.

Algorithm for Determination of Mean Flow Paths

Substitution of the assumption $\underline{r}' = 0$ into (8.2.5) leaves

$$\frac{d}{dt}\underline{r} = \underline{u}(\underline{r}). \quad (8.2.9)$$

The solution of this equation is a straightforward numerical problem once $\underline{u}$ is specified. MIXFLO provides the components of the velocity at a finite set of points on the computational mesh. The problem is how to construct $\underline{u}$ from the discrete set of velocity components. The choice of construction is not entirely arbitrary. The velocity in any flow is known to satisfy the continuity equation which in this case is

$$\nabla \cdot \underline{u} = 0. \quad (8.2.10)$$

It is clearly desirable that the constructed velocity field should also have this property. The calculated velocity components satisfy the discrete analogue of this relationship and this must be the starting point for our interpolation.

Mallinson and de Vahl Davis (1973) presented flow paths in three dimensions from a CFD calculation. They used a vector potential to calculate the velocity field, not the primitive variables (i.e. $\underline{u}$ and p) used in this work. The velocity interpolation was achieved by constructing an interpolation for the vector potential from which the local velocity can be derived which satisfies the continuity condition. The path itself was formed by integrating the equation of motion numerically within each cell. This approach is not suitable for this work because the vector potential is not available.

Handscomb (1984) has proposed a way of performing the interpolation of velocities in Cartesian coordinate systems. This involves the construction of a multidimensional spline for each velocity component such that satisfaction of the discretised continuity condition ensures that equation (8.2.10) is satisfied. The method produces a continuous function for the velocity which can be used to solve (8.2.9) numerically. Special care must be taken with walls or other impermeable features to ensure that the calculated flow line does not pass through them.

Handscomb justifies the sophistication of this spline approach on the grounds that the interpolated velocity is continuous everywhere which produces smooth flow lines. However the presentation of slightly disjoint lines does have one advantage, namely that it reminds the viewer that the produced lines are reconstructed from a finite data set and any sudden changes in direction indicate that the calculation has poorly resolved the flow in that region. There are three other drawbacks of Handscomb's approach for the presentation of flow trajectories in stirred vessels. The first is the complexity of the interpolation procedure, especially in three dimensions. Second, a numerical method is required to integrate (8.2.9). Third and most serious is that the extension of the method to cylindrical coordinate systems is non-trivial.

A practical solution to this problem, developed from a proposal by Gosman (1986), in cylindrical coordinates is given below. It is possible to generalise it to other coordinate systems, in particular Cartesian coordinates are very easy. This has already been done by Pollock (1988).

The continuity equation in cylindrical coordinates is

$$\frac{1}{r} \frac{\partial}{\partial r}(r u_r) + \frac{1}{r} \frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z} = 0. \quad (8.2.11)$$

The discrete analogue to this solved by MIXFLO is

$$2 \frac{u_{rn} r_n - u_{rs} r_s}{r_n^2 - r_s^2} + 2 \frac{u_{\theta d} - u_{\theta u}}{(r_n + r_s)(\theta_d - \theta_u)} + \frac{u_z - u_w}{z_s - z_u} = 0. \quad (8.2.12)$$

The individual coordinate directions can be decoupled by demanding that the first term in (8.2.11) is always proportional to the first term in (8.2.12) and so on. The proportionality constant, $\alpha(r, \theta, z)$, is the same for all three terms and may be a function of position. We can therefore write

$$\frac{1}{r} \frac{\partial}{\partial r} (r u_r) = 2\alpha \frac{u_{rn} r_n - u_{rs} r_s}{r_n^2 - r_s^2}, \quad (8.2.13)$$

$$\frac{1}{r} \frac{\partial u_\theta}{\partial \theta} = 2\alpha \frac{u_{\theta d} - u_{\theta u}}{(r_n + r_s)(\theta_d - \theta_u)}, \quad (8.2.14)$$

$$\frac{\partial u_z}{\partial z} = \alpha \frac{u_z - u_w}{z_s - z_u}. \quad (8.2.15)$$

We are free to choose the function α in any way so long as the integral of the resulting velocity over the face of the computational cell is equal to the volume flux used by MIXFLO. There is therefore potentially an infinite number of velocity interpolations that can be constructed.

The flux constraints are written below.

$$\int_{z_w}^{z_s} \int_{\theta_u}^{\theta_d} u_r(r_n, \theta, z) r_n d\theta dz = u_{rn} r_n (\theta_d - \theta_u) (z_s - z_w) \quad (8.2.16)$$

$$\int_{z_w}^{z_s} \int_{\theta_u}^{\theta_d} u_r(r_s, \theta, z) r_s d\theta dz = u_{rs} r_s (\theta_d - \theta_u) (z_s - z_w) \quad (8.2.17)$$

$$\int_{r_s}^{r_n} \int_{z_w}^{z_s} u_\theta(r, \theta_d, z) dr dz = u_{\theta d} (r_n - r_s) (z_s - z_w) \quad (8.2.18)$$

$$\int_{r_s}^{r_n} \int_{z_w}^{z_s} u_\theta(r, \theta_u, z) dr dz = u_{\theta u} (r_n - r_s) (z_s - z_w) \quad (8.2.19)$$

$$\int_{r_s}^{r_n} \int_{\theta_u}^{\theta_d} u_z(r, \theta, z_s) r dr d\theta = \frac{1}{2} u_{zs} (r_n^2 - r_s^2) (\theta_d - \theta_u) \quad (8.2.20)$$

$$\int_{r_s}^{r_n} \int_{\theta_u}^{\theta_d} u_z(r, \theta, z_w) r dr d\theta = \frac{1}{2} u_{zw} (r_n^2 - r_s^2) (\theta_d - \theta_u) \quad (8.2.21)$$

Handscorn's work in Cartesian coordinates is equivalent to taking α to be a quadratic function of its arguments. However for the purposes of this study we simply take

$$\alpha(r, \theta, z) = 1. \quad (8.2.22)$$

In Cartesian coordinates this is equivalent to assuming the velocity varies linearly with position in the finite volume cell. Now we solve equation (8.2.13) with (8.2.16) to obtain

$$u_r(r, \theta, z) = \frac{u_{rn} r_n - u_{rs} r_s}{r_n^2 - r_s^2} r - \frac{1}{r} \frac{u_{rn} r_s - u_{rs} r_n}{r_n^2 - r_s^2} r_n r_s, \quad (8.2.23)$$

which is consistent with (8.2.17).

The other two velocity components can be derived in an identical manner.

$$u_\theta(r, \theta, z) = 2r \frac{u_{\theta d} - u_{\theta u}}{(r_n + r_s)(\theta_d - \theta_u)} (\theta - \theta_u) + 2r \frac{u_{\theta u}}{r_n + r_s}, \quad (8.2.24)$$

$$u_z(r, \theta, z) = \frac{u_{z\theta} - u_{zw}}{z_\theta - z_w}(z - z_w) + u_{zw} \quad (8.2.25)$$

These interpolations for the velocity components will be applied in the domain

$$\begin{aligned} r_s &\leq r < r_n, \\ \theta_u &\leq \theta < \theta_d, \\ z_w &\leq z < z_\theta. \end{aligned} \quad (8.2.26)$$

Outside this domain the interpolation for the next computational cell is applied. Examination of the above three interpolated velocity components reveals that the velocity component normal to a cell face is continuous across it but the other two components are discontinuous. This may cause relatively abrupt changes in direction if the velocity components in the two cells differ appreciably which was Handscomb's concern with the simpler methods.

It is interesting to note that the interpolated value of u_θ is different from the value at the position where u_θ is stored. However the latter velocity should be regarded as the mean velocity over the whole cell face in the sense of equations (8.2.18) and (8.2.19) so the interpolation is not inconsistent with the data.

The interpolated velocity is always zero on any cell face where the velocity is zero. Therefore no special treatment of boundaries is required at the interpolation stage and flow paths will never attempt to cross boundary faces. Therefore boundaries do not need to be explicitly taken into account when applying this method. This is a highly desirable feature of this algorithm.

The next step is to integrate equations (8.2.23-25). This can be done analytically. Assuming that at time t_0 the fluid element is at position (r_0, θ_0, z_0) the evolution of the radial component is

$$\frac{dr}{dt} = u_r \quad (8.2.27)$$

where the RHS is given by (8.2.23). The solution of this equation can be shown to be

$$r(t) = \sqrt{2 \left(r_0^2 - r_n r_s \frac{u_{rn} r_s - u_{rs} r_n}{u_{rn} r_n - u_{rs} r_s} \right) e^{2 \frac{u_{rn} r_n - u_{rs} r_s}{r_n^2 - r_s^2} (t - t_0)} + 2 r_n r_s \frac{u_{rn} r_n r_s - u_{rs} r_n}{u_{rn} r_n - u_{rs} r_s}}. \quad (8.2.28)$$

The velocity in the circumferential direction is the product of the angular velocity and the radius thus

$$r \frac{d\theta}{dt} = u_\theta. \quad (8.2.29)$$

The solution to (8.2.24) and (8.2.29) is

$$\theta(t) = \left(\theta_0 - \frac{u_{\theta d} \theta_u - u_{\theta u} \theta_d}{u_{\theta d} - u_{\theta u}} \right) e^{2 \frac{u_{\theta d} - u_{\theta u}}{(r_n^2 - r_s^2)(\theta_d - \theta_u)} (t - t_0)} + \frac{u_{\theta d} \theta_u - u_{\theta u} \theta_d}{u_{\theta d} - u_{\theta u}}. \quad (8.2.30)$$

The axial position is the solution of

$$\frac{dz}{dt} = u_z \quad (8.2.31)$$

which is

$$z(t) = \left(z_0 - \frac{u_{ze} z_{ue} - u_{ze} z_e}{u_{ze} - u_{ze}} \right) S^{\frac{u_{ze} - u_{ze}}{u_{ze} - u_{ze}} (t - t_0)} + \frac{u_{ze} z_{ue} - u_{ze} z_e}{u_{ze} - u_{ze}} \quad (8.2.32)$$

The flow path is determined as follows. Identify the starting cell of the flow path. Construct the trajectory from equations (8.2.28), (8.2.30) and (8.2.32) and plot the trajectory until the path intersects the boundary of the cell and enters a new cell. Repeat the procedure for this new cell. This procedure is described in greater detail by Pollock (1988).

Example Flow Path Calculations

Flow path calculations were performed based on the 39,000 point SFCD MIXFLO calculation presented in the previous chapter. A selection of two and three dimensional projections of the flow path are shown in figures 8.2.1-3. In every case the radial position of the starting point is equal to the impeller radius (4.9cm) and in a plane containing the baffle. Different axial locations were used for the different flow paths. The path was followed for 1.3 seconds. Ten markers are placed along the path at equal time intervals of 0.16s and the end of a trajectory is marked with a circle.

A number of comments can be made concerning the trajectories presented.

- (i) The graphical presentation of flow paths provide an alternative manner of visualising the complex flow patterns in stirred vessels. In particular the structure of the upper and lower recirculation loops is apparent.
- (ii) The mean flow appears to cause some exchange of fluid between the upper and lower recirculation zones. However turbulence in the impeller jet will probably be the major means of exchange between these zones.
- (iii) Trajectories from neighbouring starting points diverge. This is particularly clear in the impeller region where a small change in axial location will cause a path to occupy a different recirculation zone.
- (iv) Circumferential motion takes place mainly in the impeller jet or very near to the axis of the tank.
- (v) The trajectory makes three to four revolutions around the lower recirculation loop for every revolution around the tank. Trajectories in the upper zone do not progress around the tank axis as smoothly as in the lower zone so the corresponding figure is not clear.

The recirculation time was investigated by counting the number of markers on the path on one revolution of the recirculation zone. This was found to be about one second for the lower recirculation loop and in the range 2.6 seconds for the upper loop. This suggests that the lower zone will mix faster than the upper zone. This is not surprising in view of its smaller volume and stronger flows.

Discussion

It has been demonstrated how the calculation of flow trajectories in stirred vessels aid in the visualisation of the flow and provide a means of estimating circulation times.

The computational procedure could also be applied in reverse. That is to say that the algorithm could be used to determine the flow trajectory that ended at a particular point. Another extension of the algorithm would be to superimpose random velocity perturbations related to the local turbulence intensity onto the mean velocity field. This could provide an alternative means to investigate the effects of turbulence on the large scale mixing in the vessel.

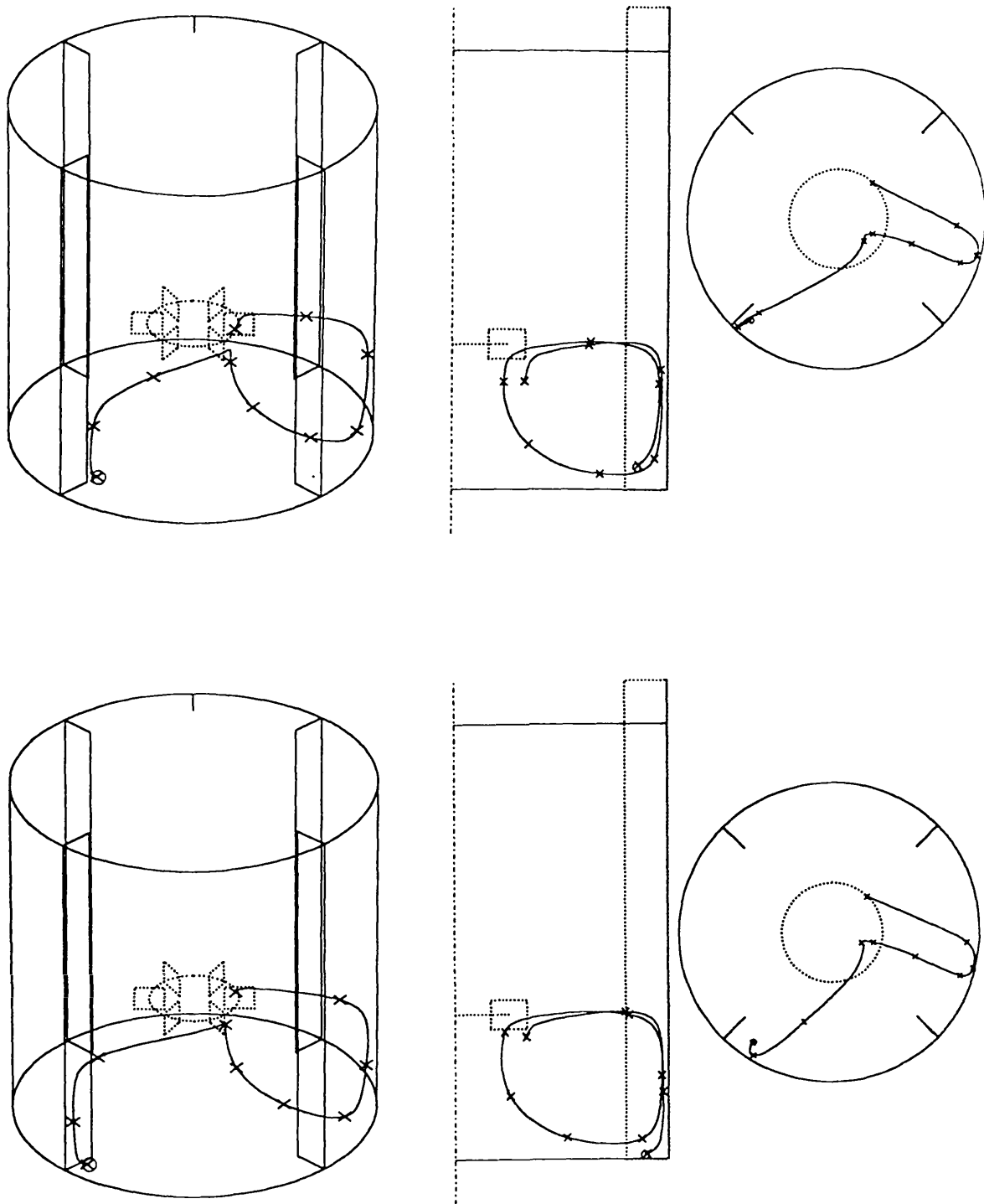


Figure 8.2.1: Projections of Flow Trajectory
 $z_0 = 7.3$ cm (top) and $z_0 = 8.3$ cm.

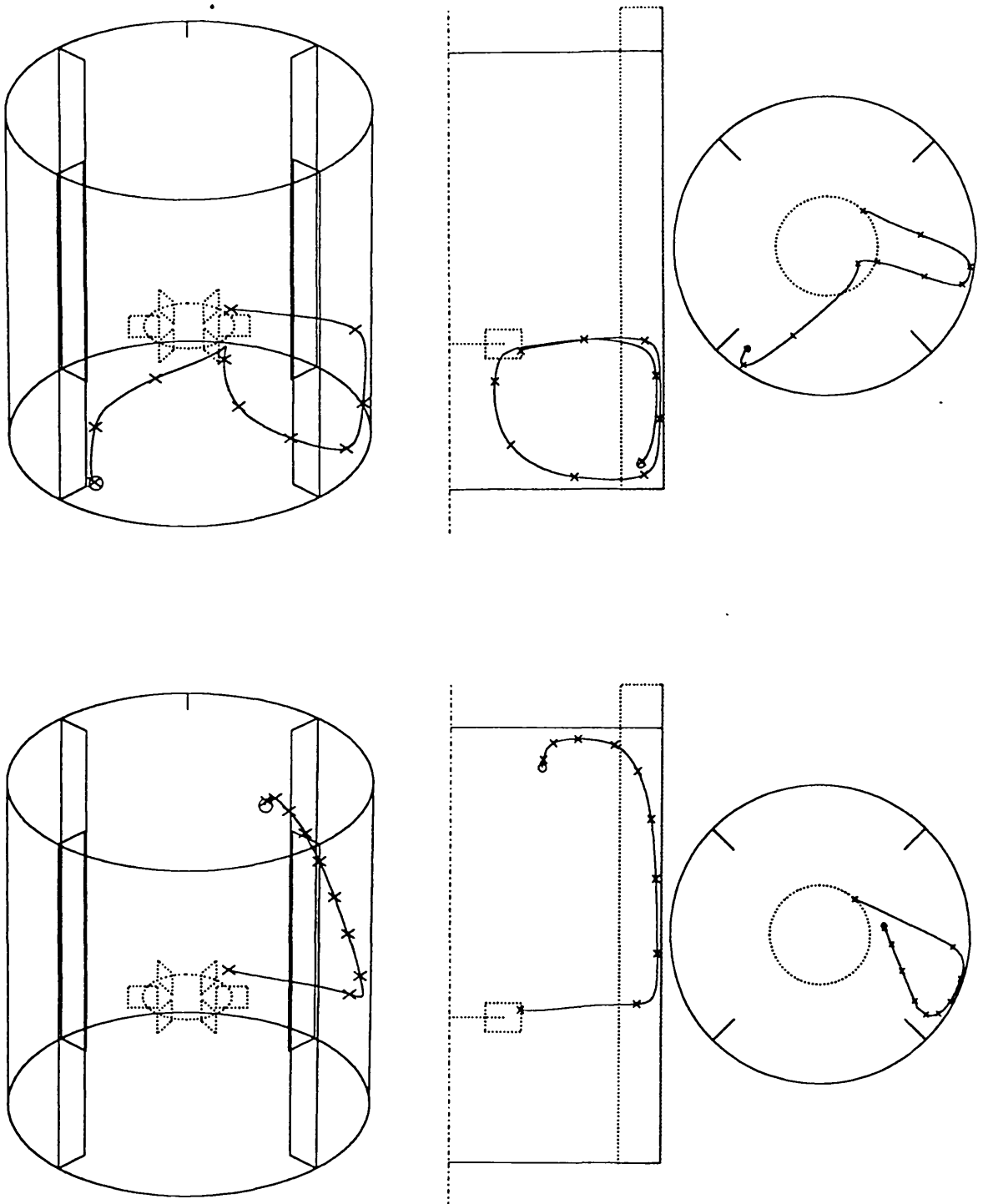


Figure 8.2.2: Projections of Flow Trajectory
 $z_0 = 9.3$ cm (top) and $z_0 = 10.3$ cm.

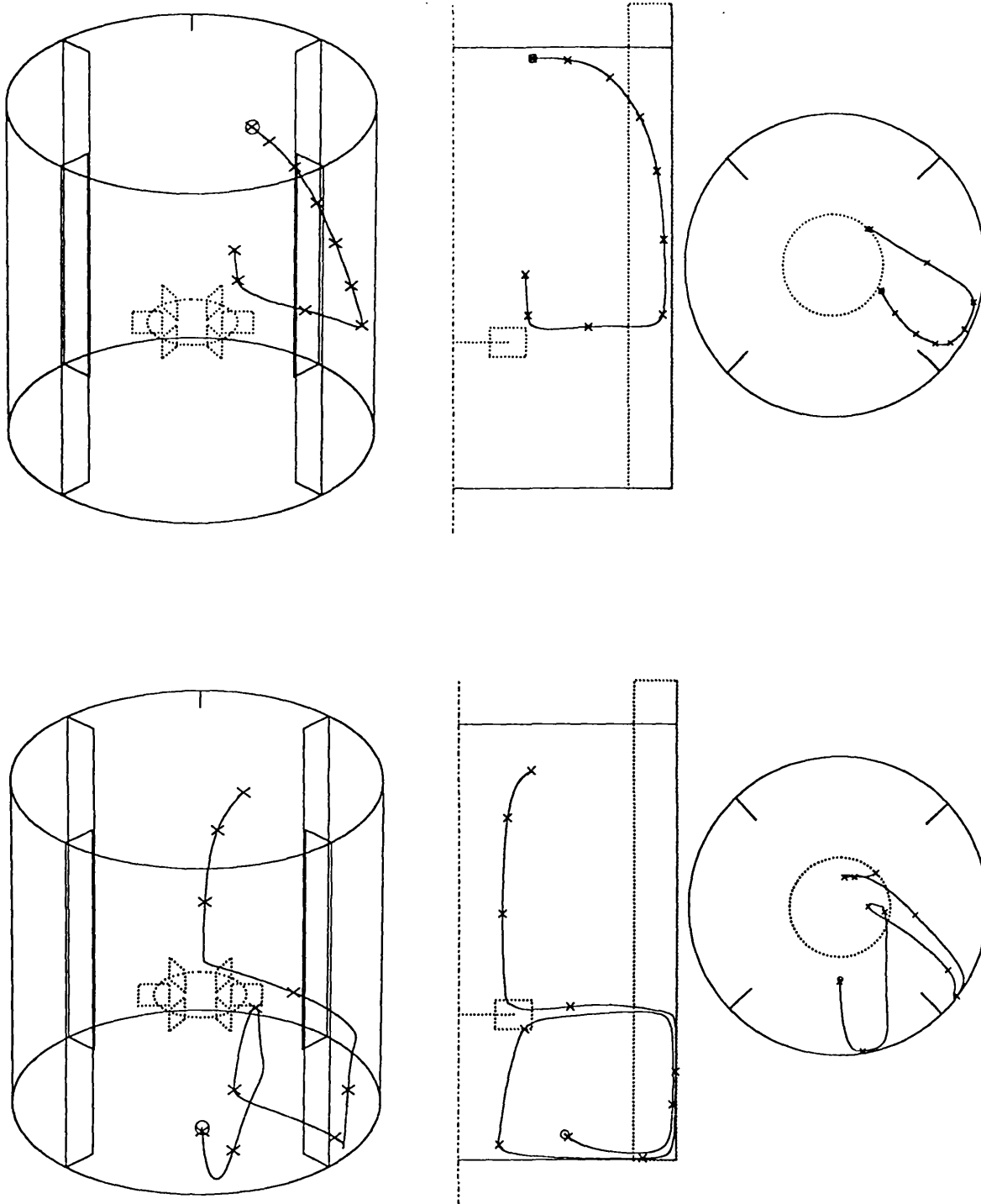


Figure 8.2.3: Projections of Flow Trajectory
 $z_0 = 14.3$ cm (top) and $z_0 = 26.3$ cm.

Appendix A.8: Derivation of the Impeller Torque Formula

This appendix derives the impeller torque formula (8.1.7). It does this by taking the vector product of the Navier-Stokes equation with position to derive the torque exerted by the impeller. The derivation is given using Cartesian coordinates and the resulting expression is then converted to cylindrical coordinates for the present purpose.

The torque due to the body forces acting on the impeller can be determined from integrating the vector product of the position and the body force over a volume containing the impeller thus

$$Tr_i = \epsilon_{ijk} \int_V x_j F_k dV. \quad (\text{A.8.1})$$

Here ϵ_{ijk} is the alternating tensor, which is defined in appendix B.8, Tr_i is a component of the torque vector, F_k is the body force exerted and V is some volume containing the impeller but no other geometrical feature that exerts a force on the fluid. The origin of the coordinate system is assumed to lie on the tank axis.

The body force F_k can be evaluated from the Navier-Stokes equation

$$\frac{\partial}{\partial x_j} (\rho u_i u_j) = \frac{\partial \sigma_{ij}}{\partial x_j} + F_i \quad (\text{A.8.2})$$

where σ_{ij} is the stress tensor.

$$\sigma_{ij} = -p\delta_{ij} + \mu_{eff} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \rho k \quad (\text{A.8.3})$$

To do this take the cross product of (A.8.2) with position to obtain

$$\epsilon_{lmj} x_m \frac{\partial}{\partial x_j} (\rho u_i u_j) = \epsilon_{lmj} x_m \frac{\partial \sigma_{ij}}{\partial x_j} + \epsilon_{lmj} x_m F_i. \quad (\text{A.8.4})$$

Advantage can be taken of the identity

$$\frac{\partial x_j}{\partial x_k} = \delta_{jk}. \quad (\text{A.8.5})$$

to rearrange (A.8.4) to

$$\epsilon_{lmj} \frac{\partial}{\partial x_j} (\rho x_m u_i u_j) - \epsilon_{lmj} \rho x_m u_i u_j \delta_{jm} = \epsilon_{lmj} \frac{\partial}{\partial x_j} (x_m \sigma_{ij}) - \epsilon_{lmj} \sigma_{ij} \delta_{jm} + \epsilon_{lmj} x_m F_i. \quad (\text{A.8.6})$$

The second term on the LHS and the second term on the RHS both involve the double contraction of a symmetric tensor with the alternating tensor which is identically zero (appendix B.8). Integration of the remainder of (A.8.6), application of the divergence theorem and rearrangement gives

$$\int_V \epsilon_{lmj} x_m F_i dV = \int_S \epsilon_{lmj} x_m (\rho u_i u_j - \sigma_{ij}) dS_j. \quad (\text{A.8.7})$$

This formula can readily be converted to cylindrical coordinates. The only component that is of interest is the axial component which gives the impeller torque formula as

$$Tr = \int_S (\rho u_r u_\theta - \sigma_{r\theta}) r dS_r. \quad (\text{8.1.7})$$

Appendix B.8: Properties of the Alternating Tensor

The alternating tensor ϵ_{ijk} is defined as the completely antisymmetric tensor of rank 3. i.e.

$$\epsilon_{ijk} = +1 \quad \text{If } ijk \text{ is an even permutation of } 123 \text{ (i.e. } ijk = 123, \quad (B.8.1) \\ 231 \text{ or } 312)$$

$$\epsilon_{ijk} = -1 \quad \text{If } ijk \text{ is an odd permutation of } 123 \text{ (i.e. } ijk = 213, \quad (B.8.2) \\ 321 \text{ or } 132)$$

$$\epsilon_{ijk} = 0 \quad \text{Otherwise} \quad (B.8.3)$$

It has the following properties.

$$\epsilon_{ijk} = -\epsilon_{ikj} = -\epsilon_{jki} = -\epsilon_{kji} \quad (B.8.4)$$

$$\epsilon_{iij} = 0 \quad (\text{No summation}) \quad (B.8.5)$$

It immediately follows from (B.8.4) that the contraction of the alternating tensor with any symmetric tensor of rank two (S_{ij}) is zero.

$$\epsilon_{ijk} S_{jk} = \frac{1}{2} (\epsilon_{ijk} S_{jk} - \epsilon_{ikj} S_{kj}) = 0 \quad (B.8.6)$$

Appendix C.8: Mean Flow Path and Path from Mean Velocity

The purpose of this appendix is to explore further the approximation that the flow path calculated from the mean velocity is the ensemble average flow path. This is done by analysing a simple picture of the local flow (figure C.8.1). The co-ordinate system is defined so that the x is in the direction of travel and y perpendicular to it.

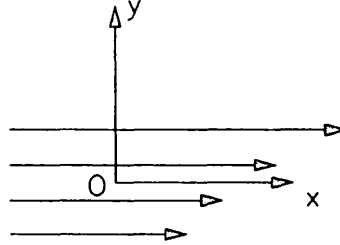


Figure C.8.1: Local Co-ordinate System for Flow about O

The mean velocity field is assumed to be independent of x and parallel to the x axis. The mean velocity, u , is a function of y .

$$u = u_0 + \alpha y + \frac{1}{2} b y^2 \quad (\text{C.8.1})$$

This can be regarded as a Taylor series expansion of the velocity about O where

$$\alpha = \frac{\partial u}{\partial y} \quad (\text{C.8.2})$$

and

$$b = \frac{\partial^2 u}{\partial y^2}. \quad (\text{C.8.3})$$

Two further assumptions are made about the velocity field. The first is that there are no perturbations in u but there are fluctuations in the velocity parallel to the y axis such that any fluid element released from O will take random y co-ordinates uniformly distributed between $\pm y_0$.

We now proceed to determine the difference between the flow paths calculated by ignoring the effects of velocity fluctuations (the mean velocity approximation) and including the velocity fluctuations. Fluid elements are released from O at time $t = 0$. The mean velocity approximation is that the ensemble average position at a later time is given by

$$x(t) = u_0 t \quad (\text{C.8.4})$$

and

$$y(t) = 0. \quad (\text{C.8.5})$$

If fluctuations are taken into account then the mean y value is zero so (C.8.5) holds. The mean velocity in the x direction, $\bar{u}$, is given by

$$\begin{aligned}\bar{u} &= \frac{1}{2y_0} \int_{-y_0}^{y_0} u(y) dy \\ &= u_0 + \frac{1}{6} b y_0^2\end{aligned}\tag{C.8.6}$$

from (C.8.1). The mean value of x at time t is therefore given by

$$x(t) = u_0 t + \frac{1}{6} b y_0^2 t\tag{C.8.7}$$

which differs from (C.8.4) by $\frac{1}{6} b y_0^2 t$. Therefore providing

$$y_0^2 \ll \frac{u_0}{b}\tag{C.8.8}$$

holds then the mean velocity approximation will be a good one.

Chapter 9: Mixing and Chemical Reaction Calculations

This chapter presents calculations performed with the MIXREACT program. Two applications are considered. The first (section 9.1) is scalar mixing in stirred tanks and the second (section 9.2) is mixing and reaction with a simple reaction scheme. Both calculations are compared with experimental data. Finally conclusions are drawn in section 9.3.

9.1 Mixing Calculations

Introduction

The purpose of this section is to investigate the performance of the computational procedure when applied to the investigation of mixing in stirred tanks. This is done by using the MIXFLO and MIXREACT programs to calculate the manner in which salt solution mixes when added to a stirred tank initially containing water. The results of these computations are compared with experimental data and the accuracy of the predictions assessed.

Description of Experiments

Revill (1987) performed a number of experiments to investigate mixing in stirred tanks. Three litres of salt solution were added to 4.39m^3 of water in a vessel whose dimensions are listed in table 9.1.1. The experiment is shown schematically in figure 9.1.1. The impeller speed was 70 rpm.

Three conductivity probes were used to measure the salt concentration at various locations in the vessel as listed in table 9.1.2. These probes were continuously logged by a micro-computer and the results normalised by the final equilibrium responses to express the salt concentrations as a percentage of the mean concentration.

The rapid addition of the salt solution was achieved in about one second by means of a vertical length of large diameter pipe terminated by a manually operated valve just above the liquid surface. The salt solution, initially in the pipe, is admitted to the tank by opening the valve at time $t = 0$. It was found that this process was reasonably reproducible.

The salt solution entering the vessel from the pipe has appreciable momentum which disturbs the flow in the region of the addition point. A plume of salt solution is formed which is believed to extend to a depth of up to 50cm (Revill, 1987). Unfortunately this plume was not characterised experimentally and no quantitative data concerning its dimensions or the salt concentrations within it are available. This causes difficulties with the formulation of appropriate initial conditions for the computations which assume that the addition takes place isokinetically (i.e. without disturbing the flow field) and instantaneously.

Description of Computations

The MIXFLO computations were performed on a mesh of 16,000 points which was the finest practical grid that could be handled on the available computer. The mesh sensitivity tests presented in chapter 7 suggest that grid independent results will not be achieved on this mesh, even with higher order differencing. The flow patterns predicted are qualitatively similar to those discussed in chapter 7.

Diameter (T_d)	1.83
Height (H_d)	1.67
Clearance (Z)	0.41
Impeller Diameter (D)	0.90
Disc Diameter (D_d)	0.65
Blade Length (L_s)	0.23
Blade Height (w)	0.18
Baffle Width (B)	0.18

Table 9.1.1: Dimensions of Stirred Tank for Mixing Experiments
The dimensions refer to figure 6.5.1.

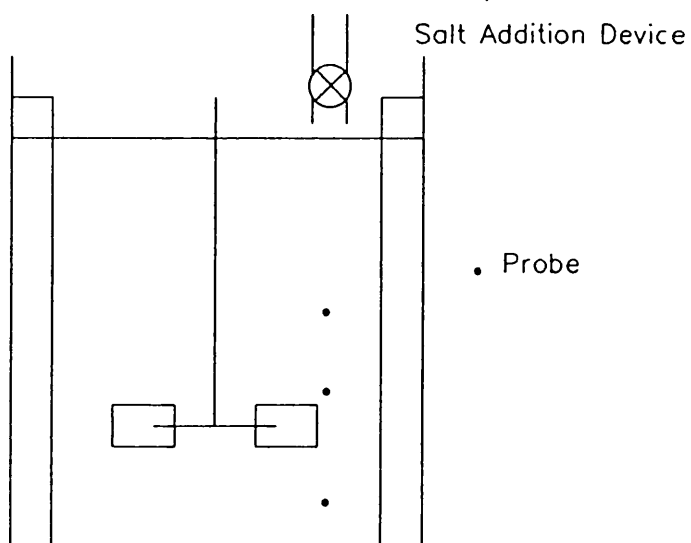


Figure 9.1.1: Schematic of Mixing and Reaction Experiments

	r	θ	z
Addition Point	0.46	30°	1.67
Probe 1	0.65	90°	1.09
Probe 2	0.65	90°	0.55
Probe 3	0.65	90°	0.15

Table 9.1.2: Addition Point and Probe Locations for Mixing Experiment
The 0° plane is midway between two baffles and the impeller rotates in the direction of increasing angle.

The MIXREACT computation of the mixing process involved the solution of a single stream fraction equation. In the absence of any experimental data concerning the initial salt plume, the initial conditions applied assume that all the salt is instantaneously and isokinetically added at the liquid surface at time $t = 0$. This is done by taking the initial stream fraction to be unity in the computational cell nearest to the addition point and zero elsewhere. The effect of the initial salt plume is that, at early times, the salt is distributed over a larger volume of the tank than predicted computationally. Consequently, the subsequent development of the concentration will differ from that observed experimentally until such time as the vessel is completely mixed. The initial conditions are therefore a significant source of error in the calculation.

The computational 'probe' results were obtained by monitoring the predicted concentration of the salt in the computational cell containing the probe as a function of time. These results are then normalised by the average salt concentration and expressed as a percentage in an identical manner to the experimental results.

The turbulent Schmidt number was taken as 0.7 and the salt diffusivity as $1.67 \times 10^{-9} \text{ m}^2/\text{s}$ (Lynch, 1985). Turbulent 'diffusion' dominates molecular diffusion so the computation is insensitive to this latter quantity. The Courant number used was 2.0 for reasons described in section 6.6. The MIXREACT calculation took approximately ten hours to complete on a SUN 4/110 Workstation.

Computational and Experimental Results

Figures 9.1.3 to 9.1.5 show the computed responses for probes at heights of 1.09, 0.55 and 0.15 metres respectively. A number of observations can be made concerning these predictions.

- (i) The largest response occurs in the 0° plane reading for the upper probe. This is also the sharpest and earliest peak seen in any of the predicted probe responses. This is caused by the counter-circulation near the surface moving the tracer fluid back towards the 0° plane where the probe is cited and being drawn down past the probe towards the impeller (figure 9.1.6), hence the early and high response.
- (ii) The tracer in the upper recirculation zone continues to be mixed into the bulk fluid. This can be seen in the high concentration in the probe at the axial location of 0.55m and angle of 0°.

- (iii) The initial peak in the lower recirculation zone is weak or non-existent. This is caused by majority of the tracer discharged from the impeller flowing into the upper recirculation loop.
- (iv) The probe responses at different angular positions differ markedly from each other. In particular the response for the highest 90° probe is quite different from the neighbouring probes at 45° and 135°.
- (v) A second weak peak is visible in some of the probe responses about four seconds after the initial peak. This is due to the tracer fluid circulating through the impeller for a second time.

Figure 9.1.6 presents the flow path calculated from the injection point. As already described, the counter-circulation draws the fluid back towards the 0° plane and down into the impeller. It is discharged back into the upper recirculation loop and the recirculation time is approximately four seconds. This is in accord with the weak initial response of the lowest probes and the time interval between peaks in the upper responses.

Figure 9.1.7 shows the comparison between the computed responses and the experimental results obtained by Revill (1987) for the probes listed in table 9.1.2. The experimental response plotted is the result of using the SMOOT algorithm (Dierckx, 1975) to smooth the average response from six experiments. These actual observations are shown in figure 9.1.8. They contain a large amount of noise and show large differences between individual observations. The measuring volume of the probes is less than one cubic centimetre so some of these differences can be attributed to turbulent concentration fluctuations. Fluctuations are also caused by electrical noise in the measuring system.

The agreement between the experimental and predicted results is poor, even allowing for the high levels of noise in the experimental results. In particular the initial peaks observed in the upper recirculation zone are not predicted. The predicted peak in the lower recirculation zone is too low and occurs too late.

It is interesting to compare the predictions for the probes at 45° with the experimental results at 90° (figure 9.1.9). These show better qualitative agreement than the predictions at 90° shown in figure 9.1.7.

A number of possible sources of differences between the experimental and predicted results can be identified.

- (i) As already noted, differences between the assumed and actual addition mechanisms are a source of error in the calculation. This could be investigated by varying the initial stream fraction field used in MIXREACT. Further experiments are also required to provide more quantitative data on the initial concentration field.
- (ii) Errors in the flow field will cause differences between the predicted and experimental mixing of passive scalars. Sources of errors in the flow fields were discussed in chapter 7. Further, the stirred tank geometry used in these mixing experiments differs from the vessel used for the LDA work presented in chapter 7 and it is possible that MIXFLO does not predict the flow in this vessel as well as that used in chapter 7.
- (iii) The application of $k - \epsilon$ modelling to account for the effect of the flow structures associated with the impeller (section 6.5) may be inappropriate. Errors in the flow field will cause the salt plume to take a different path from those in the experiments. Therefore probes that are in the experimental path but not in the computational path will show very

different responses. This may be the reason that the comparison with predicted results at 90° are poor (figure 9.1.7) while there is qualitative agreement with the predicted results at 45° (figure 9.1.9).

- (iv) The calculation of the $\overline{u'c'}$ correlation by the gradient diffusion assumption (equation (2.3.29)) may be poor. This could be due to errors in the values of k , ϵ or σ_c or simply that equation (2.3.29) is a poor approximation. This term affects the way the plume disperses and the maximum concentration in the plume. The importance of this term could be investigated by comparing its magnitude with that of the other terms in the stream fraction equation.
- (v) The grid and time step employed may not be sufficient to numerically resolve the mixing process. The tests presented in chapter 7 indicate that grid independent flow fields are not achieved with 16,000 point grids. The effect of this has been described in point (ii) above. Scalar transport calculations presented in chapter 5 show appreciable erosion of the maximum ϕ value takes place even in simple flows on relatively fine meshes. The meshes used in the mixing calculation are coarser than those used in chapter 5 and therefore the scalar mixing process is not being adequately resolved. This could be investigated by employing finer grids and lower Courant numbers when more computational resources become available.

The experimental results consist only of normalised salt concentration as a function of time in three positions in the flow field which is insufficient to characterise the three-dimensional concentration field. The development of improved experimental techniques providing details of the concentration as a function of time at more locations would guide the development of the computational method.

Another improvement would be to replace the current batch mixer with a continuous vessel. Here fluid continuously flows into the stirred tank by means of a port in a wall and leaves the tank at an equal rate through another port elsewhere. The mixing experiment would be performed by changing the fluid entering the tank from water to a salt solution and monitoring the salt concentration at various locations as at present. The flow in the vessel, including the jet from the inlet port, could be calculated by MIXFLO. The MIXREACT calculation would proceed by assigning a stream fraction value of unity to the salt solution entering the vessel and zero for the water already present. The initial conditions would be well characterised and the difficulty encountered in this study avoided.

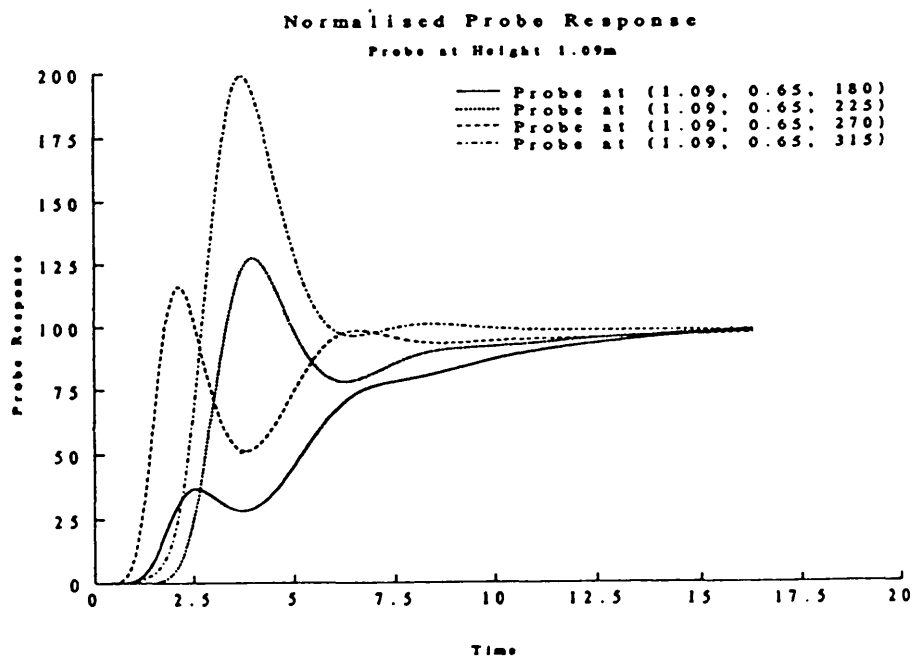
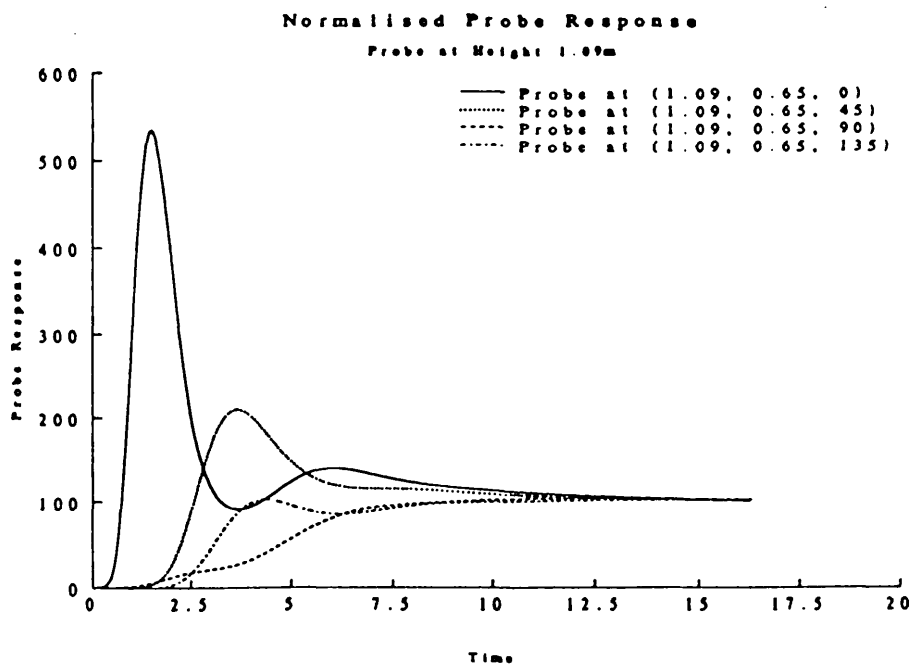


Figure 9.13: Computed Probe Responses

The probes are placed at a height of 1.09m above the bottom of the tank and at various angular positions.

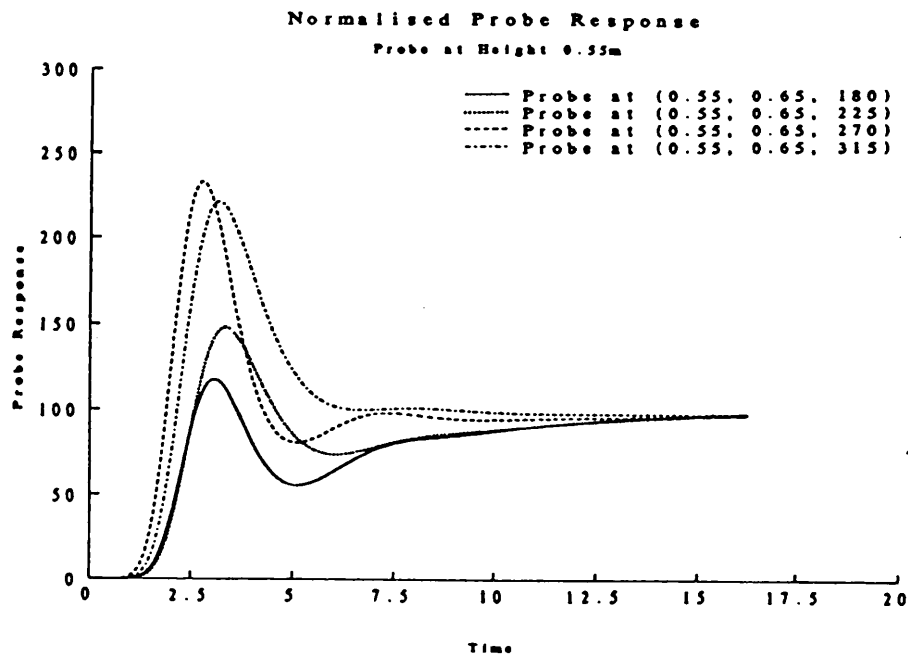
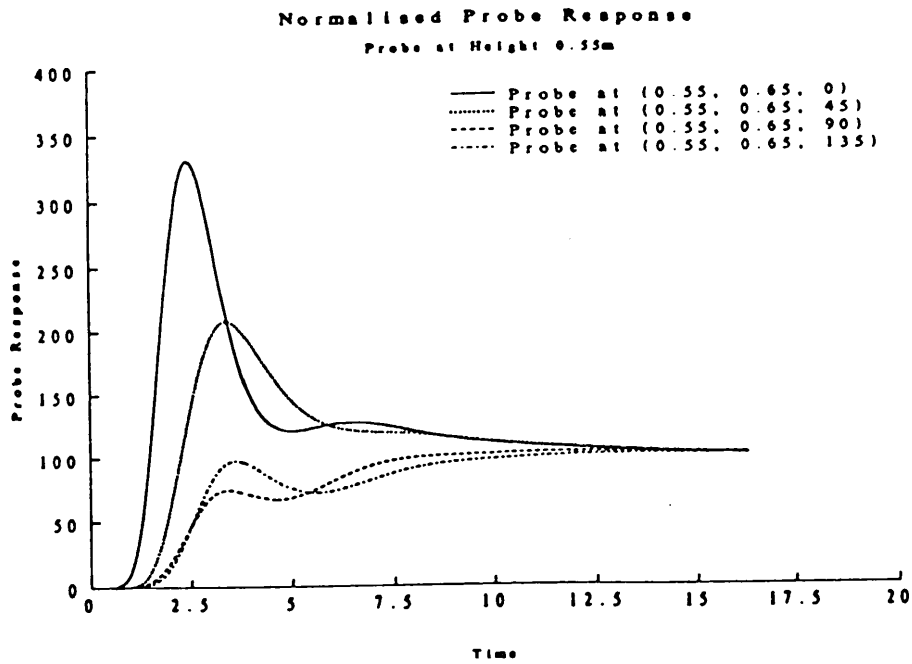


Figure 9.1.4: Computed Probe Responses
The probes are placed at a height of 0.55m above the bottom of the tank and at various angular positions.

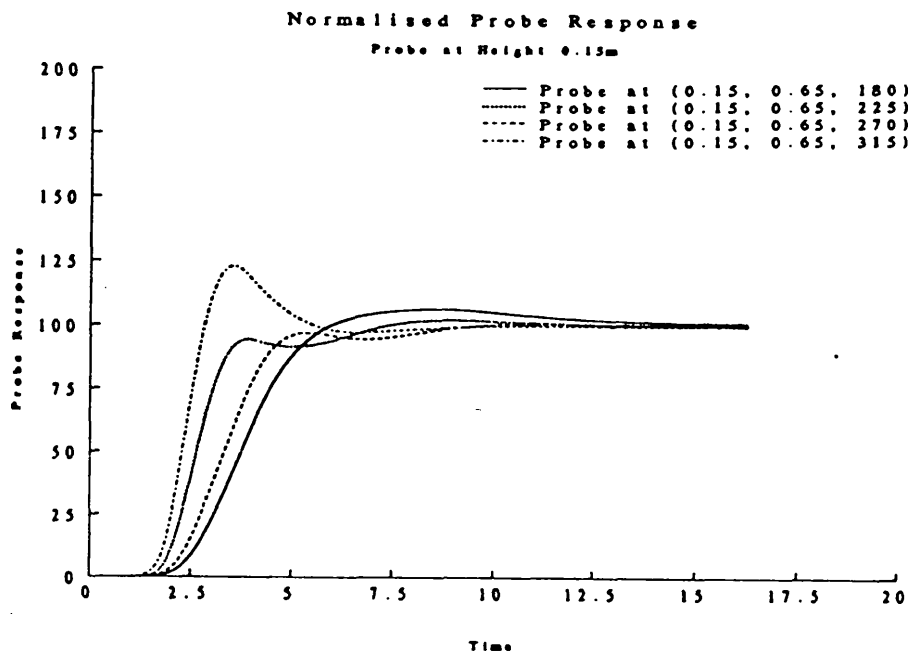
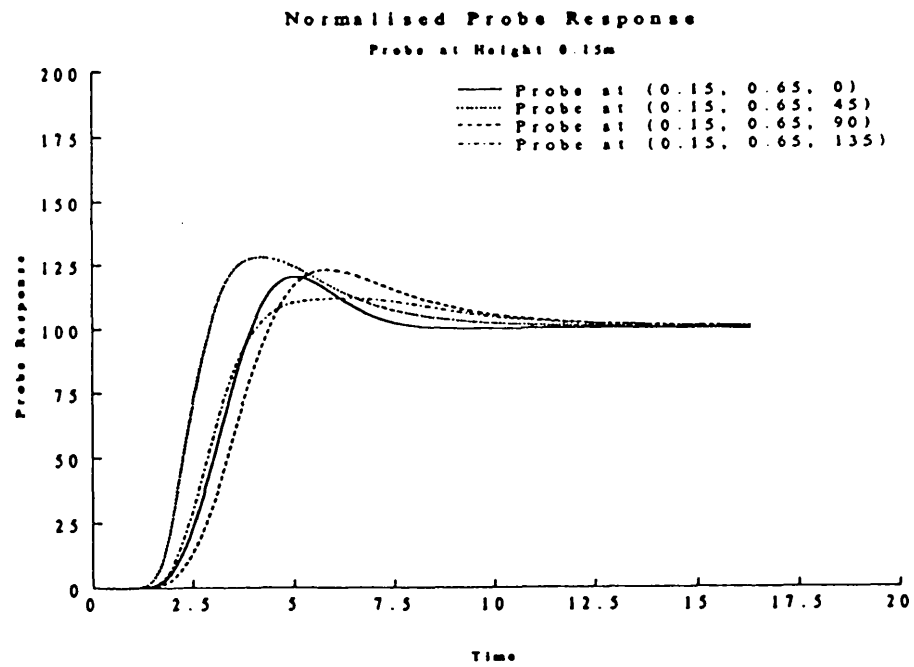


Figure 9.1.5: Computed Probe Responses

The probes are placed at a height of 0.15m above the bottom of the tank and at various angular positions.

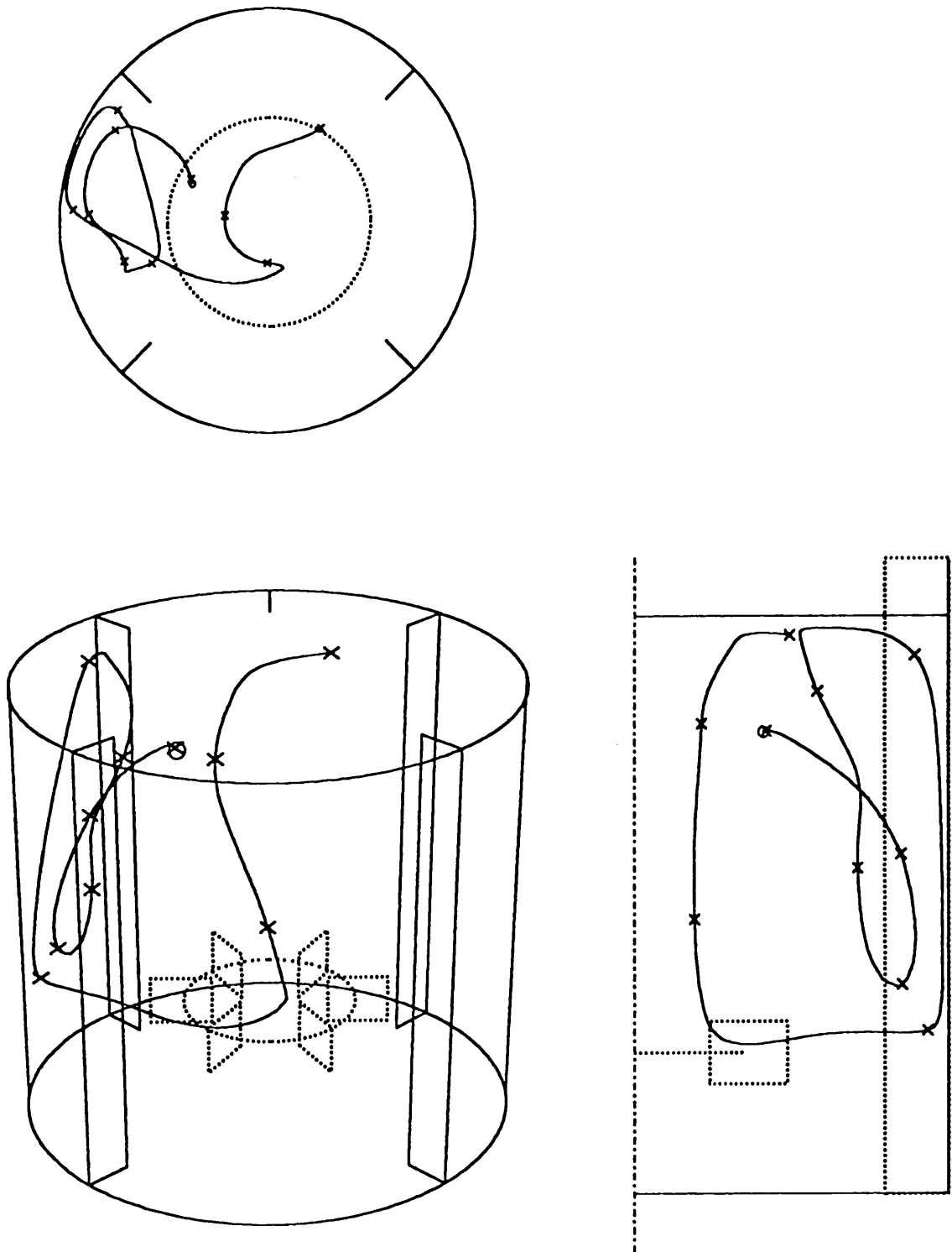
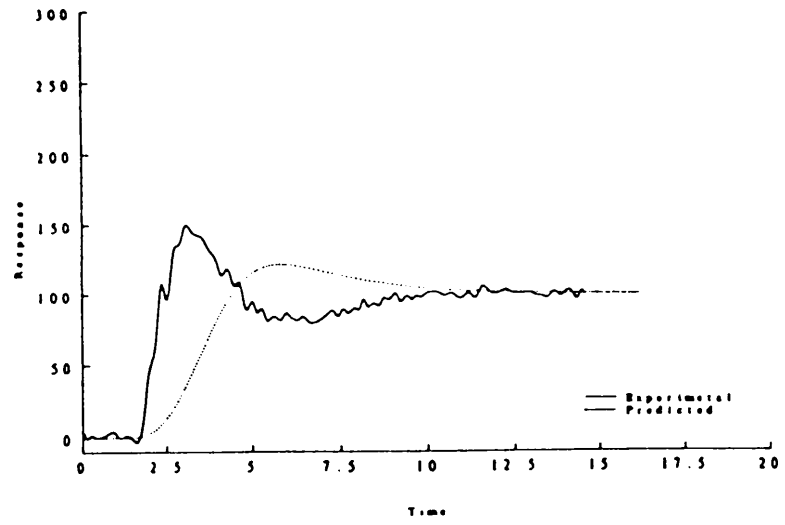
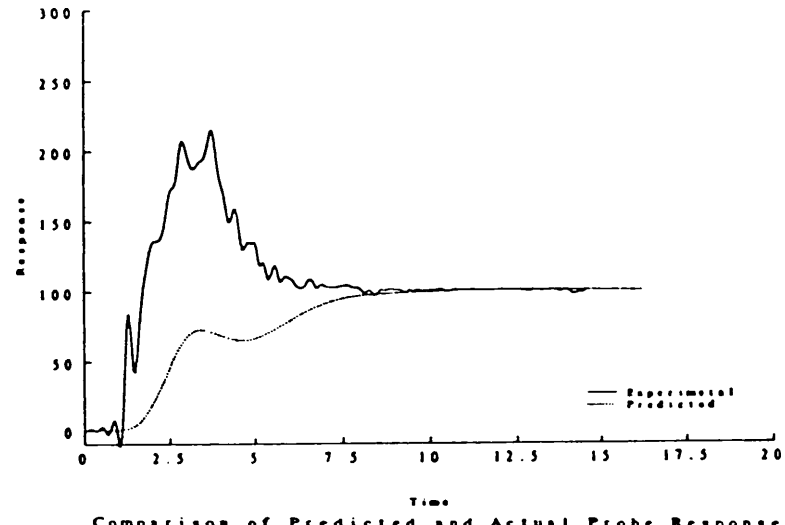


Figure 9.1.6: Flow Trajectory from Tracer Fluid Addition Point
Markers are placed at a time interval of one second.

Comparison of Predicted and Actual Probe Response
Probe at (0.15, 0.45, 90)



Comparison of Predicted and Actual Probe Response
Probe at (0.55, 0.45, 90)



Comparison of Predicted and Actual Probe Response
Probe at (1.09, 0.45, 90)

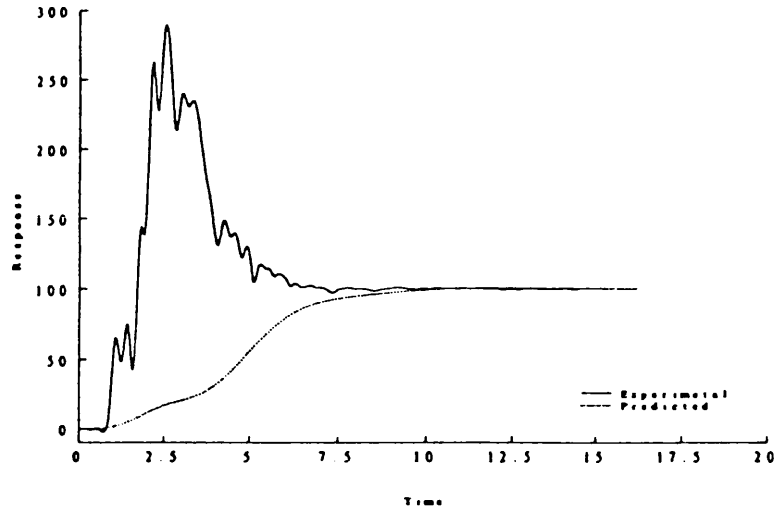
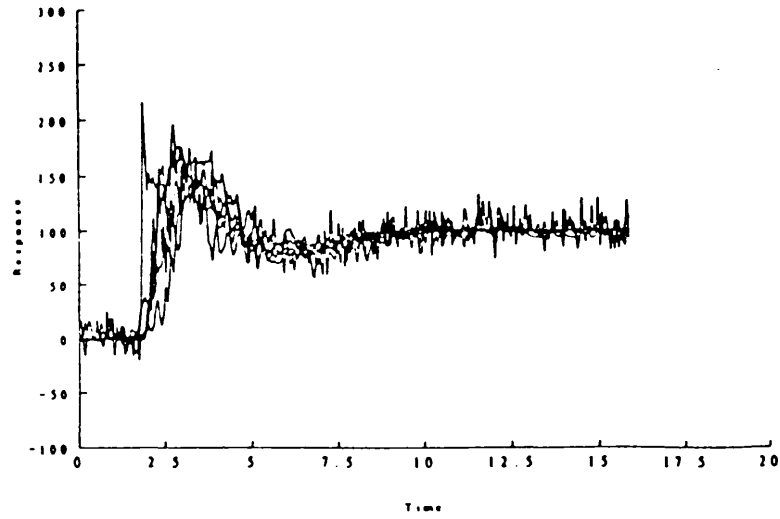


Figure 9.1.7: Comparison of Computed and Experimental Probe Responses
Computed probe results at 90° and height 0.15m (top), 0.55m (middle) and 1.09m (bottom).

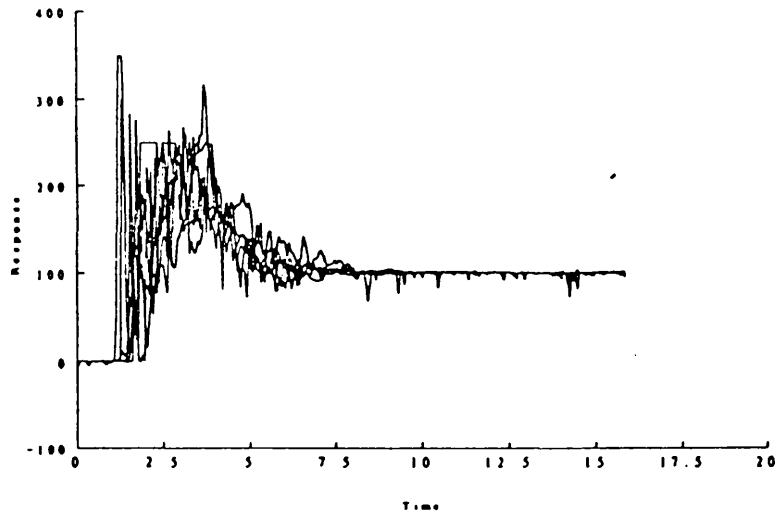
Results from Individual Experiments

Probe at 10 15, 0 45, 90



Results from Individual Experiments

Probe at 10 35, 0 65, 90



Results from Individual Experiments

Probe at 11 00, 0 45, 90

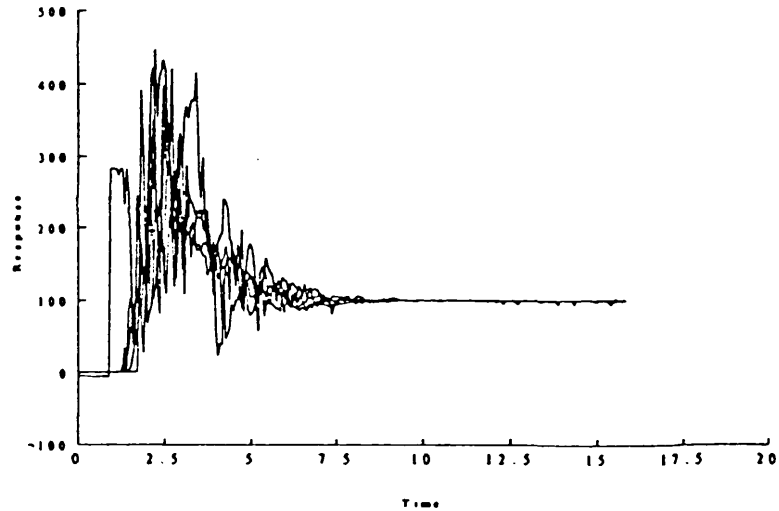
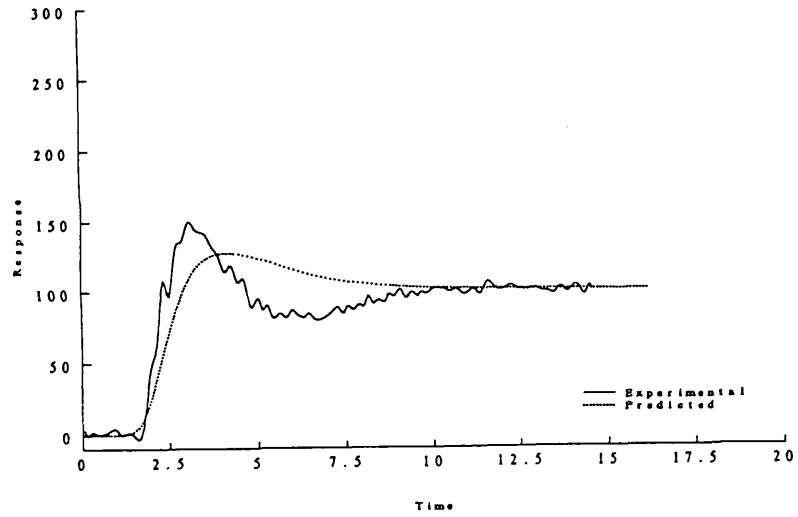
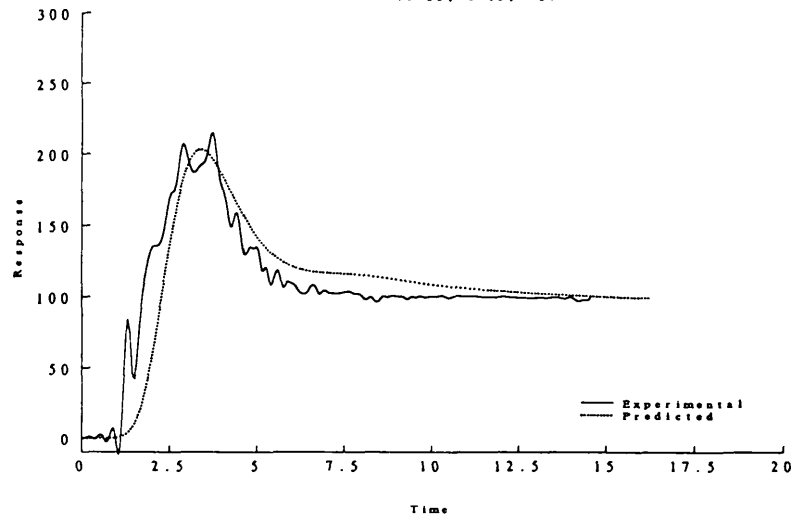


Figure 9.1.8: Actual Probe Responses

Comparison of Predicted and Actual Probe Response
Probe at (0.15, 0.65, 45)



Comparison of Predicted and Actual Probe Response
Probe at (0.55, 0.65, 45)



Comparison of Predicted and Actual Probe Response
Probe at (1.09, 0.65, 45)

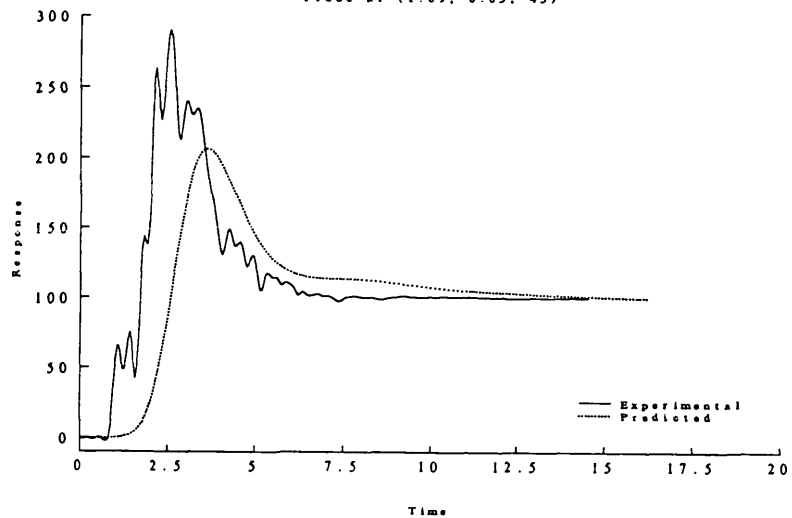


Figure 9.1.9: Comparison of Computed and Experimental Probe Responses
Computed probe results at 45° and height 0.15m (top), 0.55m (middle) and 1.09m (bottom).

9.2 Reaction Calculations

Introduction

The objective of this section is to assess the performance of the computational methods described in this thesis to the prediction of the yield of a well-characterised mixing-sensitive reaction. The reaction selected is that described by Bourne *et al* (1981) and used by Middleton *et al* (1986) to investigate mixing in stirred tanks. The reaction scheme is



Here *A* is 1-naphthol, *B* is diasotised sulphanilic acid, *R* is 4-(4'-sulphophenylazo)-1-naphthol and *S* is 2,4-bis-(4'-sulphophenylazo)-1-naphthol. The pH was buffered at 10 and the reactions performed at 25°C. Under these conditions the instantaneous rate of the first reaction is

$$\Omega_1 = k_1 [A][B] \quad (9.2.2)$$

where k_1 is 7.3×10^3 litres/mole/s. The second reaction is also second order with the reaction rate given by

$$\Omega_2 = k_2 [B][R] \quad (9.2.3)$$

where k_2 is 3.5 litres/mole/s (Bourne *et al*, 1981). (The kinetic constants reported earlier by Middleton *et al* (1986) are incorrect (Middleton, 1990)). Both (9.2.2) and (9.2.3) are of the general type of kinetics given by (2.5.3).

The amount of *S* produced in reaction (9.2.1) is very sensitive to the mixing of *A* and *B* for the following reason. If the mixing is good then *B* will preferentially react with *A* due to the fast kinetics of this reaction compared to the reaction with *R*. Therefore the yield of *S* will be low. However if the mixing is poor then there will be regions where all the available *A* will have been used to produce *R* which will react with the remaining *B* to produce a higher yield of *S*.

Description of Experiments

The experiments consisted of rapidly adding 1.500×10^{-3} moles of *B* to 1.581×10^{-3} moles of *A* in aqueous solution in the stirred tank. The dimensions of the vessel are given in table 9.2.1. *B* is added at the liquid surface midway between baffles and midway between the axis and wall of the tank by means of placing a vertical pipe containing the solution of *B* above the addition point (figure 9.1.1). The bottom of the pipe was terminated by a valve which was just above the liquid surface. The addition was achieved by opening the valve and rapidly discharging the solution into the stirred tank. This presents the same difficulty with the initial conditions as that in the previous section.

When all the *B* was consumed and the reactions ceased the concentrations of *R* and *S* were measured and the yield X_S defined as

$$X_S = \frac{2[S]}{[R] + 2[S]}. \quad (9.2.2)$$

was calculated. This is the fraction of *B* that was used to form *S*. X_S values are reported for impeller speeds of 50, 100 and 600 rpm in table 9.2.2. The uncertainty in the experimental results is about 10%. If the mixing was perfect then a X_S would be 0.006 while if the mixing was very poor then it would approach unity. It can be seen that the yield decreases with increasing impeller speed indicating that faster mixing does reduce the amount of *S* produced.

Diameter (T_d)	0.30
Height (H_d)	0.41
Clearance (Z)	0.15
Impeller Diameter (D)	0.15
Disc Diameter (D_d)	0.11
Blade Length (L_b)	0.04
Blade Height (w)	0.03
Baffle Width (B)	0.03

Table 9.2.1: Dimensions of Stirred Tank for Reaction Experiments
The dimensions refer to figure 6.5.1.

Description of Computations

The first stage of the computations consisted of calculating the flow patterns in the vessels at impeller speeds of 50, 100 and 600 rpm. This was done using a MIXFLO grid of approximately 16,000 cells. The resulting flow patterns are essentially similar to those presented in Chapter 7. The second stage was to use MIXREACT to calculate the yield of the reactions at the various speeds. These computations required the solution of a stream fraction equation and two reaction extent equations as described in section 2.4. The stoichiometric matrix n is given by equation (2.4.37) and the stream composition matrix c , by (2.4.38).

The MIXREACT computations modelled the addition of B by placing all the B in the computational cell nearest to the point of addition. The rest of the vessel was filled with the solution of A. As already mentioned, these initial conditions suffer from the same difficulty as those employed in the previous section. The concentrations and kinetic rate constants are such that the first reaction can be regarded as being rapid and the second slow. Therefore the micromixing model (equation (2.5.15)) was applied to the kinetics of the first reaction and the laminar model used for the second. Middleton *et al* (1986) argue that, under the experimental conditions being considered, the reaction yield is determined by the macro-mixing and not the micromixing. Therefore the experiment is not suitable for directly testing the micromixing model.

The Courant number employed in the calculations was 2.0 and typically about 300 time steps were required for the reaction to complete. The computations took around 24 hours of CPU time on a SUN 4-110 Workstation.

Results

The calculated yields and experimental yields from Middleton *et al* (1986) are given in table 9.2.2. The table also contains reaction yields computed by Middleton *et al* (1986) using FVM with coarser meshes (typically around 6,000 points for the flow calculation) and upwind differencing. The laminar model was used for the kinetics of the fast reaction and the initial conditions in MIXREACT were similar to those used in the current study.

Impeller Speed rpm	Experimental Yield	Predicted Yield (This Study)	Predicted Yield (Middleton <i>et al</i> , 1986)
50	0.41	0.47	0.37
100	0.24	0.36	0.27
600	0.09	0.11	0.10

Table 9.2.2: Predicted and Experimental Values of X_s

The present predictions produced yields that are consistently higher than the observed yields suggesting that the rate of mixing is under-predicted. The following comments can also be made.

- (i) The factors identified in the previous section concerning the calculation scalar mixing also apply to the mixing of A and B in this case thereby affecting the yield of the mixing-sensitive reaction.
- (ii) The experimental procedure used to add the B solution will form a plume of mixed A and B in the tank rather than the small localised volume of B assumed in the initial conditions to the calculation. The effect of this plume is to mix A and B more rapidly than would be predicted computationally and hence the calculated yield would be too high. This is consistent with the results presented in table 9.2.2.
- (iii) The model used for the micromixing kinetics of the fast reaction may cause the rate of this reaction to be under-predicted. The second reaction would therefore compete more effectively for B increasing the yield of S, again consistent with table 9.2.2. However, as noted previously, it is not expected that the reaction yield is sensitive to the micromixing model and therefore this is not likely to be the dominant source of error.

It is also interesting to note that the predictions from this study are consistently higher than those of Middleton *et al* (1986) which are in closer agreement with the experimental values. However, the use of UD and coarse grids in their calculation would result in significant amounts of numerical diffusion. This would have the effect of increasing the predicted mixing rate and therefore lowering the yield of S. Additionally, omission of the effect of turbulence on the kinetics of the fast reaction is likely to cause the predicted reaction rate to be too high which will also have the effect of decreasing the yield of S. These two factors counterbalance the effect of the uncertainties in the initial conditions, which tend to increase the predicted yield, and hence the results are closer to the experimental data.

Middleton *et al* (1986) also calculated the reaction yield on various scales of vessel from 30 litres to 900 litres. They found that conventional criteria for scaling stirred tanks such as agitator tip speed, power per unit volume and mixing time failed to account for the variation of reaction yield with scale. Greater success was found using the computational method. Therefore calculations of the type presented in this section may be of use to process engineers due to the absence of other more reliable methods of estimating reaction yields.

9.3 Conclusions

MIXREACT calculations for scalar mixing and reaction yield have been presented and compared with experimental data in this chapter. It has been shown how MIXREACT can be used in conjunction with the calculations of fluid trajectories to investigate mixing processes in stirred vessels. However, comparisons with the available experimental data are not particularly favourable. A number of possible causes for the discrepancy between the predicted and experimental results have been identified including uncertainties in the initial conditions, errors in the flow field and insufficient grid refinement. The failure to properly account for the initial plume of tracer formed being the most serious of these. Clearly, if the initial conditions are not appropriate then the subsequent development of the concentration field will not agree with that observed in experiments. There is a need to either quantify the initial conditions experimentally or preferably to design new experiments with more easily characterised initial conditions. This will enable a more complete assessment of the performance of MIXREACT to be made and will provide guidance for future development work.

The calculated reaction yields were found to be higher than those measured in experiments. However, the expected variation in reaction yield with impeller speed was observed. The calculation of the mixing of the two reagents suffers from similar problems to the scalar mixing above and the over-prediction of the reaction yield can in part be attributed to uncertainty in the initial conditions. Again, improvements in the experimental method are desirable to overcome this problem. Further work is also required to test the micromixing model with a reaction whose yield is sensitive to micromixing.

Chapter 10: Conclusions

10.1 Summary of Achievements and Findings

An existing methodology for the prediction of turbulent flows in stirred tanks has been improved by the development of a new discretisation procedure and extended to handle mixing and chemical reactions.

A new mathematical framework for the equations governing mixing and reaction was developed based on the concepts of stream fraction and reaction extent. This was designed to be computationally efficient while making the implementation of different chemical equations easy. The influence of turbulence on the kinetic rates was accounted for by using a simple model based on a micromixing time scale.

The finite volume discretisation method was introduced and the discretisation of source, accumulation and diffusion terms given. Criteria for the selection of discretisation methods for the convection terms were developed and used to review the existing literature. A satisfactory scheme was not found and it was concluded that the most promising area for the development of a new scheme was in the field of non-linear discretisation techniques.

An existing non-linear finite difference scheme provided the basis for the development of a new non-linear finite volume discretisation practice, which was called Self Filtered Central Differencing (SFCD). Efficient algorithms enabling SFCD to be easily incorporated into existing programs were also devised. SFCD was shown to satisfy the criteria for the selection of the convection term discretisation. Computational scalar transport tests were satisfactory whilst existing schemes were found seriously wanting in one or more respects.

A computational procedure for the solution of the equations governing the flow and mixing processes in a stirred tank was described. The algorithm for the solution of the unsteady mixing and reaction equations ensures that chemical species will be strictly conserved during the calculation.

Extracts from Collins (1990) detailed computations of the flow pattern in stirred tanks using both the new discretisation procedure and the previous practice of upwind differencing were shown. These results further demonstrate the clear superiority of SFCD over upwind differencing.

Comparisons of the predicted results against experimental data show that the axial and radial velocities can be accurately predicted but the circumferential less so. The turbulent kinetic energy comparisons are poor. This discrepancy is believed to be due to defects in the modelling and/or the impeller flow and not insufficient grid refinement.

Two methods for exploiting the results of fluid flow computations were developed. The first is the calculation of the power dissipated by the impeller in a stirred tank which is based on calculating the torque on the impeller shaft. The results are consistent with experimental data. The second problem is the calculation of the mean trajectory of an infinitesimal fluid element. An algorithm consistent with the continuity equation has been derived and computations of trajectories in a stirred vessel clearly demonstrate the complex three-dimensional nature of the flow. This technique can be used to aid the visualisation of the flow in a stirred tank and to provide a means of estimating circulation times.

The final phase of the study concerned the prediction of mixing and reaction in stirred tanks. Calculations of the blending of a salt solution into water were performed and the predicted salt concentration at certain points compared with experimental data. The agreement with the experimental data

was generally poor. The reason for this was attributed to uncertainties in the initial conditions, modelling of the impeller and defects in the computed flow field. However, it was shown that computations of this type can be used to assist in the understanding of mixing processes in stirred tanks.

Computations of the yield of a mixing sensitive chemical reaction were presented and compared with experimental data. The predicted yields were generally higher than the experimental yields but they obeyed the correct trend as the impeller speed was varied. The overprediction of the reaction yield is consistent with the approximations made in the initial conditions. Additionally, errors in the calculation of the mixing of the two species will contribute to the discrepancy between predicted and experimental yields. However, in view of the difficulty of correlating reaction yields with parameters such as impeller speed and vessel size, computations of this type may be of considerable use despite their modest accuracy.

Overall, it is concluded that the computational technique used for the calculation of flow in stirred tanks has been improved by the new discretisation technique. The procedure can be used for both quantitative and qualitative investigation of these flows. The extension to the calculation of mixing and reactions in stirred vessels is less successful. Nevertheless, computations of this type can be used to develop an understanding of mixing processes in a stirred tank.

10.2 Suggestions for Future Work

The methodology presented in this work would benefit from further work to improve its accuracy and economy in three main areas. These are the numerical algorithm, the impeller model and the modelling used in the mixing and reaction calculations. Each of these areas is discussed below.

Numerical Algorithm

Computations with the MIXFLO program often require a large number of iterations, particularly on finer grids. The effect of this is to make detailed calculations very expensive compared to those on coarser grids. It also effectively limits the size of the mesh that can be used. An effective resolution to this problem would significantly improve the efficiency of the technique.

The accuracy of the computational method depends on both the number and position of the grid points. This is particularly important in regions where high shear is occurring. Currently, the grid is specified at the beginning of the calculation and is kept fixed throughout it. It would be more satisfactory to move the grid points during the calculation so that they are placed in regions where higher resolution is required. The R terms of the SFCD scheme provide a possible way of doing this. It can be seen from the definition of R (equations (4.1.5) and (4.1.6)) that if there is a rapid change in the gradient between successive pairs of grid nodes that R will take either low or high positive values. If the change in gradient is small then the value will be close to unity and the value will be negative if there is a turning point. Therefore it is possible to envisage a procedure in which grid points are moved away from regions where R is close to unity and towards those places where it is positive but either $R \ll 1$ or $R \gg 1$ to improve the resolution in these areas.

The third area that could be improved is the discretisation technique used for the accumulation term. The practice adopted herein is only first order accurate in time. A second or higher order scheme would be more accurate but care is required to avoid introducing a Courant number restriction on the time step (section 3.5).

Impeller Modelling

The impeller model is a key aspect of the technique presented herein because it drives the whole flow. The current practice of averaging the effects of the impeller over many revolutions to obtain characteristic ξ values (section 6.5) is not entirely satisfactory. It is likely that one of the principle causes of inaccuracy in the flow calculation is due to the loss of information due to this process. Further, the current model is only verified for the vessel configuration presented in chapter 7. The impeller model described herein may not perform as well in vessels with different clearances, impeller diameters or reduced baffling. The third disadvantage is that the current modelling is only applicable to Rushton turbine impellers but a wide variety of other types are used in industry. It is therefore concluded that a more satisfactory approach would be to actually calculate the flow around the impeller blades as part of the computational procedure. This would add to the cost of the procedure but the benefits could be very considerable.

Mixing and Reaction Modelling

There is scope for considerable improvement to this aspect of the computations. Improvements in the impeller modelling resulting in more accurate flow predictions will undoubtedly be of benefit to this type of calculation, but other aspects also need attention. The most important area to tackle is the initial conditions used at the beginning of the calculation. A satisfactory resolution to this issue will require improvements to the experimental techniques to provide quantitative measurements that can be used to formulate the initial conditions.

A second area that requires further attention is the simplistic model of micromixing adopted for this study. A major weakness is that it considers the mixing of only two chemical species undergoing a single reaction. In many situations two or more competitive and/or sequential reactions will be occurring with the consequence that the micromixing process is considerably more complex than the model suggests. Problems of this type have been addressed in the modelling of combustion processes (e.g. Jones and Whitelaw, 1982) and it may be fruitful to consider extending practices employed in this field to liquid phase reactions.

References

- Agarwal, R. K. (1981), 'A Third-Order-Accurate Upwind Scheme for Navier-Stokes Solutions at High Reynolds Numbers', AIAA 19th Aerospace Sciences Meeting, St. Louis, Missouri. AIAA-81-0112.
- Aggarwal, S. K. (1987), 'Numerical Study of Convection-Diffusion-Reaction Equations for large Damkohler and Cell Reynolds Numbers', *Num Heat Transfer* 11 pp143-164.
- Allen, D. N. and Southwell, R. V. (1955), 'Relaxation Methods Applied to Determine the Motion in Two Dimensions of a Viscous Fluid Past a Fixed Cylinder', *Q J Appl Math* 8 pp129-145.
- Arfken, G (1970), Mathematical Methods for Physicists, 2nd Edition, Academic Press.
- Atias, M., Wolfshtein, M. and Israeli, M. (1977), 'Efficiency of Navier-Stokes Solvers', *AIAA J* 15 pp263-266.
- Atkins, P. W. (1986), Physical Chemistry, 3rd Edition, Oxford University Press.
- Baldyga, J. and Bourne, J. R. (1986), 'Principles of Mixing', Chapter 6 of Encyclopedia of Fluid Mechanics: Volume 1 - Flow Phenomena and Measurement, Gulf Publishing, Houston, Texas.
- Beier, R. A., de Ris, J. and Baum, H. R. (1983), 'Accuracy of Finite Difference Methods in Recirculating Flows', *Num Heat Transfer* 6 pp283-302.
- Bolour-Froushan, A. H. (1986), 'Prediction of Single-Phase Turbulent Flow in Agitated Mixing Vessels', PhD. Thesis, University of London.
- Book, D. L., Boris, J. B. and Hain, K. (1975), 'Flux Corrected Transport II: Generalisation of the Method', *J Comp Phys* 18 p248.
- Boris, J. B. and Book, D. L. (1973), 'Flux Corrected Transport I: SHASTA, a Transport Algorithm that Works', *J Comp Phys* 11 p38.
- Bourne, J. R., Kozicki, F. and Rys, P. (1981), 'Mixing and Fast Reactions I: Test Reactions to Determine Segregation', *Chem Eng Sci* 36 pp1643-1648.
- Bradley, D., Missaghi, M. and Chin, S. B. (1988), 'Taylor Series Approach to Numerical Accuracy and a Third Order Scheme for Strong Convective Flows', *Comp Mthds Appl Mech Eng* 69 pp133-151.
- Castro, I. P. and Jones, J. M. (1987), 'Studies in Numerical Computations of Recirculating Flows', *Int J Num Mthds Fluids* 7 pp793-823.
- Chapman, M. (1981), 'FRAM - Non-Linear Damping Algorithms for the Continuity Equation', *J Comp Phys* 44 pp84-103.
- Chen, H.-C. and Chen, C.-J. (undated), 'Development of Finite Analytic Method for Unsteady Three-Dimensional Navier-Stokes Equations', source of publication not known.
- Chen, C.-J., Naseri-Neshat, H. and Ho, K. (1981), 'Finite Analytic Numerical Solution of Heat Transfer in Two Dimensional Cavity Flow', *Num Heat Transfer* 4 pp179-197.
- Claus, R. W., Neely, G. M. and Syed, S. A. (1985), 'Reducing Numerical Diffusion for Incompressible Flow Calculations', NASA Tech Memo 83621.
- Collins, P. (1990), PhD thesis in preparation, University of London.
- Corrsin, S. (1964), 'The Isotropic Turbulent Mixer: Part II. Arbitrary Schmidt Number', *AIChE J* 10 pp870-877.

- Courant, R., Isaacson, E. and Rees, M. (1952), 'On the Solution of Non-Linear Hyperbolic Equations by Finite Differences', *Comm Pure Appld Math* 5 pp243-255.
- Curtis, A. R. (1987), 'Accurate Conservative Discretisation of Transport Equations Retaining Diagonal Dominance', AERE Report R1253.
- Denis, S. C. R. and Hudson, J. D. (1978), 'A Difference Method for Solving the Navier-Stokes Equations', in Numerical Methods in Laminar and Turbulent Flow, Pentech Press, Plymouth.
- Dierckx, P. (1975), 'An Algorithm for Smoothing, Differentiation and Integration of Experimental Data Using Spline Functions', *J Comp Appl Maths* 1 pp165-183.
- Forsythe G. E. and Wasow W. (1960), Finite Difference Methods for Partial Differential Equations, Wiley, New York.
- Gaskell, P. H. and Lau, A. K. C. (1988), 'Curvature Compensated Transport: SMART; A New Boundedness Preserving Transport Algorithm', *Int J Num Mthds in Fluids* 8 June 1988.
- Georgopoulos, P. G. and Seinfeld, J. H. (1986), 'Mathematical Modelling of Turbulent Reacting Plumes - I. General Theory and Model Formulation', *Atmospheric Environment* 20 pp1791-1807.
- Gosman, A. D. (1986), private communication.
- Gosman, A. D. and Lai, K. Y. M. (1982), 'Finite Difference Approximations for the Transport and Navier-Stokes Equations', Fluids Section Report FS/82/18, Imperial College, London.
- Gosman, A. D., Pun, W. M., Runchal, A. K., Spalding, D. B. and Wolfshtein, M. (1969), Heat and Mass Transfer in Recirculating Flows, Academic Press, London.
- Gupta, M. M. and Manohar, R. P. (1978), 'A Critique of a Second-Order Upwind Scheme for Viscous Flow Problems', *ALAA J* 16 pp759-761.
- Gupta, M. M. (1983), 'A Survey of Some Second-Order Difference Schemes for the Steady-State Convection-Diffusion Equation', *Int J Num Mthds Fluids* 3 pp319-331.
- Hackman, L. P., Raithby, G. D. and Strong, A. B. (1984), 'Numerical Predictions of Flows over Backward Facing Steps', *Int J Num Mthds Fluids* 4 pp711-724.
- Hah, C. (1983), 'Calculation of Various Diffuser Flows with Inlet Swirl and Inlet Distortion Effects', *ALAA J* 21 pp1127-1133.
- Hah, C. (1984), 'A Navier-Stokes Analysis of Three-Dimensional Turbulent Flows Inside Turbine Blade Rows at Design and Off-Design Conditions', *J Eng Gas Turbines and Power (Trans ASME)* 106 pp421-429.
- Han, T., Humphrey, J. A. C. and Launder, B. E. (1981), 'A Comparison of HYBRID and Quadratic-Upstream Differencing in High Reynolds Number Elliptic Flows', *Comp Mthds Appl Mech Eng* 29 pp81-95.
- Handscorn, D. C. (1983), 'Spline Representation of Incompressible Flow', *IMA J Numerical Analysis* 4 pp491-502.
- Hassan, Y. A. and Kim, J. H. (1985a), 'A Navier-Stokes Analysis of Three Dimensional Negatively Buoyant Jet Injected into a Hot Cross Flow', *Nucl Science and Eng* 89 pp70-78.
- Hassan, Y. A. and Kim, J. H. (1985b), 'Computational Investigation of Fluid and Thermal Mixing of the EPRI/CREARE One-Fifth-Scale Facility', *Nucl Tech* 68 pp395-407.

- Hassan, Y. A., Rice, J. G. and Kim, J. H. (1984), 'An Improved Multi-Dimensional Finite Difference Scheme for Predicting Stratified Horizontal Pipe Flow', *Nucl Tech* 65 pp454-461.
- Hassan, Y. A., Rice, J. G. and Kim, J. H. (1983a), 'A Stable Mass Flow-Weighted Two-Dimensional Skew Upwind Scheme', *Num Heat Transfer* 6 pp395-408.
- Hassan, Y. A., Rice, J. G. and Kim, J. H. (1983b), 'Comparison of Measured and Predicted Thermal Mixing Tests Using Improved Finite Difference Technique', *Nucl Eng Des* 76 pp153-160.
- Hirsch, C. (1988), Numerical Computation of Internal and External Flows: Volume 1 - Fundamentals of Discretisation, John Wiley and Sons.
- Hockey, R. M. (1990) PhD thesis, University of London (in preparation).
- Huang, P. G., Launder, B. E. and Leschziner, M. A. (1983), 'Discretisation of Non-Linear Convection Processes: A Broad Range Comparison of Four Schemes', UMIST report TFD/83/1.
- Huh, K. Y., Golay M. W. and Manno (1986), 'A Method for the Reduction of Numerical Diffusion in the Donor Cell Treatment of Convection', *Int J Num Mthds Fluids*.
- Huh, K. Y. and Golay M. W. (1983), 'Treatment of Physical and Numerical Diffusion', MIT-EL-83-011.
- Issa, R. I. (1983), 'Solution of Implicitly Discretised Flow Equations by Operator Splitting', Imperial College Mechanical Engineering Department, Fluid Section Report FS/82/15.
- Jones, W. P. and Whitelaw, J. H. (1982), 'Calculation Methods for Reacting Turbulent Flows: A Review', *Combustion and Flame* 48 pp1-26.
- Joseph, M. (1983), 'Finite Difference Representations of Vorticity Transport', *Comp Mthds Appld Mech Eng* 39 pp107-116.
- Kinney, R. B. and Mahdi, H. S. (1988), 'A New Finite-Volume Approach with Adaptive Upwind Convection', *Int J Num Mthds Eng* 26 pp1326-1343.
- Lai, K. Y. M. (1983), 'Numerical Analysis of Fluid Transport Phenomena', PhD Thesis, Univ of London.
- Landau, L. D. and Lifshitz, E. M. (1963), Statistical Physics, Part 1, Pergamon Press.
- Landau, L. D. and Lifshitz, E. M. (1959), Fluid Mechanics, Pergamon Press.
- Launder, B. E. and Spalding, D. B. (1974), 'The Numerical Computation of Turbulent Flows', *Comp Mthds Appld Mech Eng* 3 pp269-289.
- Launder, B. E., Morse, A., Rodi, W. and Spalding, D. B. (1972), 'The Prediction of Free Shear Flows - A Comparison of the Performance of Six Turbulence Models', Proceedings of NASA conference on Free Shear Flows, Langley. Also available as Imperial College Mechanical Engineering Report TM/TN/A/19.
- Leonard, B. P. (1981), 'A Survey of Finite Differences with Upwinding for Numerical Modelling of the Incompressible Convective Diffusion Equation', in Recent Advances in Numerical Methods in Fluids, Volume 2: Computational Techniques in Transient and Turbulent Flow, Pineridge Press, Swansea.
- Leonard, B. P. (1979), 'A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation', *Comp Mthds Appld Mech Eng* 19 pp59-98.

- Leonard, B. P., Leschziner, M. A. and McGuirk, J. (1978), 'Third-Order Finite Difference Method for Steady Two-Dimensional Convection', Proceedings of the First International Conference on Numerical Methods in Laminar and Turbulent Flow, Eds Taylor *et al*, Pentech Press, London.
- Leschziner, M. A. (1980), 'Practical Evaluation of Three Finite Difference Schemes for the Computation of Steady-State Recirculating Flows.' *Comp Mthds Appld Mech Eng* 23 pp293-312.
- Leschziner, M. A. and Rodi, W. (1981), 'Calculation of Annular and Twin Parallel Jets Using Various Discretisation Schemes and Turbulence Model Variations', *J Fluids Eng (Trans ASME)* 103 pp352-359.
- Leschziner, M. A. (1979), 'A Comparison Between Three Finite-Difference Schemes for the Computation of Steady-State Recirculating Flows', Report SFB 80/T/146, SFB 80, Univ of Karlsruhe, W. Germany.
- Libby, P. A. and Bray, K. N. C. (1981), 'Countergradient Diffusion in Premixed Turbulent Flames', *ALAA J* 19 pp205-213.
- Lillington, J. N. (1981), 'A Vector Upstream Differencing Scheme for Problems in Fluid Flow Involving Significant Source Terms in Steady-State Linear Systems', *Int J Num Mthds Fluids* 1 pp3-16.
- Liu, C.-H. and Barklelew, C. H. (1986), 'Numerical Analysis of Jet Stirred Reactors with Turbulent Flows and Homogeneous Reactions', *AIChE J* 32 pp1813-1820.
- Looney, M. K. (1987), private communications.
- Lumley, J. L. (1975), 'Modelling of Turbulent Flux of Passive Scalar Quantities in Inhomogenous Flows', *Physics of Fluids* 18 pp619-621.
- Lynch, P. M. (1985), 'Mixing Time Program', private communication.
- Lynch, P. M. and Parker, I. B. (1982), 'Mark 3 Disc Turbine Impeller Model', private communication.
- Mallinson, G. D. and de Vahl Davis, G. (1973), 'The Method of False Transients for the Solution of Coupled Elliptic Equations', *J Comp Phys* 12 pp435-461.
- McGuirk, J. J., Taylor, A. M. K. P. and Whitelaw, J. H. (1981), 'The Assessment of Numerical Diffusion of Upwind-Difference Calculations of Turbulent Recirculating Flows', Fluid Section report FS/81/30, Imperial College, London. Presented at Third Symposium on Turbulent Shear Flows, Univ of California, Davis in September 1981.
- Middleton, J. C. (1990), private communication.
- Middleton, J. C., Pierce, F. and Lynch, P. M. (1986), 'Computations of Flow Fields and Complex Reaction Yield in Turbulent Stirred Reactors, and Comparison with Experimental Data', *Chem Eng Res Des* 64 pp18-22.
- Nouri, J. M. (1988), 'Effects of Mixing Vessel Size and Confinement on Flow Characteristics in Stirred Reactors', Fluid Section Report FS/88/34, Mechanical Engineering Department, Imperial College, London.
- Owen, A. (1984), 'Artificial Diffusion in the Numerical Modelling of the Advective Transport of Salinity', *Appl Math Modelling* 8 pp116-120.
- Patankar, S. V. (1980), Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York.
- Patel, M., Cross, M. and Markatos, N. C. (1988), 'Assessment of Flow Oriented Schemes for Reducing False Diffusion', *Int J Num Mthds Eng* 26 pp2279-2304.

- Patel, M. K., Cross, M., Markatos, N. C. and Mace, A. C. H. (1987), 'An Evaluation of Eleven Discretisation Schemes for Predicting Elliptic Flow and Heat Transfer in Supersonic Jets', *Int J Heat Mass Transfer* 30 pp1907-1925.
- Patel, M. K. and Markatos, N. C. (1986), 'An Evaluation of Eight Discretisation Schemes for Two-Dimensional Convection-Diffusion Equations', *Int J Num Mthds Fluids* 6 pp129-154.
- Patel, M. K., Markatos, N. C. and Cross, M. (1985a), 'A Critical Evaluation of Seven Discretisation Schemes for Convection-Diffusion Problems.' *Int J Num Mthds Fluids* 5 pp225-244.
- Patel, M. K., Markatos, N. C. and Cross, M. (1985b), 'Method of Reducing False-Diffusion Errors in Convection-Diffusion Problems.' *Appl Math Modelling* 9 pp302-306.
- Patterson, G. K. (1981), 'Application of Turbulent Fundamentals to Reactor Design and Scale Up', *Chem Eng Comm* 8 pp25-52.
- Peric, M. (1986), PhD Thesis, Univ of London.
- Pohorecki, R. and Baldyga, J. (1983), 'A New Model of Micromixing in Chemical Reactors 1. General Development and Application to a Tubular Reactor', *Ind Chem Fundam* 22 pp392-397.
- Politis, S. (1989a), private communication.
- Politis, S. (1989b), PhD Thesis, University of London.
- Pollard, A. and Siu, A. L.-W. (1982), 'The Calculation of some Laminar Flows Using Various Discretisation Schemes', *Comp Mthds Appld Mech Eng* 35 pp293-313.
- Pollock, D. W. (1988), 'Semi-Analytical Computation of Path Lines for Finite Difference Models', *Ground Water* 26 pp743-750.
- Prakash, C. (1984), 'Application of the Locally Analytic Differencing Scheme to Some Test Problems for the Convection-Diffusion Equation', *Num Heat Transfer* 7 pp165-182.
- Price, H. S., Varga, R. S. and Warren, J. E. (1966), 'Application of Oscillation Matrices to Diffusion-Convection Equations', *J Maths Physics* 45 pp301-311.
- Raithby, G. D. (1976), 'Skew Upstream Differencing Schemes for Problems Involving Fluid Flow', *Comp Meths Appl Mech Eng* 9 pp153-164.
- Revill, (1987), private communication.
- Rhie, C. M. and Chow, W. L. (1982), 'A Numerical Study of the Turbulent Flow Past an Isolated Airfoil with Trailing Edge Separation', AIAA 82 0998.
- Runchal, A. K. (1987), 'CONDIF: A Modified Central-Difference Scheme for Convective Flows', *Int J Num Mthds Eng* 24 pp1593-1608.
- Scarborough J. B. (1958), Numerical Mathematical Analysis, 4th ed. John Hopkins Press, Baltimore.
- Sharif, M. A. R. and Busnaina, A. A. (1988), 'An Investigation into the Numerical Dispersion Problem of the Skew Upwind Finite Difference Scheme', *Appl Math Modelling* 12 pp98-108.
- Shyy, W. (1983), 'Study of Finite Difference Schemes Approximations to Steady-State Convection-Dominated Flow Problems', General Electric Technical Information Series Report 83CRD189.

- Shyy, W. (1984), 'A Note on Assessing Finite Difference Procedures for Large Peclet/Reynolds Number Flow Calculation.' Third Int Conf on Boundary and Interior Layers: Computational and Asymptotic Methods.
- Shyy, W. and Correa, S. M. (1985), 'A Systematic Comparison of Several Numerical Schemes for Complex Flow Calculations', AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada. AIAA-85-0440.
- Smith, R. M. and Hutton, A. G. (1982), 'The Numerical Treatment of Advection: A Performance Comparison of Current Methods', *Numerical Heat Transfer* 5 pp439-461.
- Spalding, B. D. (1972), 'A Novel Finite Difference Formulation for Differential Expressions Involving Both First and Second Derivatives', *Int J Num Mthds Eng* 4 pp551-559.
- Stewart, I. (1981), *Concepts of Modern Mathematics*, Penguin Books, Harmondsworth, England.
- Stubley, G. D., Raithby, G. D. and Strong, A. B. (1980), 'Proposal for a New Discrete Method Based on an Assessment of Discretisation Errors', *Num Heat Transfer* 3 pp411-428.
- Syed, S. and Chiapetta, L. (1985), 'Finite Difference Methods for Reducing Numerical Diffusion in TEACH-Type Calculations.' AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada. AIAA-85-0057.
- Syed, S., Gosman, A. D. and Peric, M. (1985), 'Assessment of Discretisation Schemes to Reduce Numerical Diffusion in the Calculation of Complex Flows', AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada. AIAA-85-0441.
- Thom, A. (1933), 'The Flow Past Cylinders at Low Speeds', *Proc Roy Soc of London* A141, p651.
- Timin, T. and Esmail, N. (1983), 'A Comparative study of Central and Upwind Difference Schemes Using the Primitive Variables', *Int J Num Mthds Fluids* 3 pp295-305.
- Too, W. O. and Sparrow, E. M. (1987), 'The Transportive Property and Convective Numerical Stability of the Steady State Convection-Diffusion Equation', *Numerical Heat Transfer* 11 pp491-497.
- Tsui, Y. Y. (1987) PhD Thesis, University of London.
- Uhl, V. W. and Gray, J. B. (1966), *Mixing: Theory and Practice. Volume 1*, Academic Press, New York.
- Vanka, S. P. (1987), 'Second Order Upwind Differencing in a Recirculating Flow', *AIAA J* 25 pp1435-1441.
- Wilkes, N. S. and Thompson, C. P. (1983), 'An Evaluation of Higher-Order Upwind Differencing for Elliptic Flow Problems', AERE report CSS 137.
- Wong, H. H. and Raithby, G. D. (1979), 'Improved Finite-Difference Methods Based on a Critical Evaluation of the Approximation Errors.' *Num Heat Transfer* 2 pp139-163.
- Yianneskis, M. (1982), 'Flow Characteristics in a Standard Configuration Stirred Reactor', Imperial College Mech. Eng. Dept Report FS/82/23.
- Yianneskis, M., Popielek, Z. and Whitelaw, J. H. (1987), 'An Experimental Study of the Steady and Unsteady Flow Characteristics of Stirred Reactors', *J Fluid Mech* 175 pp537-555.
- Zalesak, S. T. (1979), 'Fully Multidimensional Flux Corrected Transport Algorithms for Fluids', *J Comp Phys* 31 pp335-362.