

Cranfield University



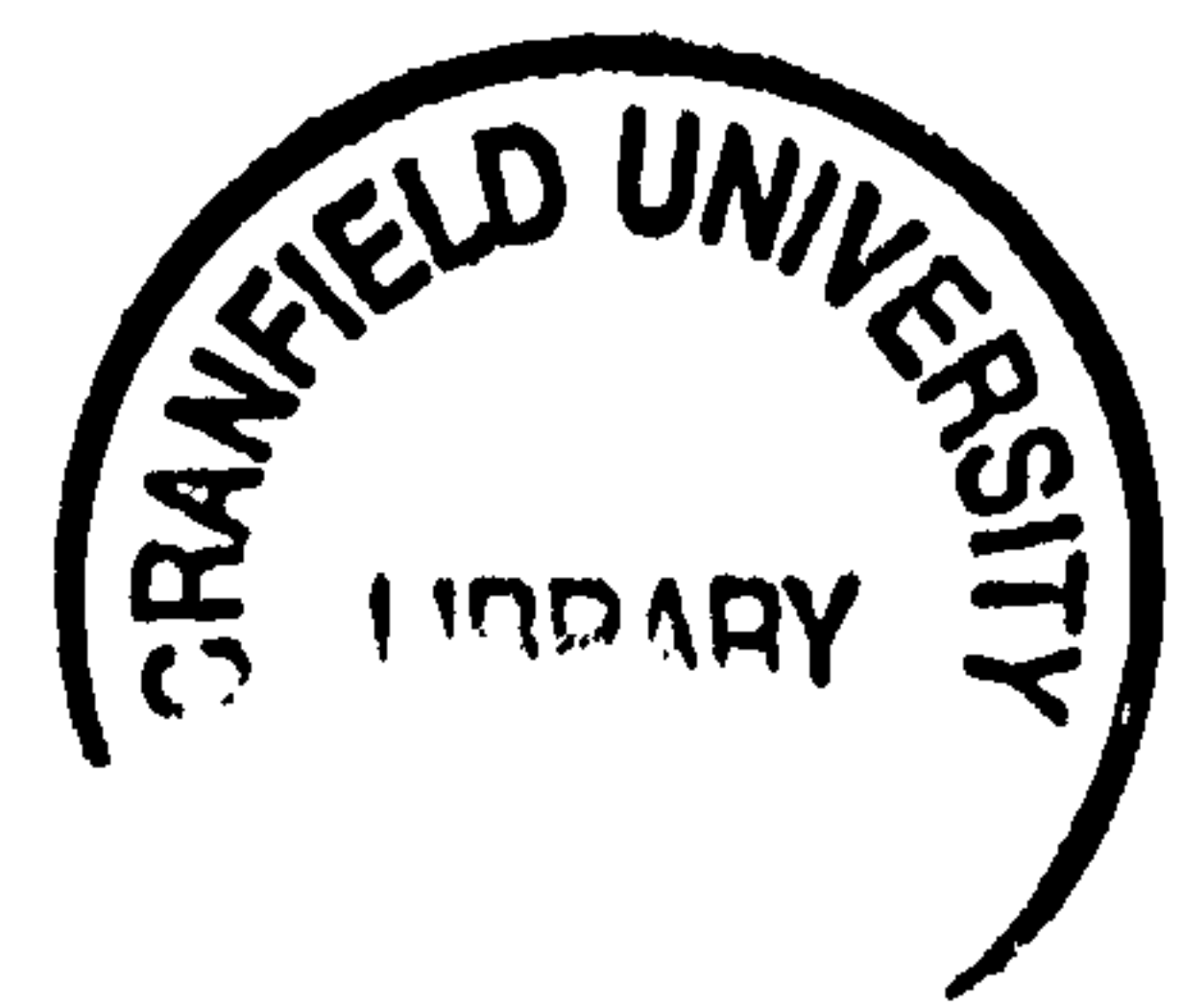
Patrick A. Lezeau

An Adaptive
Quasi-Newton Coupled Multigrid Solver
for the Simulation of
Steady Multiphase Flows

1997

School of Mechanical Engineering

PhD Thesis



Cranfield University
School of Mechanical Engineering
Applied Mathematics and Computing Group

PhD Thesis

Academic Year 1996-7

Patrick A. Lezeau

**An Adaptive
Quasi-Newton Coupled Multigrid Solver
for the Simulation of
Steady Multiphase Flows**

Supervisor: Dr C. P. Thompson

May 1997

The present thesis is submitted
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

BLANK IN ORIGINAL

Pour Elizabeth (et al.), Carmen, Daniel et Marc – Merci

... ..

When I survey the wondrous cross
On which the Prince of Glory died,
My richest gain I count but loss,
And pour contempt on all my pride.

Forbid it, Lord that I should boast,
Save in the death of Christ my God:
All the vain things that charm me most,
I sacrifice them to His blood.

See from His head, His hands, His feet,
Sorrow and love flow mingled down:
Did e'er such love and sorrow meet,
Or thorns compose so rich a crown?

Where the whole realm of Nature mine,
that were an offering far too small;
Love so amazing, so divine,
demands my soul, my life, my all!

Isaac Watts

Acknowledgements

I would like to thank particularly my supervisor, Dr C. P. Thompson both for the guidance and encouragement I received during the course of this project, and for the many stimulating discussions which have facilitated my task. I am also indebted to Cranfield University, *malgré tout*, for the funding I received during part of this project and to Dr J. K. Reid of Rutherford Appleton Laboratory who made the automatic differentiation package AD01 available to me, without which it would have been difficult to complete this work.

I am, however, most grateful to my wife, Elizabeth, who as a linguist, read and corrected this work (twice), and still maintains she does not understand it. Her constant support, particularly during the more difficult times has been invaluable. Where would I be without you?

Final credits belong to my God, the great I AM, who has led me thus far.

Unless the LORD builds the house, its builders labour in vain. Unless the LORD watches over the city, the watchmen stand guard in vain. In vain you rise early and stay up late, toiling for food to eat – for he grants sleep to those he loves.

Psalm 127:1-2

Abstract

This thesis is concerned with the application of adaptive local quasi-Newton coupled multigrid (ALQNMG) solvers to the numerical simulation of viscous incompressible fluids, using the multi-fluid model.

The ALQNMG methodology has proven highly successful for single phase flows [1], leading to solution algorithms which are: (i) robust, (ii) efficient and (iii) accurate. Its extension to multiphase flows is very challenging because the governing equations are mathematically complex and their solutions are subject to constraints. The solver presented here has therefore required a considerable number of specific algorithmic developments.

The outline of the thesis is as follows: firstly, the modelling and simulation of multiphase flows are reviewed, together with the different numerical techniques implemented in the solver. Finite volume discrete multiphase equations are then derived on structured, staggered grids. Next, having specified the solution algorithm, we consider the accuracy of the solver. Results from several test cases of varying complexity are compared with those of a widely used commercial CFD package and good agreement is obtained.

The question of performance is then addressed in detail, both in terms of robustness and speed of convergence. Good accelerations are obtained using the multigrid method but the convergence rates are often not grid-independent. The most likely explanation is that the discrete operators are highly non-linear and therefore have different characteristics on different grids. Furthermore, the solution algorithm is shown to not handle certain multiphase diffusive terms very well. Convergence rates are much faster than those achieved by single grid solvers and commercial codes, typically by one order of magnitude and often more, although the solver is not fully optimal.

Finally, adaption is considered. Grids are generated automatically which facilitates the use of the code and allows error control. It is confirmed that multigrid methods offer a good framework for the implementation of adaption. Considerable gains in speed and memory usage, by one further order of magnitude, are achieved.

Contents

Abstract	i
Acknowledgements	vii
Table of Contents	vii
List of Figures	xii
List of Tables	xxviii
Notation	xxxi
1 Introduction	1
2 A review of Relevant Literature	9
2.1 Introduction and Thesis Rationale	9
2.2 Basic Multiphase Flow Models	12
2.2.1 The Multi-Fluid Model	12
2.2.2 Derivation of the Multi-Fluid Equations for the Two-Phase Flow of Incompressible Fluids	13
2.2.3 Discussion of the Multi-Fluid Equations	17
2.2.4 Summary of the Multi-Fluid Equations	18
2.2.5 Limiting Case 1: the Homogeneous Model	19
2.2.6 Limiting Case 2: the Drift Flux Model	19
2.3 Numerical Simulation of Multiphase Flows	19
2.3.1 Brief Overview of Numerical Issues in the Simulation of Multiphase Flows	19
2.3.2 The SETS Method	21
2.3.3 Example of an Fully Implicit Scheme: IPSA	21
2.4 Coupled Solvers for the (Steady) Navier-Stokes Equations	23
2.5 Multigrid Methods	25
2.5.1 A Limitation of Traditional Iterative Solvers for Systems of Alge- braic Equations	26
2.5.2 Fundamental Concepts of Multigrid Methods: A Qualitative Approach	29
2.5.3 The Two-level Algorithm and the Correction Storage Scheme	30
2.5.4 Multigrid Cycling Strategies	39
2.5.5 Multigrid Convergence Rates for Linear Problems	42
2.5.6 Full Approximation Storage Scheme	42
2.5.7 FAS and Non-Linearity	44

2.5.8	Defect, Truncation Error and Adaption in the FAS Framework . . .	44
2.5.9	Multigrid Methods for Non-Elliptic PDEs	46
2.6	Adaptive Solution of Partial Differential Equations	47
2.7	Conclusions	49
3	Governing Equations and their Discretisation	51
3.1	Issues and Choices Made in Discretisation	51
3.1.1	Advantages and Disadvantages of Staggered Grids for Fluid Flow Computations	52
3.1.2	Finite Volume Discretisation Using Hybrid Schemes	53
3.1.3	Solver Robustness	59
3.2	Discretisation of the Steady Navier-Stokes Equations for an Incompressible Fluid	60
3.2.1	Governing Equations and Preliminary Remarks	60
3.2.2	Discretisation of the Continuity Equation	61
3.2.3	Discretisation of the Equation for Horizontal Momentum	62
3.2.4	Discretisation of the Equation for Vertical Momentum	67
3.2.5	Summary of the Results	67
3.3	Discretisation of the Multi-fluid Equations	69
3.3.1	Governing Equations for Multiphase Flows	69
3.3.2	Governing Equations in Scalar Variable Form	71
3.3.3	General Discretisation methodology	73
3.3.4	Discretisation of the Multiphase Continuity Equation	74
3.3.5	Discretisation of the Equation for Horizontal Momentum	75
3.3.6	Discretisation of the Vertical Momentum Equation	79
3.3.7	Summary of the Results	80
3.4	Conclusions	82
4	The pamg and pamg-multiphase Solvers: Description and Differences	85
4.1	Introduction	85
4.2	The pamg Single Phase Algorithm	86
4.2.1	The Local Quasi-Newton Coupled Solver	86
4.2.2	The pamg FAS Multigrid Algorithm – Definition of the Grid Transfer Operators	90
4.2.3	Boundary Conditions in pamg	94
4.2.4	Adaption in pamg	94
4.3	From pamg to pamg-multiphase: Algorithmic Issues	97
4.3.1	A Globally Convergent Quasi-Newton Solver for Multiphase Flows .	97
4.3.2	Stable Interpolation Procedures on Staggered Grids	103
4.3.3	Interpolation Operators for Multiphase Flows and Implications for Adaption	104
4.3.4	Boundary Conditions	109
4.3.5	Adaption in pamg-multiphase	109
4.3.6	Summary of Algorithmic Issues	110
4.4	From pamg to pamg-multiphase: Computational Issues	110
4.4.1	Methodology Adopted	111
4.4.2	Phased Modification Plan and Testing	112
4.4.3	Run Times and Optimisation	113
4.4.4	Computer Language Issues	113

4.5	Conclusions	114
5	pamg-multiphase Computations on Uniform Grids: Validation and Performance	115
5.1	Introduction	115
5.2	Single Phase Validation Problems	117
5.2.1	Problem 1: Pseudo-Two-Phase Channel Flow	117
5.2.2	Problem 2: Single Phase T-Junction with 2 Inlets	122
5.2.3	Problem 3: Single Phase flow through a Backward-Facing Step	128
5.3	Multiphase Test Problems	134
5.3.1	Problem 1: Two-Phase Channel Flow	134
5.3.2	Problem 2: Two-phase Flow Through a T-Junction with Two Inlets	153
5.3.3	Test Problem 3: Two-phase Backward-Facing Step	169
5.3.4	Estimates of the Truncation Errors	188
5.3.5	Conclusions	190
5.3.6	Volume Fraction Maps Obtained from Pamg-multiphase Solutions	191
5.4	The performance of the pamg-multiphase Solver	193
5.4.1	Introduction	193
5.4.2	Multigrid Acceleration	195
5.4.3	Multigrid Convergence Factors and Grid Resolution	201
5.4.4	Comparison with pamg Convergence Factors	205
5.4.5	Brief Investigation of Multigrid Cycling Strategies	214
5.4.6	Effect of Under Relaxation	215
5.4.7	Effect of Multigrid Cycling Parameters	218
5.4.8	Some Experiments on Interpolation Operators for the Volume Fractions	222
5.4.9	Effect of the Source Terms for Inter-Phase Momentum Transfers	230
5.4.10	Convergence Factors in "Difficult" Flow Regions	232
5.4.11	Some Experiments on Reynolds Numbers Effects	241
5.4.12	Two-Phase Flow through a Longer Channel – Effect of Outlet Boundary Conditions	251
5.4.13	Effect of Different Formulations of the Stress Tensor Flux	255
5.4.14	More on the Viscous Tensor	271
5.5	Conclusions	278
6	pamg-multiphase Computations on Adaptive Grids: Accuracy and Benefits	283
6.1	Introduction	283
6.2	Adaptive Computation of the Two-Phase Channel Flow Problem	284
6.2.1	Effect of the Different Transfer Procedures on Uniform Grids	285
6.2.2	Adaptive A(2,3) Results	288
6.2.3	Adaptive A(3,4) Results	294
6.2.4	Adaptive A(3,5) and A(3,6) Results	298
6.3	Adaptive Computations of the Two-Phase T-Junction Problem	304
6.4	Interpolation Procedures, Conservation of Mass and Convergence	319
6.5	Estimation of the Performance Gain from Adaptive Gridding	320
6.6	Conclusions	322
6.7	Uniform and Adaptive Grids for the pamg and pamg-multiphase Solvers	324
6.7.1	Channel Geometry	324

6.7.2	T-junction Geometry	327
6.7.3	Backward-Facing Step Geometry	329
7	Future Work	331
8	Conclusions	333
A	One-Dimensional Steady Two-Phase Flow	339
A.1	Model Equations	339
A.2	Discretisation and Hybrid Schemes	339
A.3	Multigrid Solutions	347
B	Solution Profiles for a Single Phase Entry Flow	349
	Bibliography	353
	Final Credits	361

List of Figures

2.1	Basic two-level scheme for a one-dimensional diffusion problem: Coefficients c_k as a (continuous) function of k/N	36
2.2	Basic two-level scheme for a one-dimensional diffusion problem: Coefficients s_k as a (continuous) function of k/N	36
2.3	Basic two-level scheme for a one-dimensional diffusion problem: Eigenvalue $\lambda(k)$ of the iteration matrix P as a (continuous) function of k/N	37
2.4	Basic two-level scheme for a one-dimensional diffusion problem: Attenuation factor for low frequency modes k as a (continuous) function of k/N	37
2.5	Basic two-level scheme for a one-dimensional diffusion problem: Attenuation factor for high frequency modes $N - k$ as a (continuous) function of k/N	38
2.6	Four level multigrid V-Cycle	41
2.7	Four level multigrid W-Cycles	41
2.8	Four level multigrid F-Cycles	41
2.9	Four level multigrid FMG algorithm	41
3.1	Staggered grid used for the discretisation of the Navier-Stokes equations	61
3.2	Control volume for the integration of the Navier-Stokes continuity equation	62
3.3	Control volume for the integration of the Navier-Stokes horizontal momentum equation	63
3.4	Control volume for the integration of the Navier-Stokes vertical momentum equation	68
3.5	Staggered grid used for the discretisation of the multi-fluid equations	74
3.6	Control volume for the integration of the multi-fluid continuity equation	75
3.7	Control volume for the integration of the multi-fluid horizontal momentum equation	76
3.8	Control volume for the integration of the multi-fluid vertical momentum equation	79
4.1	Discrete values for the restriction of the horizontal velocities	90
4.2	Discrete values for the restriction of the pressures	91
4.3	Discrete values for the prolongation of the horizontal velocities	92
4.4	Arrangement of variables at grid interfaces	92
4.5	Conservative multiphase transfer procedure	108
5.1	Geometrical representation of the channel flow problem	119
5.2	Computational grid for the channel flow problem – Uniform level 3 grid (768 cells, $\Delta x = \Delta y = 0.0625$)	119

5.3	Channel flow problem – Coordinate system and main sections used to give the solution profiles	120
5.4	Pseudo multiphase channel flow – Streamlines for the pamg-multiphase solution	120
5.5	Pseudo multiphase channel flow – Velocity profile along the line $x = 2.5$. .	120
5.6	Pseudo multiphase channel flow – Pressure drop along the line $y = 0.5$. . .	121
5.7	Geometrical representation of the T-junction problem	122
5.8	T-junction Problem – Coordinate system and main sections used to give the solution profiles	123
5.9	T-junction problem – Uniform grid at level 3 (2336 cells, $\Delta x = \Delta y = 0.0625$)	123
5.10	Single phase T-junction problem – Streamlines for the pamg-multiphase solution	124
5.11	Single phase flow through a T-junction – Horizontal velocity profiles along the line $x = 3.5$ on level 3 uniform grids for different solvers	124
5.12	Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.5$ on uniform grids for different solvers	125
5.13	Single phase flow through a T-junction – Horizontal velocity profiles along the line $x = 3.5$ on adaptive (3,4) grids	125
5.14	Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.5$ on adaptive (3,4) grids	126
5.15	Single phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.5$ on adaptive (3,4) grids	126
5.16	Single phase flow through a T-junction – Velocity profile along the line $y = 0.5$ on adaptive (3,4) grids – Note that the computational grids are identical even though the number of interpolated point differ.	127
5.17	Diagrammatic description of the problem of a two-phase flow through a backward facing step	128
5.18	Single phase backward-facing step problem – Coordinate system and main sections used to give solution profiles	129
5.19	Single phase backward-facing problem – Uniform grid at level 3 (9216 cells, $\Delta x = 0.03125$, $\Delta y = 0.046875$)	129
5.20	Single phase T-junction problem – Streamlines for the pamg-multiphase solution (shortened in the horizontal direction)	130
5.21	Single phase backward-facing problem – Horizontal velocity profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4	131
5.22	Single phase backward-facing problem – Vertical velocity profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4	131
5.23	Single phase backward-facing problem – Pressure profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4	132
5.24	Single phase backward-facing problem – Horizontal velocity profile along the line $x = 3.8$ – Comparison of the pamg-multiphase and CFX 4.1 solutions on a uniform level 3 grid	132
5.25	Single phase backward-facing problem – Vertical velocity profile along the line $x = 3.8$ – Comparison of the pamg-multiphase and CFX 4.1 solutions on a uniform level 3 grid	133

5.26	Two-phase channel flow – Streamlines for phase 1	136
5.27	Two-phase channel flow – Streamlines for phase 2	136
5.28	Two-phase channel flow – pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.5$	136
5.29	Two-phase channel flow – pamg-multiphase results – Volume fraction profiles along the line $y = 0.5$	137
5.30	Two-phase channel flow – pamg-multiphase results – Horizontal velocity profiles along the line $x = 2.5$	137
5.31	Two-phase channel flow – pamg-multiphase results – Vertical velocity profiles along the line $x = 2.5$	138
5.32	Two-phase channel flow – pamg-multiphase results – Volume fraction profile for phase 1 along the line $x = 2.5$	138
5.33	Two-phase channel flow – pamg-multiphase results – Volume fraction profile for phase 2 along the line $x = 2.5$	139
5.34	Two-phase channel flow – pamg-multiphase results – Volume fraction profile for phase 1 along the line $x = 2.5$ for the following set of physical properties: $(\mu_1 = 0.01, \rho_1 = 1)$ and $(\mu_2 = 0.005, \rho_2 = 1)$	139
5.35	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)	141
5.36	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ (Phase 1) – See Figure 5.37	141
5.37	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ (Phase 1) after shifting . .	142
5.38	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Volume fraction profiles along the line $y = 0.5$ (Phase 1)	142
5.39	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $x = 2.5$ (Phase 1)	143
5.40	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $x = 2.5$ (Phase 1) – Note the scale	143
5.41	Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Volume fraction profiles along the line $x = 2.5$ (Phase 1)	144
5.42	Two-phase channel flow – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)	146
5.43	Two-phase channel flow – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $y = 0.5$ (Phase 1)	146
5.44	Two-phase channel flow – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 2.5$ (Phase 1)	147
5.45	Two-phase channel flow – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $x = 2.5$ (Phase 1)	147
5.46	Two-phase channel flow – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $x = 2.5$ (Phase 1)	148
5.47	Two-phase channel flow – Grid independence for CFX 4.1 results – Volume fraction profiles along the line $x = 0.5$ (Phase 1) – Compare with Figure 5.43	148
5.48	Two-phase channel flow – Grid independence for CFX 4.1 results – Volume fraction profiles along the line $x = 2.5$ (Phase 1) – Compare with Figure 5.46	149
5.49	Horizontal velocity profile when inter-phase momentum transfer are modelled, along the line $x = 2.5$ – Compare with Figure 5.30	151

5.50	Vertical velocity profile when inter-phase momentum transfer are modelled, along the line $x = 2.5$ – Compare with Figure 5.31	151
5.51	Horizontal velocity profile when inter-phase momentum transfer are modelled, along the line $y = 0.5$ – Compare with Figure 5.28	152
5.52	Comparison of the volume fraction profiles with and without inter-phase momentum transfer terms, along the line $x = 2.5$	152
5.53	Multiphase T-junction problem – Streamlines for phase 1	155
5.54	Multiphase T-junction problem – Streamlines for phase 2 – Note the recirculation region	155
5.55	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ – Note the problems experienced by CFX 4.1 at the outlet, see below Figure 5.56 . . .	155
5.56	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ (near the outlet – $6 \leq x \leq 7$) showing the innacurate solution generated by CFX 4.1 .	156
5.57	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $y = 0.5$ – Note again the poor quality of the CFX 4.1 solution at the outlet	156
5.58	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$	157
5.59	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ after shifting	157
5.60	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $x = 3.5$, just downstream of the recirculation region	158
5.61	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $x = 3.5$, just downstream of the recirculation region – See Figure 5.62	158
5.62	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical momentum profiles along the line $x = 3.5$, just downstream of the recirculation region – See Figure 5.61	159
5.63	Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Volume fraction profiles along the line $x = 3.5$, just downstream of the recirculation region for phase I (i.e. the denser phase)	159
5.64	Multiphase T-junction problem – CFX 4.1 solution – Horizontal velocity profiles along the line $y = 0.5$ for different outlet boundary conditions, showing their effect on the quality of the solution	160
5.65	Multiphase T-junction problem – CFX 4.1 solution – Horizontal velocity profiles along the line $y = 0.5$ for different outlet boundary conditions – Detail of the solutions near the outlet showing the effect of the outlet boundary conditions	160
5.66	Multiphase T-junction problem – CFX 4.1 solution – Vertical velocity profile along the line $y = 0.5$ for different outlet boundary conditions showing their effect on the quality of the solution	161
5.67	Multiphase T-junction problem – CFX 4.1 solution – Volume fraction contours for phase 1	161
5.68	Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)	162

5.69 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $x = 2.5$ (Phase I) 162

5.70 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 3.0$ (Phase I) 163

5.71 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $x = 3.0$ (Phase I) 163

5.72 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $x = 3.0$ (Phase I) 164

5.73 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $x = 3.0$ (Phase II) 164

5.74 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.0$ (Phase II) 165

5.75 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $y = 0.0$ (Phase I) 165

5.76 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $y = 0.0$ (Phase I) showing that the volume fractions can be grid-dependent in certain regions 166

5.77 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 6.5$ (Phase I) 166

5.78 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 6.5$ (Phase I) 167

5.79 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 6.5$ (Phase II) 167

5.80 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Vertical velocity profiles along the line $x = 6.5$ (Phase II) 168

5.81 Multiphase T-junction problem – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $x = 6.5$ (Phase I) 168

5.82 Diagrammatic description of the problem of a two-phase flow through a backward facing step 170

5.83 Multiphase backward-facing step problem – Coordinate system and main sections 170

5.84 Multiphase backward-facing step problem – Uniform grid at level 3 (3072 cells, $\Delta x = 0.09375$, $\Delta y = 0.0625$) 171

5.85 Multiphase backward-facing step problem – Streamlines for phase 1 173

5.86 Multiphase backward-facing step problem – Streamlines for phase 2 173

5.87 Multiphase backward-facing step problem – Horizontal velocity profiles along the line $y = 0.5$ showing the fluid deceleration through the step 174

5.88 Multiphase backward-facing step problem – Vertical velocity profiles along the line $y = 0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 174

5.89 Multiphase backward-facing step problem – Volume fraction profiles along the line $y = 0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 175

5.90 Multiphase backward-facing step problem – Pressure profiles along the line $y = 0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 . . . 175

5.91 Multiphase backward-facing step problem – Horizontal velocity profiles along the line $y = -0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 176

5.92	Multiphase backward-facing step problem – Vertical velocity profiles along the line $y = -0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3	176
5.93	Multiphase backward-facing step problem – Volume fraction profiles along the line $y = -0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3, showing the separation of the phases just downstream of the step . .	177
5.94	Multiphase backward-facing step problem – Horizontal velocity profiles along the line $x = 3.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 showing the effect of the recirculation zone on the values of primitive variables	177
5.95	Multiphase backward-facing step problem – Vertical velocity profiles along the line $x = 3.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3 showing the effect of the recirculation zone on the values of primitive variables	178
5.96	Multiphase backward-facing step problem – Horizontal momentum profiles along the line $x = 3.5$ for phase II – Comparison of pamg-multiphase solutions at levels 2 and 3	178
5.97	Multiphase backward-facing step problem – Vertical momentum profiles along the line $x = 3.5$ for phase II – Comparison of pamg-multiphase solutions at levels 2 and 3	179
5.98	Multiphase backward-facing step problem – Volume fraction profiles along the line $x = 3.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3	179
5.99	Multiphase backward-facing step problem – Horizontal velocity profiles along the line $x = 10.0$ – Comparison of pamg-multiphase solutions at levels 2 and 3	180
5.100	Multiphase backward-facing step problem – Vertical velocity profiles along the line $x = 10.0$ – Comparison of pamg-multiphase solutions at levels 2 and 3 – Note the scale	180
5.101	Multiphase backward-facing step problem – Volume fraction profiles along the line $x = 10.0$ – Comparison of pamg-multiphase solutions at levels 2 and 3	181
5.102	Two-phase backward-facing problem – Horizontal velocity profile along the line $x = 3.5$ (phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids	182
5.103	Two-phase backward-facing problem – Vertical velocity profile along the line $x = 3.5$ (phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids	182
5.104	Two-phase backward-facing problem – Volume fraction profile along the line $x = 3.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids)	183
5.105	Two-phase backward-facing problem – Horizontal velocity profile along the line $x = 10.0$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids	183
5.106	Two-phase backward-facing problem – Volume fraction profile along the line $x = 10.0$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – See text for further comments	184

5.107 Two-phase backward-facing problem – Vertical velocity profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – Note the slight spurious oscillations in the CFX 4.1 solution near the outlet 184

5.108 Two-phase backward-facing problem – Pressure profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids 185

5.109 Two-phase backward-facing problem – Pressure profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids after shifting 185

5.110 Two-phase backward-facing problem – Volume fraction profile along the line $x = 10.0$ (Phase 2) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids 186

5.111 Two-phase backward-facing problem – Horizontal velocity profile along the line $y = -0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids 186

5.112 Two-phase backward-facing problem – Volume fraction profile along the line $y = -0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – Note the slight spurious oscillations in the CFX 4.1 solution near the outlet 187

5.113 Two-phase backward-facing problem – CFX 4.1 solution – contours for the volume fraction of phase 1 – Note the slight spurious oscillations 187

5.114 Two-phase channel flow – Volume fraction map for phase 1 – See page 134 . 191

5.115 Multiphase T-junction problem – Volume fraction map for phase 1 – See page 153 191

5.116 Multiphase backward-facing step problem – Volume fraction map for phase 1 (see page 169) showing the exclusion from the region behind the step of the lighter phase (phase 2) 192

5.117 Multiphase backward-facing step problem at higher Reynolds numbers – Volume fraction map for phase 1 (see page 233) showing more extended recirculation zones 192

5.118 Two-phase channel flow – Convergence of solution for multi-grid computations and the equivalent single grid computation 197

5.119 Two-phase channel flow – Volume fraction profiles along the line $x = 2.5$ – Comparison of multigrid and single grid solutions 197

5.120 Multiphase T-junction problem – Comparison of the convergence histories for single and multigrid computations at level 3 198

5.121 Multiphase T-junction problem – Comparison of the convergences histories for pamg-multiphase and CFX 4.1 (with default under-relaxation) 199

5.122 Multiphase T-junction problem – Comparison of the convergence histories for pamg-multiphase and CFX 4.1 for the initial iterations, showing that pamg-multiphase is always faster than CFX 4.1 200

5.123 Multiphase T-junction problem – Comparison of the convergence histories for CFX 4.1 solutions and for different under-relaxation strategies. In each case, the under-relaxation factors are either the default values or set to $\lambda_r = 0.1$ 200

5.124 Two-phase channel flow – Convergence of solution for different numbers of grids 204

5.125	Multiphase T-junction problem – Convergence factors for multi-grid computations at different levels	204
5.126	Convergence histories of the <code>pamg-multiphase</code> solver for the pseudo-multiphase channel flow without relaxation	207
5.127	Convergence histories for a <i>single phase</i> channel problem – single phase <code>pamg</code> solver	208
5.128	Convergence histories for a <i>single phase</i> channel problem – <code>pamg</code> multiphase with automatic differentiation Jacobians	208
5.129	Convergence histories for a single phase channel problem – <code>pamg</code> multiphase with approximate Jacobians and without under-relaxation	209
5.130	Convergence histories for a single phase channel problem – <code>pamg</code> multiphase with automatic differentiation Jacobian and under-relaxation factor ($\lambda_r = 0.6$)	209
5.131	Convergence histories for a pseudo two-phase pipe problem – <code>pamg</code> multiphase with automatic differentiation Jacobian and under-relaxation showing mesh independent convergence with under-relaxation	210
5.132	Single phase entry flow – Convergence histories with the <code>pamg</code> code for different (uniform) grid levels	212
5.133	Single phase T-junction – Convergence histories with the <code>pamg</code> code for different (adaptive) grid levels	213
5.134	Two-phase channel flow – Convergence of solution on a level 5 grid for two cycling strategies: the “W” and “F” cycles	214
5.135	Multiphase channel flow – Convergence history for 2 grid computations depending on the amount of under-relaxation	216
5.136	Multiphase channel flow – Convergence history for 3 grid computations depending on the amount of under-relaxation	216
5.137	Multiphase channel flow – Convergence history for 4 grid computations depending on the amount of under-relaxation	217
5.138	Two-phase channel flow – Comparison of the convergence histories of a level 3 computation for different number of relaxation sweeps after prolongation and restriction	220
5.139	Two-phase channel flow – Effect on the convergence history of a level 3 computation for different numbers of relaxation sweeps after prolongation and restriction	220
5.140	Two-phase channel flow – Convergence histories as a function of computational work for a level 3 computation and for different numbers of relaxation sweeps after prolongation and restriction	221
5.141	Two-phase channel flow – Comparison of the convergence histories of a level 5 computation for different numbers of relaxation sweeps after prolongation and restriction	221
5.142	Bilinear interpolation for the volume fractions for interior cells	223
5.143	Bilinear interpolation for the volume fractions for border cells	224
5.144	Two-phase channel flow problem – Convergence history on a level 4 grid according to the order of interpolation of the volume fraction correction to fine grids	226
5.145	Two-phase channel flow problem – Convergence history on a level 4 grid according to the number of fine grid relaxations sweeps $\nu_r = \nu_p$, for second order prolongation of the volume fraction corrections	226

5.146 Two-phase channel flow problem – Convergence history on finer grids for second order prolongation of the volume fraction corrections 227

5.147 Two-phase channel flow problem – Convergence history on a level 3 grid according to the order of interpolation of the volume fraction correction to fine grids 227

5.148 Two-phase channel flow problem – Convergence history on a level 3 grid according to the number of fine grid relaxations sweeps $\nu_r = \nu_p$, for second order prolongation of the volume fraction corrections 228

5.149 Two-phase channel flow problem – Convergence history on coarser grids for second order prolongation of the volume fraction corrections – $\nu_r = \nu_p = 2$. 228

5.150 Two-phase channel flow problem – Convergence history on coarser grids for second order prolongation of the volume fraction corrections – $\nu_r = \nu_p = 4$. 229

5.151 Two-phase channel flow problem – Comparison of the convergence history for finer and coarser grids with second order interpolation of the volume fractions and $\nu_r = \nu_p = 2$ 229

5.152 Convergence history when inter-phase momentum transfer terms are included 231

5.153 Two-phase channel flow problem – Volume fraction profiles along the line $x = 2.5$ for different grid levels when inter-phase momentum transfer terms are included 231

5.154 Two-phase flow through a T-junction – Volume fraction profile along the line $x = 6.5$ at different grid levels showing a “dry-out” zone, i.e. a complete separation of the flow which is more established at level 5 – See page 123 for the geometry of the problem 234

5.155 Two-phase T-junction problem – Convergence factors for multi-grid computations at different levels. See text for details concerning the irregularities in the convergence history at level 5 235

5.156 Multiphase backward-facing step problem ($Re_1 = 10$ and $Re_2 = 7.5$) – Convergence history on level 2 and 3 grids showing reasonable grid independence 235

5.157 Multiphase backward-facing step problem – Convergence history on a level 3 grid for $Re_1 = 100$ and $Re_2 = 75$ showing the divergence of the `pamg-multiphase` solver because both phases are recirculating and the problem is therefore ill-posed mathematically 236

5.158 Multiphase backward-facing step problem – Convergence history for single grid computation with $\Delta x = \Delta y = 0.625$, equivalent to a level 3 multigrid computation ($Re_1 = 100$ and $Re_2 = 75$) – Compare with Figure 5.157: the multigrid solver is approximately 50 times faster than the single grid solver 236

5.159 Multiphase backward-facing step problem at $Re_1 = 100$ and $Re_2 = 75$ – Streamlines for phase 1 – Compare with Figure 5.85 237

5.160 Multiphase backward-facing step problem at $Re_1 = 100$ and $Re_2 = 75$ – Streamlines for phase 2 – Compare with Figure 5.86 237

5.161 Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 1) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established recirculation region at high Reynolds numbers 237

5.162 Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 2) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing that a recirculation zone only occurs at higher Reynolds numbers 238

5.163	Multiphase backward-facing step problem – Volume fraction profiles (for phase 1) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established single phase regions behind the step at high Reynolds number	238
5.164	Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 1) along the line $x = 3.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established recirculation region at high Reynolds numbers	239
5.165	Multiphase backward-facing step problem – Volume fraction profiles (for phase 1) along the line $x = 3.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing in the top-left corner, a recirculating region where both phases are present	239
5.166	Multiphase backward-facing step problem – Horizontal velocity profiles (for both phases) along the line $x = 3.5$ for high Reynolds numbers ($Re_1 = 100$ and $Re_2 = 75$), showing the presence of two recirculation zones up to $y = -0.5$	240
5.167	Two-phase channel flow problem – Horizontal velocity profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1) . . .	244
5.168	Two-phase channel flow problem – Pressure profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1)	244
5.169	Two-phase channel flow problem – Volume fraction profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1)	245
5.170	Two-phase channel flow problem – Horizontal velocity profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1) . . .	245
5.171	Two-phase channel flow problem – Vertical velocity profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1)	246
5.172	Two-phase channel flow problem – Volume fraction profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1)	246
5.173	Two-phase channel flow problem – Convergence History on level 5 grids for different Reynolds Numbers and a first order interpolation for the fine grid volume fraction corrections	247
5.174	Two-phase channel flow problem – Convergence History on level 5 grids for different Reynolds Numbers and a second-order like interpolation for the fine grid volume fraction corrections	247
5.175	Two-phase channel flow problem – Convergence History for a flow at $Re = 1$, on different grid levels	248
5.176	Two-phase channel flow problem – Convergence history for a single grid computation for flows at $Re = 1$ and $Re = 100$ – The grid size is $\Delta x = \Delta y = 0.015625$ and the computation is equivalent to a level 5 multigrid computation.	248
5.177	Low Reynolds number flow through a T-junction – Volume fraction profile along the line $x = 6.5$ for different grid level showing an increased amount of diffusion (compare with Figure 5.81)	249
5.178	Low Reynolds number flow through a T-junction – convergence history for different grid levels	249
5.179	Multiphase flow through a T-junction – convergence history on a level 3 grid for different Reynolds numbers	250
5.180	Multiphase flow through a T-junction – convergence history on a level 4 grid for different Reynolds numbers	250

5.181	Two-phase channel flow – Extended computational domain – Horizontal velocity profile along the line $y = 0.5$	251
5.182	Two-phase channel flow – Extended computational domain – Volume fraction profiles along the line $y = 0.5$	252
5.183	Two-phase channel flow – Horizontal velocity profiles for phase 1 along the line $y = 0.5$ for $0 \leq x \leq 3$ for different length of the computational domain .	252
5.184	Two-phase channel flow – Volume fraction profiles for phase 1 along the line $y = 0.5$ for $0 \leq x \leq 3$ for different lengths of the computational domain	253
5.185	Two-phase pipe channel – Volume fraction profiles for phase 1 along the line $x = 2.5$ for computational domain of different lengths	253
5.186	Two-phase channel flow – Extended computational domain – Convergence history for various grid levels – Compare with Figure 5.124	254
5.187	Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1	259
5.188	Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 2	260
5.189	Two-phase channel flow – Comparison of pressure profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$	260
5.190	Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1	261
5.191	Two-phase channel flow – Comparison of horizontal velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1	261
5.192	Two-phase channel flow – Comparison of horizontal velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 2	262
5.193	Two-phase channel flow – Comparison of vertical velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1	262
5.194	Two-phase channel flow – Comparison of vertical velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 2	263
5.195	Two-phase channel flow – Comparison of volume fraction profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1	263
5.196	Two-phase channel flow – Comparison of volume fraction profiles along the line $x = 2.5$ for the partial diffusion model at different grid levels – Phase 1	264
5.197	Two-phase channel flow – Comparison of the convergence histories for the partial diffusion model at different grid levels	264
5.198	Two-phase channel flow – Comparison of the convergence histories for the full diffusion model at different grid levels	265
5.199	Two-phase channel flow – Comparison of the convergence histories for the partial diffusion model at different grid levels	265

5.200	Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 2 grid	266
5.201	Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 3 grid	266
5.202	Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	267
5.203	Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 5 grid	267
5.204	Multiphase T-junction problem – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	268
5.205	Multiphase T-junction problem – Comparison of volume fraction profiles along the line $y = 0.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	268
5.206	Multiphase T-junction problem – Comparison of volume fraction profiles along the line $x = 3.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	269
5.207	Multiphase T-junction problem – Comparison of vertical velocity profiles along the line $x = 6.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	269
5.208	Multiphase T-junction problem – Comparison of the volume fraction profiles along the line $x = 6.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid	270
5.209	Multiphase T-junction problem – Convergence history for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid, showing the stabilising effect of the extra diffusive terms	270
5.210	Pseudo two-phase entry flow – Horizontal velocity along $y = 0.5$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$	274
5.211	Pseudo two-phase entry flow – Horizontal velocity along $x = 0.15$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$	274
5.212	Pseudo two-phase entry flow – Horizontal velocity along $x = 0.15$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$	275
5.213	Pseudo two-phase entry flow – Convergence histories on a level 4 grid for different formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$	275
5.214	Pseudo two-phase channel flow with the partial formulation of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Convergence history for grid levels 2, 3, 4 and 5	276
5.215	Pseudo two-phase phase channel flow with the full formulation of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Convergence history for grid levels 2, 3, 4 and 5	276
5.216	Single phase pipe flow with simplified stress tensor – Convergence history for grid levels 3 and 5	277
6.1	Multiphase channel flow problem – Comparison of the vertical velocity profiles (for phase 1) along the line $x = 2.5$ obtained on a level 3 uniform grid, for different transfer operators between grids	286

6.2	Multiphase channel flow problem – Comparison of the volume fraction profiles (for phase 1) along the line $x = 2.5$ obtained on a level 3 uniform grid, for different transfer operators between grids, showing that the fine grid solution is independent from the transfer operators	286
6.3	Multiphase channel flow problem – Comparison of the convergence rates of computations on a level 3 uniform grid for different transfer operators between grids	287
6.4	Multiphase channel flow problem – Comparison of the horizontal velocity profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids	289
6.5	Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids showing the error caused by misplaced grid interfaces	290
6.6	Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids at different stages of the solution algorithm, showing that the development of solution jumps – Note the volume fraction “kink” near the inlet associated with a coarse/fine grid interface	291
6.7	Multiphase channel flow problem – Comparison of the vertical velocity profile (for phase 1) along the line $x = 0.15$ obtained on a (2,3) adaptive grid, for different transfer operators between grids	291
6.8	Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $x = 0.15$ obtained on a (2,3) adaptive grid, for different transfer operators between grids	292
6.9	Multiphase channel flow problem – Convergence histories for adaptive A(2,3) computations and for different transfer operators between grids – See text for details.	293
6.10	Multiphase channel flow problem – Horizontal velocity profile along the line $y = 0.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures	295
6.11	Multiphase channel flow problem – Volume fraction profile along the line $y = 0.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures	295
6.12	Multiphase channel flow problem – Volume fraction profile along the line $x = 2.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures – Note that the number of interpolation points is not related to the grid size	296
6.13	Multiphase channel flow problem – Convergence Rates for adaptive (3,4) computations with $\lambda_r = 0.8$	296
6.14	Multiphase channel flow problem – Convergence Rates for adaptive (3,4) computations with $\lambda_r = 0.6$	297
6.15	Refinement pattern based on truncation error alone	298
6.16	Refinement pattern based on truncation error plus forced refinement	299
6.17	Multiphase channel flow problem – Horizontal velocity profiles along the line $y = 0.9$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	299

6.18	Multiphase channel flow problem – Volume fraction profiles along the line $y = 0.9$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	300
6.19	Multiphase channel flow problem – Horizontal velocity profiles along the line $y = 0.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	300
6.20	Multiphase channel flow problem – Volume fraction profiles along the line $x = 0.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	301
6.21	Multiphase channel flow problem – Volume fraction profiles along the line $x = 2.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	301
6.22	Multiphase channel flow problem – Vertical velocity profiles along the line $x = 2.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels	302
6.23	Multiphase channel flow problem – Comparison of the volume fraction profiles along the line $x = 2.5$ for adaptive A(3,6) and A(3,5) pamg-multiphase computations with solutions obtained on level 5 uniform grids with different solvers (pamg-multiphase and CFX 4.1) – Note that the number of interpolation points is not related to the grid size	302
6.24	Multiphase channel flow problem – Comparison of convergence history for adaptive A(3,6) and adaptive A(3,5) pamg-multiphase computations with uniform grid equivalents	303
6.25	Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase I) on uniform and adaptive grids	307
6.26	Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase II) on uniform and adaptive grids	307
6.27	Two-phase flow through a T-junction – Volume fraction profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids	308
6.28	Two-phase flow through a T-junction – Horizontal velocity profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids	308
6.29	Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 6.5$ (Phase II) on uniform and adaptive grids	309
6.30	Two-phase flow through a T-junction – Volume fraction profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids	309
6.31	Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase I) on different adaptive grids	310
6.32	Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase II) on uniform and adaptive grids	310
6.33	Two-phase flow through a T-junction – Convergence histories for different adaptive grid computations and for $\gamma = 2$ – See text for explanations about the convergence history of the (3,6) computation.	311
6.34	Single phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.5$ on uniform and adaptive grids ($\gamma = 2.0$)	311
6.35	Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ on uniform and adaptive grids ($\gamma = 2.0$)	312
6.36	Two-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$)	312

6.37 Two-phase phase flow through a T-junction – Vertical velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids($\gamma = 1$) 313

6.38 Single-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ for different levels of uniform grids 313

6.39 Two-phase flow through a T-junction – Horizontal momentum profiles along the line $y = 0.05$ (phase 1) for different adaptive grids ($\gamma = 1$), plotted on a logarithmic scale 314

6.40 Two-phase flow through a T-junction – Horizontal momentum profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$) 314

6.41 Two-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 2$) 315

6.42 Single phase flow through a T-junction – Volume fraction profiles along the line $y = 0.05$ (phase 2) for different adaptive grids($\gamma = 2$) 316

6.43 Two-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$) – Compare with Figure 6.41 316

6.44 Single phase flow through a T-junction – Volume fraction profiles along the line $y = 0.05$ (phase 2) for different adaptive grids($\gamma = 1$) – Compare with Figure 6.42 317

6.45 Two-phase flow through a T-junction – Convergence rates for an A(3,6) computation (and intermediate computations) for $\gamma = 1.0$ with the “no-change” strategy for handling the fine grid volume fraction corrections – Compare with Figure 6.33 317

6.46 Two-phase flow through a T-junction – Convergence rates for an A(3,6) computation (and intermediate computations) for $\gamma = 1.0$ with the “cut-off” strategy for handling the fine grid volume fraction corrections – Compare with Figure 6.45 318

6.47 Numerical coupling introduced by the computation of horizontal mass fluxes through a computational cell 320

6.48 Uniform grid (level 3) for the channel problem 324

6.49 Adaptive (2,3) Grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$ 324

6.50 Adaptive (2,3) Grid for the multiphase channel problem – Conservative interpolation – $\gamma = 0.35$ 325

6.51 Adaptive (3,4) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$ 325

6.52 Adaptive (3,4) grid for the multiphase channel problem – Conservative interpolation – $\gamma = 0.35$ 325

6.53 Adaptive (3,5) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$ 325

6.54 Adaptive (3,6) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$ 326

6.55 Uniform grid (level 3) for the T-junction problem 327

6.56 Adaptive (3,4) grid for the T-junction problem – $\gamma = 2.0$ 327

6.57 Adaptive (3,5) grid for the T-junction problem – $\gamma = 2.0$ 327

6.58 Adaptive (3,6) grid for the T-junction problem – $\gamma = 2.0$ 328

6.59 Adaptive (3,6) grid for the T-junction problem – $\gamma = 1.0$ 328

6.60 Adaptive (3,4) grid for the single phase T-junction problem – $\gamma = 2.0$. . . 328

6.61	Adaptive (3,6) grid for the single phase T-junction problem – $\gamma = 2.0$. . .	328
6.62	Uniform grid (level 3) for the multiphase backward-facing step problem . .	329
6.63	Adaptive (2,4) grid for the single phase backward-facing step problem – $\gamma = 2.0$	329
A.1	One-dimensional pseudo-two-phase flow problem with $l = 4.9$ – Converged pressure field obtained with central differencing	342
A.2	One-dimensional pseudo-two-phase flow problem with $l = 5.1$ – Converged pressure field obtained with central differencing	342
A.3	One-dimensional pseudo-two-phase flow problem with $l = 5.1$ – Converged pressure field obtained with hybrid differencing	343
A.4	Velocity solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing	343
A.5	Pressure solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing	344
A.6	Volume fraction solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing	344
A.7	Converged pressure solution for the one-dimensional multiphase flow prob- lem – $l=4.0$ – Central differencing	345
A.8	Approximation of the pressure solution for the one-dimensional multiphase flow problem after 600 iterations – $l=4.1$ – Central differencing	345
A.9	Converged pressure solution for the one-dimensional multiphase flow prob- lem – $l=4.1$ – Hybrid differencing	346
A.10	One-dimensional multiphase flow problem – Multi-grid convergence rates for different grid levels	347
A.11	One-dimensional multiphase flow problem – Comparison of the convergence rates for multi-grid and single grid computations	348
B.1	Pseudo two-phase entry flow – Horizontal velocity along $y = 0.5$ for different finest grid levels	349
B.2	Pseudo two-phase entry flow – Pressure along $y = 0.5$ for different finest grid levels	350
B.3	Pseudo two-phase entry flow – Horizontal velocity along $y = 0.1$ for different finest grid levels	350
B.4	Pseudo two-phase entry flow – Vertical velocity along $y = 0.1$ for different finest grid levels	351
B.5	Pseudo two-phase entry flow – Vertical velocity along $x = 0.15$ for different finest grid levels	351
B.6	Pseudo two-phase entry flow – Horizontal velocity along $x = 2.5$ for different finest grid levels	352
B.7	Pseudo two-phase entry flow – Vertical velocity along $x = 2.5$ for different finest grid levels	352

List of Tables

4.1	Simple fortran90 program illustrating the use of the AD01 package for automatic differentiation	100
5.1	Two-phase channel flow – Estimation of the truncation error for different grid levels	189
5.2	Two-phase T-junction problem – Estimation of the truncation error for different grid levels	189
5.3	Two-phase backward-facing step problem – Estimation of the truncation error for different grid levels	189
5.4	Two-phase channel flow – Estimation of the (point-wise) truncation error for different grid levels near the median line of the channel – The use of a staggered grid prevents the estimation to be done at the same point on different grids	189
5.5	Average convergence factors for the two phase channel flow problem	201
5.6	Average convergence factors for the two phase channel flow, depending on the value of the average residual	202
5.7	Average convergence factor for the two phase T-junction problem	202
5.8	Single phase pipe flow – Exact (automatic differentiation) Jacobian for a level 3 cell ($x = 2.036125 \pm 0.03125$, $y = 0.5036125 \pm 0.03125$) – See Section 4.2.1 for the arrangement of equations and unknowns in the Jacobian	206
5.9	Single phase pipe flow – Approximate Jacobian for a level 3 cell ($x = 2.036125 \pm 0.03125$, $y = 0.5036125 \pm 0.03125$) – See Section 4.2.1 for the arrangement of equations and unknowns in the Jacobian	207
5.10	Single phase entry flow – Average convergence factor for pamg computations on different grid levels	212
5.11	Two-phase channel flow problem – Asymptotic convergence factors for different finest levels and for second order prolongation of the volume fraction corrections	225
5.12	Multiphase backward-facing step problem – Comparison of convergence factors on level 3 and 4 grids for low Reynolds numbers	233
5.13	Multigrid convergence factors for the two phase flow through a T-junction at lower Reynolds numbers	243
5.14	Multiphase channel flow problem – Comparison of global and asymptotic convergence factors for computations on different finest levels for equations with full and partial stress tensor flux $\nabla \cdot (r\mathbf{T})$ and for $Re = 100$	257
5.15	Comparison of the <i>average</i> and <i>maximum</i> condition numbers of the Newton correction systems on different grid levels, and for computations on different finest levels, with the partial stress tensor flux $\nabla \cdot (r\mathbf{T})$	258

5.16	Comparison of the maximum modulus of the vertical velocity for phase 1 along the line $x = 0.15$ on level 4 grids, for different system of equations . . .	272
5.17	Asymptotic convergence factor for the single phase entry flow problem on level 4 grids – Comparison of different system of equations.	272
5.18	Pseudo-two-phase (parabolic) channel flow problem – Comparison of asymptotic convergence factors for different formulations of the stress tensor flux $\nabla \cdot (rT)$	272
5.19	Single phase channel flow problem – Comparison of asymptotic convergence factors for different formulations of the diffusive flux	273
6.1	Multiphase flow through a T-junction – Estimates of the Condition numbers at different levels for an A(3,6) adaptive computation with $\gamma = 2$	306
6.2	Comparison of computational work required to solve the channel problem and the T-junction problem on adaptive and uniform grids	321
6.3	Comparison of memory required to solve the channel problem and the T-junction problem on adaptive and uniform grids	321
A.1	One-dimensional multiphase flow problem – Convergence factors at different grid levels	347

Notation

\mathbf{v}	Velocity vector
\hat{u}	Horizontal component of the velocity vector
v	Vertical component of the velocity vector
p	Pressure
r	Volume fraction
q^u	Horizontal mass flux
q^v	Vertical mass flux
\mathbf{T}	Viscous stress tensor
$c_{\alpha\beta}$	Inter-phase momentum transfer coefficient
C_D	Drag coefficient
$d_{\alpha\beta}$	Inter-facial length
M_α	Inter-phase momentum transfer term
Re	Reynolds number
Re_c	Cell-Reynolds number
T	Temperature
Q	Heat flux
P	Peclet number
P_c	Cell-Peclet number
M	Number of phases
N	Number of discrete unknowns in a system
\mathcal{L}	Continuous differential operator
\mathcal{L}^h or \mathcal{L}^k	Discrete operator
\mathbf{u}	Generic vector
$\mathbf{u}^h, \mathbf{u}^h$ or \mathbf{u}^k	Exact solution of a discrete operator
$\tilde{\mathbf{u}}^h, \tilde{\mathbf{u}}^h$ or $\tilde{\mathbf{u}}^k$	Approximate solution of a discrete operator
\mathbf{e}^h	Error vector
\mathbf{r}^h	Residual vector
\mathbf{w}	Eigenvector of a matrix
\mathbf{I}	Identity tensor
I	Identity matrix
I_k^{k-1}	Restriction operator for the residual
\hat{I}_k^{k-1}	Restriction operator for the solution
I_{k-1}^k	Prolongation operator for the corrections to the solution
J	Jacobian matrix
L_2	Vector norm ($\sqrt{\sum x_i^2}$)
X_α	Phase function (see page 14)

Greek letters:

Δx	Horizontal mesh size
Δy	Vertical mesh size
ξ	Non-dimensional length
$\Delta \xi$	Non-dimensional mesh size
ρ	Density
μ	Viscosity
ν	Kinematic viscosity
ν_r	Number of relaxation sweeps before restriction
ν_p	Number of relaxation sweeps after prolongation
ν_c	Number of relaxation sweeps on the coarsest grid
λ_r	Under-relaxation factor
λ	Eigenvalue of a matrix
κ	Finest grid in a multigrid computation
Ω^h or Ω^k	Grid identifier
γ	Refinement parameter
τ^k	Truncation error on the level k grid
τ_k^{k-1}	Defect on the level $k - 1$ grid
Φ	Discrete unknowns for the local quasi-Newton solver
$\Delta \Phi$	Newton corrections
Γ_α	Inter-phase mass transfer
Δp	SIMPLE-type pressure correction

Subscripts:

α	Phase identifier
int	Inter-facial quantity
i	Horizontal index for a two-dimensional grid function
j	Vertical index for a two-dimensional grid function

Superscripts:

h	Generic mesh size
k	Grid level in a multigrid computation
T	Transpose operation
(c)	Coarse grid value
(f)	Fine grid value
(n)	Iteration number for an iterative process
u	Relative to the horizontal momentum equation
v	Relative to the horizontal momentum equation
c	Relative to the continuity equation
$*$	Approximate value
(cor)	Corrected value

Other symbols:

$[\bullet]_x$	Partial derivative with respect to the variable x
$[\bullet]_y$	Partial derivative with respect to the variable y
$[\bullet]_t$	Partial derivative with respect to time
$\ \bullet\ $	Vector norm
$ \bullet $	Jump condition
$\tilde{\bullet}$	Phasic averaged quantity
$\hat{\bullet}$	Mass averaged quantity
∇	Gradient operator
$\nabla\cdot$	Divergence operator
\otimes	Vector product ($\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T$) where \mathbf{u} and \mathbf{v} are column vectors

Chapter 1

Introduction

The physics of fluid flow covers a large range of phenomena. It involves a wide range of length and time scales. Its proper understanding is crucial for many applications in science and engineering. Hence, a large number of models of varying complexity have been developed. Very often, the phenomena of interest are described by governing equations which are partial differential equations. The numerical solution of such equations is difficult since it can require considerable experience and lies at a meeting point between at least three distinct disciplines: physics (e.g. [2]), mathematics (e.g. [3]) and numerical analysis (e.g. [4]). This topic has received a great deal of attention since the early 50s and resulted in the development of a new and very active field, *Computational Fluid Dynamics* (CFD) (see, for instance, references [5, 6, 7]). Currently, many fundamental issues can be considered solved. However, many questions are still outstanding and progress is ongoing. As more computing power becomes available, problems of increasing complexity are being investigated.

From the point of view of this thesis, a fundamental division exists between single phase and multiphase flows. In the first case, a single substance is flowing. This may be a pure element or an homogeneous mixture such as air. In the second case, several fluids or conceptually similar materials are flowing at the same time. Once this second possibility is considered, the physics becomes extremely complex. In any study of multiphase phenomena, a thorough understanding must be gained not only of the behaviour of each fluid but also of their *interactions*. This will often involve many physical mechanisms. A further difficulty arises since multiphase flows occur in a wide range of processes, both natural and industrial, from soil erosion to oil transport through pipelines. A unified methodology for their study is therefore highly desirable but difficult to define.

The CFD of single phase flow is very advanced. In cases where the physics leads to relatively simple models, such as inviscid computations or laminar viscous flow, the technology is sufficiently developed to provide industry with real design tools. Means to deal with more complex problems such as turbulence modelling or com-

bustion are available but still require powerful computers. Single phase CFD may now be considered as mature technology on two accounts: firstly, the fundamental technology is now well established and secondly, general purpose commercial codes are widely available.

The situation regarding multiphase flows is very different. Although solution algorithms have been available for some time, it is broadly acknowledged that the performance of present multiphase solvers is generally disappointing and prevents the detailed study of many flows of industrial interest.

The majority view among researchers and practitioners is that the problem is mainly one of modelling. Whilst the mathematical models for single phase flows embodied in the framework of the Navier-Stokes equations are well established, the formulation of correct multi-phase models, even for specific applications, is still an open question. Since the mid-seventies, a general model — the *multi-fluid equations* [8] — has been available. In the same way as some turbulence models are derived, these equations are obtained *rigorously* by averaging. Nevertheless, this model has raised a large number of mathematical issues.

It is well known that the equations for the unsteady case are *ill-posed* [9] in a mathematical sense. This is best defined by reference to the opposite property, that of well-posedness. A well-posed problem is one for which (i) a solution exists, (ii) is (locally in solution space) unique and (iii) depends continuously on boundary conditions and other conditions. In the case of *transient* multiphase flows, solutions do not depend continuously on boundary conditions because error growth is made possible by the appearance of complex eigenvalues. This very unusual characteristic suggests that the basic model is fundamentally flawed, either due to simplifications and approximation, or because important physical factors are ignored altogether. It is very often argued that if the physics were correctly modelled, the corresponding equations would be well-posed and their solution would not raise difficulties. We restrict ourselves to the steady problem, for which this issue does not arise.

By contrast, in the present thesis, the underlying opinion implicitly expressed is that the numerical simulation of multiphase flow is very challenging even for correct models. It is acknowledged without any reservations that progress on the modelling front — both from theoretical and experimental viewpoints — is crucial for the development of algorithms suitable for industrial multiphase CFD. However, in the author's opinion, given a “good” — i.e. well-posed — system of governing equations, obtaining a “good” solver is not a trivial task. A “good” solver should be:

- *Robust*, so that it successfully simulates a wide range of flows without relying on user expertise and/or experimentation because the domain of convergence is good;
- *Efficient*, so that the computational cost of the solution is proportional to the amount of physical change in the solution, in keeping with the so-called golden

rule [10];

- *Accurate*, so that the discrete solution is as close to the solution of the continuous equations as computational cost will allow;
- *Self-adaptive*, so that the information on the solution obtained as the algorithm proceeds can be used to direct the computational effort *without external intervention* from the user.

For single phase flows, one such “good” solver can be constructed using quasi-Newton coupled solvers associated with multigrid methods. This class of methods has been developed during the last fifteen years to correct a fundamental weakness of traditional solution algorithms for fluid flows. The local coupling between the flow variables is not treated accurately because each equation in the system is usually iterated independently from the others. Newton solvers are a possible alternative. In this case, the global system of non-linear equations is solved by Newton’s method so that the coupling between the variables is now handled in a very accurate way. However, large linear systems are generated so that the computational cost can be heavy unless the sparseness of the Jacobian matrices can be exploited. *Even if the sparseness of the Jacobian is exploited to the full, these methods require a large amount of computing power.* This variant is sometimes referred to as full quasi-Newton (FQN). By contrast, in local quasi-Newton solvers (LQN), the global non-linear system is solved block by block so that only *local* Jacobians are needed. The correction systems are very small and can be solved using direct methods such as Gaussian elimination. By construction, LQN solvers concentrate on the local coupling and neglect the global coupling. Evidence accumulated to this day suggests that it is the former which is crucial for rapid solution algorithms [1]. LQN are often combined with multigrid methods which provide very significant acceleration. Multigrid complements the quasi-Newton solver by quickly resolving the long wavelengths. Alternatively, one can say that the multigrid method deals with the global coupling. Multigrid can be seen as an acceleration technique but this viewpoint can also be reversed: the LQN solver is then seen as an efficient smoother for the multigrid method.

Quasi-Newton coupled multigrid solvers have a very interesting property in that they are optimal order-wise: their complexity is $\mathcal{O}(N)$ where N is the number of discrete unknowns. The reason for this is that convergence rates are often grid-independent.

The variant of multigrid implemented is the *Full Approximation Storage* (FAS) method. Although relatively rarely used, it has two advantages: firstly, it is directly applicable to non-linear problems and secondly, it automatically provides estimates for the truncation error of the solution. This information can be used to refine grids automatically where appropriate. Hence, automatic error control can be achieved. Furthermore, FAS and more generally multi-level methods, provide an attractive framework for adaptive computations: adaption can be easily implemented by prolongation on sub-domains.

The methodology outlined here has been implemented in a computer code called *pamg*, which is described in reference [1, 11, 12] (see also [13, 11]). This code solves the two-dimensional Navier-Stokes equations for viscous incompressible fluids for Cartesian geometries. *pamg* has been thoroughly validated and is highly efficient and robust.

At the heart of the present thesis is the conviction that a similar approach could lead to a multiphase solver which is equally robust and efficient. This can be argued by considering that for multiphase flows the local coupling between variables is more complex than in the single phase case and crucial to the success of a numerical algorithm.

Given this general outlook, our main goal was to develop and apply adaptive local quasi-Newton coupled multigrid (ALQNMG) solvers to multiphase flows and investigate the robustness and efficiency of the resulting solver. Starting from the single phase code *pamg*, the expected end product was another code, *pamg-multiphase*, offering the same level of algorithm development for steady two dimensional flows. New issues are: (i) the system of equations is much larger and very strongly non-linear and its properties are not completely understood and (ii) the set of physically acceptable solutions is more restricted due to constraints which exists on the realisability of some unknowns.

In Chapter Two the thesis rationale is given, the physical modelling and numerical simulation of multiphase flows are briefly examined and the mathematical basis for the different numerical techniques implemented in *pamg* and *pamg-multiphase* are given. Since this field of research is very large and much information is sensitive, this literature survey does not aim to be exhaustive. Rather, it seeks to highlight the most important issues arising in the simulation of multiphase flows and give technical arguments to explain why the ALQNMG approach performs well as a solver for fluid flows.

Chapter Three deals with the problem of discretisation. Both the governing partial differential equations and the discrete algebraic equations solved by *pamg* and *pamg-multiphase* are considered, the latter being derived using a finite volume approach. Important features of the discretisation, the use of staggered grids and hybrid schemes, are then discussed and the discretisation process is described in detail. The implications of this process for the performance of the solution algorithm, namely the robustness of the solver, are investigated. Furthermore, specific characteristics of the multiphase equations — namely the presence of cross derivative terms in the diffusive fluxes — which could have a bearing on the convergence rates of the algorithms have been observed and are discussed.

Chapter Four describes the solution algorithms implemented in *pamg* and *pamg-multiphase*. We are mainly concerned with the numerical and computational developments which have been necessary in order to extend the single phase solver to multiphase flows. The single phase solution algorithm is described, in order to provide a reference

point for the subsequent discussion of the multiphase modifications. The design of the multiphase algorithm has required the careful consideration of many issues. These include:

- the derivation of consistent discretisations;
- the treatment of the non-linearity by line-searching;
- the use of automatic differentiation to obtain accurate local Jacobians;
- the application of multigrid transfers which only generate physically realisable approximations (in sharp contrast to single phase flows, some variables used in the description of multiphase flows are restricted to a certain range);
- the design of conservative interpolation operators in connection with adaption.

To the author's knowledge, these solutions are entirely original in this context.

Chapter Five seeks to establish the correctness of *pamg-multiphase* in the sense that it gives correct solutions for some test flow problems of varying degrees of complexity. Comparisons were drawn not with experiments but with two thoroughly validated codes: a commercial CFD package CFX 4.1 and, where appropriate, the single phase version of *pamg*. This can be considered as a good first stage for validation and is consistent with the numerical emphasis of the present work. Validation against experimental data may have been useful but would have required the consideration of modelling errors which are not in the scope of this thesis. In a second part, the performance of *pamg-multiphase* is discussed in detail. The main conclusions are that the LQN solver is robust and that multi-gridding significantly accelerates the computations. Of course, the level of performance is not as high as for single phase flows. In particular, it is more difficult to achieve grid independent convergence rates. As the grid size become finer, convergence rates are often degraded, at least in parts of the convergence histories.

This is not altogether surprising given the complexity of the equations solved. We investigate in detail the factors which may influence the convergence rates of the method. These include:

- the impact of multiphase source terms;
- the influence of characteristic (and difficult) multiphase flow patterns, particularly regions where the phases are totally separated;
- the presence of singularities in re-circulation zones;
- the effect of diffusive effects present in the governing equations, particularly the impact of terms specific to multiphase flows

- the selection of the algorithm parameters
- the accuracy of the grid transfers.

It is the author's opinion that the root cause of the dependency of the convergence rates on the grid size reflects the fact that the governing equations are highly non-linear. In non-linear problems, it is necessary to approximate the discrete operator as well as its solution. This can require significant computational effort particularly if the operators on different grid are very different. Secondly, it is shown that some multiphase diffusive terms are not efficiently handled (in relative terms) by the multigrid procedure. It remains that even when it is not optimal order-wise, the `pamg-multiphase` solver is still very efficient and much faster than single grid methods. This mirrors the situation for hyperbolic equations: multigrid methods in their basic forms are successful acceleration techniques but they are not optimal solver order-wise [14].

Finally, Chapter Six deals with computations on adaptive grids as opposed to the uniform grids used in Chapter Five. The distinctive feature here is that adaption is performed *automatically* and *dynamically* as the computation proceeds. Besides reducing the computational cost, this type of grid refinement allows good error control. The refinement algorithm is finely intertwined with the FAS method: the latter provides estimates of the truncation error very cheaply while the former exerts an influence on the definition of the multigrid transfer operators. It is shown that for multiphase flows, adaptive computations are very beneficial in terms of performance without significant loss of accuracy even if the transfers do *not* conserve numerical fluxes.

The present thesis has been written with three principal objectives: firstly, to present our new solution algorithm, secondly to demonstrate that `pamg-multiphase` is an essentially correct implementation of a FAS multigrid method for the steady multi-fluid equations in two dimensions and thirdly to investigate the potential of such an approach in providing efficient, robust, accurate and adaptive simulations of multiphase flows. The presentation of numerical results thus has a large relative weight and actually gives a rather distorted image of the work which has been carried out during the course of this project. The most delicate aspect, by far has been the development of the software, as is to be expected for any numerical work of this type. The extension of the single phase code to multiphase problems was performed in, it is hoped, a careful way but the fact remains that `pamg-multiphase`, like most numerical software, has proved difficult to write. In the author's opinion, this is mainly due to the combination of a set of equations (both differential and discrete) of relatively unknown properties with numerical algorithms – multigrid methods – whose implementation, in all its detail, is quite complex. However, the experience gained from writing such a piece of software has been very valuable.

To conclude then, the following quotation from reference [15, pages 224–225] is an effective summary of the author's struggle during the course of the project:

A [...] numerical analyst tries a multigrid solver on a new problem. He knows the basics, he has seen it implemented on another problem, so he has no trouble writing the program. He gets results showing a certain rate of convergence, perhaps improving a former rate obtained with a one-grid program. Now he is confronted with the question: Is this real multigrid efficiency? Or is it many times slower, due to some conceptual error or programming bug? The algorithm has many parts and aspects [...]. A single error (a wrong scheme or a bug) in any of these parts may degrade the whole performance very much, but is still likely to give an improvement over a one-grid method, misleading the analyst to believe that he has done a good job. How can the error be suspected and detected? How can one distinguish between various possible troubles?

Chapter 2

A review of Relevant Literature

2.1 Introduction and Thesis Rationale

The study of multiphase flows has attracted a lot of attention from theoretical, experimental and numerical standpoints and, consequently, the body of literature pertaining to this field is very large. Numerical simulation has been attempted for a long time, at least from the mid-sixties (see, for instance, [16] and the reviews [9, 17]). References [18, 19, 20, 21, 22, 23, 24, 25] are example of more recent numerical work. Many studies are specific to particular applications (e.g. Fluidized Beds [26], Nuclear Reactor safety analysis [27, 28, 29, 30], Oil Pipelines [31, 32, 33], Particulate Flows [34, 35] or even Geology [36]).

In Section 2.2, different multiphase models are reviewed: we discuss the multi-fluid model, illustrate its derivation and mention some of its limiting cases. In keeping with our stated aim of designing a robust and efficient solver, we only raise modelling issues to the extent that they influence solution strategies. Modelling questions are obviously very varied and they are of crucial importance to the study of any multiphase flows but they are not within our scope here.

Next, the numerical simulation of multiphase flows is discussed in Section 2.3. The challenges it poses are illustrated, particularly those in transient flows. Furthermore, examples of widely used solution algorithms for multiphase flows — namely the IPSA and SETS methods — are described. These procedures have been chosen as representative of the many schemes available.

Having discussed the governing equations, we then present the mathematical foundations of the different numerical techniques we propose to use for their solution. The objectives set for the numerical work are as follows:

- (i) to demonstrate the applicability of quasi-Newton coupled multigrid (FAS)

solvers to multiphase systems, and more particularly, the multi-fluid model.

- (ii) to quantify as much as possible the performance benefits in terms of efficiency and robustness that such solvers bring.
- (iii) to implement *automatic* space adaption in order to make efficient use of computer resources and allow error control.

This is a direct extension of the numerous studies which have been conducted for single phase Navier-Stokes flows. In particular, the current work builds on the work presented in references [1, 11, 12]. The coupled approach to the simulation of fluid flow is discussed in Section 2.4. In a *coupled solver*, the flow equations are solved simultaneously while in a *segregated* approach, of which the SIMPLE algorithm is a prototype, they are iterated individually. Coupled solvers are faster because the non-linear coupling between the unknowns is accurately treated, but they are usually more complex to implement. Local quasi-Newton coupled solvers (LQN) are a particular variant of coupled solvers. They are based on an accurate treatment of the flow variables *locally* rather than globally. This is achieved by a block solution of full Newton system written on sub-domains, usually single computational cells.

In the author's view, it is not only possible but also advantageous to apply a (quasi-Newton) coupled approach to multiphase flows. The main argument is that the coupling between unknowns is tighter and more non-linear than in the case of single phase flows. Taking this *local* coupling into account may therefore be crucial for a robust and efficient numerical algorithm. As in the single phase case, embedding the coupled solver in a multigrid method should then lead to a very efficient solver since global as well as local coupling will be taken into account. Obviously, the success of multigrid depends on (i) how well the LQN solver will perform as a smoother and (ii) how much the discretised equations are amenable to multigrid solutions. Section 2.5 reviews the fundamental concepts of multigrid methods: the limitations of traditional iterative methods are examined, the basic ideas of multigrid methods — seen as acceleration techniques — are presented. Both the Correction Storage and Full Approximation Storage schemes are defined, as well as the different cycling strategies. Finally, the issue of multigrid grid convergence rates is investigated together with the limitations of multigrid methods for non-elliptic equations.

It seems logical and prudent to follow the methodology used for single phase flows and first to consider steady systems. For many single phase flows, a steady state solution encapsulates most of the physics relevant to the flows. By contrast, many multiphase flows of interest are unsteady by nature. Transient phenomena may play a vital part in determining subsequent flow patterns and there are many problems where a steady state solution does not exist or is unstable (e.g. terrain-induced slugging in a multiphase pipeline). In some cases, steady state problems may not even be well-posed in a mathematical sense. If the flow admits recirculation zones, several distinct solutions are possible. By definition, there is no mass transfer between the main flow and the recirculation zones so that it is not possible to fix the

mass fraction of the different phases in the recirculation zone. This a phenomenon which is specific to multiphase flows. It does not occur for single phase flows. We will return to this as it affects some of our results.

It is the author's view that the steady state solver developed in the course of this project could be directly extended to certain important unsteady flows where implicit schemes are a competitive alternative to explicit formulations. Indeed research work is currently undertaken in AMAC with this very aim in mind (see Reference [37]). The design of an efficient steady state solver can actually be considered to be significantly harder than a transient solver, even if implicit schemes are used. This is due to the fact that attempting a direct steady state solution is akin to performing a transient computation with a time step of infinite length. In a proper steady state solver, the final solution is usually very different from the initial guess whereas the solutions provided by a implicit time marching procedure at two consecutive time steps are much closer to each other. If *pang-multiphase* was extended to unsteady problems, a further implication of this last comment is that the multigrid part of the algorithm may not provide large acceleration if long wave lengths do not change appreciably between time-steps, even if coarse grids are only occasionally visited.

Due to the intrinsically transient nature of many multiphase phenomena, most – if not all – of the published numerical work on multiphase systems deals with the simulation of *transient* flows. In this context, many issues arise. These will be briefly reviewed in Section 2.3 but are far less important for steady simulations. Crucially, the ill-conditioning of the transient multi-fluid equations is not a cause for concern.

The approach outlined above is believed to be entirely original. Indeed, only one study combining multiphase flow solutions and a multigrid method was found: Reference [38] (but see also [23]). In this paper however, the equations are those of the homogeneous models, a simplification of the multi-fluid model. Furthermore, the multigrid method is limited to solving a Poisson equation for the pressure. Numerical techniques are more developed for multiphase flows through porous media. In this case, however, the applicable equation, the so-called d'Arcy's equation [39, 40] is considerably simpler because convective effects are ignored.

Finally, Section 2.6 is devoted to space adaptation in the solution of PDEs. Fundamentally, the basic concept is to vary the grid size throughout the domain in order to resolve each feature of the solution with a sufficient, but not excessive, level of precision. As mentioned previously, this is a very important functionality since a robust adaptive algorithm will very significantly reduce the requirements in memory and computing time for a given accuracy. In this context, the term "robust" refers to an algorithm possessing capabilities for automatic grid refinement in sensible regions. In this way, truncation errors can be controlled and the need for user input and expertise can be minimised. The FAS multigrid method will here be used as a corner-stone to provide accurate refinement estimators based on the truncation error and to minimise the cost of fine grid computations. This again significantly extends the single phase methodology presented in [1], as issues pertaining to the

conservation of mass fluxes arise.

2.2 Basic Multiphase Flow Models

2.2.1 The Multi-Fluid Model

The physics of multiphase flows is covered in detail by many text books and monographs (for instance [41, 42, 43]). Multiphase flows occur in a very wide variety of forms but it is a basic assumption in this field that there is a large degree of commonality between different types of multiphase flows.

This assumption is obviously inspired by the example of single phase flow. These are governed by the system of Navier-Stokes equations (see for instance [44]) which express the conservation of mass, momentum and energy in an Eulerian framework i.e. based on the concept of control volume. It is the availability of this general purpose set of equations which, in the author's view, has allowed much of the progress achieved in the understanding of fluid flows.

The *multi-fluid* model is a largely successful attempt at providing a general (Eulerian) framework for the mathematical description of multiphase flows. Similarly to the the system of the Navier-Stokes equations, the multi-fluid equations express the conservation of mass, momentum and energy using the principle of continuum mechanics. Hence, the multi-fluid equations are not a complete description of a multiphase flow. They need to be supplemented by suitable constitutive relationships which govern the way the phases interact with each other as well as with themselves. This is a crucial and difficult question in multiphase modelling since the number of these constitutive relationships is large and their complexity increased, compared with the single phase case.

Although the multi-fluid model is derived in a rigorous way, its correctness is not absolutely established. This is due to the approximations and simplifications made to obtain a workable set of equations.

In the basic model, it is assumed that each phase can be described as a continuum, and therefore governed by partial differential equations expressing the conservation of mass, momentum and energy for single phase flows i.e. the Navier-Stokes equations. The phases are separated by *interfaces* at which conservation is also to be enforced using *jump conditions* similar to those imposed at shocks in aerodynamics. These are based on interface velocities which they actually define. Note that at a given time, at each point which is not on the interface, only *one* set of governing equations applies, depending on the phase which "occupies" the point at that time.

This is referred to as the *microscopic* level of description. In many cases, it provides

more information than is necessary to correctly model the physics and it is almost always too complex for numerical simulation.

The multi-fluid equations are then derived by averaging the microscopic equations. Therefore they provide a macroscopic level of description which ignores many details of the flow but is simple enough to be handled numerically. Averaging introduces the concept of *inter-penetrating continua*: all phases are assumed to be present at each location. Consequently, conservation laws for mass, momentum and energy are derived for each phase and for all points. The multi-fluid model introduces an extra dependent variable to characterise each phase: the *volume fraction*, defined as the proportion of volume occupied around each point by a given phase. The volume fraction arises naturally from the averaging process.

Another approach to multiphase modelling is to consider the motion of the mixture and to *postulate* independent balance equations for the mixture. This model [43, 9] appears to be most suited to dispersed flows.

The multi-fluid equations (see equations 2.12 to 2.14) are superficially similar to the Navier-Stokes equations: for each phase, conservation equations for mass, momentum and energy are written. Convective and diffusive terms have comparable structures. However, the multi-fluid equations are mathematically much more complex. They are characterised by:

- a high degree of non-linearity, since the volume fraction is introduced as a factor in every term;
- the presence of inter-phase transfer terms;
- the existence of constraints on the set of physically realisable solutions (the volume fraction must be between 0 and 1).

An inspiration for the multi-fluid methodology is provided by the averaging approach to turbulence (see [45]), in which the so-called Reynolds stresses are introduced to model the turbulent energy. The standard reference dealing with the multi-fluid model is [8] but many shorter papers are also available [46, 29, 30, 47, 48]. The review [17] provides the main results while reference [9] includes a derivation of the one-dimensional multi-fluid equations for a simple stratified flow. Finally, the review [49] provides an elegant derivation and discusses many issues relevant to the mathematical modelling of multi-phase flows.

2.2.2 Derivation of the Multi-Fluid Equations for the Two-Phase Flow of Incompressible Fluids

Following [49], the main stages of the derivation of the multi-fluid model are given here. The present study is concerned with incompressible fluids so we do not consider

the conservation of energy but rather focus our attention on the conservation of mass and momentum. Furthermore, the derivation is given for a two phase flow but can be readily extended to multiphase flows.

At a microscopic level of description, a multiphase flow is a set of mutually exclusive regions in which a single phase flow occurs. These regions are separated by interfaces. A domain is considered and single phase governing equations are assumed to apply at each point (although the flowing fluids may be different at different points). This is true except at interfaces where jump conditions are applied:

The (single phase) equations of motion are:

- Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = 0 \quad (2.1)$$

- Conservation of linear momentum ¹:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho (\mathbf{v} \otimes \mathbf{v}) = \nabla \cdot \mathbf{T} + \rho \mathbf{f} \quad (2.2)$$

- Conservation of angular momentum:

$$\mathbf{T} = \mathbf{T}^T \quad (2.3)$$

At phase interfaces, the following jump conditions are applied

$$|\rho(\mathbf{v} - \mathbf{v}_{int})| = 0 \quad (2.4)$$

and

$$|\rho(\mathbf{v} - \mathbf{v}_{int}) \cdot \mathbf{n} - \mathbf{T} \cdot \mathbf{n}| = \sigma \kappa \mathbf{n} \quad (2.5)$$

for the conservation of mass and momentum respectively.

$|\bullet|$ denotes the jump across the interface, \mathbf{v}_i is the velocity of the interface, σ is the surface tension, κ the average curvature of the interface, and \mathbf{n} is the unit normal, oriented in such a way that if $|f| = f^\beta - f^\alpha$ then \mathbf{n} points from phase α to phase β . \mathbf{T} is the stress tensor and must be supplied by an appropriate constitutive relationship.

In order to perform the averaging, the phase function $X_\alpha(\mathbf{x}, t)$ is introduced and defined as follows:

$$X_\alpha(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is in phase } \alpha \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

Finally, let $\langle \rangle$ denote an averaging process. Commonly used averaging operators are:

¹The vector operation $\mathbf{u} \otimes \mathbf{v}$ denotes the product $\mathbf{u}\mathbf{v}^T$, where \mathbf{u} and \mathbf{v} are column vectors.

- Time average:

$$\langle f \rangle (\mathbf{x}, t) = \frac{1}{T} \int_{t-T}^t f(\mathbf{x}, t') dt'$$

- Space average:

$$\langle f \rangle (\mathbf{x}, t) = \int \int \int_{R^3} f(\mathbf{x}', t) d\mathbf{x}'$$

- Ensemble average:

$$\langle f \rangle (\mathbf{x}, t) = \frac{1}{N} \sum_{n=1}^N f_n(\mathbf{x}, t)$$

where f_n denotes a realisation of f over a set of possible equivalent realisations.

The averaging operators are assumed to satisfy the following properties [49, 50, 45]:

- Reynolds' rules:

$$\begin{aligned} \langle f + g \rangle &= \langle f \rangle + \langle g \rangle \\ \langle \langle f \rangle g \rangle &= \langle f \rangle \langle g \rangle \\ \langle c \rangle &= c \end{aligned}$$

where c is constant

- Leibnitz's rule:

$$\left\langle \frac{\partial f}{\partial t} \right\rangle = \frac{\partial}{\partial t} \langle f \rangle$$

- Gauss' rule:

$$\left\langle \frac{\partial f}{\partial x_i} \right\rangle = \frac{\partial}{\partial x_i} \langle f \rangle$$

The phase function will be treated as a *generalised function* [51]. It can then be shown that, in the sense of generalised functions,

$$\frac{\partial X_\alpha}{\partial t} + \mathbf{v}_{int} \nabla X_\alpha = 0 \quad (2.7)$$

∇X_α is also a generalised function which is zero everywhere except at interfaces where its "value" is related to the average inter-facial area per unit volume.

In order to derive averaged equations for the motion of each phase, the balance equations (2.1) and (2.2) are multiplied by the appropriate phase function X_α . Averaging is performed on the product and results in the multi-fluid governing equation for phase α . For instance, for the conservation of mass, the procedure is as follows. Start with:

$$X_\alpha \frac{\partial \rho}{\partial t} + X_\alpha \nabla \cdot \rho \mathbf{v} = 0,$$

since:

$$\begin{aligned} X_\alpha \frac{\partial \rho}{\partial t} &= \frac{\partial X_\alpha \rho}{\partial t} - \rho \frac{\partial X_\alpha}{\partial t} \\ &= \frac{\partial X_\alpha \rho}{\partial t} + \rho \mathbf{v}_{int} \cdot \nabla X_\alpha \end{aligned}$$

and

$$X_\alpha \nabla \cdot \rho \mathbf{v} = \nabla \cdot (X_\alpha \rho \mathbf{v}) - \rho \mathbf{v} \cdot \nabla X_\alpha$$

we have:

$$\frac{\partial}{\partial t} \langle X_\alpha \rho \rangle + \nabla \cdot \langle X_\alpha \rho \mathbf{v} \rangle = \langle [\rho(\mathbf{v} - \mathbf{v}_{int})]_\alpha \cdot \nabla X_\alpha \rangle \quad (2.8)$$

where $[\rho(\mathbf{v} - \mathbf{v}_{int})]_\alpha$ takes the limiting value at the interface for phase α .

When the momentum equation is considered, a similar treatment yields:

$$\begin{aligned} \frac{\partial}{\partial t} \langle X_\alpha \rho \mathbf{v} \rangle + \nabla \cdot \langle X_\alpha \rho \mathbf{v} \otimes \mathbf{v} \rangle &= \nabla \cdot \langle X_\alpha T \rangle + \langle X_\alpha \rho f \rangle + \\ &\langle [\rho(\mathbf{v} - \mathbf{v}_{int}) - T]_\alpha \cdot \nabla X_\alpha \rangle \end{aligned} \quad (2.9)$$

Here,

$$\langle [\rho(\mathbf{v} - \mathbf{v}_{int})]_\alpha \cdot \nabla X_\alpha \rangle = \Gamma_\alpha$$

and

$$\langle [\rho(\mathbf{v} - \mathbf{v}_{int}) - T]_\alpha \cdot \nabla X_\alpha \rangle = M_\alpha$$

are source terms which arise from inter-facial effects. They model the mass and momentum transfers which can occur between different phases. By now considering the jump conditions (2.4) and (2.5), it can be shown that:

$$\sum_{\alpha=1}^2 \Gamma_\alpha = 0$$

and

$$\sum_{\alpha=1}^2 M_\alpha = M_m.$$

The first of these two relationships expresses the fact that the total mass in the system is conserved – no mass is created – while the second means that momentum balances must take into account inter-facial forces and particularly, the surface tension.

The volume fraction is then defined as:

$$r_\alpha = \langle X_\alpha \rangle.$$

Next, the concept of *phasic average* and *mass-weighted average* of a quantity ϕ are introduced respectively by:

$$\tilde{\phi}_\alpha = \frac{\langle X_\alpha \phi \rangle}{r_\alpha}$$

and

$$\hat{\phi}_\alpha = \frac{\langle X_\alpha \rho \phi \rangle}{r_\alpha \tilde{\rho}_\alpha}.$$

The averaged equations (2.8) and (2.9) become (noting that different averaging are applied to different quantities):

$$\frac{\partial r_\alpha \tilde{\rho}_\alpha}{\partial t} + \nabla \cdot (r_\alpha \tilde{\rho}_\alpha \hat{\mathbf{v}}_\alpha) = \Gamma_\alpha \quad (2.10)$$

$$\frac{\partial r_\alpha \tilde{\rho}_\alpha \hat{\mathbf{v}}_\alpha}{\partial t} + \nabla \cdot (r_\alpha \tilde{\rho}_\alpha \hat{\mathbf{v}}_\alpha \otimes \hat{\mathbf{v}}_\alpha) = -r_\alpha \tilde{\mathbf{T}} + \mathbf{M}_\alpha \quad (2.11)$$

where the body force f has been neglected.

It is usual to write $\tilde{\mathbf{T}}_\alpha$ in terms of a pressure plus extra stresses giving:

$$\tilde{\mathbf{T}}_\alpha = -\tilde{p}_\alpha \mathbf{I} + \tilde{\boldsymbol{\tau}}_\alpha$$

Constitutive relationships have to be supplied for the stress tensor $\tilde{\boldsymbol{\tau}}_\alpha$:

Similarly, the inter-facial momentum transfer term \mathbf{M}_α is usually separated into various parts:

$$\mathbf{M}_\alpha = \Gamma_\alpha \mathbf{v}_{\alpha,int} - p_{\alpha,int} \nabla r_\alpha + \mathbf{M}_\alpha^d$$

where \mathbf{M}_α^d is the inter-facial force density, $p_{\alpha,int}$ is the inter-facial pressure for phase α , and $\mathbf{v}_{\alpha,int}$ is the inter-facial velocity for phase α . These last three quantities have to be specified by constitutive relationships. In the second part of [49], the problem of specifying suitable constitutive relationships is considered.

Finally, the so-called single pressure model [9] is obtained by setting, for all α ,

$$\tilde{p}_\alpha = p_{\alpha,int} = p.$$

The single pressure model is advantageous because it eliminates the need for constitutive relationships for the pressures. However, it appears that the ill-conditioning of the multi-fluid equations has its roots in this simplification.

2.2.3 Discussion of the Multi-Fluid Equations

The multi-fluid equations provide a general framework for the modelling of multiphase flows. By themselves however, they constitute only a very incomplete description of the physics involved in multiphase flows. In order to be useful, they need to be supplemented by constitutive relationships which will supply the missing information. This will concern mainly diffusion tensors as well as inter-phase transfer terms. Providing good constitutive relationships is a crucial issue in multiphase modelling. They will be highly dependent on flow regimes. For instance, transfer terms in stratified flows are very different from those of bubbly flow. It may be said

that different flow regimes are characterised by different constitutive relationships. A further difficulty is that constitutive relationships will usually be functions of a large number of physical parameters.

Traditionally, constitutive relationships have been obtained from a large amount of experimental correlations but a more theoretical approach as outlined in reference [49] is also possible.

2.2.4 Summary of the Multi-Fluid Equations

A simplified version, for compressible flows, of the model developed by Ishii [8] can be found in [52]. For each phase, it takes the following form:

- Continuity Equation:

$$\frac{\partial r_\alpha \rho_\alpha}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{v}_\alpha) = \Gamma_\alpha. \quad (2.12)$$

- Momentum Equation:

$$\frac{\partial r_\alpha \rho_\alpha \mathbf{v}_\alpha}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{v}_\alpha \otimes \mathbf{v}_\alpha) = -r_\alpha \nabla p_\alpha + \nabla \cdot r_\alpha (\mathbf{T}_\alpha + \mathbf{T}_\alpha^t) + r_\alpha \rho_\alpha \mathbf{g} + \mathbf{v}_{\alpha, int} \Gamma_\alpha + \mathbf{M}_{int, \alpha} - \nabla r_\alpha \cdot \mathbf{T}_{int}. \quad (2.13)$$

- Enthalpy Energy Equation:

$$\frac{\partial r_\alpha \rho_\alpha H_\alpha}{\partial t} + \nabla \cdot (r_\alpha \rho_\alpha \mathbf{v}_\alpha H_\alpha) = -\nabla \cdot r_\alpha (\mathbf{q}_\alpha + \mathbf{q}_\alpha^t) + r_\alpha \frac{D_\alpha}{Dt} p_\alpha + H_{\alpha, int} \Gamma_\alpha + \frac{q_{\alpha, int}}{L_s} + \Phi_\alpha. \quad (2.14)$$

where:

- Γ_α is the mass transfer to phase α ;
- $\mathbf{M}_{i, \alpha}$ is the inter-facial drag;
- \mathbf{T}_α and \mathbf{T}_α^t are the laminar and turbulent shear stress tensors;
- \mathbf{q}_α and \mathbf{q}_α^t are the laminar and turbulent heat fluxes;
- $q_{\alpha, i}$ is the inter-facial heat flux;
- Φ_α is the heat source term;
- L_s is the length scale at the interface, the ratio $1/L_s$ has the physical meaning of the an inter-facial area per unit volume;
- D_α/Dt denotes the convective velocity for phase α ;
- the subscript *int* denotes inter-facial values.

2.2.5 Limiting Case 1: the Homogeneous Model

In this case, the phases are assumed to share the same velocity and energy fields [9]. In this way, the momentum equations can be summed over all phases to give a single momentum transport equation of the form:

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \nabla \cdot (\rho(\mathbf{v} \otimes \mathbf{v}) - \mu \mathbf{T}) = -\nabla p + \rho f \quad (2.15)$$

where ρ and μ can be considered to be the density and viscosity of the mixture:

$$\rho = \sum_{\alpha=1}^M r_{\alpha} \rho_{\alpha}$$

$$\mu = \sum_{\alpha=1}^M r_{\alpha} \mu_{\alpha}$$

If surface tension effects are neglected, as is the case for equation (2.15), then the inter-facial source terms cancel out. The energy equations can also be added together in a similar way but the continuity equations remain distinct, allowing the volume fractions to be determined. The homogeneous model is adopted if (i) there are large momentum and energy transfer terms so that the velocities and energy for each phase equalise quickly or (ii) if the flow is strongly stratified under the influence of gravity. The VOF (Volume Of Fluid) method [53] is an algorithm based on these equations, derived from the MAC (Marker And Cell) method [54]. It is particularly suited to flows with a free surface where the position of the interface is of interest.

2.2.6 Limiting Case 2: the Drift Flux Model

The Drift Flux Model [41, 30, 9] is applicable to problems where the physics is mainly contained in the motion of the phases relative to each other, such as a gas bubble rising in a column of liquid. In this model, a total momentum equation is considered. However, the phases are allowed to have different velocity fields so that a constitutive relationship for the velocity difference has to be supplied to close the problem.

2.3 Numerical Simulation of Multiphase Flows

2.3.1 Brief Overview of Numerical Issues in the Simulation of Multiphase Flows

The numerical simulation of multiphase flows is a large subject, covered in detail in References [9] and [17]. In many transient applications, the differential terms con-

nected with the viscous effects are neglected and it is necessary to solve conservation equations of the type:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{g}$$

which are similar to the well-known Euler equations of Aerodynamics. References [55, 56], for instance, can be consulted for the theory of hyperbolic conservation laws. Their numerical simulation is well covered by many books (e.g. [57, 58, 4]).

In the particular case of multiphase flows, there are, however, three fundamental issues which need to be addressed.

- Firstly, equations (2.12) to (2.14) are ill-posed (see, for instance, References [49, 9, 17, 59, 60]). This refers to the fact that the Jacobian of \mathbf{f} , with respect to the conservative variables \mathbf{u} , can have complex eigenvalues for certain regions of phase space. In such cases, the amplitude of all error modes increases with time over the entire range of frequencies. Their numerical solutions will therefore tend to exhibit non-physical oscillations unless diffusion reduces the error. Usually a sufficient amount of numerical diffusion is added by making use of cell-donor or upwind differencing. However, this has the obvious drawback of also blurring the solution features since the schemes are then first order accurate in space. The ill-posedness of the equation can be traced to the use of a single pressure model. Consequently, the reformulation of the multi-fluid model to eliminate the ill-posedness of the equation is an important topic of research. Adding viscous terms does not completely cure the problem: on sufficiently fine grids, there will always be some wavelengths which have an amplification factor larger than 1.
- Secondly, in many cases, the flux function \mathbf{f} is not available. In other words, the governing equations cannot be rewritten in *conservative* form. This complicates the implementation of most numerical schemes available for the solution of hyperbolic conservation laws. The non-conservative character of the equations also has implications for the modelling of the speed at which fronts propagate.
- Thirdly, in many two phase flows, phenomena of interest propagate at a velocity much lower than the speed of sound. Explicit methods, which would be the simplest means of providing time-accurate numerical solutions, are therefore hugely penalised because of their stability limit, which is related to the speed of sound in the system. A large number of very small explicit steps are necessary to model a phenomenon which varies relatively slowly. In such cases, implicit computations should have a role to play. This blending of explicit and implicit schemes for the numerical simulation is also a current area of research. A traditional approach to remove the stability constraint imposed by the speed of sound, is to use semi-implicit methods [61, 62]. The pressure terms which are responsible for the CFL limitation (see for instance [63, pages 87-91]) are treated implicitly together with the inter-phase coupling, while

the convective terms are treated explicitly. It is actually possible to derive an elliptic equation for the pressure, linking it to its value for neighbouring cells. Fast multigrid solvers can then be easily applied to the pressure equation.

Significant effort has been devoted to the design of efficient numerical schemes for the accurate solution of the Euler equations such as TVD schemes [64] and Roe-type Riemann solvers [65]. It would be very beneficial if these could be extended to the conservation laws which are applicable to multiphase flows. Assuming that a well-posed set of equations can be found, the multiphase conservation laws have two distinguishing features compared with the Euler equations, which makes such an adaptation non-trivial:

- There are source terms which render the characteristics curved.
- The equations do not admit a maximum principle since new extrema can appear in the solutions. This is a feature shared by the simpler shallow water equations.

As a result, simpler monotonic schemes may be more appropriate to multiphase flows. Accuracy can then be enhanced by adaptive algorithms.

Having discussed some of the issues arising in the simulation of multiphase flows, we now present two well-established solution algorithms. These have been chosen as representative of specific techniques developed for multiphase flows.

2.3.2 The SETS Method

SETS, which stands for *Stability Enhancing Two Step* method, is described in Reference [27]. It was originally developed for nuclear reactor safety analysis and has been incorporated in the TRAC family of codes. The SETS method is also the basic solver in the PLAC code [66] for pipeline analysis. SETS was designed to overcome a basic limitation of semi-implicit schemes, namely, the fact that they are not unconditionally stable but require the CFL number based on the *convective* velocity to be less than 1. SETS is designed to extend the region of stability by the addition of a corrector or stabiliser step to the basic semi-implicit scheme. During the stabiliser step, corrections to the unknowns are computed and the convective terms are treated implicitly. The performance penalty incurred is not too large.

2.3.3 Example of an Fully Implicit Scheme: IPSA

The IPSA method – which stands for *Inter Phase Slip Algorithm* – is a well known *implicit* method for the solution of the (transient) multi-fluid equations. It is used by at least two major commercial CFD codes, CFX 4.1 [67] and fluent [68].

Reference [69] appears to be the main reference for the description of the IPSA algorithm but it focuses on the main features of the method and omits the details of some derivations.

It is basically an extension of the SIMPLE [70] family of schemes to multi-phase flows. As with SIMPLE, there are a number of variants for IPSA. The original IPSA-PEA method as used in `fluent` [68, page 9-25] is briefly described:

- The equations are solved on a staggered grid and discretised using a finite volume approach;
- The algorithm is segregated in nature: given an approximate solution on each cell, each equation is solved in turn to give a correction for a particular value. Volume fraction corrections are obtained from the continuity equations. Velocities correction arise from the momentum equations. Finally the pressure correction is computed from the overall continuity equation, which is obtained by adding all the continuity equations together.

Like the SIMPLE algorithm, the over-all continuity equation is manipulated to obtain a linear relationship between the pressure correction at one point and its nearest neighbours. This relationship uses the differential coefficients of velocity variations with pressure. These are obtained from the momentum equations.

If the energy is included, then the corresponding equations lead to enthalpy corrections.

- One of the central aspects of IPSA is the *Partial Elimination Algorithm* (PEA). This is introduced in order to take into account the strong local coupling between the phases, hence accelerating convergence. It involves algebraic manipulations of the momentum equations so that the velocities for each phase are eliminated from the momentum equations for all other phases.
- In a single IPSA cycle, values are corrected in the following order:
 1. enthalpy
 2. volume fractions
 3. velocities
 4. pressure

IPSA and SIMPLE are examples of segregated solvers. They treat different equations separately and rely on global linearisations. As a result, these methods are plagued by poor convergence rates. Coupled solvers were specifically developed to overcome these difficulties.

2.4 Coupled Solvers for the (Steady) Navier-Stokes Equations

Since we have now discussed the equations we wish to solve, we will now focus our attention on the techniques it is proposed we will use to design our solution algorithm. The coupled solver methodology is first examined before multigrid methods are introduced.

Many solvers for the incompressible Navier-Stokes equations rely on a *segregated* approach where velocities and pressure corrections are computed separately. A typical example is the SIMPLE algorithm [70] and its variants [71]. The SIMPLE algorithm, in its original form, is a pseudo-transient method where the steady state solution is obtained by marching in “time”. Given an approximation of the solution, the basic steps to be taken in order to obtain the next approximate are [72, pages 357–368]:

1. Given approximations u^n , v^n and p^n of the velocity and pressure field, solve the momentum equations (implicitly) to obtain an approximate velocity solution u^* and v^* .
2. Compute a pressure correction Δp so that the velocity fields $u^{n+1} \equiv u^* + u^{(cor)}$ and $v^{n+1} \equiv v^* + v^{(cor)}$ satisfy the continuity equation. Here $u^{(cor)}$ and $v^{(cor)}$ are the velocity corrections associated with the pressure correction. Δp satisfies a Poisson equation obtained from the continuity equation, with a right-hand-side depending on u^* and v^* . Once Δp is known, $u^{(cor)}$ and $v^{(cor)}$ can be computed.
3. Update the velocities: $u^{n+1} \equiv u^* + u^{(cor)}$ and $v^{n+1} \equiv v^* + v^{(cor)}$.
4. Update the pressures: $p^{n+1} \equiv p^n + \Delta p$.

The fundamental difficulty which one meets when solving the steady Navier-Stokes equations for fluid flows, both in the compressible and incompressible cases, is that there is a strong coupling between the pressure field and the velocity fields via the continuity equation.² SIMPLE-like solvers linearise the equations over the entire domain to obtain an approximate relationship between pressure corrections and velocity corrections. This global linearisation is often of poor quality and the convergence rate of the method is slow.

Another issue is that the flow variables are closely coupled together. In many flow applications, it is common that some of the derivatives:

$$\frac{\partial f_i}{\partial x_j} \tag{2.16}$$

²For compressible flows, the pressure can be introduced explicitly in the continuity equation using the equation of state while for incompressible flow, the coupling is implicit. However, the numerical difficulties are identical.

which form the entries of the Jacobian, take large values. Here the x_j are the dependent flow variables.

In such situations, segregated solvers tend to over-correct some values. Since information about the Jacobians is not incorporated, a correction is obtained from one equation and its effect on other equations, which can be significant if the derivatives take large values, is ignored. It is often said that the correction to one equation throws the others off-balance. The usual cure is to under-relax the computations so that the correction steps (but also the convergence rates) are reduced.

Coupled solvers, (see for instance [73, 74, 75]), are specifically designed to overcome these difficulties. In a typical implementation, the equations are discretised on a staggered grid using a finite volume approach. The discrete momentum and continuity equations, written on a group of cells, are solved together by Newton's method. These equations form a non-linear subsystem:

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}$$

for the vector \mathbf{x} of unknowns written on the group of cells. It is necessary to have an approximation \mathbf{J} of the Jacobian of \mathbf{f} in order to apply Newton's method.

If desired, the equations for the entire domain can be solved together. In this case, the matrices to be handled are large and their degree of sparseness is important for performance. Usually iterative methods of the conjugate gradient type are used to solve (2.4). See for instance Reference [74].

Another approach is to consider a coupled system written on a single cell of the computational domain [73, 1]. In this case, only local coupling is considered and the sub-systems are much smaller (5 by 5 for a basic Navier-Stokes solver in two dimensions) and a block iterative method is used to solve the complete set of equations. A strong body of evidence suggests that it is the accurate treatment of the local coupling which is crucial for improved convergence rates and robustness. This is confirmed by the performance of the pang Navier-Stokes solver [1].

In most implementations, the basic coupled solver is usually embedded in a multigrid method (of which it forms the basic smoother) and this combination forms an efficient, nearly optimal, iterative solver. Two facts are crucial to the success of this approach:

- The Newton linearisation is of sufficiently good quality to rapidly eliminate the high frequencies in the error because it is local in nature. It therefore performs well as a multigrid smoother. The cost of the Newton method is not very large because only the local coupling is taken into account so that block systems are small.
- The multigrid method greatly accelerates the convergence of the quasi-Newton single grid solver since it provides a framework in which the global coupling

can be quickly taken into account using the coarse grid computations. A dual point of view is to consider that the multigrid method resolves the long wavelengths efficiently while the LQN solver deals with the short wavelengths.

For such solvers, the local coupling is treated efficiently since all the residuals are reduced at the same time. Indeed the Newton correction is a descent direction for the Euclidean norm of the residual vector (see Section 4.3.1).

A good example is that of buoyant flows [13]. If a segregated solver is used, the amount of under-relaxation necessary to obtain convergent computations is inversely proportional to the Rayleigh number which can be big. In other words, the pseudo time-step taken by a segregated solver is limited by the buoyant time-scales. On the other hand, a coupled solver removes the need for under-relaxation.

2.5 Multigrid Methods

Multigrid methods can simply be seen as an acceleration technique for the iterative solution of algebraic systems of equations, or, in their full generality, as a consistent framework enabling the solution of a given problem to be found at a nearly optimal cost. The basic idea is to define a sequence of “grids”³, and to cycle through the different grids in the course of the computations so that the features of the solution are resolved on the most suitable grid. By so doing, the solution algorithm is made almost optimal.

In the words of Achi Brandt (reference [76, page 1]), multigrid methods satisfy the so-called *golden rule*:

The amount of computational work should be proportional to the amount of real physical change in the system. Stalling numerical processes must be wrong.

In the context of finite difference/finite element analysis, multigrid methods can be considered to be a device for resolving solutions on the coarsest grid possible: low frequency features will be resolved on coarse grids while high frequency features will require finer grids.

The amount of literature pertaining to multigrid methods is vast. The following will discuss some books and articles which may be consulted as a starting point.

³The first applications of multigrid methods have been in the field of finite difference analysis, where the concept of a hierarchy of grids with different grid size arises naturally. In the present thesis, it is only this geometric interpretation which is being used. However, multigrid methods, or more precisely multilevel methods, have been applied to problems which have no association with physical grids. The “grid” refers then to some abstract data organisation.

Reference [77] is the initial paper in the field of multigrid. Reference [10] discusses the fundamental ideas of multigrid and defines its vocabulary, while reference [76] is a very complete and practical guide to the implementation of multigrid methods. Reference [78] is an example of a more theoretical treatment, while reference [79], which is closely followed in parts of this section, provides a clear introduction to the fundamental concepts of multigrid and a useful list of suggested readings. Reference [80] brings together a set of important papers on multigrid methods. In particular, the introductory paper [81] was very useful in providing a reasonably detailed overview of the topic. Finally, references [82, 83] deal with iterative solvers for system of linear equations and provide a good background to multigrid theory.

Sections 2.5.1 to 2.5.9 form a brief overview of multigrid methods. The focus is on concepts which are relevant for the implementation of multigrid in `pamg` and `pamg-multiphase`. First, we illustrate a fundamental limitation of traditional solvers: as the number of unknowns increases, convergence rates are degraded. Multigrid methods are then introduced qualitatively as a means of remedying this shortcoming. The two-level correction scheme is then described in detail, and its convergence rates estimated for a model problem (Section 2.5.3). The different multigrid cycling strategies are examined in Section 2.5.4, while the salient property of the grid independence of multigrid convergence rates is addressed in Section 2.5.5. The FAS variant of multigrid is introduced in Section 2.5.6, while Section 2.5.7 and 2.5.8 are devoted to its application to non-linear problems and adaptive computation respectively. We conclude by discussing the limitations of multigrid methods for non-elliptic problems (Section 2.5.9).

2.5.1 A Limitation of Traditional Iterative Solvers for Systems of Algebraic Equations

Traditional iterative schemes for the solution of linear systems of equations are marred by their poor asymptotic convergence rates. Fundamentally, traditional relaxation methods are local (the Preconditioned Conjugate Gradient (PCG) scheme being an exception), hence iterative solvers remove the high frequency error modes better than they eliminate those at lower frequencies. As the computations progress, the convergence rate is limited by the low-frequency modes in the error. The reason for this phenomena is that low frequencies are usually associated with eigenvalues of the iteration matrix which are close to unity (see [79]).

Let \mathbf{u} be the solution of the linear system:

$$A\mathbf{u} = \mathbf{b}. \quad (2.17)$$

An iterative scheme for solving (2.17) can be written in the following form, having split the matrix A into relevant parts:

$$\tilde{\mathbf{u}}^{n+1} = P\tilde{\mathbf{u}}^n + \mathbf{g} \quad (2.18)$$

where $\tilde{\mathbf{u}}^n$ and $\tilde{\mathbf{u}}^{n+1}$ are successive approximations of \mathbf{u} and P is the iteration matrix. If we define the *error* as follows:

$$\mathbf{e}^n = \mathbf{u} - \tilde{\mathbf{u}}^n,$$

and use the fact that the iterative scheme is consistent⁴, successive error vectors are given:

$$\mathbf{e}^n = P^n \mathbf{e}^0$$

where \mathbf{e}^0 is the initial error. \mathbf{e}^n can then be expanded on the basis of the k eigenvectors \mathbf{w}_k of P (assumed independent):

$$\mathbf{e}^n = \sum_{k=1}^n e_k^n \mathbf{w}_k = \sum_{k=1}^n e_k^0 (\lambda_k)^n \mathbf{w}_k \quad (2.19)$$

where λ_k are the n eigenvalues of P . Hence, the k^{th} mode of the error vector is reduced by a factor λ_k at each iteration. If λ_k is close to unity, the mode is therefore damped very slowly. Equation (2.19) also shows that after a few iterations, the error is proportional to the dominating eigenvector of the system.

A related question is that of providing a measure for the speed at which the iterative process converges. Equation 2.19 indicates that the convergence is limited by the largest eigenvalue of P in modulus. This motivates the introduction of the concept of *spectral radius* of a matrix. If a vector norm, and its associated matrix norm are introduced, then the error after n iterations is related to the initial error by the following relationship:

$$\|\mathbf{e}^n\| = \|P\|^n \|\mathbf{e}^0\|.$$

The convergence condition that after M iterations the norm of the error is reduced by a factor of 10^{-d} is approximately satisfied if:

$$[\rho(P)]^M \leq 10^{-d}$$

where $\rho(P)$, the spectral radius of the matrix P , is defined as:

$$\rho(P) = \max_i |\lambda_i(P)|$$

The iterative method given by (2.18) will converge if and only if $\rho(P) < 1$. However, the convergence can be very slow if $\rho(P)$ tends to 1. $\rho(P)$ is called the *convergence factor* and the quantity $-\log_{10}(\rho(P))$ is called the *convergence rate*.

We now put this in context with a simple example.

⁴That is, it does not correct the exact discrete solution: $\mathbf{u} = P\mathbf{u} + \mathbf{g}$.

($k < N/2$) of the error are associated with eigenvalues which are close to 1 and are therefore responsible for the slow convergence rates (see Figure 2.3).

This characteristic is not significantly dependent on the value of the relaxation factor which tends to affect the convergence rates for the higher frequency modes. Indeed the reason why the weighted Jacobi method is preferable to the basic Jacobi procedure is that the latter performs best for middle frequencies but reduces both the low and the high frequencies very slowly. For $\omega = 1$, we have:

$$\lambda_1(P) = 1 - 2 \sin^2 \left(\frac{\pi}{2N} \right) = 1 - \mathcal{O} \left(\frac{1}{N^2} \right)$$

and

$$\lambda_{N-1}(P) = 1 - 2 \cos^2 \left(\frac{\pi}{2N} \right) = -1 + \mathcal{O} \left(\frac{1}{N^2} \right).$$

2.5.2 Fundamental Concepts of Multigrid Methods: A Qualitative Approach

Multigrid techniques are based on the observation that an error vector which only contains low frequency modes on a given grid has high frequency components on a coarser grid, provided no aliasing is introduced. Aliasing will be avoided if the error is smooth enough to be approximated meaningfully on a coarser grid.

Often this can be achieved by applying a few iterations of a traditional solver. This will eliminate the high frequency modes in the error until they can be neglected, compared with the low frequency modes. At this point, the computation is then transferred to a coarser grid. The objective is to quickly approximate the fine grid error by taking advantage of the fact that it will contain a significant amount of high frequency modes on the coarse grid. The coarse grid correction is then transferred back to the fine grid where it will greatly reduce the residual because it is a good approximation of the error on the fine grid.

The interpolation operators used for this last transfer will result in errors in the approximation but these will be of a high frequency nature on the fine grid because they are local effects and consequently, the fine grid smoother will quickly eliminate them. The mathematical expression for these concepts is the *Correction Scheme* which is described in Section 2.5.3.

The process of coarse grid corrections can be nested with further transfers to coarser grids. As a result, the convergence rate of the method remains large because error modes are removed on the most suitable grid i.e. the coarsest grid on which they can be resolved.

A multigrid method for a linear problem should converge in $\mathcal{O}(N)$ iterations, where N is the number of discrete unknowns. This is better than both preconditioned

conjugated gradient methods ($\mathcal{O}(N^{5/4})$ iterations for one particular problem and one particular pre-conditioner) and SOR methods with optimal relaxation ($\mathcal{O}(N^{3/2})$ iterations) and makes it an optimal process (order-wise) for the solution of linear algebraic equations.

The fact that multigrid methods converge in $\mathcal{O}(N)$ iterations is associated with the property that convergence rates are grid-independent. This can be proved for linear problems (see Reference [84] and Section 2.5.5) and is often observed for non-linear problems. The convergence rates of pang for single phase Navier-Stokes flows (see Section 5.4.4) are a good example. Reference [84] also addresses the problem of multigrid convergence for non-linear problems.

2.5.3 The Two-level Algorithm and the Correction Storage Scheme

The so-called two-level scheme is the simplest and most fundamental multigrid algorithm. It only involves two grids, a fine grid with spacing h and a coarse grid with spacing $2h$, but all multigrid methods are merely extensions or variations of it. The different elements of the two-level scheme are first described. Following [79], the two-level scheme is then analysed in terms of Fourier modes in order to obtain the so-called *spectral* picture of multigrid. This gives insights into the behaviour of multigrid methods and highlights the reasons for their efficiency.

Elements of Multigrid Before we begin a detailed description of multigrid, we need to introduce some notation. Given a grid Ω^h where h is the meshsize, the solution u of the problem:

$$A^h u^h = f^h \quad (2.25)$$

is sought. Let \tilde{u}_0^h be an initial approximation of u^h and consider an iterative procedure to solve equation (2.25) defined by the two matrices P^h and Q^h . Hence, successive approximations to the solution are generated by application of:

$$\tilde{u}_{n+1}^h = P^h \tilde{u}_n^h + Q^h f^h.$$

In the context of multigrid method, the iterative procedure is known as the *relaxation* step. Alternatively, the relaxation procedure is referred to as the smoother, since its main aim is to remove the high frequency components of the error:

$$e^h = u^h - \tilde{u}^h.$$

as efficiently as possible. As seen in Section 2.5.1, many of the traditional iterative solvers are suitable for this task.

The first stage of a multigrid cycle is to perform ν_1 relaxation sweeps on the initial estimate \tilde{u}_0^h to obtain an improved approximation $\tilde{u}_{\nu_1}^h$. Its associated error should

grid (see below). Since the coarse grid problem is defined in purely algebraic terms, it leads to a class of multigrid methods known as *Algebraic Multigrid*. Such formulations for the coarse grid problems are well established from a theoretical point of view. Equation (2.28) is known as the Galerkin condition. It forms the first of two so-called *variational properties*. The second property,

$$I_h^{2h} = c \left(I_{2h}^h \right) \quad c \in \mathfrak{R}$$

is a relationship satisfied by the interpolation operator and the full-weighting operator. The variational properties greatly facilitate the analysis of multigrid methods. In algebraic multigrid, other definitions of the coarse grid operator are possible.

When equation (2.25) arises from the discretisation of a partial differential problem, the alternative is to define A^{2h} as the discretisation of the same partial differential problem on the coarse grid. This could be referred to as *geometrical multigrid*. An advantage of geometrical multigrid is that the evaluation of A^{2h} is usually less costly than for algebraic multigrid. In the case of the model problem (2.20), both the geometrical and algebraic approach lead to the same expression for A^{2h} .

We will leave aside the problem of solving (2.27) for the moment. Since the grid is coarser, it is an easier problem than (2.25) and in particular, if iterative methods are used, convergence rates should be improved. Once an estimate $\tilde{\mathbf{e}}^{2h}$ for \mathbf{e}^{2h} is available, it is transferred back to the coarse grid. This is the process of *prolongation*. The approximation $\tilde{\mathbf{u}}^h$ is updated by applying the rule:

$$\tilde{\mathbf{u}}^h \leftarrow \tilde{\mathbf{u}}^h + I_{2h}^h \tilde{\mathbf{e}}^{2h}.$$

Since the problem is linear, the errors on grids Ω^h and Ω^{2h} are very similar and the approximation $\tilde{\mathbf{u}}^h$ should be significantly improved. I_{2h}^h is an interpolation operator used to transfer the coarse grid correction to the fine grid:

$$\mathbf{v}^h = I_{2h}^h \mathbf{v}^{2h}.$$

In the one-dimensional case, a common choice is defined by:

$$\begin{aligned} v_{2j}^h &= v_j^{2h} \\ v_{2j+1}^h &= \frac{1}{2}(v_j^{2h} + v_{j+1}^{2h}) \end{aligned} \quad 0 \leq j \leq \frac{N}{2} - 1 \quad (2.29)$$

Similarly to the restriction operator I_h^{2h} , the prolongation operator I_{2h}^h can be written

Hence, a single application of the two-level scheme can be considered as a single operation on Ω^h :

$$\tilde{\mathbf{u}}^h \equiv P^\nu \tilde{\mathbf{u}}^h + I_{2h}^h (A^{2h})^{-1} I_h^{2h} (\mathbf{f}^h - A^h \tilde{\mathbf{u}}^h)$$

Now, the exact solution \mathbf{u} is unchanged by the coarse grid correction and it can be deduced that:

$$\mathbf{e}^h \equiv [I - I_{2h}^h (A^{2h})^{-1} I_h^{2h} A^h] P^\nu \mathbf{e}^h \quad (2.30)$$

Equation (2.30) defines how the error is affected by one application of the correction storage scheme. Let M_h^{2h} be the amplification matrix of the error:

$$M_h^{2h} = [I - I_{2h}^h (A^{2h})^{-1} I_h^{2h} A^h] P^\nu \quad (2.31)$$

M_h^{2h} can be considered as the iteration matrix of the two level scheme.

We now consider our model problem (2.20). The error vector \mathbf{e}^h is projected on the modes of A^h , i.e. the eigenvectors \mathbf{w}^k given by equation (2.23), and the effect of M_h^{2h} on the modes of \mathbf{e}^h is analysed in order to assess the effects of the two-level scheme, mode by mode.

The various components of M_h^{2h} are considered in turn. First, the effect of the transfer operators on the modes of A^h should be investigated. It is useful to separate the modes of A^h in low frequencies ($1 \leq k \leq N/2$) and high frequencies ($N/2 < k \leq N-1$). A high frequency mode \mathbf{w}_{N-k} can be associated to each low frequency mode \mathbf{w}_k , and the pair $\{\mathbf{w}_k; \mathbf{w}_{N-k}\}$ are referred to as *complementary modes*. In other words, \mathbf{w}_k and \mathbf{w}_{N-k} are harmonics.

If the full weighting operator is applied to the k^{th} mode of A^h , it can be shown that:

$$I_h^{2h} \mathbf{w}_k^h = \cos^2 \left(\frac{k\pi}{2N} \right) \mathbf{w}_k^{2h}, \quad 1 \leq k \leq \frac{N}{2} \quad (2.32)$$

Similarly, on the complementary node, we have:

$$I_h^{2h} \mathbf{w}_{N-k}^h = -\sin^2 \left(\frac{k\pi}{2N} \right) \mathbf{w}_k^{2h}, \quad 1 \leq k \leq \frac{N}{2} \quad (2.33)$$

The full weighting operator transforms both complementary modes k and $N-k$ into the k^{th} mode of A^{2h} , so that if the error component on \mathbf{w}_{N-k} is not null, then aliasing will be introduced.

On the other hand, the interpolation operator I_{2h}^h transforms the k^{th} mode of A^{2h} into a fine grid vector which has non-zero components on both the k^{th} and the $N-k^{\text{th}}$ modes of A^h . It can be shown that:

$$I_{2h}^h \mathbf{w}_k^{2h} = \cos^2 \left(\frac{k\pi}{2N} \right) \mathbf{w}_k^h - \sin^2 \left(\frac{k\pi}{2N} \right) \mathbf{w}_{N-k}^h, \quad 1 \leq k \leq \frac{N}{2} \quad (2.34)$$

Combined with the fine grid relaxation, it can be shown that the effect of the coarse grid correction scheme acts on the complementary nodes as follows:

$$\begin{aligned} \mathbf{w}_k &\rightarrow (\lambda(k))^\nu s_k \mathbf{w}_k + (\lambda(k))^\nu s_k \mathbf{w}_{N-k} \\ \mathbf{w}_{N-k} &\rightarrow (\lambda(N-k))^\nu c_k \mathbf{w}_k + (\lambda(N-k))^\nu c_k \mathbf{w}_{N-k} \end{aligned} \quad 1 \leq k \leq \frac{N}{2} \quad (2.35)$$

where $\lambda(k)$ is the eigenvalue of P associated with the k^{th} mode \mathbf{w}_k :

$$\lambda(k) = 1 - 2\omega \sin^2 \left(\frac{k\pi}{2N} \right), \quad 1 \leq k \leq N-1,$$

The coefficients c_k and s_k are defined respectively as:

$$c_k = \cos^2 \left(\frac{k\pi}{2N} \right)$$

and

$$s_k = \sin^2 \left(\frac{k\pi}{2N} \right).$$

Figures 2.1 to 2.5 plot these different quantities as functions of K/N , considered as a continuous variable, and for a particular value of the relaxation parameter ($\omega = 2/3$). The low frequency modes \mathbf{w}_k are quickly removed by the coarse grid correction scheme since the s_k take small values for k small. In fact:

$$s_k = \mathcal{O} \left(\frac{k^2}{N^2} \right), \quad k \ll N$$

and dominates over $\lambda(k)$. The high frequency modes \mathbf{w}_{N-k} are also quickly removed. However, the mechanism is different. This time, for $k \ll N$, we have,

$$c_k = 1 - \mathcal{O} \left(\frac{k^2}{N^2} \right), \quad k \ll N$$

but $\lambda(N-k)$ tends to zero (see Figure 2.3) and dominates over c_k . In other words, for a given pair of complementary modes, the low frequency mode is removed on the coarse grid while the high frequency harmonic is removed on the fine grid.

As Figures 2.4 and 2.5 indicate, the attenuation coefficients are largest for $k = N/2$. On any given grid, the convergence is actually limited by the mid-range frequencies, for which convergence rates are grid-independent.

The spectral theory outlined here, gives a good, realistic, and yet relatively simple image of the behaviour of a multigrid method. A complementary point of view can also be developed, as in [79], based on fundamental concepts of linear algebra, namely the range and null-space of linear operators. Briggs refers to this theory as the *algebraic* picture of the multigrid procedure. Combined with the spectral picture, it provides theoretical reasons why multigrid is necessarily very fast.

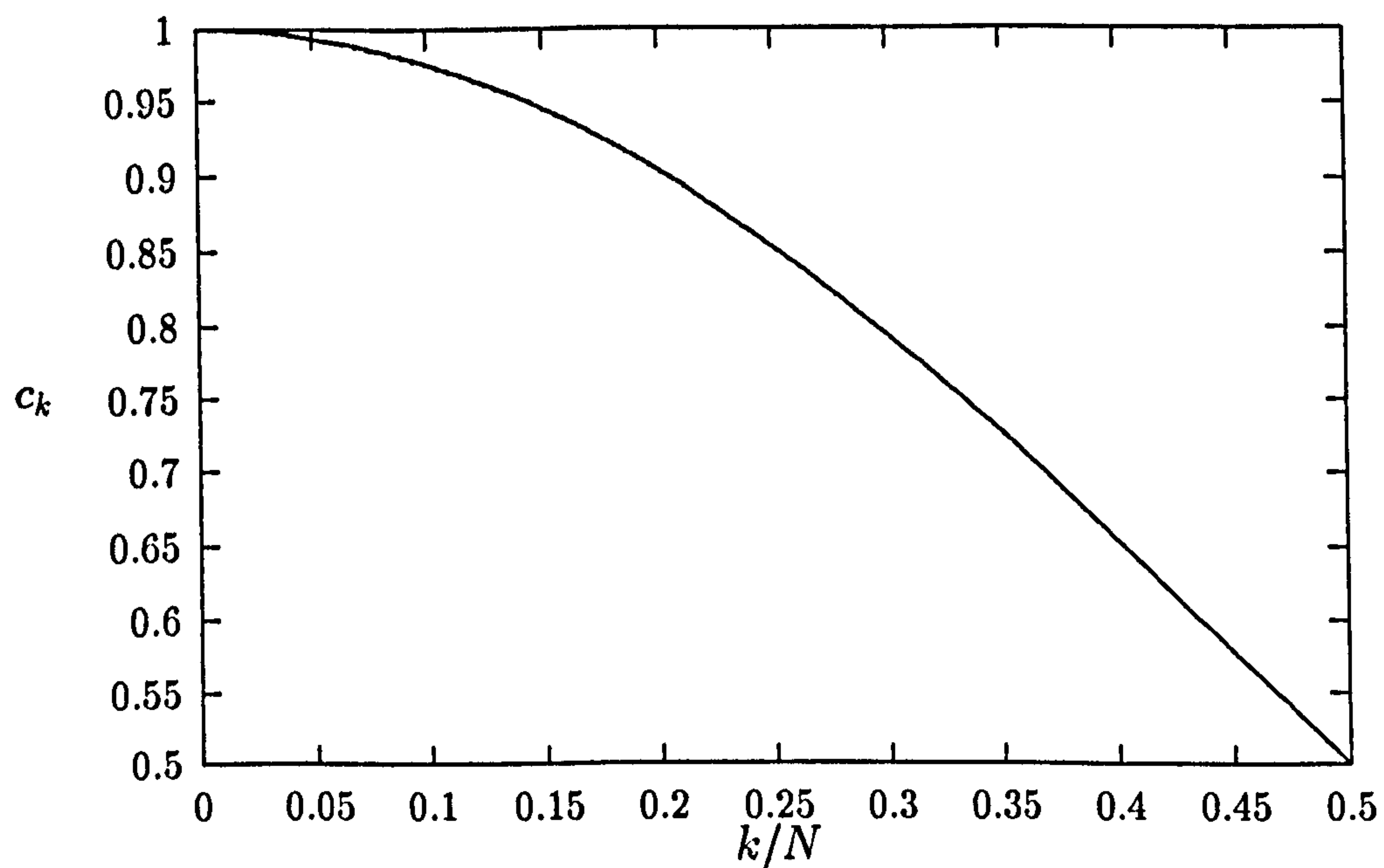


Figure 2.1: Basic two-level scheme for a one-dimensional diffusion problem: Coefficients c_k as a (continuous) function of k/N .

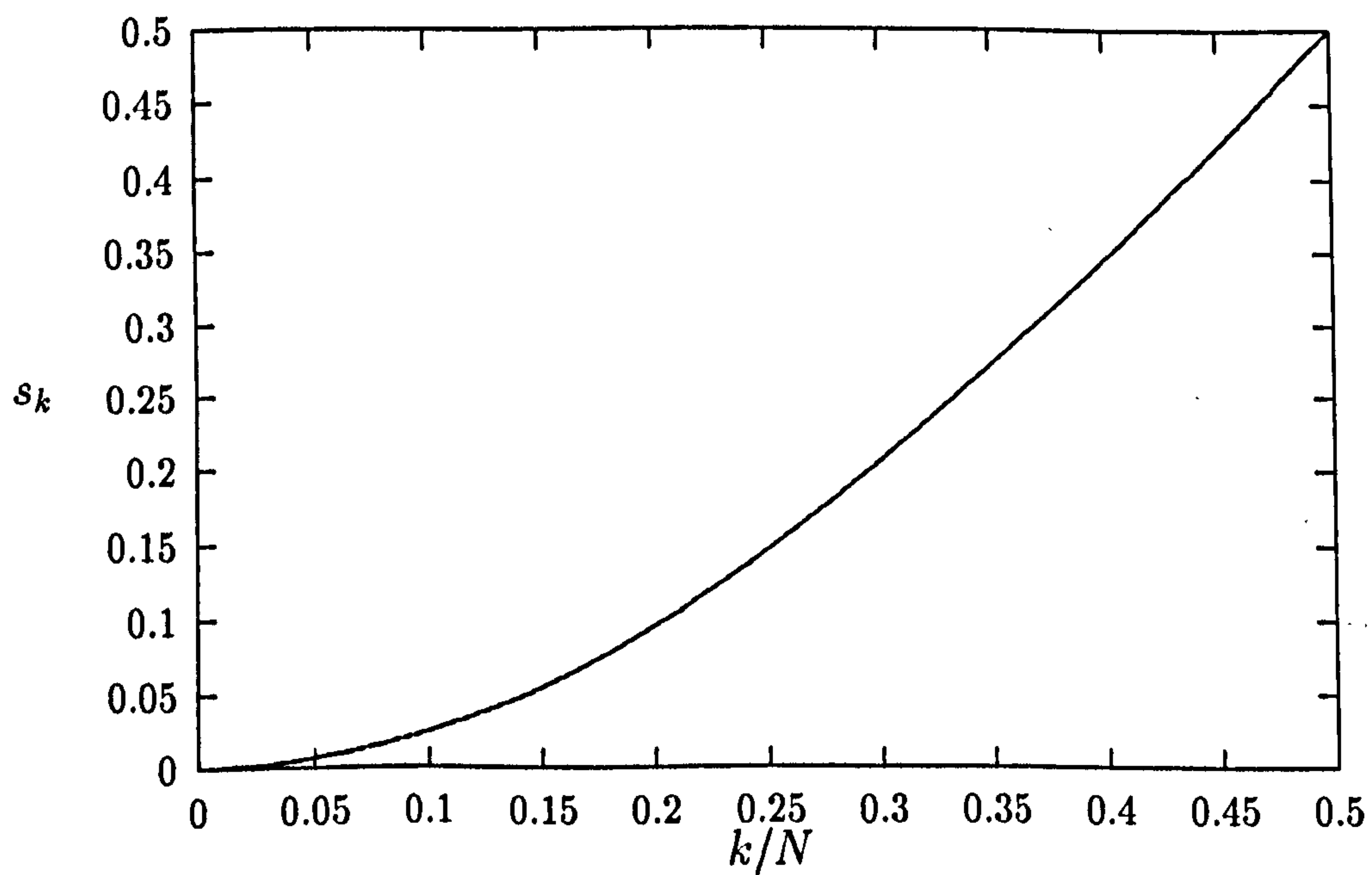


Figure 2.2: Basic two-level scheme for a one-dimensional diffusion problem: Coefficients s_k as a (continuous) function of k/N .

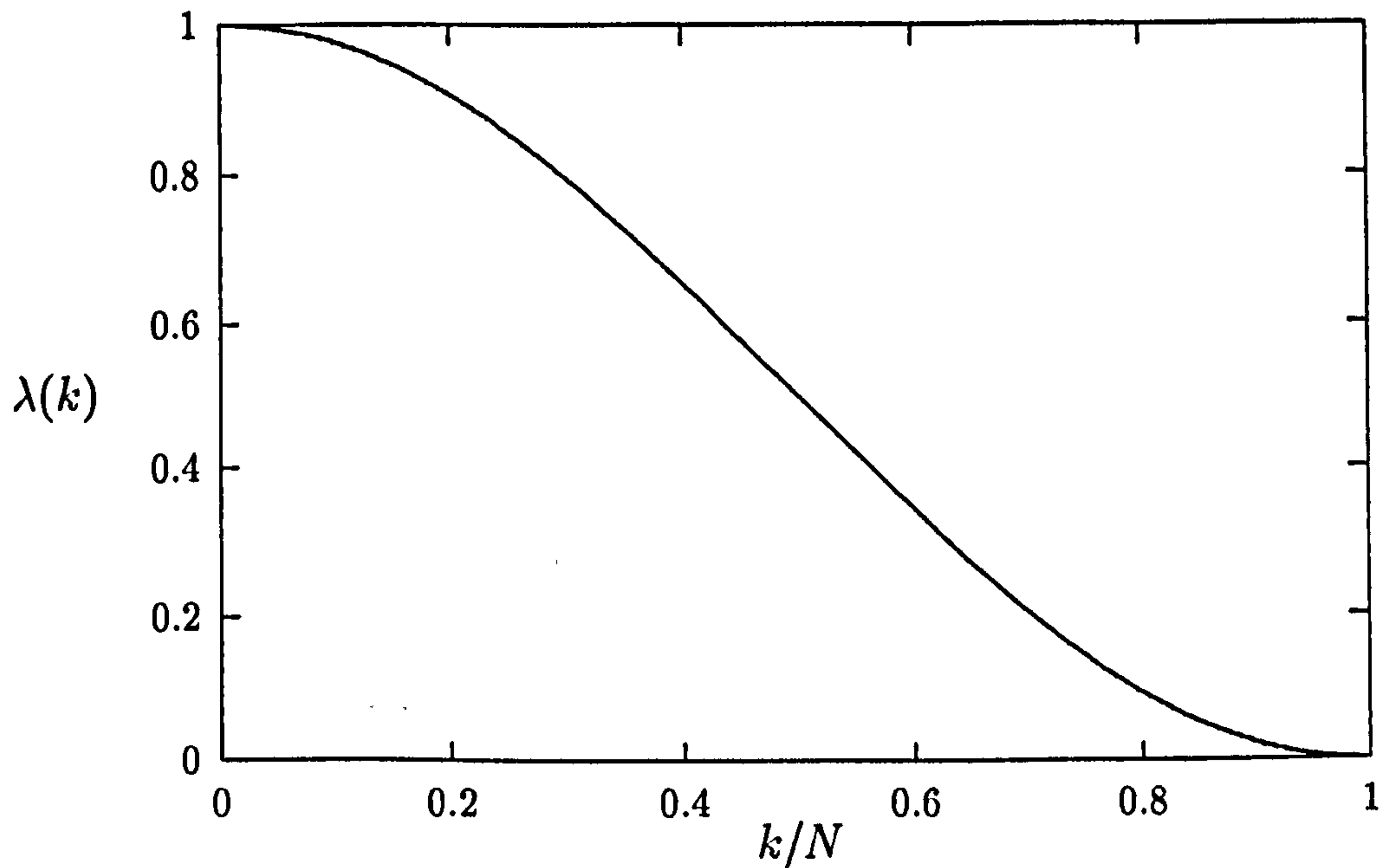


Figure 2.3: Basic two-level scheme for a one-dimensional diffusion problem: Eigenvalue $\lambda(k)$ of the iteration matrix P as a (continuous) function of k/N .

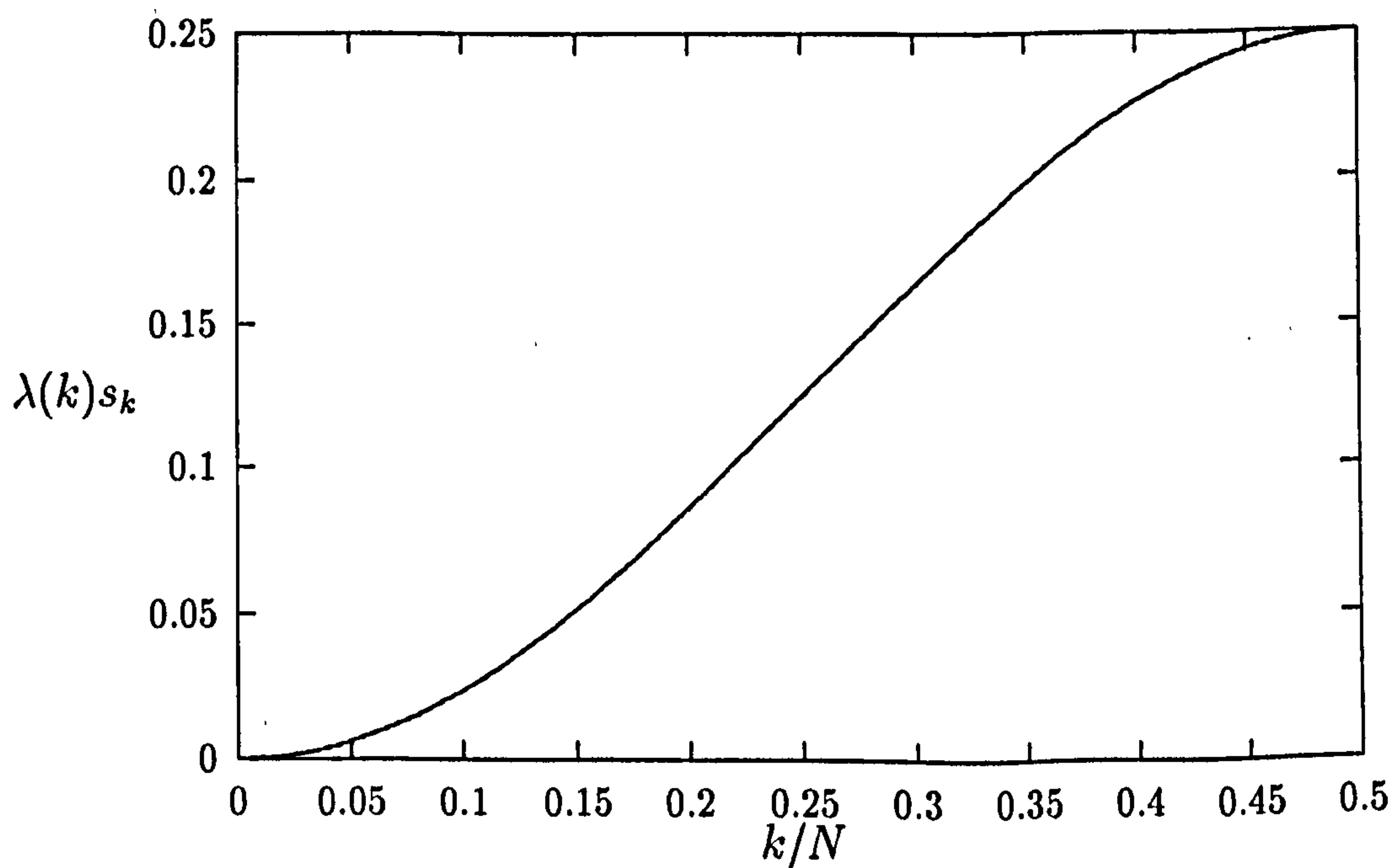


Figure 2.4: Basic two-level scheme for a one-dimensional diffusion problem: Attenuation factor for low frequency modes k as a (continuous) function of k/N .

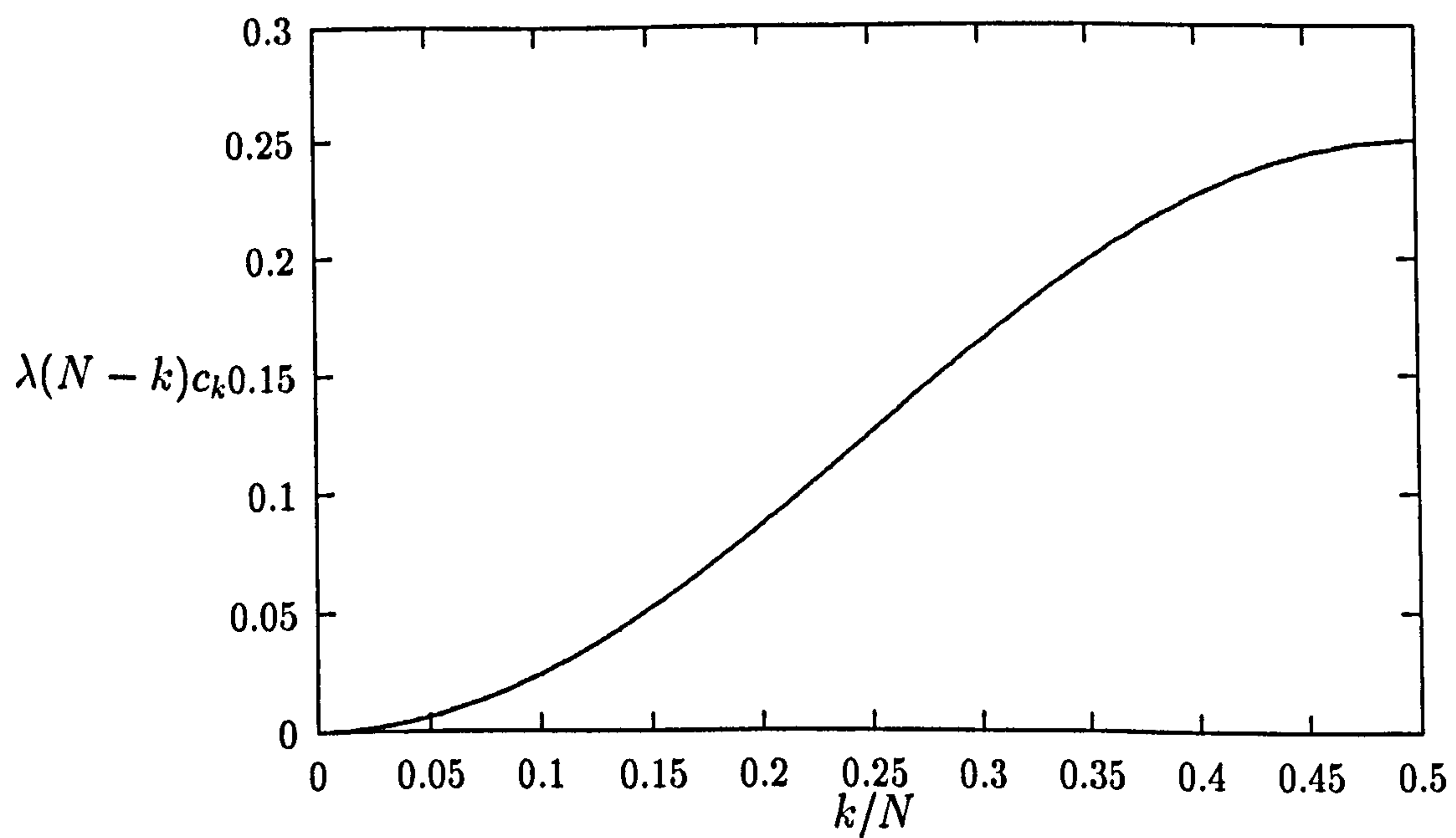


Figure 2.5: Basic two-level scheme for a one-dimensional diffusion problem: Attenuation factor for high frequency modes $N - k$ as a (continuous) function of k/N .

2.5.4 Multigrid Cycling Strategies

Up to this point, the coarse grid problem has been defined but we have not indicated how it may be solved. In fact, it was assumed that an exact solution was available, which is an idealisation. In principle, any direct or iterative method is suitable. Obviously, the coarse grid problem does not differ qualitatively from the fine grid problem and can be solved using a still coarser grid with spacing $4h$. This recursive application of the two-level algorithm gives rise to proper multigrid algorithms, and more specifically to the so-called *V-cycle* (see Figure 2.6). It is only on the coarsest grid that the problem needs to be solved using traditional methods. At this stage it should involve only a small number of unknowns and the use of a direct method is usually recommended.

In a *V-cycle*, as in all other multigrid methods, the cost of the algorithm is very dependent on the amount of fine grid relaxation sweeps $\nu_1 + \nu_2$ performed at each visit of a grid. Sufficient relaxation should take place to eliminate high frequencies but too many relaxation sweeps i.e. relaxing an already smooth error vector may not accelerate convergence and greatly penalise the performance. Choosing the right number of relaxation sweeps is clearly problem-dependent, but in practice small values for ν_1 and ν_2 (in the order of 2) work well.

An alternative to the *V-cycle* is to apply the two-level algorithm twice in order to solve the coarse grid problem. This results in the *W-cycle* (see Figure 2.7). A *W-cycle* costs more than a *V-cycle* but the coarse grid correction is of better quality so less cycles are required to solve a problem. *W-cycles* are usually more robust than *V-cycles* for non-linear problems.

A third cycling strategy is the *F-cycle* [81, pages 52–53]. It is quite similar to the *W-cycle* in the sense that coarse grid computations are repeated in order to improve the quality of the coarse grid correction (see Figure 2.8). However, it is less costly than the *W-cycle*.

V- and *F-cycles* are (*asymptotically*) *optimal* iterative methods in the sense that the computational work required to achieve a fixed accuracy is proportional to the number of discrete unknowns [81, page 4].

Another approach to accelerating the convergence of iterative methods is to provide good starting guesses. This is the concept of *Nested Iteration*, which combined with multigrid methods leads to the *Full Multigrid Method* (FMG) (see, for instance, [81, pages 71–76] and also [79, 76]). A FMG method achieves the discretisation accuracy — i.e. $\|\mathcal{L}^h(\tilde{u}^h) - \mathcal{L}^h(u^h)\| < \|\tau^h\|$ where τ^h is the truncation error — for an amount of computational work which is still proportional to the number of discrete unknowns, if the order of accuracy of the grid transfers is sufficiently high.

The basic idea is to solve the coarse grid problem as accurately as possible before the fine grid problem is attempted. A FMG method starts on the coarse grid and works

itself up to the finest level by repeated application of prolongations and multigrid iterations. See Figure 2.9 for an example based on the V-cycles as the underlying multigrid method. The F-cycle can be considered as a simple example of FMG method.

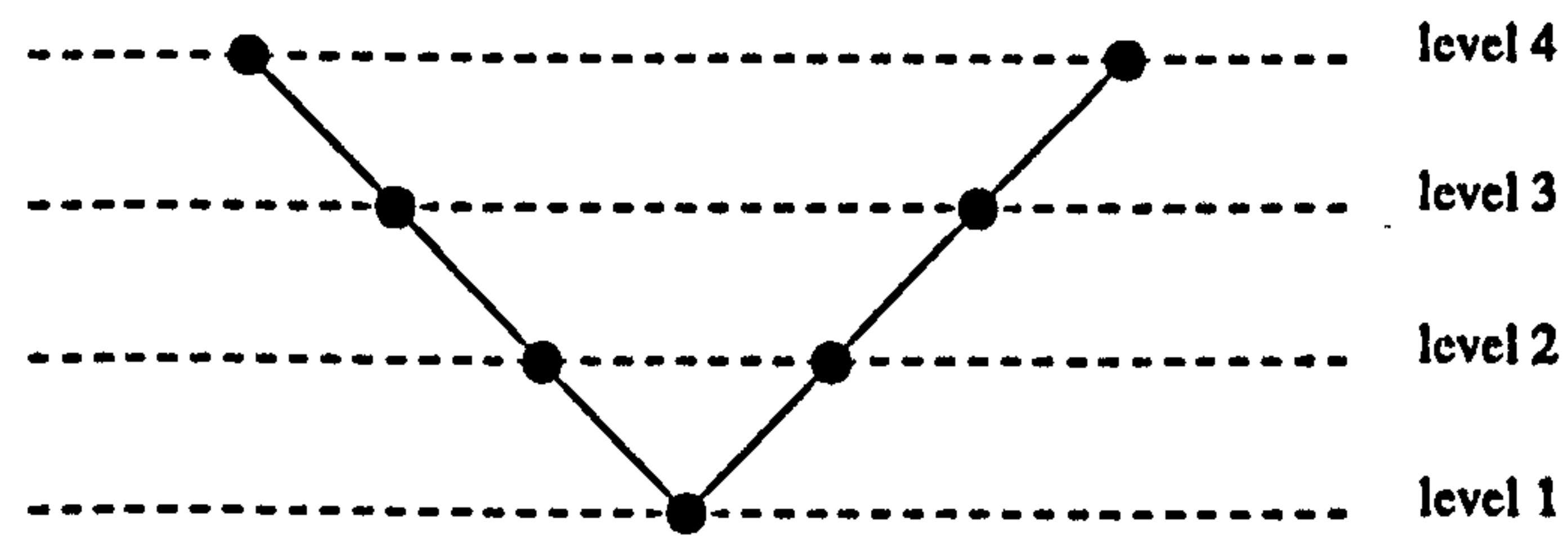


Figure 2.6: Four level multigrid V-Cycle

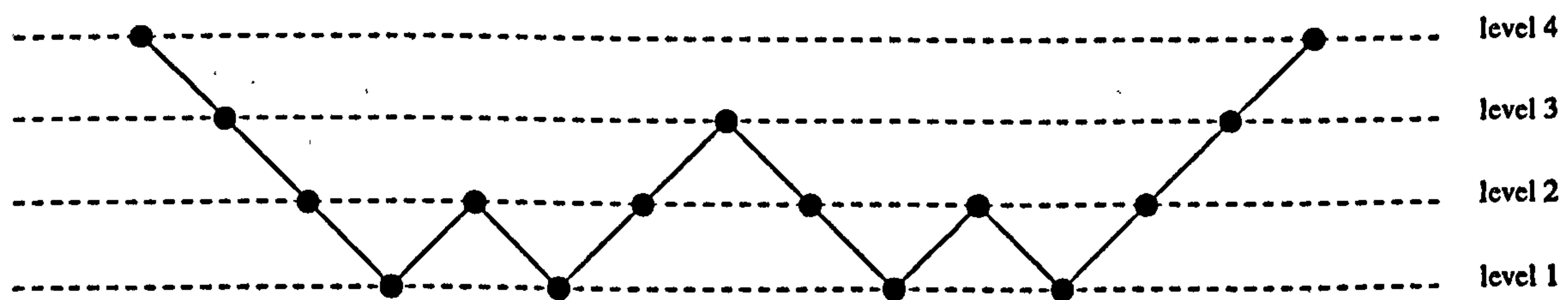


Figure 2.7: Four level multigrid W-Cycles

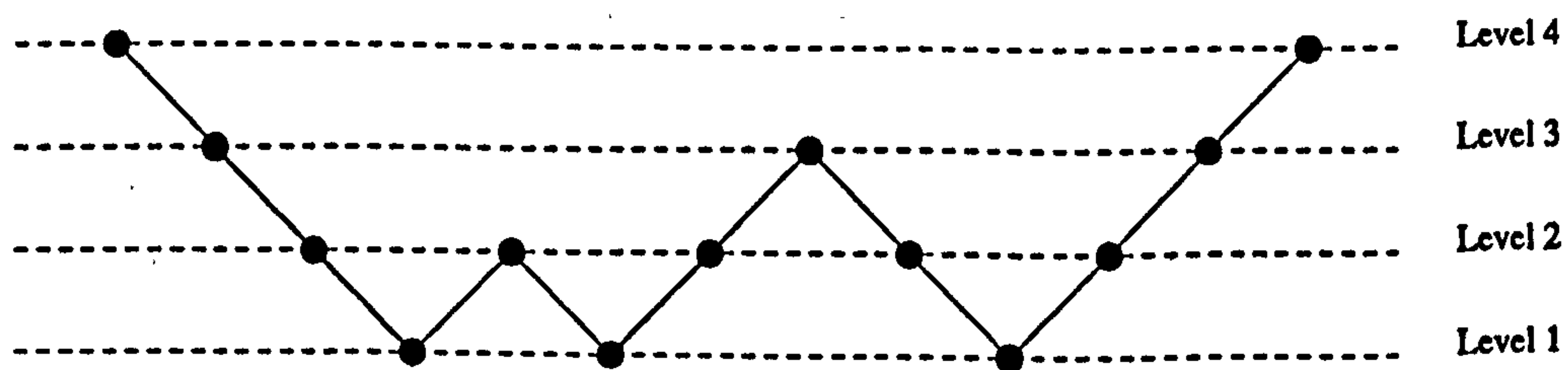


Figure 2.8: Four level multigrid F-Cycles

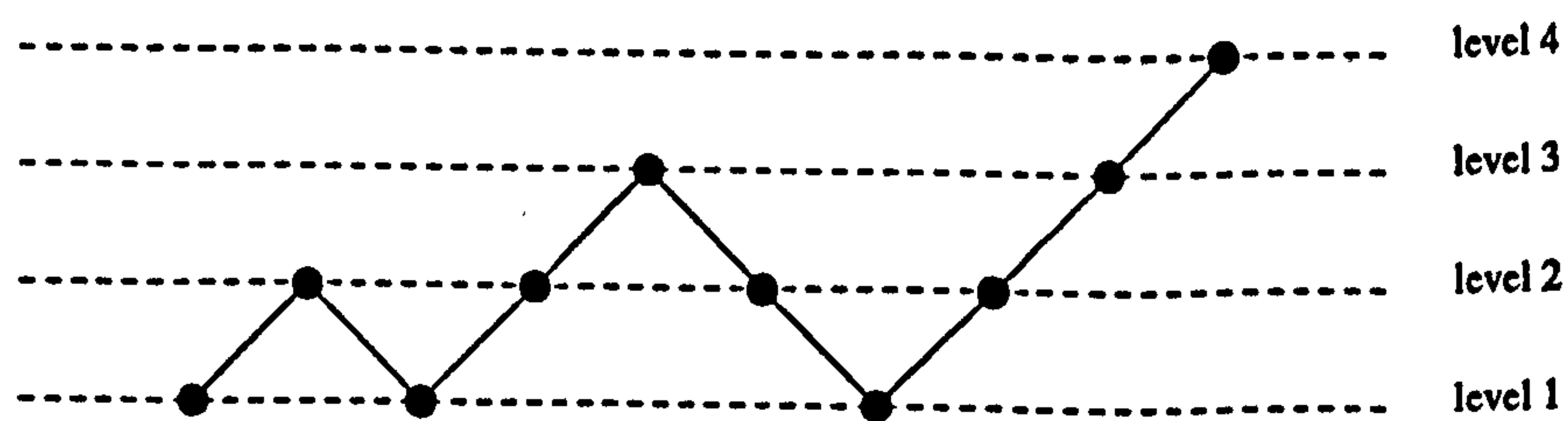


Figure 2.9: Four level multigrid FMG algorithm

2.5.5 Multigrid Convergence Rates for Linear Problems

It can be proven for a general linear problem that the convergence rate of the two-level algorithm is independent from the grid size. See Reference [84] for a detailed discussion and, for instance, Reference [78, pages 17-19] which gives one such proof in the context of finite element analysis. The fundamental concepts in the proof given in [84] are summarised below. Alternative theoretical approaches can be found in [81].

The two-level scheme given by the iteration matrix (2.31) is central to the analysis of multigrid iterations. Since multigrid methods converge in very few cycles, the spectral radius $\rho(M_h^{2h})$, which governs *asymptotic* convergence rates is not a satisfactory measure of the two level scheme behaviour. The proof relies on estimates of the matrix norm $\|M_h^{2h}\|$ where $\|\bullet\|$ refers to the Euclidean norm. Once a good approximate for $\|M_h^{2h}\|$ is available, the error $\|\tilde{u}^h - u^h\|$ can be bounded using the properties of matrix norms. The convergence rates are derived from two conditions:

- the *Smoothing Property* which expresses in terms of vector norms the fact that the relaxation procedure is really smoothing. Several procedures are investigated in Reference [84, pages 191–197].
- the *Approximation Property* which ensures that the fine grid correction is sufficiently well approximated by the coarse grid correction. This property is difficult to prove in the context of finite difference analysis but arises naturally from finite element formulations (see [84, pages 187–191]).

The multigrid iteration matrix is obtained by recursive application of the two grid iteration matrix, which can then be regarded as the two-level iteration matrix plus a perturbation. This formulation is then used to prove the grid independent convergence rates.

2.5.6 Full Approximation Storage Scheme

The correction storage scheme presented in Section 2.5.3 is only applicable to linear problems because the residual equation (2.26) on which it is based is only valid for such problems. The FAS (Full Approximation Storage) variant of multigrid methods is specifically designed to solve non-linear problems by extending the correction scheme.

Given a sequence of grids Ω^k with $k = 1, \dots, \kappa$, the solution u^κ to the non-linear system of algebraic equations:

$$\mathcal{L}^\kappa(u^\kappa) = f^\kappa.$$

is sought on the finest grid Ω^κ .

In FAS, as with CS, given an approximation \tilde{u}_n^κ of the solution, we seek to compute a correction $\delta\tilde{u}^\kappa$ to give the next approximation of the solution: $\tilde{u}_{n+1}^\kappa = \tilde{u}_n^\kappa + \delta\tilde{u}^\kappa$. The correction is assumed to contain only low frequency modes and consequently is computed on the next coarser grid $\Omega^{\kappa-1}$. This coarse grid correction is then interpolated back to the fine grid using a prolongation operator $I_{\kappa-1}^\kappa$.

The difference between the FAS and CS schemes arises from the definition of the coarse grid problem. In the CS scheme, the coarse grid correction is obtained by solving the residual equation. As a result, only the fine grid residuals need to be interpolated to the coarse grid. In FAS, the coarse grid problem has to be modified due to the non-linearity of the problem. Moreover, the discrete operators \mathcal{L}^h are now functions of the discrete solution u^k .

The following problem is therefore posed on the immediately coarser grid $\kappa - 1$:

$$\mathcal{L}^{\kappa-1}(\tilde{u}^{\kappa-1}) = f^{\kappa-1} \quad (2.36)$$

with

$$f^{\kappa-1} \equiv I_{\kappa}^{\kappa-1}(f^\kappa - \mathcal{L}^\kappa(\tilde{u}^\kappa)) + \mathcal{L}^{\kappa-1}(I_{\kappa}^{\kappa-1}\tilde{u}^\kappa) \quad (2.37)$$

Equation (2.37) is obtained by writing a fine grid equation for the correction $\delta\tilde{u}^\kappa$ to the current approximation \tilde{u}^κ , and transferring it to the coarser grid $\Omega^{\kappa-1}$ (see Section 2.5.7).

After (2.36) has been solved, a coarse grid correction defined as:

$$\delta\tilde{u}^{\kappa-1} \equiv \tilde{u}^{\kappa-1} - I_{\kappa}^{\kappa-1}(\tilde{u}^\kappa) \quad (2.38)$$

is computed and transferred back to the fine grid using the transfer operator $I_{\kappa-1}^\kappa$.

Here a two grid algorithm has been described. It is obvious that equation (2.36) can itself be solved more efficiently using the next coarser grid $\Omega^{\kappa-2}$. The recursion is exactly similar to that which has been discussed for the correction storage scheme. In particular, the V , W and F cycles are identical.

In order to implement a FAS method, it is therefore necessary to restrict both the residual and the approximation of the fine grid solution to the coarse grid. The operators may be different. \hat{I}_k^{k-1} will denote the operator operating on the solution, while I_k^{k-1} will denote, as above, the residual operator. In its general form, the coarse grid correction procedure for FAS is therefore:

$$\mathcal{L}^{k-1}(\tilde{u}^{k-1}) = f^{k-1} \equiv I_k^{k-1}(f^k - \mathcal{L}^k(\tilde{u}^k)) + \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{u}^k) \quad (2.39)$$

and

$$\tilde{u}^k \equiv \tilde{u}^k + I_{k-1}^k(\tilde{u}^{k-1} - \hat{I}_{k-1}^k u^k) \quad (2.40)$$

The coarse grid problem represented by equations (2.39) and (2.40) is consistent: as the fine grid residual $f^k - \mathcal{L}^k(\tilde{u}^k)$ tends to 0, f^{k-1} tends to $\mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{u}^k)$, the coarse

grid solution \tilde{u}^{k-1} tends to $\hat{I}_k^{k-1}(\tilde{u}^{k-1})$ and finally, the coarse grid correction tends to zero.

Hence, f^k (for $k < \kappa - 1$ at least) tends to a non-identically zero quantity: the *defect* which can be viewed as an approximation of the coarse grid truncation error [76, 10] (see Section 2.5.8). This property of FAS is central to the adaptive algorithms developed both in single phase (see [1]) and in the present study.

2.5.7 FAS and Non-Linearity

It is often claimed that FAS is directly applicable to non-linear problems and that global linearisations are not only unnecessary but also harmful to its efficiency [76, pages 80–84]. A very considerable body of evidence, particularly in the domain of fluid flows, indicates that this is indeed the case. The level of performance obtained with `pamg`, alone, would be strong evidence.

First, we derive the formulae for FAS. For a non-linear problem, the residual equation written on the fine grid Ω^k is:

$$\mathcal{L}^k(u^k) - \mathcal{L}^k(\tilde{u}^k) = f^k - \mathcal{L}^k(\tilde{u}^k) = r^k$$

A correction δu^k is sought such that $\tilde{u}^k + \delta u^k \approx u^k$, therefore the fine grid equation for the correction is:

$$\mathcal{L}^k(\tilde{u}^k + \delta u^k) - \mathcal{L}^k(\tilde{u}^k) = f^k - \mathcal{L}^k(\tilde{u}^k) = r^k$$

This equation is then transferred to the coarse grid Ω^{k-1} : \mathcal{L}^k , \tilde{u}^k and r^k are replaced by \mathcal{L}^{k-1} , $\hat{I}_k^{k-1}\tilde{u}^k$ and $I_k^{k-1}r^k$ respectively. Setting $\tilde{u}^{k-1} \equiv \hat{I}_k^{k-1}\tilde{u}^k + \delta u^{k-1}$, the FAS algorithm is then obtained.

FAS and CS are exactly equivalent for linear problems. FAS can be thought of as a mechanism for approximating the non-linear operator \mathcal{L}^{k-1} which corresponds to the fine grid problem (see Section 2.5.8). In this sense, it is rigorously valid. When \mathcal{L} is non-linear, the discrete operators \mathcal{L}^{k-1} and \mathcal{L}^k are not known. As the computation proceeds good approximations are obtained, but it remains that the coarse grid correction may not constitute a very good fine grid correction if the non-linearities are strong. Furthermore, the process of approximating the non-linear operators will require time.

2.5.8 Defect, Truncation Error and Adaption in the FAS Framework

One can view both the CS and FAS schemes as procedures for resolving long wavelengths on coarse grids. FAS, however, conveys supplementary information about

the differential equations being solved. According to this dual interpretation [76, page 82–83], the fine grid can be viewed as a device to calculate the correction to the coarse grid solution.

First, the defect τ_k^{k-1} is introduced and defined as:

$$\tau_k^{k-1} \equiv \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{u}^k) - I_k^{k-1}(\mathcal{L}^k\tilde{u}^k). \quad (2.41)$$

Assuming the restriction operator I_k^{k-1} is linear, the FAS correction scheme can be rewritten in terms of the defect as:

$$\mathcal{L}^{k-1}\tilde{u}^{k-1} = f^{k-1} + \tau_k^{k-1} \quad (2.42)$$

where $f^{k-1} = I_h^{k-1}f^k$. At convergence, $\tilde{u}^{k-1} = \hat{I}_k^{k-1}\tilde{u}^k$ so that τ_k^{k-1} can be interpreted as a fine to coarse grid correction designed to make the two solutions fit together.

τ_k^{k-1} can also be interpreted as an approximation of the truncation error on the grid $\kappa - 1$ where κ is the finest grid available. The truncation error on a grid $\Omega^{\kappa-1}$ is defined as:

$$T^{\kappa-1} \equiv \mathcal{L}^{\kappa-1}(\tilde{u}^{\kappa-1}) - \mathcal{L}(u)$$

If the computational error:

$$f^\kappa - \mathcal{L}^\kappa(\tilde{u}^\kappa)$$

is small enough so that it can be neglected compared with other errors, then:

$$\mathcal{L}^{\kappa-1}(\tilde{u}^{\kappa-1}) \approx \mathcal{L}^{\kappa-1}(\hat{I}_\kappa^{\kappa-1}\tilde{u}^\kappa),$$

using (2.37) written for the converged solution.

On the grid $\Omega^{\kappa-1}$ alone, the following approximation can also be performed:

$$[\mathcal{L}(u)]^{\kappa-1} \approx I_\kappa^{\kappa-1}(\mathcal{L}^\kappa\tilde{u}^\kappa).$$

This is not applicable to coarser grids Ω^k with $1 \leq k < \kappa - 1$ because the solution \tilde{u}^{k+1} converges to a corrected version of the original differential equation.

In effect, for the purpose of approximating the truncation error of the grid $\Omega^{\kappa-1}$, the difference between \tilde{u}^κ and u is neglected, together with the difference between the non-discrete operator \mathcal{L} and its discretisation \mathcal{L}^κ on the grid Ω^κ .

Hence,

$$\tau_\kappa^{\kappa-1} \approx T^{\kappa-1}$$

and the defect can be interpreted as an approximation of the truncation error on the grid $\Omega^{\kappa-1}$. Also, note that $f^{\kappa-1} = \tau_\kappa^{\kappa-1}$ if $f^\kappa = 0$ (which can always be imposed by applying a translation to the operator \mathcal{L}^κ).

In other words, the FAS scheme intrinsically provides an estimate of the truncation error at no extra computational cost. This estimate can be used as a reliable error estimator to design a robust refinement algorithm ⁶ (see for instance [1]).

The FAS method appears to be ideal for adaptive computations on two counts:

1. At no extra cost, it provides an estimate of the truncation error which can be used for accurate error control via grid refinement.
2. As the grid is refined, the deterioration of the convergence rate is either eliminated for linear problems, or at the very least considerably alleviated, thus minimising the numerical cost of adaption.

The implementation of adaption in `pamg` and `pamg-multiphase` is somewhat complex (see Sections 4.2.4 and 4.3.5 together with [1, 85]) and, crucially, exerts an influence on the definition of the interpolation operators, particularly for single phase flows. Reference [86] discusses similar issues.

2.5.9 Multigrid Methods for Non-Elliptic PDEs

Multigrid methods have been primarily developed for elliptic boundary value problems but they were soon adapted for other types of differential equations. First and foremost, multigrid methods were used to obtain steady state solutions to the Euler equations of gas dynamics (see for instance [87, 88, 89]). In transient applications where implicit schemes are competitive i.e. when the solution is time-marched to steady state or when the problem is such that sonic velocities are much larger than convective velocities (see Section 2.3), multigrid solvers also offer benefits.

The success of multigrid methods crucially depends upon the discretisation applied to the (continuous) differential equations. The discretisation should be *h-elliptic*. This is defined by Brandt in [10]. Another discussion can be found in [76]. The concept, which involves the symbols of the discretisation, may appear rather technical but, as explained in [90], the basic idea is that a discretisation with good *h-ellipticity* is one which introduces strong local coupling between variables so that relaxation smoothers can quickly reduce the high frequency components of the error. It is therefore quite logical that good multigrid convergence depends on the measure of *h-ellipticity* of the discretisation.

Mulder (references [89, 14]) showed that multigrid methods for hyperbolic equations *in multi-dimensions* can be hindered by a breaking down of *h-ellipticity* in one of the coordinate directions. This phenomena is referred to as *Strong Alignment* by Brandt [10, page 27], and is common in anisotropic problems such as $u_{xx} + \epsilon u_{yy} = 0$ for

⁶i.e. an algorithm which never refines in regions where refinement leads to inaccuracies.

$\epsilon \rightarrow 0$. Strong alignment is not due to a problem being locally one-dimensional but rather is caused by a problem being one-dimensional in one of the grid directions.

In such cases, high frequencies in the transverse directions will not be smoothed by relaxation on the fine grid since variables are decoupled in the transverse direction. Furthermore, they will not be removed by the coarse grid correction since it will either be strongly aliased on the coarse grid, or will cancel out due to the averaging introduced by the restriction operators. Convergence will usually be forced by the propagation of boundary data to the interior but this is a highly grid-dependent process and as a result, grid independence convergence rates are lost.

Mulder in reference [90] proposes two remedies for this undesirable phenomenon: (i) global relaxation schemes, such as line relaxation or Incomplete LU factorisation, which are exact in the case of strong alignment; or (ii) *semi-coarsening* where restrictions are only performed in the direction of the strongest coupling so that aliasing is avoided in the transverse direction. The error in the transverse direction is finally removed by an exact solver on the coarsest grid.

In Sections 5.4.13 and 5.4.14, we investigate novel but similar phenomena in the context of the multi-fluid equations.

2.6 Adaptive Solution of Partial Differential Equations

There are many applications in fluid dynamics where the solutions exhibit small regions of very rapid variations which are often crucial for determining the overall features of the flow. Shocks in Aerodynamics and boundary layers in Hydrodynamics are typical examples.

In such cases, adaptive computations are very attractive because they allow large savings of both computing time and storage. The local grid spacing is adjusted so that the resolution is optimal i.e. just fine enough to accurately resolve the features of the solutions without incurring a large computational cost. In this framework, it is also possible to bound the truncation error of the discrete solution.

In the author's view, adaption could be also be very beneficial to the simulation of multiphase flows in both the steady and transient cases. In particular, slug flows in long pipelines have many interesting properties from the point of view of adaption (see [91]). The present thesis deals with the steady case, and work is currently undertaken for transient, one-dimensional multiphase flows [37].

The topic of adaptive techniques for the solution of partial differential equations is another area about which a vast amount of literature is available. References [92, 93, 86, 94] are examples of the studies carried out. The main types of differ-

ential equations, elliptic [95], parabolic [96] and hyperbolic [93], are all amenable to adaptive computations but the details of the implementation will vary greatly according to the type of equations. In transient simulations, space adaption will often be complemented by *local time stepping* where different time steps are taken at different locations on the grid. This is implemented either on physical grounds, if there are regions where the solution varies faster in time than in others, or on numerical grounds to ensure the stability of the integration schemes. Theoretical results exist proving the convergence of such methods in both the hyperbolic [97] and parabolic [96] cases.

In this section, the aim is to formulate some key issues as found in reference [1] which discuss the implementation of adaption in *pamg*. See also reference [12]

A key objective is to perform grid refinement in an *automatic* fashion whilst the computations proceed, rather than as a pre-processing step which is dependent on user expertise. This feature also allows automatic error control.

The goal of automatic and dynamic gridding raises the question of accurately estimating the errors in an approximation to the solution sought. In many cases, simple indicators such as solution gradients are used because of their low computational costs. In *pamg*, however, the FAS multigrid defect is used to estimate the error (see Section 2.5.8). Reference [12] shows that the defect performs very well as an error estimator for single phase flows. Furthermore, in the FAS framework, its computational cost is very low.

Two other important issues also arise:

1. How is the interface between grids treated and what are the implications in terms of accuracy? Reference [1] shows that for incompressible Navier-Stokes flows it is very important that the fluxes should be conserved across grid interfaces. In particular, if mass fluxes are not conserved, the solution algorithm will not converge because the residual vector does not entirely lie in the range of the differential operator.

Obviously, it is also important that grid refinement should occur only in regions where the solution varies relatively slowly, so that the unavoidable interpolations performed at grid boundaries do not affect overall accuracy.

2. What are the data structures required to implement adaption efficiently? Adaptive codes are by essence considerably more complex than codes which use uniform grids. In order to draw maximum benefits from adaption, it is important to minimise the associated overheads (grid management, solution transfers and error estimations). As noted above, the multigrid framework is very suitable to adaptive techniques and in *pamg*, the code complexity was reduced by considering structured grids decomposed in quadrants which are self-similar under refinement.

2.7 Conclusions

In this chapter, mathematical models for multiphase flows have been reviewed, paying particular attention to the multi-fluid equations because of their generality. These are the equations solved by `pamg-multiphase`. The multi-fluid equations are obtained by averaging. This methodology has been illustrated for a simple two-phase flow. Issues pertaining to their numerical resolution have also been discussed: although their form is similar to that of the single phase Navier-Stokes equations, the multi-fluid equations are actually significantly more complex due to (i) their degree of non-linearity; (ii) the presence of source terms; and (iii) restrictions on physically admissible solutions. Another important property is that the multi-fluid equations are ill-posed in the unsteady case: error growth is allowed due to complex eigenvalues. This is clearly irrelevant for *steady* flows but other types of ill-posedness are possible, particularly when the flow contains recirculation zones.

The main objective of the present thesis is to design an efficient and robust algorithm for the simulation of steady flows. The mathematical concepts undergirding the local quasi-Newton coupled multigrid approach have been reviewed. Local quasi-Newton solvers have been compared to more traditional segregated solvers such as the SIMPLE procedure, and shown to accurately take the local coupling of variables into account. It has also been established that multigrid methods, for their part, allow an efficient treatment of the long-wavelength modes in the solution, by resolving them on the coarsest possible grid. Local quasi-Newton coupled solvers and multigrid methods are therefore naturally complementary.

The fundamental concepts of multigrid have been reviewed and illustrated. These include the two-level scheme and its effects on error modes, the possible cycling strategies and multigrid convergence rates. The variant of multigrid which is considered here is the FAS method. It has two desirable features: (i) it is directly applicable to non-linear problems; and (ii) it provides estimates for the truncation error of the discretisation which can actually form the basis for *automatic* grid refinement. Some limitations of multigrid methods are also given. Although FAS is applicable to non-linear problem, good multigrid convergence rates depend upon the fine and coarse grid operators and solutions being reasonably close to each other. Another potential problem arises for hyperbolic and strongly anisotropic elliptic systems of equations in multi-dimensions. That is, if the problem becomes locally one-dimensional and aligned with the grid, then the convergence of the multigrid method will be greatly degraded. Finally, automatic grid refinement and error control have been discussed. Key issues for the successful implementation of a good adaptive algorithm — flux conservation, error estimation and the design of efficient data structures — have been highlighted.

Multiphase flows are highly non-linear. It is therefore likely that the local coupling between flow variables is even more important than in the single phase case. Quasi-Newton solvers may therefore be very beneficial. As in the single phase case, it is

natural to embed it in a multigrid method. The non-linearity of the equations is not contradictory to the application of multigrid although complex multiphase solutions may appear very different on grids of different resolution, which may slow down convergence.

In this review, the applicability of a quasi-Newton coupled multigrid approach to multiphase simulation has been established. In the next chapter, the multi-fluid equations are discretised. Solution algorithms for both the single and multi phase cases are then described. Their accuracy and efficiency will then be assessed.

Chapter 3

Governing Equations and their Discretisation

3.1 Issues and Choices Made in Discretisation

Physical models for fluid flows are usually formulated, in an Eulerian framework, in terms of partial differential equations — the *governing equations* — which express the physical principle that mass, momentum and energy are conserved during the motion. Their solutions are continuous functions. By contrast, numerical analysis codes solve discrete algebraic equations. *Discretisation* is the process by which a correspondence between the two representations is established and is therefore the very first stage in the numerical simulation of a physical phenomena. Finite difference analysis and finite element analysis, for instance, are means of replacing the continuous problem by a discrete one whose solution is close to the solution of the original problem.

A good discretisation possesses two major properties:

- It is accurate in the sense that it minimises the truncation error introduced by representing the continuous problem on a discrete grid. An effective discretisation will ensure high accuracy with compact molecules.
- It leads to algebraic equations which can be solved quickly and reliably, due to inherited properties such as monotonicity.

This chapter discusses the equations — both discrete and continuous — which are solved by `pamg` and `pamg-multiphase`, as well as the specific methodology used to perform the discretisation. Discrete difference equations are obtained by applying the finite volume method to the governing partial differential equations. This ensures, amongst other things, that the principle of conservation is automatically

satisfied for the discrete problem as well as the continuous problem. The discretisation is performed, for both single phase and multi-phase flows, on staggered grids which are well suited to the computation of incompressible flows. The hybrid scheme introduced by Spalding [98] (see also, for instance, [99]) is used to derive discrete equations which can be accurate up to order 2 without generating non-physical oscillations. Both the use of staggered grids and hybrid schemes are design choices which are key factors for the robustness and efficiency of the solution algorithms used in `pamg` and `pamg-multiphase`. They are discussed in the remainder of this section. In particular, an argument to explain the robustness of the solution algorithms is given, based on the use of hybrid schemes in a multigrid framework.

Secondly, the discrete equations implemented in the single phase code `pamg` are derived in detail. This facilitates the subsequent treatment of the multiphase equations which follows an identical methodology. Although in theory it does not raise any difficulty, the discretisation of the equations which constitute the multi-fluid model needs to be executed with extreme care, and was indeed an important task to be performed in the course of this research.

The single and multi-phase equations differ in some significant aspects which are highlighted in the course of this chapter. Attention is devoted to the treatment of the volume fractions and source terms. Furthermore, it has been noticed during this project that the diffusive fluxes in the multi-phase momentum equation introduces cross derivative terms which are hyperbolic-like. In view of Section 2.5.9, which discusses the limitations of multigrid methods for hyperbolic problems, these cross derivative terms may have significant effects on the convergence rates of the `pamg-multiphase` multigrid solver. These cross-derivative terms are discussed in detail in Section 3.3.2, in anticipation of the experimental results presented in Sections 5.4.13 and 5.4.14.

3.1.1 Advantages and Disadvantages of Staggered Grids for Fluid Flow Computations

Staggered grids (introduced in reference [54]) where velocities and pressures are not defined at the same grid points, have been used for some time in computational fluid dynamics. If collocated grids are used for the simulation of incompressible flows, then one observes that with the widely used central differencing, the pressure field and the velocity field are numerically decoupled at every grid point (see [58, pages 666–669]). This decoupling generates high frequency oscillations which have to be removed by the addition of an artificial diffusion term in the continuity equation.

Staggered grids remedy this problem by ensuring that consecutive pressure grid points are used to approximate pressure gradients. Hence, they remove the need for the artificial viscosity. This is very satisfactory since the artificial viscosity coefficient is an empirical parameter which has no physical basis and yet affects the

solution of the equations. Used in connection with multi-grid methods, staggered grids, however, raise the difficulty of having to use different transfer operators for different quantities. In other words, several grids have to be transferred from one level to the other as opposed to just transferring one. For similar reasons it is also harder to deal with staggered grids for complex geometries. These limitations are balanced by the fact that staggered grids are very useful in the context of adaptation. They greatly facilitate the imposition of the constraint that mass fluxes across grid interface should be conserved. That is, on staggered grids, the discrete single phase continuity equations can be solved exactly so that global mass conservation can (and must) be rigidly enforced. Furthermore, the author's experience also suggests that the use of staggered grids is a key factor in the success of coupled solvers. It can be argued that the discretisation on a staggered grid improves the conditioning of the local (numerical) Jacobians.

The benefits derived from the use of a staggered grid for multiphase flow are less obvious. The decoupling of the pressure and velocity fields is avoided but it has been observed that certain discretisations of the continuity equations are unstable (see Section 4.3.2). Furthermore, mass conservation across grid interfaces can be problematical (see Section 4.3.3) and requires special interpolation operators for grid transfers.

Figures 3.1 and 3.5 show the staggered grids used in `pamg` and `pamg-multiphase` respectively.

3.1.2 Finite Volume Discretisation Using Hybrid Schemes

Many transport problems involve both convection and diffusion mechanisms. For example the transport of momentum and energy in fluid flows falls neatly in that category. Mathematically, the equations contain both first and second order spatial derivatives.

When steady state solutions are sought, it is well known that depending on the size of the grid and the differencing scheme used, non-physical oscillations can be generated because the schemes are not monotonic i.e. they can introduce new extrema in the approximate solution of the equations being solved ¹. Upwind schemes [98] aim to always be monotonic and of optimal order of accuracy.

The development of more complicated non-linear schemes which are monotonic and more accurate (in practice) than the upwind scheme is possible. However, the principal advantage of hybrid schemes lies in their simplicity and in the compactness of their computational molecule. In the remainder of this section, we discuss the

¹It can be proved that a linear monotonic scheme is necessarily only first order accurate (see for instance [58, Chapter 21]). Therefore all linear second-order accurate, or higher, are non-monotonic whether they are centred or upwinded.

properties of hybrid schemes following and expanding reference [98].

A simple model problem is provided by the convection and diffusion of heat in a one-dimensional media. This physical phenomenon can be considered to be governed by the following non-dimensional ordinary differential equation:

$$P \frac{dT}{d\xi} - \frac{d^2T}{d\xi^2} = 0. \quad (3.1)$$

The Peclet number P is defined as:

$$P = \frac{c_p G L}{\lambda}.$$

It measures the relative weight of convection and diffusion for a characteristic length L . c_p is the constant pressure heat capacity of the material, G is the mass flow rate of material per unit cross sectional area, and λ is the thermal conductivity of the material.

When solutions of equation (3.1) are sought on discrete grids, an important parameter is the *cell-Peclet number* defined as:

$$P_c = \frac{c_p G \Delta x}{\lambda} = P \Delta \xi$$

In other words, the cell-Peclet Number corresponds to taking the characteristic length equal to the grid size $\Delta \xi$.

In the case of fluid flow, it is the convection and diffusion of *momentum* which are of interest. The situation is very parallel, except that the Peclet numbers are replaced by the Reynolds number:

$$Re = \frac{\rho u L}{\mu} \quad (3.2)$$

and cell-Reynolds number:

$$Re_c = Re \Delta \xi$$

respectively. The numerator in (3.2) measures the magnitude of inertia forces while the denominator is proportional to the viscous forces. The analysis for fluid flow, however, is complicated by the fact that the convective terms become non-linear.

In the framework of finite differences, equation (3.1) can either be discretised by central differencing or by upwind differencing. The former is second order accurate while the latter is only first order accurate. It is a well-known fact that central differencing generates non-physical oscillations in the discrete solution if $|P_c| > 2$. This can be established by considering the coefficients of the discretisation (see below). The oscillations are correct solutions of the discrete equations but violate the maximum principle which operates on the governing ODE.

By contrast, upwind differencing leads to oscillation-free solutions because it introduces *numerical diffusion* [100, pages 280–281] in the discrete equations which

removes any oscillation in the solution. Another way to look at this problem is to assert that upwind differencing is unconditionally monotonic while central differencing is only monotonic for $|P_c| < 2$. If coarse grids are used, that is if $|P_c| > 2$, upwind differencing can be preferable even though its order of accuracy is less. The fundamental idea behind hybrid differencing is to automatically switch between upwind and central differencing depending on the value of the cell-Peclet number, or its equivalent for other problems, so that a monotonic discretisation with “optimal” accuracy is always obtained.

Before applying hybrid differencing to the model equation (3.1), the advantages and disadvantages of hybrid schemes are discussed, particularly in the context of multigrid methods. There are two limitations associated with hybrid schemes [101]:

- Firstly, the presence of numerical diffusion may remove features from the solution, which are physically significant. Strictly speaking, the hybrid scheme is second-order accurate since the order of accuracy is obtained by taking a limit for vanishing grid spacing, for which it is equivalent to the central scheme. However, the point of introducing an hybrid scheme is to be able to use coarse grids and in these circumstances, the hybrid scheme is only first order accurate.
- Secondly, in multi-dimensional problems, the hybrid approach does not work well if the angle between the main direction of the flow and the grid is close to 45 degrees, since strong numerical diffusion is then introduced in all grid directions.

A possible approach is to adopt a second order upwind scheme such as QUICK [99], at the cost of extending the stencil for the formulation of the finite differences. It should be noted however that QUICK (together with other second order – one-sided – upwind schemes) is not monotonic and will therefore generate some oscillations and overshoots.

Within the framework of multigrid methods, the appraisal of hybrid schemes is somewhat different. Independent of the size of the finest grid, multigrid methods usually involve very coarse grids on which central differencing would be unstable. The choice of an hybrid scheme is very attractive in this context on two accounts: (i) the implementation is simple and only involves a three-point stencil, and (ii) the order of accuracy of the solution is not altered. This is because the coarse grid computations only act as an accelerating device for the determination of the “true” solution defined on the finest grid. If practical, the final grid will be chosen so that it is sufficiently fine for central differencing to be stable.

When used in a multigrid framework, hybrid schemes can be considered to be, *in practice*, second order accurate. The numerical diffusion added on coarse grids may have an adverse effect on the convergence rates of the multigrid method since coarse grid solutions may be very different from the fine grid solutions. However, one crucial point for the success of multigrid methods is to avoid high frequency oscillations in

the solution on coarser grids ², a property which hybrid differencing provides at very little cost.

Appendix A gives some results for the application of hybrid schemes to a one-dimensional two-phase flow.

Finite Volume Discretisations using Hybrid Schemes Finite volume discretisations are based on the integral formulation of the governing partial differential equations. The central concept is therefore that of (numerical) *flux*.

For the model problem, equation (3.1) can be integrated and the result is:

$$Q(\xi) = PT - \frac{dT}{d\xi} \quad (3.3)$$

$Q(\xi)$ is the heat flux at the point ξ . A discretisation can be obtained by (i) approximating Q in terms of the discrete grid functions, and (ii) using the principle of the conservation of heat (i.e. the fact that through a control volume, the total flux of heat is equal to the heat sources in the control volume). Such an application of conservation principles lies at the heart of the finite volume approach.

Consider a uniform grid of size $\Delta\xi$. The temperature T_i is defined at cell centres. For the cell (i), a finite volume discretisation of (3.3) is:

$$\frac{Q_{i+1/2} - Q_{i-1/2}}{\Delta\xi} = 0 \quad (3.4)$$

$Q_{i+1/2}$ and $Q_{i-1/2}$ can be approximated using the temperatures T_{i-1} , T_i , T_{i+1} . Different approximations will make equation (3.4) equivalent to finite difference formula obtained by central and upwind differencing.

The first derivative in equation (3.3) is approximated by central differences:

$$\left[\frac{dT}{d\xi} \right]_{\xi=i+1/2} \approx \frac{T_{i+1} - T_i}{\Delta\xi}$$

There are two possibilities for the approximation of the convective term PT : geometric (centred) interpolation of T or upwind interpolation.

If geometric interpolation is chosen,

$$T_{i+1/2} \approx \frac{T_i + T_{i+1}}{2}$$

The heat fluxes on the cell edges are therefore:

$$Q_{i-1/2} = P \frac{T_{i-1} + T_i}{2} + \frac{T_i - T_{i-1}}{\Delta\xi}$$

²Oscillations on the coarser grids imply that low frequency components are not removed by the coarse grid correction.

$$Q_{i+1/2} = P \frac{T_i + T_{i+1}}{2} + \frac{T_{i+1} - T_i}{\Delta\xi}$$

Equating the two fluxes, and dividing by $\Delta\xi$ yields:

$$P \frac{T_{i+1} - T_{i-1}}{2\Delta\xi} - \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta\xi^2} = 0. \quad (3.5)$$

Using a finite difference approach and working from Taylor's series leads to an equivalent result.

If upwind interpolation is chosen, the direction in which the heat is convected is important. If the heat flow from ξ_{i-1} to ξ_{i+1} , we take:

$$T_{i-1/2} \approx T_{i-1} \quad T_{i+1/2} \approx T_i.$$

The fluxes are:

$$Q_{i-1/2} = PT_{i-1} + \frac{T_i - T_{i-1}}{\Delta\xi}$$

$$Q_{i+1/2} = PT_i + \frac{T_{i+1} - T_i}{\Delta\xi}$$

which result in the following discretisation:

$$P \frac{T_i - T_{i-1}}{\Delta\xi} - \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta\xi^2} = 0. \quad (3.6)$$

If the heat flows in the opposite direction,

$$T_{i-1/2} \approx T_i \quad T_{i+1/2} \approx T_{i+1}.$$

The fluxes are:

$$Q_{i-1/2} = PT_i + \frac{T_i - T_{i-1}}{\Delta\xi}$$

$$Q_{i+1/2} = PT_{i+1} + \frac{T_{i+1} - T_i}{\Delta\xi}$$

which result in the following discretisation:

$$P \frac{T_{i+1} - T_i}{\Delta\xi} - \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta\xi^2} = 0.$$

Standard (finite difference) schemes are again obtained. This is usually true on uniform grids.

Another key concept is that the direction of the heat flow at $\xi_{i\pm 1/2}$ is given by the sign of P : Physically, P is proportional to the mass flow rate of material in the positive x -direction. If $P > 0$ the heat flow from ξ_{i-1} to ξ_{i+1} . If $P < 0$, the heat flows from ξ_{i+1} to ξ_{i-1} .

Hybrid differencing is obtained by choosing different expressions for the fluxes according to the value of P :

- if $|P| < 2/\Delta\xi$, convective effects are not dominant and centred differences are used. In other words, we take:

$$Q_{i+1/2} = P \frac{T_i + T_{i+1/2}}{2} + \frac{T_{i+1} - T_i}{\Delta\xi}$$

- if $P_c = P\Delta\xi > 2$, the diffusive term is ignored because convective effects are dominant. Because heat flows from ξ_i to ξ_{i+1} ,

$$Q_{i+1/2} = PT_i$$

- if $P_c = P\Delta\xi < 2$, the diffusive term is also ignored but as heat flows from ξ_{i+1} to ξ_i , we take:

$$Q_{i+1/2} = PT_{i+1}$$

In the last two cases, the physical diffusion is neglected in the discrete approximation of the fluxes. This is justified on the grounds that the numerical diffusion introduced by upwinding totally dominates the physical diffusion. Our model problem is linear and P is a constant. In non-linear problems the convective term depends on the flow solution but upwinding is conceptually identical.

Assuming Dirichlet boundary conditions, the central difference scheme can be represented by the following matrix:

$$\frac{1}{\Delta\xi} \begin{pmatrix} -\frac{P}{2} - \frac{1}{\Delta\xi} & \frac{2}{\Delta\xi} & \frac{P}{2} - \frac{1}{\Delta\xi} & & & & \\ & \frac{2}{\Delta\xi} & \frac{P}{2} - \frac{1}{\Delta\xi} & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & -\frac{P}{2} - \frac{1}{\Delta\xi} & \frac{2}{\Delta\xi} & \frac{P}{2} - \frac{1}{\Delta\xi} & \\ & & & & -\frac{P}{2} - \frac{1}{\Delta\xi} & \frac{2}{\Delta\xi} & \end{pmatrix} \quad (3.7)$$

For $|P| > 2/\Delta\xi$, i.e. $|P_c| > 2$, the matrix is non-longer diagonally dominant, which explains why the maximum principle is no longer respected by the discrete solution. This can be easily shown by the following argument:

Given T_{i+1} and T_{i-1} , a non-oscillatory solution will be achieved if the maximum principle is respected, that is, if either:

$$T_{i+1} > T_i > T_{i-1}$$

or:

$$T_{i-1} > T_i > T_{i+1}$$

are true. Assume $T_{i+1} = 0$ and $T_{i-1} = 1$, then the solution of (3.5) for T_i is:

$$\begin{aligned} T_i &= \frac{1}{2} \left(\frac{P}{2} \Delta\xi + 1 \right) \\ &= \frac{1}{2} \left(\frac{P_c}{2} + 1 \right). \end{aligned}$$

If $P_c > 2$, then $T_i > 1$, while if $P_c < -2$, $T_i < 0$. In either case, the solution becomes oscillatory. On the other hand, if $-2 < P_c < 2$, then $0 < T_i < 1$ and the maximum principle is satisfied.

If upwind differencing is used and $P > 0$, the equivalent matrix is:

$$\frac{1}{\Delta\xi} \begin{pmatrix} P + \frac{2}{\Delta\xi} & -\frac{1}{\Delta\xi} & & & & & \\ -P - \frac{1}{\Delta\xi} & P + \frac{2}{\Delta\xi} & -\frac{1}{\Delta\xi} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & -P - \frac{1}{\Delta\xi} & P + \frac{2}{\Delta\xi} & -\frac{1}{\Delta\xi} & & \\ & & & -P - \frac{1}{\Delta\xi} & P + \frac{2}{\Delta\xi} & -\frac{1}{\Delta\xi} & \\ & & & & -P - \frac{1}{\Delta\xi} & P + \frac{2}{\Delta\xi} & \\ & & & & & & \ddots \end{pmatrix} \quad (3.8)$$

which is (weakly) diagonally dominant for all positive value of P . If $T_{i-1} = 1$ and $T_{i+1} = 0$, the solution of equation (3.6) is:

$$T_i = \frac{P + \frac{1}{\Delta\xi}}{P + \frac{2}{\Delta\xi}}$$

which is bounded by 0 and 1 for all positive values of P so that the maximum principle is always satisfied.

3.1.3 Solver Robustness

Having discussed the properties of the hybrid scheme, we investigate its influence on the performance of the solver. The single phase solver `pamg` has proved to be very robust: the probability that the algorithm will converge is very high even in cases where there are several distinct solutions (see for instance the sudden expansion problem in reference [1]). In particular, convergence is not dependent on the initial guess. This is a crucial advantage for any code since it reduces the amount of educated guess needed to obtain a solution. Unfortunately, many iterative solvers are very sensitive to the quality of the initial guess.

The reason why `pamg` and `pamg-multiphase` are robust is that they incorporate an *implicit continuation method*. In the context of inviscid flows, a continuation method involves obtaining different solutions for decreasing values of the viscosity and taking the limit for zero viscosity. Continuation methods are highly stable due to the influence of the viscosity — which damps errors — without being inaccurate since a limit is taken. The combination of the multigrid algorithm and hybrid differencing leads to a similar effect since the computations start on the coarsest grid. As grids get finer, less and less artificial viscosity is added by hybrid differencing.

3.2 Discretisation of the Steady Navier-Stokes Equations for an Incompressible Fluid

3.2.1 Governing Equations and Preliminary Remarks

The two-dimensional steady flow of an incompressible viscous Newtonian fluid is governed by the following form of the Navier-Stokes equations (see for instance [44]):

- Conservation of mass:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.9)$$

- Conservation of horizontal momentum:

$$\frac{\partial uu}{\partial x} + \frac{\partial vu}{\partial y} - \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = -\frac{\partial p}{\partial x} \quad (3.10)$$

- Conservation of vertical momentum:

$$\frac{\partial uv}{\partial x} + \frac{\partial vv}{\partial y} - \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) = -\frac{\partial p}{\partial y} \quad (3.11)$$

In this section, these governing equations are discretised on a staggered grid (shown in Figure 3.1) where the velocities are defined at the edges of cells and the pressure at the centre. A finite volume approach is followed and hybrid schemes are used. The result is the difference equations implemented in `pamg`.

The continuity equation (3.9) does not pose any problems. In fact, because the discretisation is performed on a staggered grid, the discrete *single-phase* continuity equation can be solved exactly for each cell without the need for any interpolation. Hybrid schemes are introduced in the momentum equations.

In the derivation of multi-dimensional finite volume discretisations, an important tool is *Gauss' Theorem*, which relates the volume integral of the divergence of a vector field to its flux through the enclosing surface. Mathematically, given a closed volume \mathcal{V} and its boundary surface \mathcal{S} , then for any vector field \mathbf{f} differentiable over \mathcal{V} ,

$$\int_{\mathcal{V}} \nabla \cdot \mathbf{f} dv = \int_{\mathcal{S}} \mathbf{f} \cdot \hat{\mathbf{n}} ds. \quad (3.12)$$

In this formula, $\hat{\mathbf{n}}$ is the *outward* unit normal to the surface \mathcal{S} . See [102] for more details.

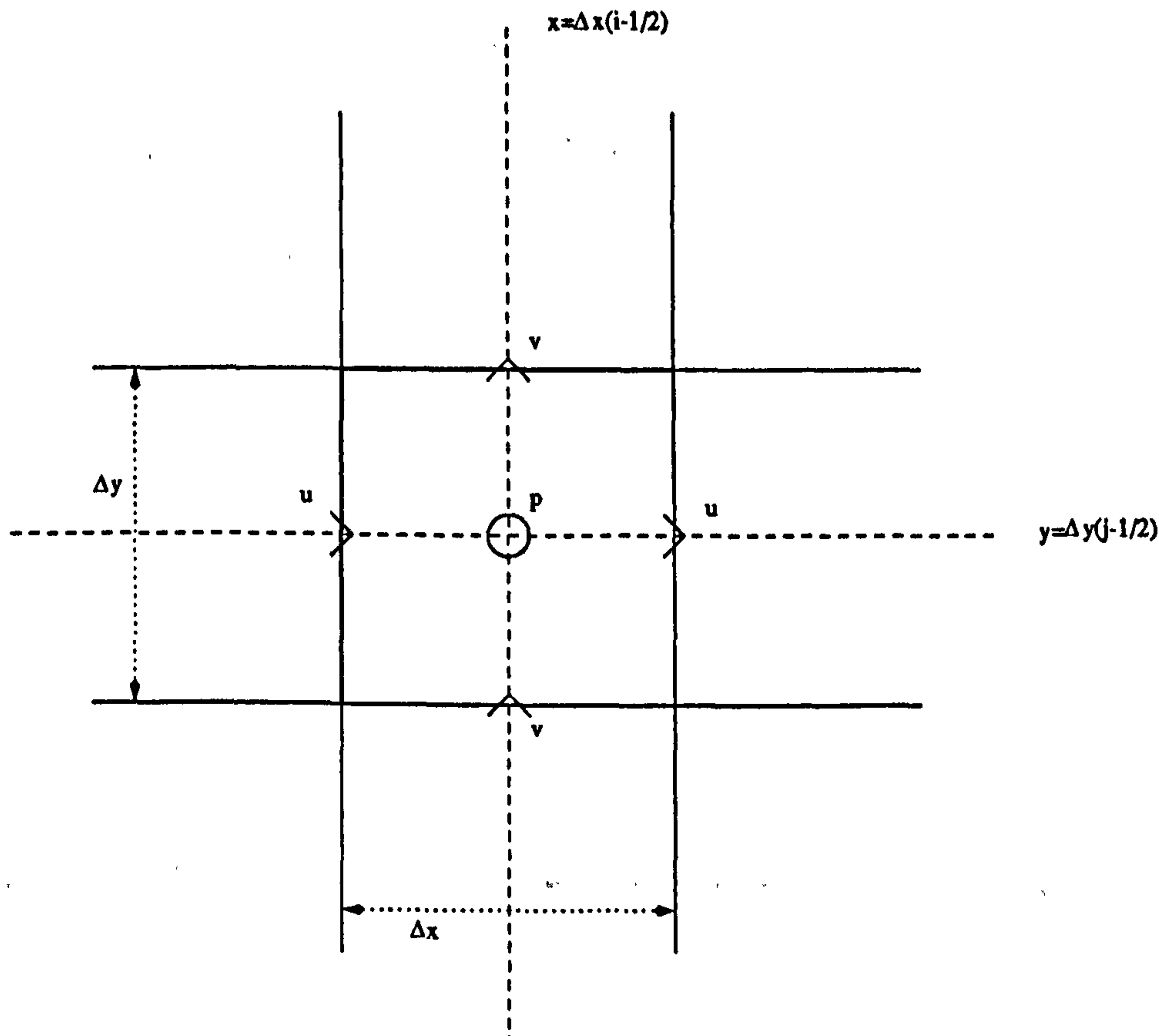


Figure 3.1: Staggered grid used for the discretisation of the Navier-Stokes equations

3.2.2 Discretisation of the Continuity Equation

The derivation by integral means of the finite difference formula expressing the conservation of mass is quite straight forward: equation (3.9) is integrated over one cell (as shown in Figure 3.2) and the integral scaled by the area of the cell, that is $1/(\Delta x \Delta y)$.

The scaling makes the discrete equations more suitable to multigrid methods because the residuals of the discrete equations are then independent from the grid-size. Another benefit is that the discrete equations obtained are identical to those resulting from a more traditional finite difference approach.

The details of the computations are as follows. The integration leads to:

$$\frac{1}{\Delta x \Delta y} \int_{x_1}^{x_2} \int_{y_1}^{y_2} (u_x + v_y) dx dy = 0$$

and after application of Gauss' theorem,

$$\frac{1}{\Delta x \Delta y} \left(\int_{x_1}^{x_2} [v]_{y_1}^{y_2} dx + \int_{y_1}^{y_2} [u]_{x_1}^{x_2} dy \right) = 0 \quad (3.13)$$

Next, the integrals appearing in the right-hand-side of equation (3.13) are approxi-

mated by quadrature formulae, giving respectively,

$$\Delta y(u_{i+1/2,j} - u_{i-1/2,j}) \quad \text{and} \quad \Delta x(v_{i,j+1/2} - v_{i,j-1/2}).$$

The traditional discretisation of the continuity equation on staggered grids results immediately:

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0$$

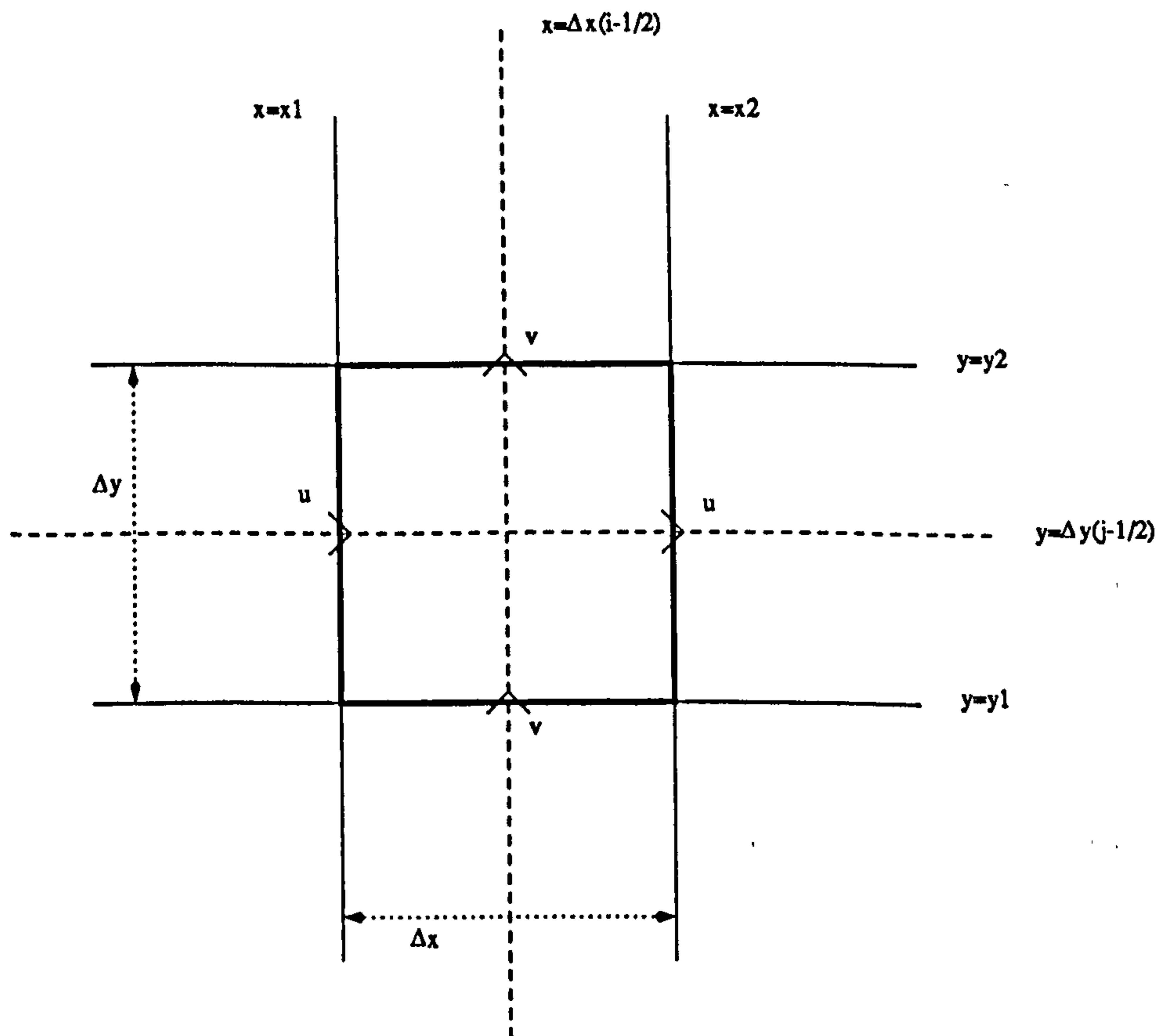


Figure 3.2: Control volume for the integration of the Navier-Stokes continuity equation

3.2.3 Discretisation of the Equation for Horizontal Momentum

The starting point for the discretisation of the equation for horizontal momentum is to rewrite the left-hand-side of (3.10) as the divergence of a vector (hence the usefulness of a conservative formulation):

$$\nabla \cdot \begin{pmatrix} uu - \nu \frac{\partial u}{\partial x} \\ uv - \nu \frac{\partial u}{\partial y} \end{pmatrix} = -\frac{\partial p}{\partial x} \quad (3.14)$$

Equation (3.14) is integrated over a control volume of size $\Delta x \times \Delta y$, i.e. the same size as computational cells, but centred around the point $(i + 1/2, j)$ (see Figure 3.3). Here again, the integral is scaled by $1/(\Delta x \Delta y)$.

Since a coupled solver is used to solve the equations, for each computational cell, another discrete equation for the horizontal momentum is written by considering a control volume of the same size but centred around the point $(i - 1/2, j)$. This is done so that the two equations for horizontal momentum correspond to the two horizontal velocities defined on each cell. The derivation of this second equation is identical to the procedure given in this section.

A discretisation for the right-hand-side of equation (3.14) is given before hybrid schemes are applied to the left-hand-side.

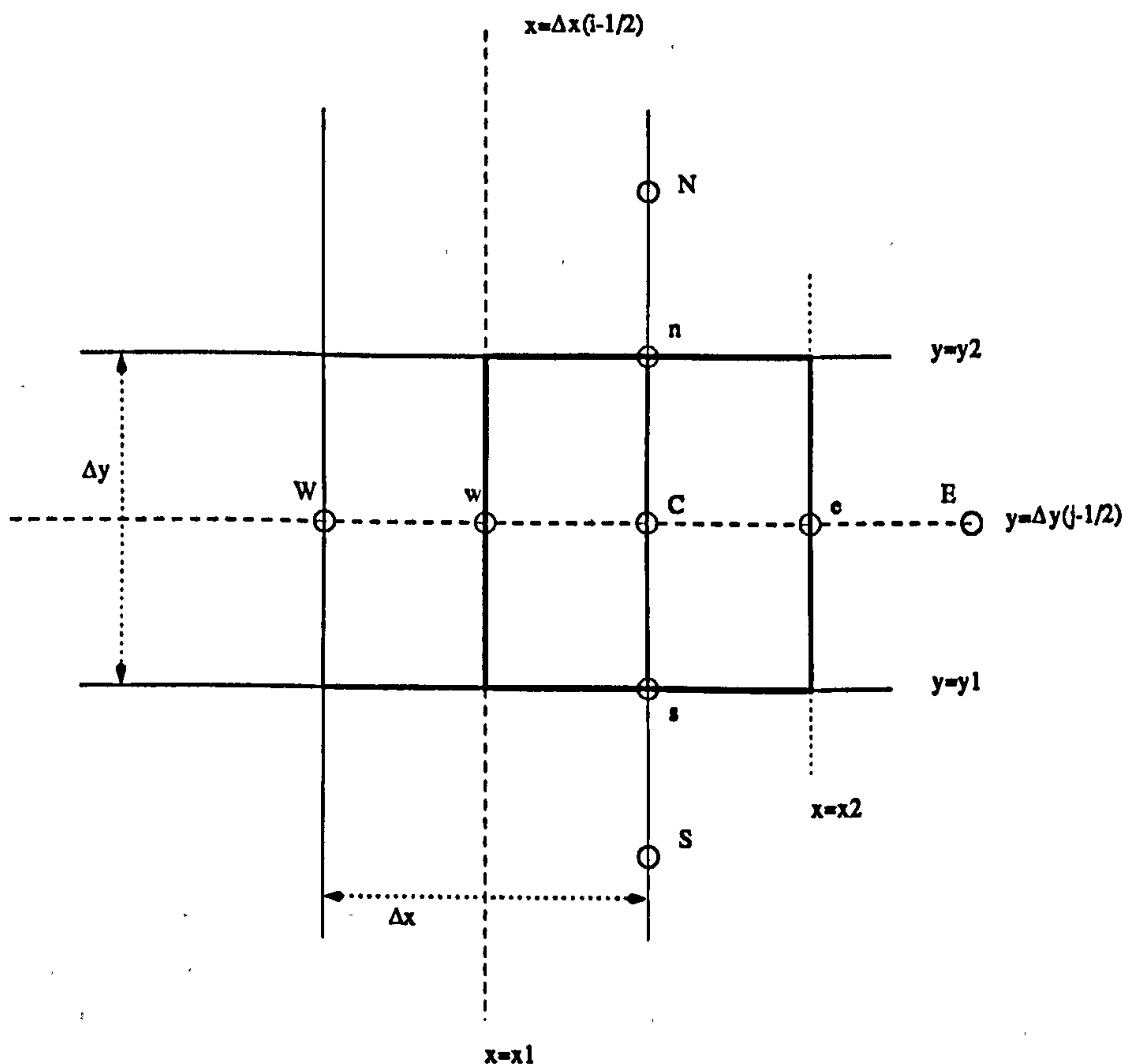


Figure 3.3: Control volume for the integration of the Navier-Stokes horizontal momentum equation

Discretisation of the Right-Hand-Side The integration of the right-hand-side is very easily performed. We have

$$\begin{aligned} \int_{x_1}^{x_2} \frac{\partial p}{\partial x} dx &= [p]_{x_1}^{x_2} \\ &= p_e - p_w \end{aligned} \tag{3.15}$$

so that:

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \int_{y_1}^{y_2} \int_{x_1}^{x_2} \frac{\partial p}{\partial x} dx dy &= \frac{1}{\Delta x \Delta y} \int_{y_1}^{y_2} (p_e - p_w) dy \\ &\approx \frac{1}{\Delta x} (p_e - p_w) \end{aligned} \quad (3.16)$$

For the quadrature formula (3.16), no interpolation is necessary because the discrete pressure grid function is defined at all the required points.

Discretisation of the Left-Hand-Side The application of Gauss' theorem leads to:

$$\frac{1}{\Delta x \Delta y} \int_V \nabla \cdot \begin{pmatrix} uu - \nu \frac{\partial u}{\partial x} \\ uv - \nu \frac{\partial u}{\partial y} \end{pmatrix} dx dy = I_e + I_w + I_n + I_s$$

where:

$$I_e = \frac{1}{\Delta x \Delta y} \left[\int_{y_1}^{y_2} uu - \nu \frac{\partial u}{\partial x} \right]_{x=x_2} \quad (3.17)$$

$$I_w = -\frac{1}{\Delta x \Delta y} \left[\int_{y_1}^{y_2} uu - \nu \frac{\partial u}{\partial x} \right]_{x=x_1} \quad (3.18)$$

$$I_n = \frac{1}{\Delta x \Delta y} \left[\int_{x_1}^{x_2} uv - \nu \frac{\partial u}{\partial y} \right]_{y=y_2} \quad (3.19)$$

$$I_s = -\frac{1}{\Delta x \Delta y} \left[\int_{x_1}^{x_2} uv - \nu \frac{\partial u}{\partial y} \right]_{y=y_1} \quad (3.20)$$

I_e , I_w , I_n and I_s are the horizontal momentum fluxes through the edges of the control volume. In order to obtain discrete equations, the integrals have to be replaced by quadrature formulae, which will involve first order derivatives of u . As in the model problem (see Section 3.1.2), these will be approximated by central finite difference formulae.

For instance, we have for I_e ,

$$\left[\int_{y_1}^{y_2} uu - \nu \frac{\partial u}{\partial x} \right]_{x=x_2} \approx \left(u_e u_e - \nu \frac{u_E - u_C}{\Delta x} \right) \Delta y.$$

The other integrals - I_w , I_n and I_s - are all treated in a similar way.

The following coefficients are then introduced:

$$C_e^u = \frac{1}{2\Delta x} u_e$$

$$\begin{aligned}
 C_w^u &= \frac{1}{2\Delta x} u_w \\
 C_n^u &= \frac{1}{2\Delta y} u_n \\
 C_s^u &= \frac{1}{2\Delta y} u_s \\
 D_e^u &= D_w^u = \frac{\nu}{\Delta x^2} \\
 D_n^u &= D_s^u = \frac{\nu}{\Delta y^2}
 \end{aligned}$$

so that the momentum fluxes can be written as:

$$I_e = 2C_e^u u_e - D_e^u (u_E - u_C) \quad (3.21)$$

$$I_w = -2C_w^u u_w + D_w^u (u_C - u_W) \quad (3.22)$$

$$I_n = 2C_n^u u_n - D_n^u (u_N - u_C) \quad (3.23)$$

$$I_s = -2C_s^u u_s + D_s^u (u_C - u_S). \quad (3.24)$$

$$(3.25)$$

The coefficients C_e^u , C_n^u , C_w^u , C_s^u involve velocities at points where the corresponding grid functions are not defined. Consequently, geometric interpolation is used to obtain approximations. For instance,

$$C_e^u = \frac{1}{2\Delta x} u_e \approx \frac{1}{4\Delta x} (u_{i+1/2,j} + u_{i+3/2,j}). \quad (3.26)$$

Similarly,

$$C_n^u = \frac{1}{4\Delta y} (v_{i,j+1/2} + v_{i+1,j+1/2}), \quad (3.27)$$

$$C_w^u = \frac{1}{4\Delta x} (u_{i+1/2,j} + u_{i-1/2,j}), \quad (3.28)$$

$$C_s^u = \frac{1}{4\Delta x} (v_{i,j-1/2} + v_{i+1,j-1/2}). \quad (3.29)$$

As for the model problem (Section 3.1.2), different finite volume schemes will be obtained for different approximations of the convective terms. In particular, different approximations of u_e , u_w , u_n and u_s will lead to the central, upwind or hybrid schemes. Since the coefficients C_i and D_i are measures of the convective and diffusive effects respectively, the switching between a central and an upwind discretisation is done according to their relative weights.

This procedure is illustrated in the case of I_e and applies directly to the other fluxes. If $|C_e^u| < D_e^u$, central differencing is stable and u_e is interpolated as:

$$u_e \approx \frac{u_E + u_C}{2}.$$

The discrete flux is therefore:

$$\begin{aligned} I_e &= C_e^u(u_E + u_C) - D_e^u(u_E - u_C) \\ &= u_E(C_e^u - D_e^u) + u_C(C_e^u + D_e^u). \end{aligned} \quad (3.30)$$

If $|C_e^u| > D_e^u$, the convective terms are dominant. Central differencing is unstable and needs to be replaced by upwind differencing. Since numerical diffusion introduced by the upwinding dominates the physical diffusion, the latter is neglected so that only the convective flux is taken into account. Upwind differencing is achieved by an upwind interpolation of u_e . Given that $C_e^u = u_e$, its signs give the local direction of the flow and can be used to determine the direction of the up-winding:

$$u_e \approx \begin{cases} u_E & \text{if } C_e < 0 \\ u_C & \text{if } C_e > 0 \end{cases}$$

The discrete (upwind) flux is then given by:

$$\begin{aligned} I_e &= \begin{cases} 2C_e^u u_E & \text{if } C_e < 0 \\ 2C_e^u u_C & \text{if } C_e > 0 \end{cases} \\ &= u_E(C_e^u - |C_e^u|) + u_C(C_e^u + |C_e^u|). \end{aligned} \quad (3.31)$$

Combining equations (3.30) and (3.31), the hybrid discretisation of the flux I_e can be written as:

$$\begin{aligned} I_e &= -\frac{1}{2}[(D_e + |C_e| + |D_e - |C_e||) - C_e]u_E + \\ &\quad \frac{1}{2}[(D_e + |C_e| + |D_e - |C_e||) + C_e]u_C \\ &= -[\max(D_e, |C_e|) - C_e]u_E + [\max(D_e, |C_e|) + C_e]u_C \end{aligned} \quad (3.32)$$

and similarly, the other fluxes are:

$$\begin{aligned} I_w &= -(\max(D_w^u, |C_w^u|) + C_w^u)u_W + (\max(D_w^u, |C_w^u|) - C_w^u)u_C \\ I_n &= -(\max(D_n^u, |C_n^u|) - C_n^u)u_N + (\max(D_n^u, |C_n^u|) + C_n^u)u_C \\ I_s &= -(\max(D_s^u, |C_s^u|) + C_s^u)u_S + (\max(D_s^u, |C_s^u|) - C_s^u)u_C \end{aligned}$$

At this juncture, the following coefficients are introduced:

$$\begin{aligned} A_E^u &= \max(|C_e^u|, D_e^u) - C_e^u \\ A_W^u &= \max(|C_w^u|, D_n^u) + C_w^u \\ A_N^u &= \max(|C_n^u|, D_n^u) - C_n^u \\ A_S^u &= \max(|C_s^u|, D_s^u) + C_s^u \end{aligned}$$

Under the condition that:

$$C_e^u - C_w^u + C_n^u - C_s^u = 0, \quad (3.33)$$

the conservation of discrete horizontal momentum can be written as:

$$A_C^u u_C = A_E^u u_E + A_N^u u_N + A_W^u u_W + A_S^u u_S - \frac{1}{\Delta x} (p_e - p_w) \quad (3.34)$$

with:

$$A_C^u = A_E^u + A_N^u + A_W^u + A_S^u \quad (3.35)$$

Equation (3.33) expresses the physical fact that mass should be conserved in the control volume used to derive the momentum equations. Indeed, equation (3.33) implies:

$$\Delta y u_e - \Delta y u_w + \Delta x u_n - \Delta x u_s = 0$$

which is a discrete approximation of the mass flux through the control volume.

3.2.4 Discretisation of the Equation for Vertical Momentum

The discretisation of equation for vertical momentum proceeds along very similar lines. The starting point is to rewrite equation (3.11) in the following form:

$$\nabla \cdot \left(\begin{array}{c} uv - \nu \frac{\partial v}{\partial x} \\ vv - \nu \frac{\partial v}{\partial y} \end{array} \right) = -\frac{\partial p}{\partial y} \quad (3.36)$$

The control volume used to integrate equation (3.36) is shown in Figure 3.4. Its size is also $\Delta x \times \Delta y$ and it is centred around the point $(i, j+1/2)$. In direct parallel to the horizontal momentum, another discrete vertical momentum equation is written for each cell by integrating (3.36) on a similar control volume centred around $(i, j-1/2)$.

3.2.5 Summary of the Results

The steady Navier-Stokes equations for an incompressible viscous fluid, discretised on a staggered grid by the finite volume method using hybrid schemes, as implemented in the `pang` code, are:

$$\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} = 0 \quad (3.37)$$

$$A_C^u u_{i+1/2,j} = A_E^u u_{i+3/2,j} + A_N^u u_{i+1/2,j+1} + A_W^u u_{i-1/2,j} + A_S^u u_{i+1/2,j-1} - \frac{1}{\Delta x} (p_{i+1,j} - p_{i,j}) \quad (3.38)$$

$$A_C^v v_{i,j+1/2} = A_E^v v_{i+1,j+1/2} + A_N^v v_{i,j+3/2} + A_W^v v_{i-1,j+1/2} + A_S^v v_{i,j-1/2} - \frac{1}{\Delta y} (p_{i,j+1} - p_{i,j}) \quad (3.39)$$

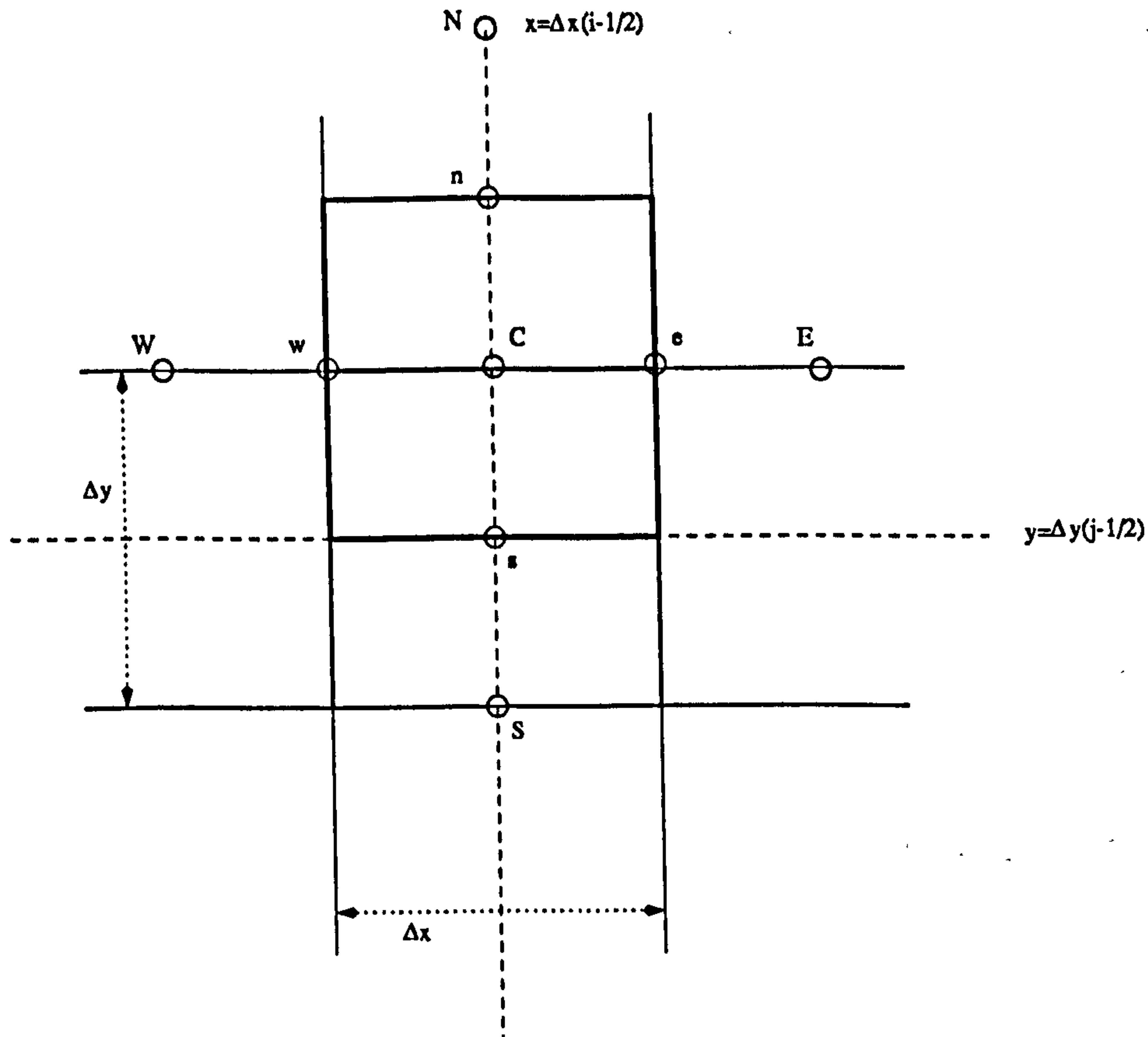


Figure 3.4: Control volume for the integration of the Navier-Stokes vertical momentum equation

with:

$$A_C^u = A_E^u + A_N^u + A_W^u + A_S^u \quad (3.40)$$

$$A_E^u = \max(|C_e^u|, D_e^u) - C_e^u \quad (3.41)$$

$$A_W^u = \max(|C_w^u|, D_n^u) + C_w^u \quad (3.42)$$

$$A_N^u = \max(|C_n^u|, D_n^u) - C_n^u \quad (3.43)$$

$$A_S^u = \max(|C_s^u|, D_s^u) + C_s^u \quad (3.44)$$

$$C_e^u = \frac{1}{4\Delta x} (u_{i+3/2,j} + u_{i+1/2,j}) \quad (3.45)$$

$$C_n^u = \frac{1}{4\Delta y} (v_{i+1,j+1/2} + v_{i,j+1/2}) \quad (3.46)$$

$$C_w^u = \frac{1}{4\Delta x} (u_{i+1/2,j} + u_{i-1/2,j}) \quad (3.47)$$

$$C_s^u = \frac{1}{4\Delta y} (v_{i+1,j-1/2} + v_{i,j-1/2}) \quad (3.48)$$

$$D_e^u = D_w^u = \frac{\nu}{\Delta x^2} \quad (3.49)$$

$$D_n^u = D_s^u = \frac{\nu}{\Delta y^2} \quad (3.50)$$

$$A_C^v = A_E^v + A_N^v + A_W^v + A_S^v \quad (3.51)$$

$$A_E^v = \max(|C_e^v|, D_e^v) - C_e^v \quad (3.52)$$

$$A_W^v = \max(|C_w^v|, D_n^v) + C_w^v \quad (3.53)$$

$$A_N^v = \max(|C_n^v|, D_n^v) - C_n^v \quad (3.54)$$

$$A_S^v = \max(|C_s^v|, D_s^v) + C_s^v \quad (3.55)$$

$$C_e^v = \frac{1}{4\Delta x}(u_{i+1/2,j+1} + u_{i+1/2,j}) \quad (3.56)$$

$$C_n^v = \frac{1}{4\Delta y}(v_{i,j+3/2} + v_{i,j+1/2}) \quad (3.57)$$

$$C_w^v = \frac{1}{4\Delta x}(u_{i-1/2,j+1} + u_{i-1/2,j}) \quad (3.58)$$

$$C_s^v = \frac{1}{4\Delta x}(v_{i,j+1/2} + v_{i,j-1/2}) \quad (3.59)$$

$$D_e^v = D_w^v = \frac{\nu}{\Delta x^2} \quad (3.60)$$

$$D_n^v = D_s^v = \frac{\nu}{\Delta y^2} \quad (3.61)$$

3.3 Discretisation of the Multi-fluid Equations

3.3.1 Governing Equations for Multiphase Flows

pamg-multiphase is applicable to the simulation of steady multiphase flows of M viscous incompressible fluids, where $M \geq 1$. The assumption that the fluids are incompressible allows the energy balance (equation 2.14) to be dropped altogether from the system of governing equations

In order to further simplify the model, it is assumed that there is no transfer of mass between the phases. This functionality could very easily be added later since the code is designed to allow mass transfers. Momentum transfer is allowed and is modelled by simple algebraic source terms. Under these conditions, the multi-fluid equations (2.12) to (2.13) can be written as follows:

- M continuity equations of the form:

$$\nabla \cdot (r_\alpha \mathbf{v}_\alpha) = 0 \quad (3.62)$$

- M momentum equations of the form:

$$\nabla \cdot (r_\alpha (\mathbf{v}_\alpha \otimes \mathbf{v}_\alpha - \mathbf{T}_\alpha)) = -r_\alpha \frac{1}{\rho_\alpha} \nabla p_\alpha + \frac{1}{\rho_\alpha} \sum_{\beta=1}^M c_{\alpha\beta} (\mathbf{v}_\beta - \mathbf{v}_\alpha) \quad (3.63)$$

In this form, the momentum equation has been divided by the density of the phase using the fact the fluids are incompressible.

- M constitutive relationships for the viscous stresses:

$$\mathbf{T}_\alpha = \nu_\alpha(\nabla\mathbf{v}_\alpha + (\nabla\mathbf{v}_\alpha)^T) \quad (3.64)$$

These constitutive relationships can be thought of as “direct” extensions of the single phase constitutive relationships for Newtonian fluids. Different types of multiphase flows may require different expressions for \mathbf{T}_α . Note that \mathbf{T}_α is symmetric ensuring the conservation of angular momentum.

- One closure relationship for the volume fraction:

$$\sum_{\alpha=1}^M r_\alpha = 1 \quad (3.65)$$

- $M - 1$ closure relationships for the pressures:

$$p_1 = p_2 = \dots = p_M \quad (3.66)$$

This constitutes a system of $4M$ equations for $4M$ unknowns: r_α , u_α , v_α and p_α for each phase. This simplified system encapsulates the difficulties associated with the simulation of steady state multiphase flows without being unnecessarily burdened — for our purpose — by modelling issues.

The algebraic source terms:

$$c_{\alpha\beta}(\mathbf{v}_\beta - \mathbf{v}_\alpha) \quad (3.67)$$

model the momentum transfers. They indicate the amount of momentum transferred to phase α by phase β . Different models for the momentum transfer will lead to different values for the coefficients $c_{\alpha\beta}$. Furthermore, $c_{\alpha\beta}$ will, in general, depend on a large number of flow parameters. In droplet flows for instance, it may be a function of the droplet concentration and the droplet shape and average size.

In this study a simple model known as the mixture model [67] is adopted for which the momentum transfer coefficients are given by:

$$c_{\alpha\beta} = c_{\beta\alpha} = \begin{cases} \frac{C_D}{d_{\alpha\beta}} \rho_{\alpha\beta} r_\alpha r_\beta |\mathbf{v}_\beta - \mathbf{v}_\alpha| & \text{if } \alpha \neq \beta, \\ 0 & \text{otherwise.} \end{cases} \quad (3.68)$$

In the present work, the drag coefficient C_D and the inter-facial length scale $d_{\alpha\beta}$ are assumed to be constants. $\rho_{\alpha\beta}$, the mixture specific mass, is defined as:

$$\rho_{\alpha\beta} = r_\alpha \rho_\alpha + r_\beta \rho_\beta$$

This model, which is very simple, has been chosen in order to facilitate comparisons with our reference commercial CFD code, CFX 4.1 [67]. Here again more complex models could easily be added later.

Algebraic source terms are very often used for multiphase simulation. It is possible to include *differential* terms in the momentum transfer terms. One such example is the virtual mass force (see, for instance, reference [49]). However, algebraic terms are more convenient because they do not interfere with the properties of the differential operator.

3.3.2 Governing Equations in Scalar Variable Form

In this section, the aim is to express equations (3.62) and (3.63) in terms of the scalar unknowns $u_\alpha, v_\alpha, r_\alpha$ and p_α for each phase. This is a necessary first step to the discretisation. It will also allow some potentially important characteristics of the diffusive term:

$$\nabla \cdot (r_\alpha \mathbf{T}_\alpha) = \nabla \cdot (r_\alpha \nu_\alpha (\nabla \mathbf{v}_\alpha + (\nabla \mathbf{v}_\alpha)^T))$$

to appear clearly as they are discussed from an analytical point of view.

In order to simplify the notations, subscripts denoting the phase will be dropped from that point on, unless they are absolutely necessary to a proper understanding of the formulae. All the equations and definitions used in the remainder of this chapter apply for all phases separately.

The task of reformulating the continuity equation (3.62) in terms of the scalar variables is almost trivial: simple vector calculus leads to:

$$\frac{\partial ru}{\partial x} + \frac{\partial rv}{\partial y} = 0$$

The convective term in the momentum equation (3.63) is quite easy to expand. The product $r\mathbf{v} \otimes \mathbf{v}$ gives the following tensor:

$$\begin{pmatrix} ruu & ruv \\ ruv & rvv \end{pmatrix}$$

The divergence of a second order tensor \mathbf{Z} is a vector defined as (using tensor notation and Einstein's convention):

$$\nabla \cdot \mathbf{Z} = \frac{\partial z_{ij}}{\partial x_j}$$

We have therefore:

$$\nabla \cdot (r(\mathbf{v} \otimes \mathbf{v})) \begin{pmatrix} (ruu)_x + (ruv)_y \\ (ruv)_x + (rvv)_y \end{pmatrix}$$

The diffusive part,

$$\nabla \cdot [r\nu (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)]$$

is slightly more complex. Firstly, the gradient of a vector field \mathbf{v} is a tensor defined as:

$$\nabla \mathbf{v} = \frac{\partial v_i}{\partial x_j}$$

so that:

$$\nabla \mathbf{v} + (\nabla \mathbf{v})^T = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} = \begin{pmatrix} 2u_x & u_y + v_x \\ u_y + v_x & 2v_y \end{pmatrix}$$

The diffusive term is therefore:

$$\nabla \cdot (r\nu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T]) = \nu \frac{\partial}{\partial x_j} \left(r \frac{\partial u_i}{\partial x_j} + r \frac{\partial u_j}{\partial x_i} \right) = \nu \begin{pmatrix} 2(ru_x)_x + [r(u_y + v_x)]_y \\ [r(v_x + u_y)]_x + 2(v_y)_y \end{pmatrix} \quad (3.69)$$

When only one phase is present, equation (3.69) can be simplified to give the usual diffusive flux for single phase Newtonian flow:

$$\begin{pmatrix} u_{xx} + u_{yy} \\ v_{xx} + v_{yy} \end{pmatrix}. \quad (3.70)$$

This is established by setting $r \equiv 1$ and expanding $\nabla \cdot (\nabla \mathbf{v} + (\nabla \mathbf{v})^T)$

$$\begin{aligned} \nabla \cdot (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) &= \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \\ &= \frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial^2 u_j}{\partial x_j \partial x_i} \\ &= \frac{\partial^2 u_i}{\partial x_j \partial x_j} \end{aligned} \quad (3.71)$$

after using the continuity equation to obtain:

$$\frac{\partial^2 u_j}{\partial x_j \partial x_i} = \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_j} \right) = 0.$$

For multiphase flows where:

$$\nabla \cdot (r [\nabla \mathbf{v} + (\nabla \mathbf{v})^T]) = \frac{\partial}{\partial x_j} \left[r \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right].$$

The corresponding simplification would be:

$$\frac{\partial}{\partial x_j} \left(r \frac{\partial u_j}{\partial x_i} \right) = 0. \quad (3.72)$$

However, equation (3.72) is not satisfied in general. Instead, we have:

$$\begin{aligned} \frac{\partial}{\partial x_j} \left(r \frac{\partial u_j}{\partial x_i} \right) &= \frac{\partial}{\partial x_j} \left(\frac{\partial r u_j}{\partial x_i} - u_j \frac{\partial r}{\partial x_i} \right) \\ &= \frac{\partial}{\partial x_i} \left(\frac{\partial r u_j}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left(u_j \frac{\partial r}{\partial x_i} \right) \\ &= -\frac{\partial}{\partial x_j} \left(u_j \frac{\partial r}{\partial x_i} \right) \end{aligned}$$

after making use of the multiphase continuity equation:

$$\frac{\partial ru_i}{\partial x_i} = 0.$$

The term:

$$\frac{\partial}{\partial x_j} \left(r \frac{\partial u_j}{\partial x_i} \right)$$

is therefore different from zero in general. It is equal to the divergence of the second order tensor

$$\left(u_j \frac{\partial r}{\partial x_i} \right) = \begin{pmatrix} ur_x & vr_y \\ ur_y & vr_y \end{pmatrix}$$

which can be thought of as a measure of the variation of momentum³ due to the spatial variation of the volume fraction. This tensor is identically zero if the volume fractions are constant in space, and in this case, $\nabla \cdot (r\mathbf{T})$ is proportional to:

$$\begin{pmatrix} u_{xx} + u_{yy} \\ v_{xx} + v_{yy} \end{pmatrix}.$$

In terms of the scalar variables u, v, r and p , the differential multi-fluid equations which are to be discretised take the following form:

$$(ru)_x + (rv)_y = 0 \tag{3.73}$$

$$\begin{aligned} (ruu)_x + (ruv)_y &- \nu[2(ru_x)_x + (ru_y)_y + (rv_x)_y] \\ &= -rp_x + c_{\alpha\beta}(u_\beta - u_\alpha) \end{aligned} \tag{3.74}$$

$$\begin{aligned} (ruv)_x + (rvv)_y &- \nu[(rv_x)_x + 2(rv_y)_y + (ru_y)_x] \\ &= -rp_y + c_{\alpha\beta}(v_\beta - v_\alpha) \end{aligned} \tag{3.75}$$

3.3.3 General Discretisation methodology

The method used in Section 3.2 to derive discrete equations for the Navier-Stokes equations can be directly extended to the multi-fluid equations (3.73) to (3.75). The staggered grid used for multiphase flows is shown in Figure 3.5. It is very similar to the staggered grid used in the single phase solver pang; the only difference concerns the volume fractions which are naturally cell-centred quantities.

As discussed in Section 4.3.2, the continuity equation needs a careful treatment. Interpolation of volume fractions are necessary for the formulation of the discrete multiphase continuity equations and certain interpolation schemes have been proved to lead to inconsistent discretisations.

As far as the momentum equations are concerned, the main difference lies in the diffusive tensor \mathbf{T} chosen in this study which contains added cross derivative terms

³The density being constant.

(see equation 3.69). These tend to couple more strongly the vertical and horizontal momentum equations than in the single phase case.

Furthermore, the coefficients D_i and C_i need to be modified to take into account the volume fractions and finally, the momentum transfer terms need to be discretised but neither of these tasks poses any difficulty.

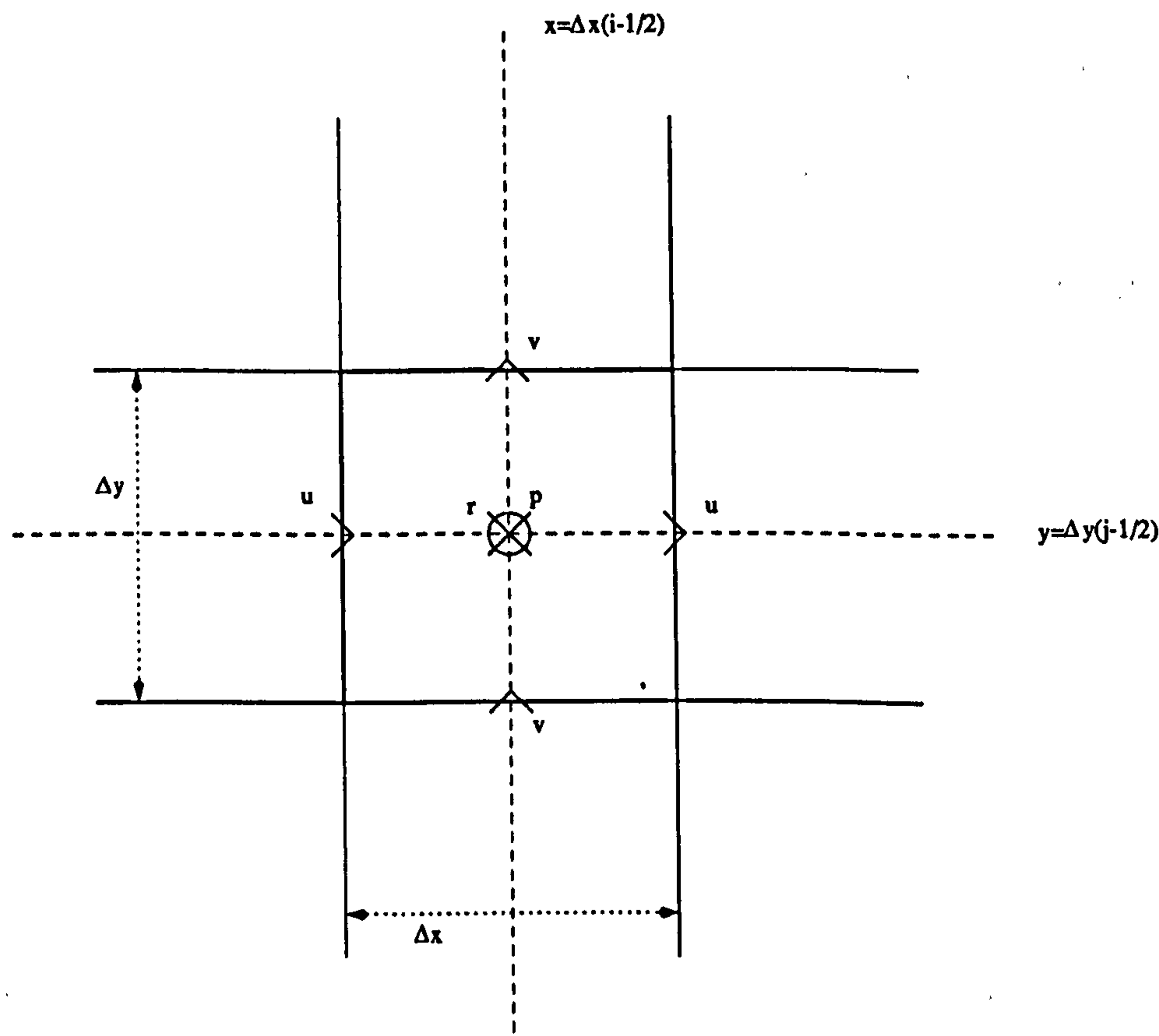


Figure 3.5: Staggered grid used for the discretisation of the multi-fluid equations

3.3.4 Discretisation of the Multiphase Continuity Equation

The methodology used for the single phase case can be directly extended to the multiphase problem. The control volumes on which the equations are integrated are identical (see Figure 3.6). The basic result is:

$$\frac{r_e u_e - r_w u_w}{\Delta x} + \frac{r_n v_n - r_s v_s}{\Delta y} = 0$$

Note that the discrete multiphase continuity equation cannot be solved exactly even on a staggered grid. The velocities are all defined at the required positions but since volume fractions are defined at cell centres, it is necessary to interpolate them to form the products ru and rv at the midpoints of cell edges. For reasons which are given in Section 4.3.2, first order upwind interpolation is chosen:

$$r_e \approx \frac{1}{2} \left[(1 - \text{sgn}(u_{i+1/2,j})) r_{i+1,j} + (\text{sgn}(u_{i+1/2,j}) + 1) r_{i,j} \right]$$

$$r_w \approx \frac{1}{2} \left[(1 - \text{sgn}(u_{i-1/2,j}))r_{i,j} + (\text{sgn}(u_{i-1/2,j}) + 1)r_{i-1,j} \right]$$

$$r_n \approx \frac{1}{2} \left[(1 - \text{sgn}(v_{i,j+1/2}))r_{i,j+1} + (\text{sgn}(v_{i,j+1/2}) + 1)r_{i,j} \right]$$

$$r_s \approx \frac{1}{2} \left[(1 - \text{sgn}(v_{i,j-1/2}))r_{i,j} + (\text{sgn}(v_{i,j-1/2}) + 1)r_{i,j-1} \right]$$

where sgn denotes the sign function:

$$\text{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

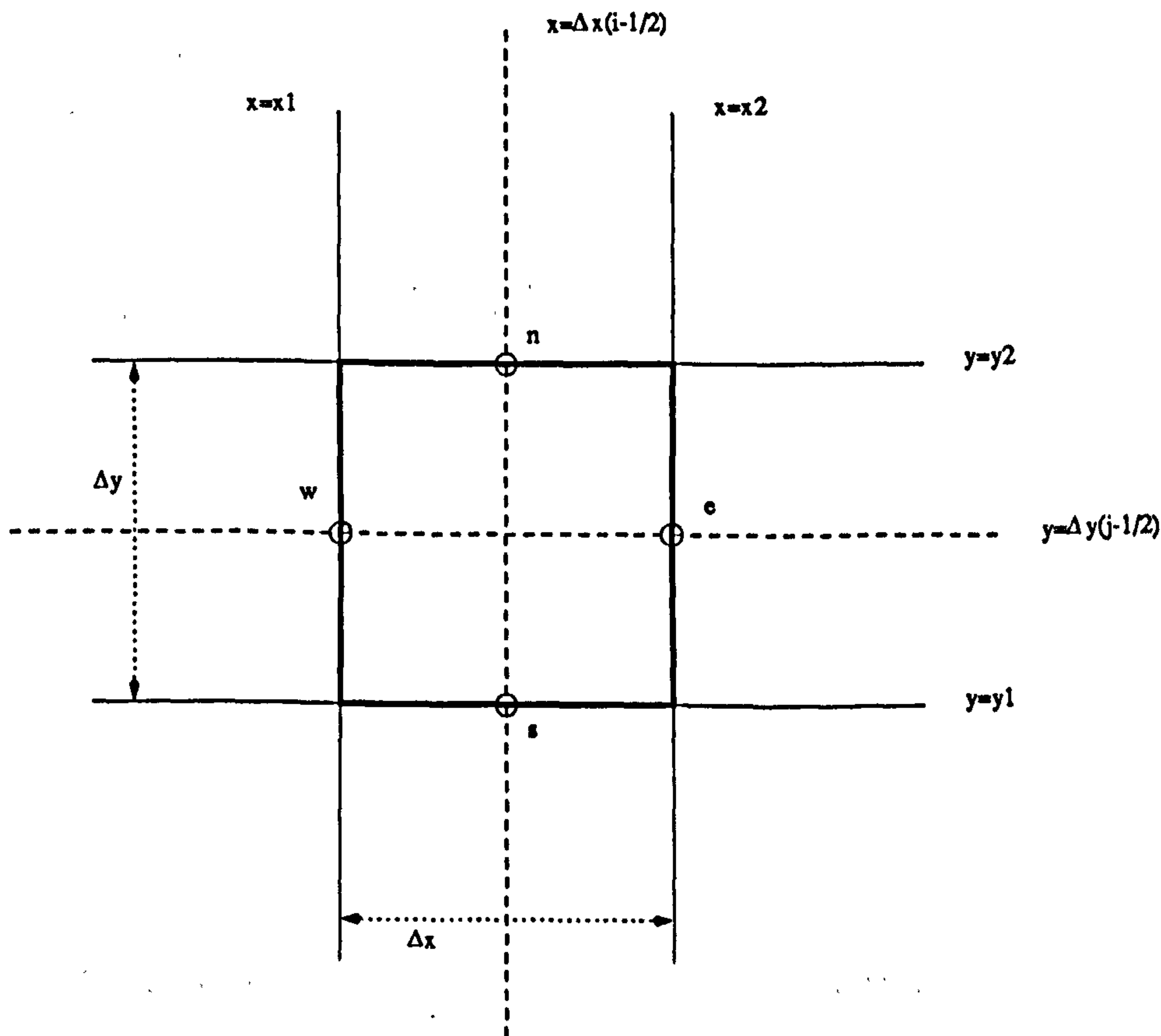


Figure 3.6: Control volume for the integration of the multi-fluid continuity equation

3.3.5 Discretisation of the Equation for Horizontal Momentum

A discretised version of the horizontal momentum equation (3.74) is defined using the technique of hybrid differencing. Vertical momentum can be handled in exactly the same way. Attention is firstly focussed on the right-hand-side whose discretisation is quite simple. A discrete approximation of the left-hand-side is then derived. The control volume used for the horizontal momentum equation is again the same as in the single phase case. It is shown in Figure 3.7.

While the discretisation of the continuity equation requires care in interpolating the grid functions, it was observed that the behaviour of the momentum equations is not as sensitive to the interpolation scheme for the volume fractions. Geometric interpolation has proved to be satisfactory and is used throughout this section.

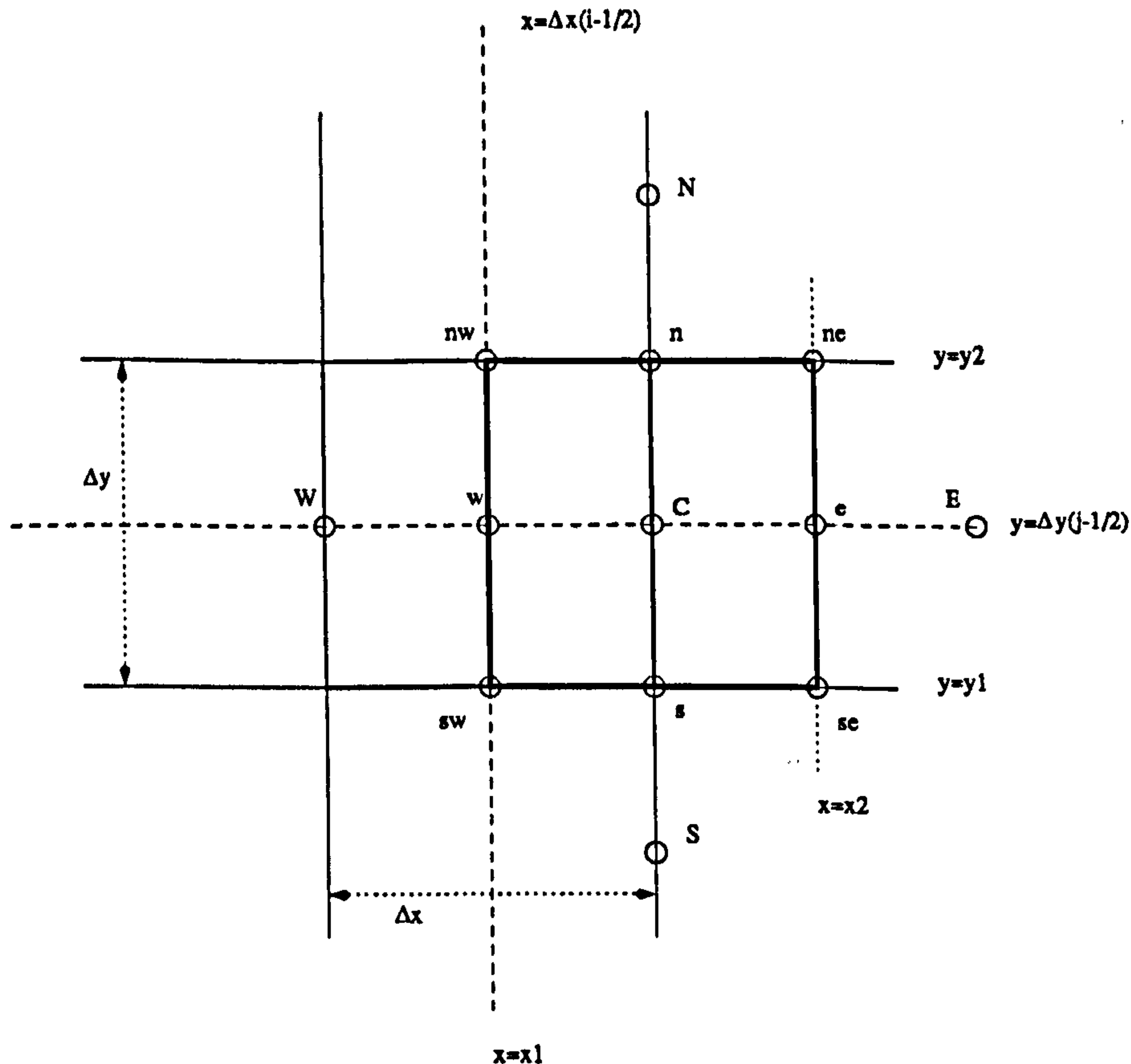


Figure 3.7: Control volume for the integration of the multi-fluid horizontal momentum equation

Discretisation of the Right-Hand-Side After integration of (3.74) over the control volume, the right-hand-side is:

$$\frac{1}{\Delta x \Delta y} \frac{1}{\rho} \left(\int_{x_1}^{x_2} \int_{y_1}^{y_2} r p_x + c_{\alpha\beta} (u_\beta - u_\alpha) dx dy \right).$$

The integration of the first term yields:

$$\begin{aligned} \frac{1}{\Delta x \Delta y} \frac{1}{\rho} \int_{x_1}^{x_2} \int_{y_1}^{y_2} r p_x dx dy &\approx \frac{1}{\Delta x \Delta y} \frac{1}{\rho} r_C \int_{y_1}^{y_2} (p_e - p_w) dy \\ &= \frac{1}{\Delta x} \frac{1}{\rho} r_C (p_e - p_s) \end{aligned}$$

The volume fraction is not directly available at point C (it is defined on the vertex of the control volume) and consequently, a simple geometric interpolation has to be

performed:

$$r_C \approx \frac{1}{2}(r_{i+1,j} + r_{i,j}) \quad (3.76)$$

The integration of the inter-phase momentum transfer terms yields simply:

$$\frac{1}{\rho_\alpha} [c_{\alpha\beta}]_C ([u_\beta]_C - [u_\alpha]_C)$$

u_β and u_α are directly available at point C . The drag coefficient $c_{\alpha\beta}$ involves:

- The modulus of the relative velocity $|\mathbf{v}_\alpha - \mathbf{v}_\beta|$. At present, for the horizontal momentum equation, the approximation:

$$|\mathbf{v}_\alpha - \mathbf{v}_\beta| \approx |u_\alpha - u_\beta|$$

is used, i.e. the vertical component of the velocity is assumed to be zero, so that here again no interpolation is necessary.

- The volume fractions $[r_\alpha]_C$ and $[r_\beta]_C$. The interpolated values have already been defined by equation (3.76).
- The mixture density, which is also available once $[r_\alpha]_C$ and $[r_\beta]_C$ have been defined.

Discretisation of the Left-Hand-Side As for single phase flows, (3.74) is rewritten as the divergence of a vector:

$$\nabla \cdot \begin{pmatrix} ruu - 2\nu ru_x \\ ruv - \nu ru_y - \nu rv_x \end{pmatrix} = -\frac{1}{\rho} r \frac{\partial p}{\partial x} + \frac{1}{\rho} c_{\alpha\beta} (u_\beta - u_\alpha) \quad (3.77)$$

This equation is integrated over the control volume, and scaled by $1/(\Delta x \Delta y)$. The application of Gauss' theorem to the left-hand-side leads then to:

$$\frac{1}{\Delta x \Delta y} \int \nabla \cdot \begin{pmatrix} ruu - 2\nu ru_x \\ ruv - \nu ru_y - \nu rv_x \end{pmatrix} dx dy = I_e + I_n + I_w + I_s \quad (3.78)$$

The I_i are the horizontal momentum fluxes across the four sides of the control volume and are defined in similar ways to equations (3.17) to (3.20):

$$\begin{aligned} I_e &= \frac{1}{\Delta x \Delta y} \left[\int_{y_1}^{y_2} ruu - 2\nu r \frac{\partial u}{\partial x} \right]_{x=x_2} \\ I_w &= -\frac{1}{\Delta x \Delta y} \left[\int_{y_1}^{y_2} ruu - 2\nu r \frac{\partial u}{\partial x} \right]_{x=x_1} \\ I_n &= \frac{1}{\Delta x \Delta y} \left[\int_{x_1}^{x_2} ruv - \nu r \frac{\partial u}{\partial y} - \nu r \frac{\partial v}{\partial x} \right]_{y=y_2} \\ I_s &= -\frac{1}{\Delta x \Delta y} \left[\int_{x_1}^{x_2} ruv - \nu r \frac{\partial u}{\partial y} - \nu r \frac{\partial v}{\partial x} \right]_{y=y_1} \end{aligned}$$

Note the extra terms:

$$\begin{aligned} V_n &= \left[\int_{x_1}^{x_2} \nu r \frac{\partial v}{\partial x} \right]_{y=y_2} \\ &\approx \frac{1}{\Delta x \Delta y} \nu r_n [v]_{x=x_1, y=y_2}^{x=x_2, y=y_2} \\ &\approx \frac{1}{\Delta x \Delta y} \nu r_n (v_{ne} - v_{nw}) \end{aligned}$$

and:

$$\begin{aligned} V_s &= \left[\int_{x_1}^{x_2} \nu r \frac{\partial v}{\partial x} \right]_{y=y_1} \\ &\approx \frac{1}{\Delta x \Delta y} \nu r_s [v]_{x=x_1, y=y_1}^{x=x_2, y=y_1} \\ &\approx \frac{1}{\Delta x \Delta y} \nu r_s (v_{se} - v_{sw}) \end{aligned}$$

which arise from the cross derivative terms in $\nabla \cdot (r\mathbf{T})$. With the following definitions of the coefficients C_i and D_i ,

$$\begin{aligned} C_e^u &= \frac{1}{2\Delta x} r_e u_e \\ C_w^u &= \frac{1}{2\Delta x} r_w u_w \\ C_n^u &= \frac{1}{2\Delta x} r_n u_n \\ C_s^u &= \frac{1}{2\Delta x} r_s u_s \\ D_e^u &= \frac{2\nu r_e}{\Delta x^2} \\ D_w^u &= \frac{2\nu r_w}{\Delta x^2} \\ D_n^u &= \frac{\nu r_n}{\Delta y^2} \\ D_s^u &= \frac{\nu r_s}{\Delta y^2}, \end{aligned}$$

expressions for the fluxes I_e , I_w , I_n and I_s can be constructed which are equivalent to equations (3.21) to (3.24):

$$I_e = 2C_e^u u_e - D_e^u (u_E - u_C) \quad (3.79)$$

$$I_w = -2C_w^u u_w + D_w^u (u_C - u_W) \quad (3.80)$$

$$I_n = 2C_n^u u_n - D_n^u (u_N - u_C) - V_n \quad (3.81)$$

$$I_s = -2C_s^u u_s + D_s^u (u_C - u_S) + V_s \quad (3.82)$$

Writing, as in the single phase case,

$$A_e = [\max(D_e, |C_e|) - C_e]u_E \quad (3.83)$$

$$A_w = [\max(D_w, |C_w|) + C_w]u_W \quad (3.84)$$

$$A_n = [\max(D_n, |C_n|) - C_n]u_N \quad (3.85)$$

$$A_s = [\max(D_s, |C_s|) + C_s]u_S \quad (3.86)$$

$$A_c = A_e + A_n + A_w + A_s, \quad (3.87)$$

The discrete form of equation (3.74), written for phase α is simply:

$$A_C^u u_C = A_E^u u_E + A_N^u u_N + A_W^u u_W + A_S^u u_S + (V_n - V_s) - \frac{1}{\Delta x} r_C (p_e - p_w) + [c_{\alpha\beta}]_C (u_\beta - u_\alpha) \quad (3.88)$$

3.3.6 Discretisation of the Vertical Momentum Equation

The control volume of Figure 3.8, centred around the point $(i, j + 1/2)$, is used for the derivation which rigorously parallels that of the horizontal momentum equation. Differences arise mainly from the cross derivative terms in $\nabla \cdot (r\mathbf{T})$ but these do not require any new techniques.

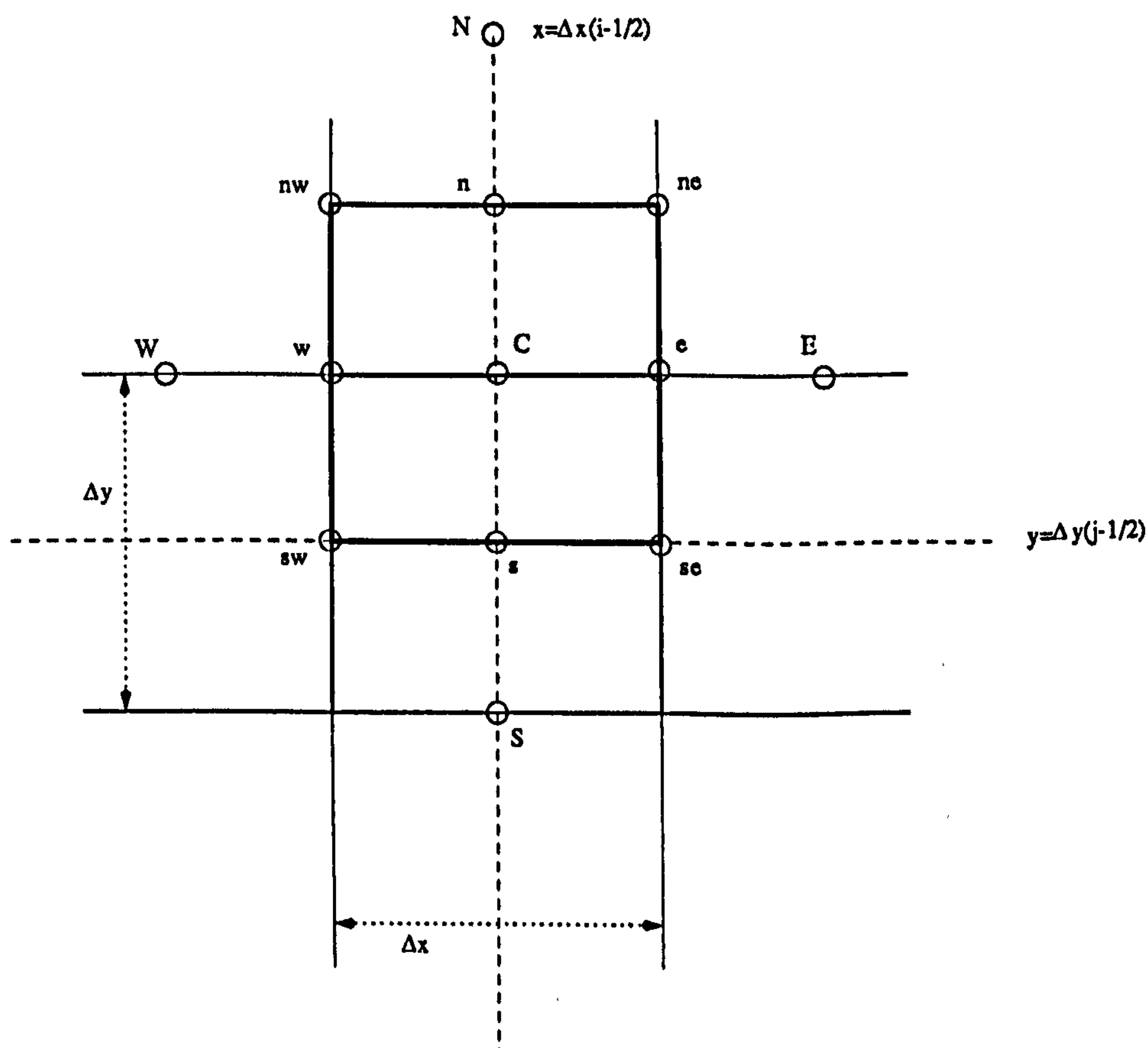


Figure 3.8: Control volume for the integration of the multi-fluid vertical momentum equation

3.3.7 Summary of the Results

Summarising the results obtained so far, the governing equations for multiphase flows of viscous incompressible fluids within the (steady) multi-fluid model, discretised on a staggered grid and as implemented in pang-multiphase are:

$$A_C^u u_{i+1/2,j} = A_E^u u_{i+3/2,j} + A_N^u u_{i+1/2,j+1} + A_W^u u_{i-1/2,j} + A_S^u u_{i+1/2,j-1} + (V_n - V_s) - \frac{r_C^u}{\rho \Delta x} (p_{i+1,j} - p_{i,j}) + \frac{1}{\rho} [c_{\alpha\beta}^u]_C ([u_\beta]_{i+1/2,j} - [u_\alpha]_{i+1/2,j}) \quad (3.89)$$

$$A_C^v v_{i,j+1/2} = A_E^v v_{i+1,j+1/2} + A_N^v v_{i,j+3/2} + A_W^v v_{i-1,j+1/2} + A_S^v v_{i,j-1/2} + (U_e - U_w) - \frac{r_C^v}{\rho \Delta y} (p_{i,j+1} - p_{i,j}) + \frac{1}{\rho} [c_{\alpha\beta}^v]_C ([v_\beta]_{i,j+1/2} - [v_\alpha]_{i,j+1/2}) \quad (3.90)$$

$$(3.91)$$

$$\frac{r_e^c u_{i+1/2,j} - r_w^c u_{i-1/2,j}}{\Delta x} + \frac{r_n^c v_{i,j+1/2} - r_s^c v_{i,j-1/2}}{\Delta y} = 0 \quad (3.92)$$

where:

$$A_C^u = A_E^u + A_N^u + A_W^u + A_S^u \quad (3.93)$$

$$A_E^u = \max(D_e^u, |C_e^u|) - C_e^u \quad (3.94)$$

$$A_W^u = \max(D_w^u, |C_w^u|) + C_w^u \quad (3.95)$$

$$A_N^u = \max(D_n^u, |C_n^u|) - C_n^u \quad (3.96)$$

$$A_S^u = \max(D_s^u, |C_s^u|) + C_s^u \quad (3.97)$$

$$C_e^u = \frac{1}{4\Delta x} (u_{i+1/2,j} + u_{i+3/2,j}) r_e^u \quad (3.98)$$

$$C_w^u = \frac{1}{4\Delta x} (u_{i-1/2,j} + u_{i+1/2,j}) r_w^u \quad (3.99)$$

$$C_n^u = \frac{1}{4\Delta y} (v_{i,j+1/2} + v_{i+1,j+1/2}) r_n^u \quad (3.100)$$

$$C_s^u = \frac{1}{4\Delta y} (v_{i,j-1/2} + v_{i+1,j-1/2}) r_s^u \quad (3.101)$$

$$D_e^u = \frac{2\nu}{\Delta x^2} r_e^u \quad (3.102)$$

$$D_w^u = \frac{2\nu}{\Delta x^2} r_w^u \quad (3.103)$$

$$D_n^u = \frac{\nu}{\Delta y^2} r_n^u \quad (3.104)$$

$$D_s^u = \frac{\nu}{\Delta y^2} r_s^u \quad (3.105)$$

$$V_n = \frac{1}{\Delta x \Delta y} \nu r_n^u (v_{i+1,j+1/2} - v_{i,j+1/2}) \quad (3.106)$$

$$V_s = \frac{1}{\Delta x \Delta y} \nu r_s^u (v_{i+1,j-1/2} - v_{i,j-1/2}) \quad (3.107)$$

$$[c_{\alpha\beta}^u]_C = \frac{C_D}{d_{\alpha\beta}} ([r_\alpha]_C^u \rho_\alpha + [r_\beta]_C^u \rho_\beta) [r_\alpha]_C^u [r_\beta]_C^u |[u_\beta]_{i+1/2,j} - [u_\alpha]_{i+1/2,j}| \quad (3.108)$$

$$A_C^v = A_E^v + A_N^v + A_W^v + A_S^v \quad (3.109)$$

$$A_E^v = \max(D_e^v, |C_e^v|) - C_e^v \quad (3.110)$$

$$A_W^v = \max(D_w^v, |C_w^v|) + C_w^v \quad (3.111)$$

$$A_N^v = \max(D_n^v, |C_n^v|) - C_n^v \quad (3.112)$$

$$A_S^v = \max(D_s^v, |C_s^v|) + C_s^v \quad (3.113)$$

$$C_e^v = \frac{1}{4\Delta x} (u_{i+1/2,j} + u_{i+1/2,j+1}) r_e^v \quad (3.114)$$

$$C_w^v = \frac{1}{4\Delta x} (u_{i-1/2,j} + u_{i-1/2,j+1}) r_w^v \quad (3.115)$$

$$C_n^v = \frac{1}{4\Delta y} (v_{i,j+1/2} + v_{i,j+3/2}) r_n^v \quad (3.116)$$

$$C_s^v = \frac{1}{4\Delta y} (v_{i,j-1/2} + v_{i,j+1/2}) r_s^v \quad (3.117)$$

$$D_e^v = \frac{\nu}{\Delta x^2} r_e^v \quad (3.118)$$

$$D_w^v = \frac{\nu}{\Delta x^2} r_w^v \quad (3.119)$$

$$D_n^v = \frac{2\nu}{\Delta y^2} r_n^v \quad (3.120)$$

$$D_s^v = \frac{2\nu}{\Delta y^2} r_s^v \quad (3.121)$$

$$U_e = \frac{1}{\Delta x \Delta y} \nu r_e^v (u_{i+1/2,j+1} - v_{i+1/2,j}) \quad (3.122)$$

$$U_w = \frac{1}{\Delta x \Delta y} \nu r_w^v (u_{i-1/2,j+1} - v_{i-1/2,j}) \quad (3.123)$$

$$[c_{\alpha\beta}^v]_C = \frac{C_D}{d_{\alpha\beta}} ([r_\alpha]_C^v \rho_\alpha + [r_\beta]_C^v \rho_\beta) [r_\alpha]_C^v [r_\beta]_C^v |[v_\beta]_{i,j+1/2} - [v_\alpha]_{i,j+1/2}| \quad (3.124)$$

$$r_e^c = \frac{1}{2} [(1 - \text{sgn}(u_{i+1/2,j})) r_{i+1,j} + (\text{sgn}(u_{i+1/2,j} - 1)) r_{i,j}] \quad (3.125)$$

$$r_w^c = \frac{1}{2} [(1 - \text{sgn}(u_{i-1/2,j})) r_{i,j} + (\text{sgn}(u_{i-1/2,j} - 1)) r_{i-1,j}] \quad (3.126)$$

$$r_n^c = \frac{1}{2} [(1 - \text{sgn}(v_{i,j+1/2})) r_{i,j+1} + (\text{sgn}(v_{i,j+1/2} - 1)) r_{i,j}] \quad (3.127)$$

$$r_s^c = \frac{1}{2} [(1 - \text{sgn}(v_{i,j-1/2})) r_{i,j} + (\text{sgn}(v_{i,j-1/2} - 1)) r_{i,j-1}] \quad (3.128)$$

$$r_e^u = r_{i+1,j} \quad (3.129)$$

$$r_w^u = r_{i,j} \quad (3.130)$$

$$r_n^u = \frac{1}{4} (r_{i,j} + r_{i+1,j} + r_{i+1,j+1} + r_{i,j+1}) \quad (3.131)$$

$$r_s^u = \frac{1}{4} (r_{i,j} + r_{i+1,j} + r_{i+1,j-1} + r_{i,j-1}) \quad (3.132)$$

$$r_e^v = \frac{1}{4} (r_{i,j} + r_{i+1,j} + r_{i+1,j+1} + r_{i,j+1}) \quad (3.133)$$

$$r_w^v = \frac{1}{4}(r_{i,j} + r_{i,j+1} + r_{i-1,j+1} + r_{i-1,j}) \quad (3.134)$$

$$r_n^v = r_{i,j+1} \quad (3.135)$$

$$r_s^v = r_{i,j} \quad (3.136)$$

$$r_C^u = \frac{1}{2}(r_{i,j} + r_{i+1,j}) \quad (3.137)$$

$$r_C^v = \frac{1}{2}(r_{i,j} + r_{i,j+1}) \quad (3.138)$$

3.4 Conclusions

In this chapter, we have defined the exact *differential* equations to which `pamg-multiphase` is meant to provide discrete solutions by supplying simple constitutive relationships in order to close the basic multi-fluid equations. The model chosen for the stress tensor introduces cross derivative terms in the momentum equations which are hyperbolic-like and could, following the argument given in Section 2.5.9, have a detrimental effect on the multigrid convergence rates.

Appropriate discrete equations have been derived for both the single phase solver `pamg` and the multiphase solver `pamg-multiphase`. The discretisation process relies on the use of staggered grids together with hybrid differencing to provide a set of non-linear algebraic equations which:

- are second order accurate for sufficiently fine grids, except for the volume fractions in the continuity equations, and monotonic on all grids so that non-physical oscillations cannot be generated;
- allow easy enforcement of conservation, particularly for mass;
- contain no added parameters such as artificial viscosity (as opposed to the numerical diffusion introduced by the discretisation);
- lead to a very robust solver when used in conjunction with a multigrid procedure because they implicitly define a continuation method when going from coarse to fine grids.

The points at which the multiphase discrete equations differ significantly from their single phase counterparts have been treated, highlighted and discussed in the course of this chapter. These include:

- the treatment of the volume fractions;
- the treatment of the source terms;
- the treatment of the momentum cross derivative terms.

Once discrete equations are available, it is possible to design a solution algorithm tailored to that particular set. This is the subject of the next chapter.

Chapter 4

The pamg and pamg-multiphase Solvers: Description and Differences

4.1 Introduction

At this juncture, we have described the physical problem at hand, motivated the solution algorithm we propose to apply and surveyed its mathematical foundations. Furthermore, we have obtained a good discretisation of the governing equations. It remains to give a detailed account of the implementation of the solution algorithm as found in the `pamg-multiphase` solver. The single phase solver `pamg` is also described as this provides a valuable and convenient reference point.

The main goal is to illustrate the similarities as well as the significant differences which exist between the two codes. The multiphase solver is considerably more complex than its single phase counterpart but their architectures are nonetheless very similar since both are adaptive quasi-Newton multigrid solvers. As mentioned in the introduction, multigrid algorithms have many parts and aspects which are inter-locked and need to be examined as a unit.

The task of extending `pamg` to obtain `pamg-multiphase` has centred mainly around algorithmic issues. Consequently, their discussion constitutes the main part of the material reported in this chapter. All the components of the multigrid solver have required careful examination and a significant number of difficulties have arisen necessitating original solutions. These include:

- consistent discretisations;
- the treatment of the non-linearity by line-searching;

- obtaining the local Jacobians by automatic differentiation;
- multigrid transfers which only generate physically realisable approximations;
- conservative interpolation operators in connection with adaptation.

A second aspect to the work carried out concerns computational issues. In the author's opinion, developing numerical software is difficult *per se*. In some senses, however, it is much easier than *modifying* numerical software. Fundamental issues in software engineering: modularity, software re-usability and validation are immediately more challenging than in other applications due to the complexity of both the data structures and program control flows. The more advanced the solution algorithm, the more potential there is for implementation errors and the more important these computational issues become. In the particular case of *pamg-multiphase*, difficulties are magnified further when it is considered that:

- multigrid methods are notoriously difficult to implement;
- the governing equations are very complex and their properties are not completely understood, thus making it is very difficult to distinguish between genuine numerical difficulties and mere implementation errors.

Defining a “good” methodology for modifying and developing the software has therefore been crucial in order to accelerate development time. The process adopted is addressed here since it is very likely to have been a key factor in the success of the present study.

4.2 The *pamg* Single Phase Algorithm

In this section we describe the various elements of the *pamg* solution algorithm in turn. These are, in order: the quasi-Newton coupled solver, the embedding FAS multigrid method, more specifically the grid transfer operators, and finally, the refinement procedure.

4.2.1 The Local Quasi-Newton Coupled Solver

Basic Idea A detailed description of the *pamg* solver can be found in [13]. This reference deals with buoyant flows so that compared with the Navier-Stokes system of equations, the temperature is an extra unknown. The problem is closed by the addition of an extra partial differential equation modelling the convection and diffusion of the temperature. The Navier-Stokes solver described here and in references

[11, 1] is essentially a subset of the solver presented in [13]. Reference [85] gives a complete description of the features of *pang*.

The *pang* quasi-Newton solver follows the procedure introduced by Vanka [73] (with others!). The velocities and pressure are corrected in a coupled manner on a cell by cell basis. Two horizontal velocities, two vertical velocities and one pressure are corrected simultaneously by solving a five by five set of equations: two horizontal momentum equations, two vertical momentum equations and one continuity equation. The discrete equations are derived by a finite volume approach on a staggered grid and use hybrid schemes. Section 3.2 considers the discretisation of the Navier-Stokes equations as implemented in *pang*.

The cells are ordered in lexicographic order and during one relaxation sweep, all cells are visited in turn. This type of iterative procedure is referred to as a Symmetrical Coupled Gauss Seidel (SCGS) procedure [73] which is now described.

Symmetrical Coupled Gauss Seidel Procedure Consider a rectangular staggered grid containing $N = n_x \times n_y$ cells. A set of grid functions $\tilde{u}_{i-1/2,j}$, $\tilde{u}_{i+1/2,j}$, $\tilde{v}_{i,j-1/2}$, $\tilde{v}_{i,j+1/2}$, $\tilde{p}_{i,j}$ are given which are approximations to the solution of the set of $5N$ discrete equations:

$$\mathbf{f}(\mathbf{x}) = A(\mathbf{x})\mathbf{x} = \mathbf{s} \quad (4.1)$$

where A is a matrix expressing the discrete operator \mathcal{L}^h . It is written as $A(\mathbf{x})$ to emphasise its dependence on the solution. The discrete operator is defined by equations (3.37) to (3.39) written for all cells (and suitably modified at boundaries). Equation (4.1) is written in block form by ordering, for instance, the cells in lexicographic order:

$$(1, 1); (1, 2); \dots; (1, n_y); (2, 1); (2, 2); \dots; (2, n_y); \dots; (n_x, 1); \dots; (n_x, n_y)$$

and with each cell we group the following set of five variables:

$$(\tilde{u}_{i-1/2,j}, \tilde{u}_{i+1/2,j}, \tilde{v}_{i,j-1/2}, \tilde{v}_{i,j+1/2}, \tilde{p}_{i,j}) \quad (4.2)$$

as a unit in order to determine the block structure. The matrix A can be written as:

$$A = D - L - U$$

where D , L , U are respectively block diagonal, block lower and block upper matrices relative to the ordering of cells.

Consider first the linear case where the coefficients in A do not depend on \mathbf{x} . This could represent the limiting case when $Re \rightarrow 0$.

The (relaxed) Gauss-Seidel method is written as:

$$D\hat{\mathbf{x}} = L\mathbf{x}^{(1)} + U\mathbf{x}^{(0)} + \mathbf{s} \quad (4.3)$$

$$\mathbf{x}^{(1)} = \lambda \hat{\mathbf{x}} + (1 - \lambda) \mathbf{x}^{(0)} \quad (4.4)$$

where $\mathbf{x}^{(0)}$ is some initial estimate of the solution vector, $\mathbf{x}^{(1)}$ is the new approximation generated by the Gauss-Seidel procedure and λ is the relaxation parameter. If a residual vector \mathbf{r} is introduced, defined as:

$$\mathbf{r} = \mathbf{s} + L\mathbf{x}^{(1)} + U\mathbf{x}^{(0)} - D\mathbf{x}^{(0)}$$

then the following equations hold:

$$D(\hat{\mathbf{x}} - \mathbf{x}^{(0)}) = \mathbf{r}$$

and

$$\frac{1}{\lambda} D(\mathbf{x}^{(1)} - \mathbf{x}^{(0)}) = \mathbf{r} \quad (4.5)$$

Equation (4.5) is solved block by block by sweeping through the cells in turn. It is slightly modified so that only the diagonal entries of D are relaxed.

Consider the cell (i, j) and let Φ be the group of five variables defined by (4.2). The task is to solve the following linear system of algebraic equations for the correction $\Delta\Phi$ to Φ :

$$\begin{aligned} \frac{1}{\lambda} D_{ij} \Delta\Phi &= \mathbf{r}_{ij} \\ \Phi^{(1)} &= \Phi^{(0)} + \Delta\Phi \end{aligned}$$

where $D_{i,j}$ is the block of D which corresponds to the cell (i, j) and $\mathbf{r}_{i,j}$ is the sub-vector of \mathbf{r} which corresponds to the cell (i, j) . This system is small and can be efficiently solved by any direct method such as Gaussian elimination.

In the non-linear case, and for a standard Gauss-Seidel procedure, the matrices D , L and U have to be evaluated at each step using the available estimates, and equation (4.3) becomes:

$$D(\mathbf{x}^{(0)})\hat{\mathbf{x}} = L(\mathbf{x}^{(0)})\mathbf{x}^{(1)} + U(\mathbf{x}^{(0)})\mathbf{x}^{(0)} + \mathbf{s}(\mathbf{x}^{(0)}) = \mathbf{r} \quad (4.6)$$

The task is to solve the vector equation:

$$D_{i,j}(\Phi^{(0)})\hat{\Phi} = \mathbf{r}_{i,j} \quad (4.7)$$

An iterative rather than direct procedure is necessary to solve (4.7). Again the system is small (5 by 5). The method of choice is therefore the Newton-Raphson method. In *pang*, $\hat{\Phi}$ is computed using a *single* Newton step. Equation (4.7) is therefore not satisfied exactly but the corrections are good enough to ensure the global convergence of the process, as a result of the implicit continuation.

In the case of a fully coupled solver, equation (4.6) is not solved on a block by block basis but the equations for *all* cells are relaxed at the same time by Newton's method. Fully coupled solvers take into account the *global* coupling between all the flow unknowns. Their main drawback is the very large size of the Jacobians. Hence,

it is crucial to exploit the sparseness of the Jacobian matrices before this approach becomes practical.

Local quasi-Newton coupled solvers, in the line of *pamg* only require *local* Jacobians. The size of the systems of equations arising from the Newton linearisation is much smaller. However, only the local coupling between variables is taken into account. There is some evidence to suggest that this local coupling dominates over the global coupling: namely, the off-diagonal blocks in the full Jacobian are small. Furthermore, solution algorithms based on the SCGS methodology are robust and efficient.

The fact that the quasi-Newton solver favours the local coupling explains why multigrid methods are useful in connection with this type of algorithm. In a nutshell, the multigrid algorithm ensures that global coupling is taken into account and therefore complements the LQN solver. A dual point of view is to consider that MG allows the quick resolution of the long wavelengths.

The Newton Step It is well known that given the vector equation:

$$\mathbf{f}(\Phi) = \mathbf{s}$$

and an initial approximation $\Phi^{(0)}$ of the solution Φ , Newton's method (see for instance [103]) updates the approximation by the procedure

$$J\Delta\Phi = -(\mathbf{f}(\Phi^{(0)}) - \mathbf{s}) \quad (4.8)$$

$$\Phi^{(1)} = \Phi^{(0)} + \Delta\Phi$$

where J is the Jacobian of \mathbf{f} evaluated at $\Phi^{(0)}$ so:

$$J_{ij} = \frac{\partial f_i}{\partial \Phi_j}(\Phi^{(0)})$$

Whenever Newton's method is implemented on a computer, the crucial issue is to obtain a good approximation of J . In the case of *pamg*, the system is defined as:

$$\mathbf{f}(\Phi) = D(\Phi)\Phi$$

and the *frozen coefficient approximation* is used:

$$J \approx D(\Phi).$$

which leads to:

$$J = D(\Phi) = \begin{pmatrix} (A_c^u)_{i-1/2,j} & 0 & 0 & 0 & \frac{1}{\Delta x} \\ 0 & (A_c^u)_{i+1/2,j} & 0 & 0 & -\frac{1}{\Delta x} \\ 0 & 0 & (A_c^v)_{i,j-1/2} & 0 & \frac{1}{\Delta y} \\ 0 & 0 & 0 & (A_c^v)_{i,j+1/2} & -\frac{1}{\Delta y} \\ \frac{1}{\Delta x} & -\frac{1}{\Delta x} & -\frac{1}{\Delta y} & \frac{1}{\Delta y} & 0 \end{pmatrix} \quad (4.9)$$

The system (4.8) is then solved by simple Gaussian elimination. The elimination is done explicitly to exploit the sparseness of D and increase the speed of the computations.

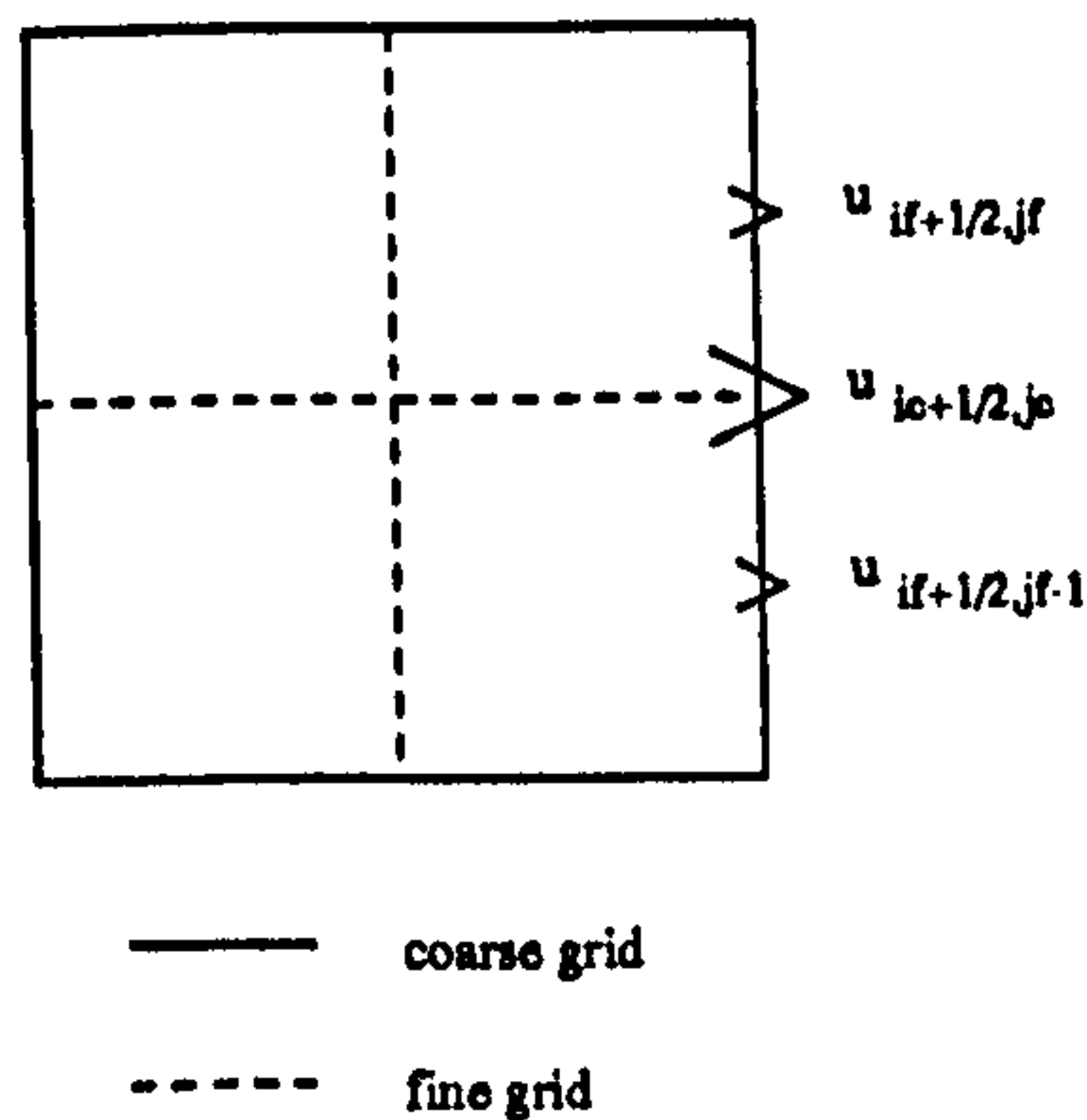


Figure 4.1: Discrete values for the restriction of the horizontal velocities

4.2.2 The pamg FAS Multigrid Algorithm – Definition of the Grid Transfer Operators

The FAS multigrid algorithm used in pamg is essentially the general algorithm described in Section 2.5.6. Its smoother, the SCGS procedure, has already been described in the previous section and it remains to specify the transfer operators chosen for single phase flows.

Restriction of the Fine Grid Approximation We have to restrict three types of quantities: the vertical and horizontal velocities, which are defined at cell edges and the pressures, which are defined at cell centres. For the velocities, second order linear interpolation is used. Letting $if = 2ic$ and $jf = 2jc$, the coarse grid values are defined in terms of the fine grid values by the following relationships:

$$u_{ic+1/2, jc}^{(c)} = \frac{1}{2} \left(u_{if+1/2, jf-1}^{(f)} + u_{if+1/2, jf}^{(f)} \right) \quad (4.10)$$

$$v_{ic, jc+1/2}^{(c)} = \frac{1}{2} \left(v_{if-1, jf+1/2}^{(f)} + v_{if, jf+1/2}^{(f)} \right) \quad (4.11)$$

where the subscripts (c) and (f) refer to the coarse and fine grid values respectively. Similarly, bilinear interpolation is used for the pressure:

$$p_{ic, jc}^{(c)} = \frac{1}{4} \left(p_{if-1, jf-1}^{(f)} + p_{if-1, jf}^{(f)} + p_{if, jf-1}^{(f)} + p_{if, jf}^{(f)} \right) \quad (4.12)$$

This last relationship applies for interior points only but it is the only restriction formula necessary for the pressures. First, no pressure boundary conditions are necessary since the continuity equation is satisfied directly using the velocities rather than a pressure based Poisson equation. At solid walls, velocities are fixed so that no transfer is necessary. Finally, for Neumann conditions, the formulae for interior nodes are used and boundary conditions are applied just after the transfer.

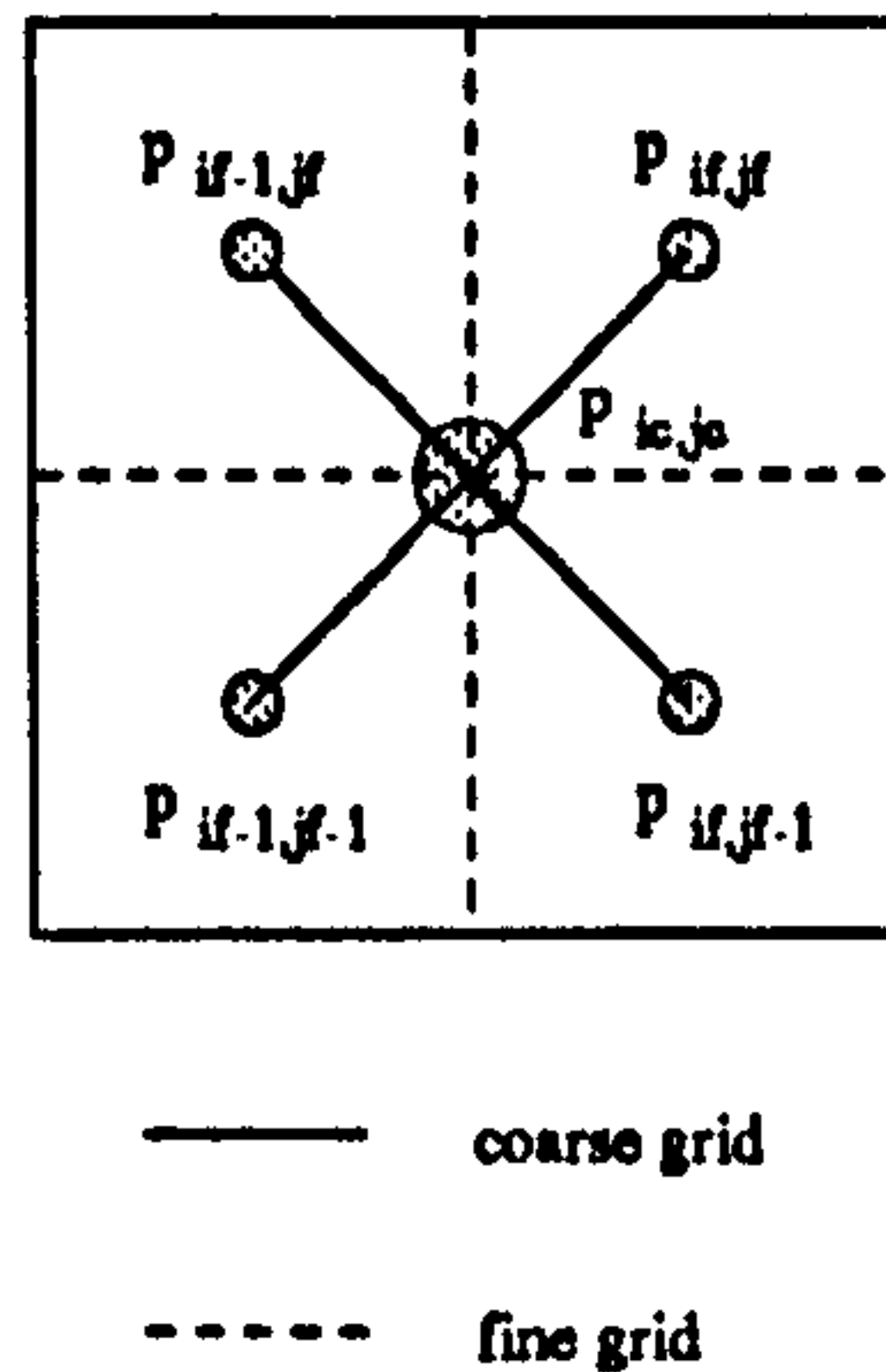


Figure 4.2: Discrete values for the restriction of the pressures

Restriction of the Fine Grid Residuals The residuals of the momentum equations are defined on cell edges. It is therefore possible to use the same operators as for the restriction of the velocities. The horizontal momentum residuals are restricted using (4.10) while the vertical momentum equations are restricted using (4.11). The pressure operator (4.12) is used for the restriction of continuity residuals to the coarse grid, since they are computed at cell centres.

Prolongation of the Corrections We need to interpolate both cell-centred and edge-defined coarse grid corrections. It is possible to transfer the velocities using standard (second order accurate) operators such as:

$$u_{if+1/2,jf}^{(f)} = \frac{3}{4}u_{ic+1/2,jc}^{(c)} + \frac{1}{4}u_{ic+1/2,jc+1}^{(c)}$$

$$u_{if+1/2,jf-1}^{(f)} = \frac{3}{4}u_{ic+1/2,jc}^{(c)} + \frac{1}{4}u_{ic+1/2,jc-1}^{(c)}$$

with, as previously, $if = 2ic$ and $jf = 2jc$. For *single phase* flows, we note that these operators are *not* conservative: the numerical mass flux on the coarse grid is not equal to the numerical mass flux on the fine grid. For computations only involving uniform grids, this is not a problem since the solution is only sought on the finest grid. Any mass variation during the grid transfers is incorporated in the defect.

For adaptive computations with composite grid involving several levels, it is necessary to conserve mass fluxes at grid interfaces, otherwise the discrete mass equation (3.37) may not admit a solution. This requirement will be satisfied by designing conservative interpolation schemes for both the restrictions of velocities values and the prolongation of velocity corrections. We note firstly, that the restriction operators described above (equations 4.10 and 4.11) conserve the mass fluxes exactly, and then describe below a prolongation operator for velocity corrections which shares the same property.

Referring to Figure 4.4, the basic idea is to write:

$$u^+ = U + \Delta U$$

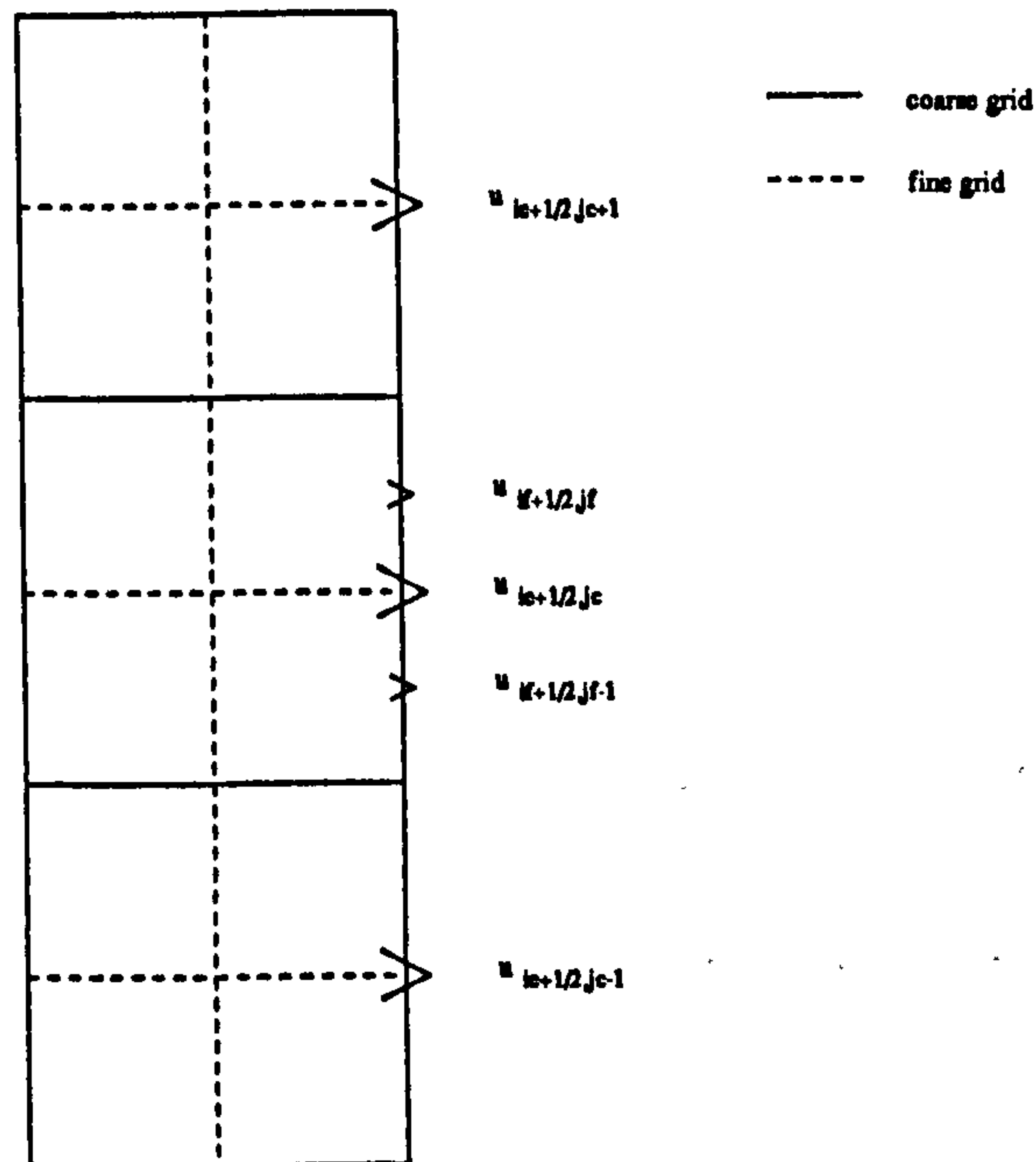


Figure 4.3: Discrete values for the prolongation of the horizontal velocities

$$u^- = U - \Delta U$$

so that the numerical mass flux on the fine grid:

$$\Delta y(u^+ + u^-) = 2\Delta y U,$$

is equal to the coarse grid flux.

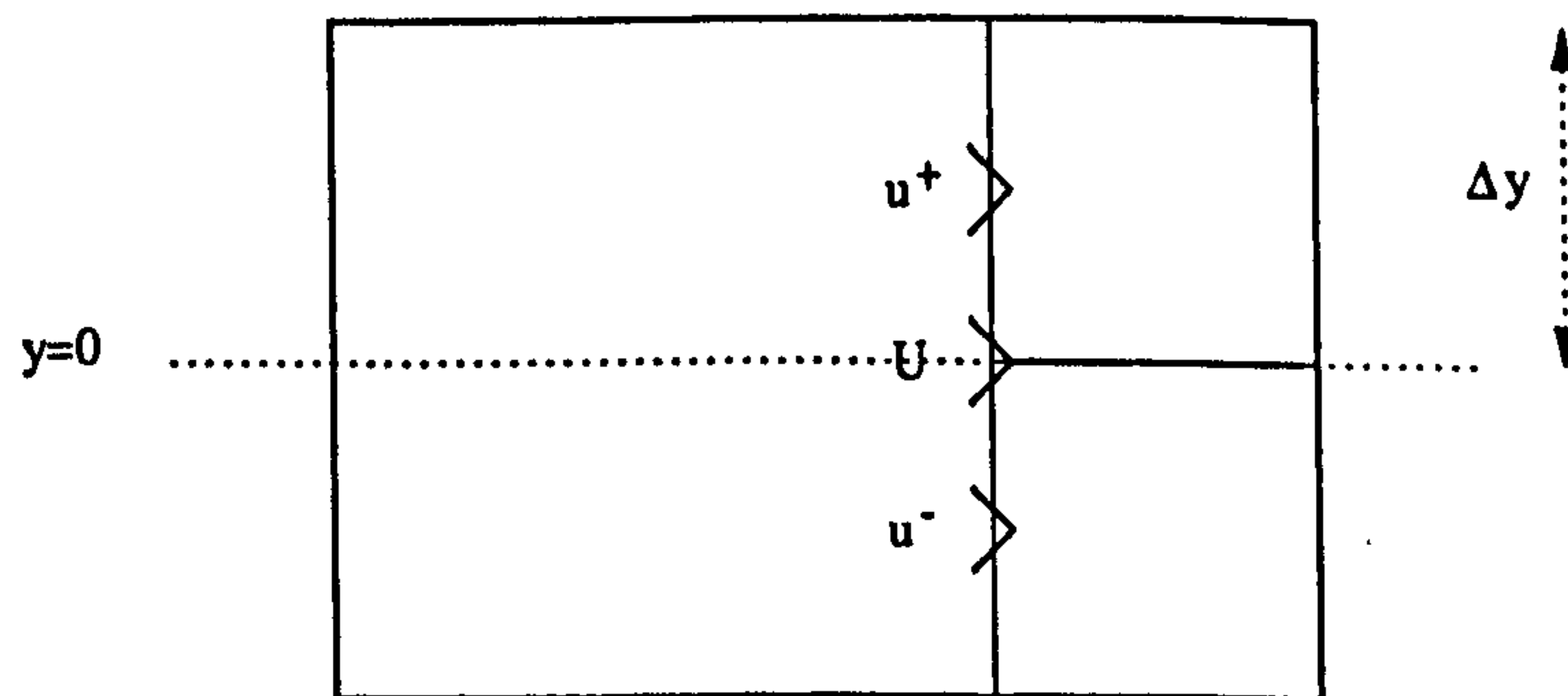


Figure 4.4: Arrangement of variables at grid interfaces

Geometrically, we can define ΔU as:

$$\Delta U \equiv \frac{\Delta y}{2} \left(\frac{dU}{dy} \right)_{y=0}$$

The interpolation is therefore completely defined when an approximate of dU/dy is specified and we simply choose:

$$\left(\frac{dU}{dy} \right)_{y=0} \approx \frac{1}{4\Delta y} (U(2\Delta y) - U(-2\Delta y))$$

so that:

$$u^+ = U(0) + \frac{1}{8}(U(2\Delta y) - U(-2\Delta y)),$$

$$u^- = U(0) - \frac{1}{8}(U(2\Delta y) - U(-2\Delta y)).$$

Noting that in FAS, we need to interpolate the corrections rather than the solution values to the fine grid, and using the notation of the previous sections, we obtain the following operators for the prolongation of the velocities:

$$\Delta u_{if+1/2,jf}^{(f)} = \Delta u_{ic+1/2,jc}^{(c)} + \frac{1}{8}\Delta u_{ic+1/2,jc+1}^{(c)} - \frac{1}{8}\Delta u_{ic+1/2,jc-1}^{(c)}, \quad (4.13)$$

$$\Delta u_{if+1/2,jf-1}^{(f)} = \Delta u_{ic+1/2,jc}^{(c)} - \frac{1}{8}\Delta u_{ic+1/2,jc+1}^{(c)} + \frac{1}{8}\Delta u_{ic+1/2,jc-1}^{(c)}, \quad (4.14)$$

$$\Delta u_{if-1/2,jf}^{(f)} = \frac{1}{2}(\Delta u_{ic+1/2,jc}^{(c)} + \Delta u_{ic-1/2,jc}^{(c)}) + \frac{1}{16}(\Delta u_{ic+1/2,jc+1}^{(c)} + \Delta u_{ic-1/2,jc+1}^{(c)}) - \frac{1}{16}(\Delta u_{ic+1/2,jc-1}^{(c)} + \Delta u_{ic-1/2,jc-1}^{(c)}), \quad (4.15)$$

$$\Delta u_{if-1/2,jf-1}^{(f)} = \frac{1}{2}(\Delta u_{ic+1/2,jc}^{(c)} + \Delta u_{ic-1/2,jc}^{(c)}) - \frac{1}{16}(\Delta u_{ic+1/2,jc+1}^{(c)} + \Delta u_{ic-1/2,jc+1}^{(c)}) + \frac{1}{16}(\Delta u_{ic+1/2,jc-1}^{(c)} + \Delta u_{ic-1/2,jc-1}^{(c)}). \quad (4.16)$$

The transfer operators for the vertical velocity are defined in a very similar way:

$$\Delta v_{if,jf+1/2}^{(f)} = \Delta v_{ic,jc+1/2}^{(c)} + \frac{1}{8}\Delta v_{ic+1,jc+1/2}^{(c)} - \frac{1}{8}\Delta v_{ic-1,jc+1/2}^{(c)}, \quad (4.17)$$

$$\Delta v_{if-1,jf+1/2}^{(f)} = \Delta v_{ic,jc+1/2}^{(c)} - \frac{1}{8}\Delta v_{ic+1,jc+1/2}^{(c)} + \frac{1}{8}\Delta v_{ic-1,jc+1/2}^{(c)}, \quad (4.18)$$

$$\Delta v_{if,jf-1/2}^{(f)} = \frac{1}{2}(\Delta v_{ic,jc-1/2}^{(c)} + \Delta v_{ic,jc+1/2}^{(c)}) + \frac{1}{16}(\Delta v_{ic+1,jc+1/2}^{(c)} + \Delta v_{ic+1,jc-1/2}^{(c)}) - \frac{1}{16}(\Delta v_{ic-1,jc+1/2}^{(c)} + \Delta v_{ic-1,jc-1/2}^{(c)}), \quad (4.19)$$

$$\Delta v_{if-1,jf-1/2}^{(f)} = \frac{1}{2}(\Delta v_{ic,jc+1/2}^{(c)} + \Delta v_{ic,jc-1/2}^{(c)}) - \frac{1}{16}(\Delta v_{ic+1,jc+1/2}^{(c)} + \Delta v_{ic+1,jc-1/2}^{(c)}) + \frac{1}{16}(\Delta v_{ic-1,jc+1/2}^{(c)} + \Delta v_{ic-1,jc-1/2}^{(c)}). \quad (4.20)$$

These formula are modified near walls.

It only remains to specify the prolongation operator for pressure corrections. Since there are no difficulty to complicate the definition of the operator, we simply choose:

$$\Delta p_{if,jf}^{(f)} = \Delta p_{ic,jc}^{(c)}, \quad (4.21)$$

$$\Delta p_{if,jf-1}^{(f)} = \Delta p_{ic,jc}^{(c)}, \quad (4.22)$$

$$\Delta p_{if-1,jf}^{(f)} = \Delta p_{ic,jc}^{(c)}, \quad (4.23)$$

$$\Delta p_{if-1,jf-1}^{(f)} = \Delta p_{ic,jc}^{(c)}, \quad (4.24)$$

which are accurate to first order. Alternatively, the following second order formulae can be adopted:

$$\Delta p_{if,jf}^{(f)} = \frac{1}{16}(9\Delta p_{ic,jc}^{(c)} + 3\Delta p_{ic+1,jc}^{(c)} + 3\Delta p_{ic,jc+1}^{(c)} + \Delta p_{ic+1,jc+1}^{(c)}), \quad (4.25)$$

$$\Delta p_{if,jf-1}^{(f)} = \frac{1}{16}(9\Delta p_{ic,jc}^{(c)} + 3\Delta p_{ic+1,jc}^{(c)} + 3\Delta p_{ic,jc-1}^{(c)} + \Delta p_{ic+1,jc-1}^{(c)}), \quad (4.26)$$

$$\Delta p_{if-1,jf}^{(f)} = \frac{1}{16}(9\Delta p_{ic,jc}^{(c)} + 3\Delta p_{ic-1,jc}^{(c)} + 3\Delta p_{ic,jc+1}^{(c)} + \Delta p_{ic-1,jc+1}^{(c)}), \quad (4.27)$$

$$\Delta p_{if-1,jf-1}^{(f)} = \frac{1}{16}(9\Delta p_{ic,jc}^{(c)} + 3\Delta p_{ic-1,jc}^{(c)} + 3\Delta p_{ic,jc-1}^{(c)} + \Delta p_{ic-1,jc-1}^{(c)}). \quad (4.28)$$

4.2.3 Boundary Conditions in pamg

The boundary conditions implemented in the pamg solver are conventional as far as the velocities are concerned: Dirichlet conditions are imposed at inlets; no-slip condition ($u = v = 0$) at solid walls, and outflow conditions for developed flow at outlets. pamg is a coupled solver. Hence there are no pressure equations to be solved and since the momentum equations are not solved at inlets and solid walls, no pressure boundary conditions are needed at such boundaries.

At outlets, however, boundary conditions for the pressure are still required. This is done *implicitly* by fixing, on the finest grid only, the pressure to zero on a vertical line just downstream of the outlet. Boundary values for the coarser grids are then obtained from the restriction operator (4.12).

4.2.4 Adaption in pamg

Adaption in pamg is very closely linked to the FAS multigrid algorithm. As discussed in Section 2.5.8, FAS is highly suitable for adaptive computations. However, the fact that adaption is built in the solver has implications for the multigrid algorithm and particularly the transfer operators. Adaption necessitates the design of prolongation and restriction operators which conserve mass across grids. Another distinctive feature of pamg is that adaptive grids are generated *automatically* as the solution proceeds.

The refinement algorithm is based on the following basic idea: given a grid at level k which may be uniform or composite, solve the equations to obtain an estimate of the truncation errors for that problem. Based on the estimate of the truncation errors, refine the grid at level k in regions of high truncation errors to obtain the composite grid at level $k + 1$. If $k + 1$ is the highest level for which the solution is sought, then solve the equations to the required tolerance. Otherwise, solve them to obtain an estimate of the truncation errors, in order to refine up to level $k + 2$.

The first corner-stone of the technique is that the FAS method provides estimates of the truncation errors, as an intrinsic part of the solution procedure. The FAS multigrid method generates a set of right-hand side functions on all grids, given by equation (see Section 2.5.6):

$$f^{k-1} \equiv I_k^{k-1}(f^k - \mathcal{L}^k(\tilde{u}^k)) + \mathcal{L}^{k-1}(\hat{I}_k^{k-1}\tilde{u}^k) \quad (4.29)$$

If κ is the finest grid level defined at a given point, then $f^{\kappa-1}$ tends to the defect:

$$\tau_\kappa^{\kappa-1} = \mathcal{L}^{\kappa-1}(I_\kappa^{\kappa-1}u^\kappa) - I_\kappa^{\kappa-1}(\mathcal{L}^\kappa u^\kappa)$$

as defined in Section 2.5.8.

$f^{\kappa-1}$ is an approximation of the truncation error on the grid $\Omega^{\kappa-1}$, and as such, provides a sound refinement estimate. It is also important to note that within the framework of FAS multigrid methods, the error estimation does not entail any extra work: the computation of $f^{\kappa-1}$ is an intrinsic part of the FAS procedure.

Once the error estimator has been chosen, the refinement algorithm is almost trivial. The details are as follows. They are strongly influenced by two factors:

- *Parallelisation.* `pamg` was designed as a parallel code. A natural approach to the parallelisation of the solution algorithm is to group together groups of neighbouring cells so that processing tasks on cells can easily be distributed across available processors and transfer minimised. These groups of cells are called *patches*.
- *Simplification of the implementation.* If refined objects are similar to the non-refined objects, then the computer code will be much easier to design. This is a strong argument in favour of structured grids.

In the present version of `pamg`, a patch contains 4×4 cells. In the context of adaptive computations, each patch is thought of as containing four *quadrants* of equal size (2×2 cells). A patch at level $k+1$ exactly matches a quadrant at level k and a locally refined grid at level $k+1$ is simply obtained by overlaying certain quadrants at level k with level $k+1$ patches. A uniform grid at level $k+1$ is simply obtained by refining all the quadrants at level k .

Obviously, it is the quadrants with large truncation errors which should be refined and in `pamg`, any non-refined quadrant at level k which satisfies:

$$[\tau_{k+1}^k] \geq \left(\frac{1}{N_{vis}} \sum_{N_{vis}} [\tau_{k+1}^k]^\gamma \right)^{\frac{1}{\gamma}} \quad (4.30)$$

is refined. In equation (4.30), the following definitions apply: $[\tau_{k+1}^k]$ is the *average* L_2 norm of the defect (computed on grid $k-1$) over the quadrant. N_{vis} is the number of

level k quadrants which are as yet unrefined, and are therefore eligible for refinement, on all levels. γ is a parameter which controls the extent of the refinement. The larger γ , the more selective the refinement algorithms as the average:

$$\frac{1}{N_{vis}} \sum_{N_{vis}} [\tau_{k+1}^k]^\gamma \quad (4.31)$$

will tend to favour the larger truncation errors. The expression (4.31) can be considered to define the γ -norm of the defect.

As indicated in Section 2.6, there are two crucial issues for a successful implementation of adaption: one is the conservation of mass fluxes at grid interfaces – this is dealt with by the multigrid transfer operators – and the second one is the location of grid interfaces. Unless the grid interfaces are located at points where the flow solution does not change significantly with respect to the grid size, discontinuities in the solution may be generated. In pang, this last condition is met by forcing refinement if the ratio of grid size is greater than 2 for any of the eight neighbours. Hence, as finer and finer grids are generated at points of high truncation error, the refinement zone keeps expanding and in most cases, grid interfaces are pushed away from regions where the flow solution is not smooth. See Figures 6.15 and 6.16 for an illustration.

Besides γ , adaptive computations are mainly defined by two parameters:

- k_u is the level of the finest uniform grid ($k_u \geq 2$)
- k_f is the finest level at which the solution is sought

For the refinement algorithm to work properly, it is important that approximate solution obtained with the uniform grid at level k_u is sufficiently representative of the exact solution to provide a reliable estimate of the truncation error. The lower k_u is, the more gains are available from adaption. However, if the initial uniform grid is too coarse, some important features of the flow may not appear on that grid and consequently, it may prove difficult to recapture them.

Solutions of the equations to obtain an estimate of the truncation errors are referred to as intermediate computations for which the level of accuracy required can be relaxed compared to the accuracy at which the “real” solution, i.e the solution on a composite grid up to level n_f , is sought. Consequently, these intermediate computations are usually cheap compared with the final application of the FAS.

A composite grid up to level k contains visible quadrant on all levels from grid k_u inclusive. In order to obtain the composite grid at level $k + 1$, all visible quadrants – not just those at level k – are eligible for refinement. Once the final grid is obtained, the flow solution is accepted when the average residual, *on the finest level*, is lower than the required tolerance. By construction of the multigrid algorithm,

this condition is sufficient to ensure that the average residual on all levels is lower than the tolerance.

When composite grids are used there are refined and unrefined quadrants at any given level. One of the advantages of the underlying FAS method is that discretised equations are independent of the refinement status of quadrants. The only difference lies in the right-hand-side terms. For cells which are part of non-refined quadrants, the right-hand-side f^k in equation (4.29) is zero (possibly after translation) while for cells in refined quadrants f_k is non-zero, generally speaking.

4.3 From pang to pang-multiphase: Algorithmic Issues

In this section, the pang-multiphase solution algorithm is described mainly by contrasting it with the single phase solver. Both solvers have the same structure, as would be expected. Apart from the number of equations and unknowns which is larger, the main differences concern the quasi-Newton solver. Obtaining a convergent method has required a much more careful treatment. Grid transfers also had to be extended but this posed no conceptual difficulty. Finally, the multiphase boundary conditions and the implementation of adaption are discussed.

4.3.1 A Globally Convergent Quasi-Newton Solver for Multiphase Flows

The pang-multiphase solver differs from its single phase counterpart in three main areas:

- firstly, the number of equations and unknowns;
- secondly, the approximation for the Jacobians;
- thirdly, the treatment of the Newton corrections.

These are examined in turn.

Equations and Unknowns In pang-multiphase as in pang, the quasi-Newton solver is based on the Symmetrical Coupled Gauss Seidel procedure described in Section 4.2.1. One advantage of this approach is that it greatly simplifies the analysis of the method since only the discrete equations solved for each cell must be defined.

If we consider a single computational cell (i, j) , we can write twelve equations for the twelve unknowns defined on that cell. For each phase, we can write a continuity equation, two horizontal momentum equations centred around the points $(i - 1/2, j)$ and $(i + 1/2, j)$ respectively, and two horizontal momentum equations centred around the points $(i, j - 1/2)$ and $(i, j + 1/2)$ respectively. The equations solved by `pamg-multiphase` are given in Section 3.3 – see page 80.

The problem is closed by:

$$[p_1]_{i,j} - [p_2]_{i,j} = 0$$

and:

$$[r_1]_{i,j} + [r_2]_{i,j} = 1.$$

The SCGS procedure requires that the local non-linear system of algebraic equations:

$$\mathbf{f}(\Phi) = 0 \quad (4.32)$$

where: ¹

$$\Phi = \left\{ \begin{array}{cccccccc} [u_1]_{i-1/2,j} & [u_1]_{i+1/2,j} & [v_1]_{i,j-1/2} & [v_1]_{i,j+1/2} & [p_1]_{i,j} & [u_2]_{i-1/2,j} & [u_2]_{i+1/2,j} \\ [v_2]_{i,j-1/2} & [v_2]_{i,j+1/2} & [p_2]_{i,j} & [r_1]_{i,j} & [r_2]_{i,j} \end{array} \right\}^T,$$

should be solved. This is achieved by applying Newton's method.

Approximations for the Jacobian If J denote the Jacobian matrix of the \mathbf{f} on the cell (i, j) , given an approximation $\Phi^{(n)}$ of the solution $\Phi^{(0)}$ of (4.32), the Newton correction $\Delta\Phi$ is defined by:

$$J\Delta\Phi = -\mathbf{f}(\Phi^{(n)}). \quad (4.33)$$

After inversion of this system, the approximation of the solution can be updated:

$$\Phi^{(n+1)} = \Phi^{(n)} + \Delta\Phi \quad (4.34)$$

Again, the key issue is to obtain an expression for J . This may be cumbersome as the number of variables increases. In addition, the presence of boundaries may complicate the derivation of correct expressions.

In some cases, the expressions for J can be approximated: in the the single phase `pamg` code for instance, J is obtained after approximating $\mathbf{f}(\Phi)$ by $A(\Phi)\Phi$, and neglecting the non-diagonal velocity entries. Experimental evidence seems to suggest that in the case of multi-phase flows, the expressions for the Jacobian need to be

¹The volume fractions are grouped together for consistency with the actual computer implementation.

much more accurate than in the single phase case as a result of the increased degree of non-linearity of the solutions. In particular, a straightforward extension of the expressions derived in [13] (see also Section 4.2.1) has proved insufficient to ensure the convergence of the method.

In the present study, we have used *automatic differentiation* to obtain expressions for the Jacobian. This is a relatively new technique which has important applications in optimisation problems (see for instance [104, 105]). It aims to deduce the derivative of an expression from its implementation in a computer language by applying the chain rule of differentiation:

$$\frac{d}{dx}(f \circ g) = \frac{dg}{dx} \times \left(\frac{df}{dx} \circ g\right)$$

Automatic differentiation relies on the fact that in a computer programme, any mathematical function is expressed as a sequence, possibly a lengthy one, of elemental operations and functions for which the differentiation rules are known. This approach differs totally from *symbolic differentiation*. In the latter case, an expression for the derivative is provided as a function of the independent variables whereas automatic differentiation provides one value for the derivative corresponding to one set of values for the independent variables. The advantage of automatic differentiation is that it can be easily incorporated in existing numerical software. Automatic differentiation may usefully be applied whenever Jacobians of complicated functions are sought — hence its use in optimisation.

Many automatic differentiation packages are implemented as pre-processors which analyse the implementation of a mathematical function in a particular computer language, apply the chain rules of differentiation to obtain the derivative, and produce a sub-programme implementing the computation of the derivatives in the same programming language, which can be FORTRAN [106] or C [107].

In this study, the package AD01 from the Harwell Subroutine Library [108] has been used. It is not a pre-processor but a collection of routines which computes the derivatives of an expression at run-time by relying on the operator overloading capabilities of `fortran90` [109]. Table 4.1 shows a simple `fortran90` programme which uses the AD01 package to differentiate automatically a polynomial.

This choice is costly in terms of performance – the automatic differentiation can account for up to 80 % of the computation time – but it has important advantages for code development.

- Firstly, it ensures that the Jacobian and the residual in (4.33) are consistent with each other, thus reducing the probability of implementation errors.
- Secondly, different discretisation options can be verified easily with minimal code writing. This has proved very useful in the course of this project.

PROGRAM TEST

```

USE HSL_ADO1_BACKWARD_DOUBLE           ! use the AD package
                                         ! (backward method,double
                                         ! precision)

INTEGER::DEGREE=1
DOUBLE PRECISION VALUE(2),GRAD(2),FUNC
TYPE (ADO1_REAL)::X(2),F=ADO1_UNDEFINED ! data manipulated by the AD
                                         ! package allowing operator
                                         ! overloading

READ(*,*)VALUE

CALL ADO1_INITIALIZE(DEGREE,X,VALUE)
WRITE(*,'(A,2ES12.4)') ' AT X=',REAL(VALUE)

F=(X(1)**4-3D0)**2 + X(2)**3           ! definition of the function
                                         ! to differentiate.

CALL ADO1_VALUE(F,FUNC)                 ! compute the value of f
WRITE(*,'(A,2ES12.4)') 'F=',REAL(FUNC)

CALL ADO1_GRAD(F,GRAD)                  ! compute its derivative
WRITE(*,'(A,2ES12.4)') 'GRAD=',REAL(GRAD)

END PROGRAM TEST

```

Table 4.1: Simple fortran90 program illustrating the use of the ADO1 package for automatic differentiation

Eventually, when a good discretisation has been obtained, the Jacobians could be computed in a more traditional way to dramatically reduce the computational cost. The code could then easily be tested against the automatic differentiation results.

Globally Convergent Newton Methods It is well known that Newton's method is not globally convergent: if the initial guess is not close enough to the actual solution, the method may fail due to the size of the corrective steps which are taken. This is particularly true if the condition number of J is large.

This behaviour is a consequence of the fact that as the degree of non-linearity of the equations increases, the domain of validity of the Newton linearisation may be reduced and the large correction steps which are usually taken when the residual is large, become more and more inaccurate.

Line-searching [110] is a simple procedure aimed at making Newton's method globally convergent. Its principle is very simple: if the Newton correction leads to an increase in the Euclidean norm of the residual, then the correction is progressively reduced until it results in a reduction of the residual.

Given the correction $\Delta\Phi$ defined by (4.33), we define the new approximation of the solution by:

$$\Phi^{(n+1)} = \Phi^{(n)} + \lambda\Delta\Phi \quad (4.35)$$

The scaling factor λ is chosen so that the correction reduces the Euclidean norm of \mathbf{f} :

$$\|\Phi^{(n+1)}\|_2 < \|\Phi^{(n)}\|_2 \quad (4.36)$$

It is always possible to find λ for which (4.36) is satisfied because the direction of the Newton correction is a descent direction for the Euclidean norm. Let:

$$f = \frac{1}{2}\mathbf{f}\cdot\mathbf{f}$$

then the gradient of f is:

$$\nabla f = J^T\mathbf{f}. \quad (4.37)$$

Using tensor notation, $f = 1/2f_i f_i$ and:

$$(\nabla f)_i = \frac{1}{2} \frac{\partial f_j f_j}{\partial x_i} = f_j \frac{\partial f_j}{\partial x_i}$$

which establishes (4.37) given that:

$$J^T = \frac{\partial f_j}{\partial x_i}$$

Therefore, we have:

$$\nabla f \cdot \Delta\Phi = (J\mathbf{f})^T(-J^{-1}\mathbf{f}) = -\mathbf{f}^T\mathbf{f} = -\mathbf{f}\cdot\mathbf{f}$$

This shows that $\nabla f \cdot \Delta\Phi$ is always negative. In other words, the Newton correction has a negative component along the direction of the gradient of f . Consequently, f must decrease in the direction of the Newton correction $\Delta\Phi$. The only exception is the case where the approximation is a local minimum for f . However, in practice, this does not appear to be cause any difficulty for our problems.

The main question which remains to be answered is how to choose λ efficiently. It is possible to choose λ such that $\|\Phi^{(n+1)}\|_2$ is minimised. This is optimal but not necessary. Any correction which reduces the residual is acceptable. Hence, the computational cost of the algorithm can be minimised. In our implementation, we first try to apply the full Newton correction in order to benefit from the optimal quadratic convergence rate near the solution. If the full Newton step does not reduce the residual, the correction step is progressively reduced until a satisfactory value of λ is found. The successive trial values for λ are obtained by the following procedure, described in [110]:

f is considered to be a function of λ only (in the direction given by the Newton correction):

$$f(\lambda) = f(\Phi_n + \lambda\Delta\mathbf{x}_n)$$

and information available about f is used to model it as a quadratic or a cubic polynomial on the interval $[0, 1]$. The chosen value for λ is the value which minimises the polynomial. If $f(\lambda) < f(0)$, the value of λ is accepted. Otherwise, f is modelled by another polynomial and the minimisation process is performed again until a suitable value for λ is found.

The polynomial modelling f is constructed in the following way: it can be seen as a function of λ only. We then know the value of the function at $\lambda = 0$ and $\lambda = 1$ (i.e. the full Newton step). In addition, we know the derivative $f'(\lambda)$ at $\lambda = 0$ because $\nabla f = J^T \mathbf{f}$. f can therefore be modelled as a quadratic. Minimisation yields a value λ_1 . If this corrected step is still not acceptable, we then know the value of the function at three points: $\lambda = 0, \lambda_1, 1$ in addition with $f'(0)$ and f can be modelled as a cubic. In general, f will be modelled by a cubic, taking into account the function values at $\lambda = 0, \lambda_m, \lambda_{m-1}$ together with the derivative $f'(0)$.

For *single phase* flows, Newton's method with the approximate Jacobian given by equation (4.9), can be used directly. Line-searching appears to be unnecessary. By contrast, for *multiphase flows*, line-searching appears to be always necessary. This effect may be due to the increased degree of non-linearity of the equations, but it is also a consequence of the approximation chosen for the Jacobians. If exact numerical Jacobians, obtained by automatic differentiation, are used for single phase flows, then line-searching is also necessary.

Under-Relaxation The line-searching algorithm described above can be considered to be an "intelligent" form of under-relaxation. The amount by which the correction step is reduced by the line search depends on how good the correction is. In this context, there may be no need for further under-relaxation. Experimental evidence suggests otherwise (see section 5.4.6): it has proved beneficial, and sometimes necessary, to relax the correction after line-searching. This is implemented very simply. A further relaxation parameter λ_r is introduced and equation (4.34) is modified to:

$$\Phi^{(n+1)} = \Phi^{(n)} + \lambda_r LS(\Delta\Phi) \quad (4.38)$$

where $LS(\Delta\Phi)$ is the correction resulting from the application of the line-searching procedure. We require that the SCGS procedure should have good smoothing properties for a successful application of multigrid. Under-relaxation greatly enhances this. Typically, we choose $0.5 \leq \lambda_r \leq 1.0$. Hence, the level of under-relaxation required is far less severe than in the case of most segregated solvers.

4.3.2 Stable Interpolation Procedures on Staggered Grids

In Section 3.3.4, it has been shown that the discrete multiphase continuity equations for the cell (i, j) can be written:

$$\frac{[ru]_{i+1/2,j} - [ru]_{i-1/2,j}}{\Delta x} + \frac{[rv]_{i,j+1/2} - [rv]_{i,j-1/2}}{\Delta y} = 0 \quad (4.39)$$

Since we use a staggered grid, every volume fraction has to be interpolated. A first obvious choice is geometric interpolation:

$$r_{i+1/2,j} \equiv \frac{1}{2}(r_{i+1,j} + r_{i,j})$$

$$r_{i-1/2,j} \equiv \frac{1}{2}(r_{i-1,j} + r_{i,j})$$

$$r_{i,j+1/2} \equiv \frac{1}{2}(r_{i,j+1} + r_{i,j})$$

$$r_{i,j-1/2} \equiv \frac{1}{2}(r_{i,j-1} + r_{i,j})$$

which is second order accurate. However, if (4.39) is then rewritten as:

$$\begin{aligned} & \left(\frac{1}{2\Delta x} u_{i+1/2,j} \right) r_{i+1,j} - \left(\frac{1}{2\Delta x} u_{i-1/2,j} \right) r_{i-1,j} + \\ & \left(\frac{1}{2\Delta y} v_{i,j+1/2} \right) r_{i,j+1} - \left(\frac{1}{2\Delta y} v_{i,j-1/2} \right) r_{i,j-1} + \\ & \left[\frac{1}{2\Delta x} (u_{i+1/2,j} - u_{i-1/2,j}) + \frac{1}{2\Delta y} (v_{i,j+1/2} - v_{i,j-1/2}) \right] r_{i,j} = 0, \end{aligned}$$

it appears immediately that as $\Delta x, \Delta y \rightarrow 0$, the derivatives for the (scaled) mass flux with respect to the volume fraction r_{ij} :

$$\frac{1}{2\Delta x} (u_{i+1/2,j} - u_{i-1/2,j}) + \frac{1}{2\Delta y} (v_{i,j+1/2} - v_{i,j-1/2}) \quad (4.40)$$

tends to zero for incompressible flows. By contrast, the derivative of the continuous mass flux is non-zero in general:

$$\frac{\partial}{\partial r}(ru + rv) = u + v$$

Geometric interpolation for the volume fractions in the continuity equations generates instabilities.

In order to get a consistent discretisation, we used a first order accurate upwind interpolation for the volume fraction:

$$\begin{aligned}
r_{i+1/2,j} &\equiv \frac{1}{2} \left[(1 - \text{sgn}(u_{i+1/2,j}))r_{i+1,j} + (\text{sgn}(u_{i+1/2,j}) + 1)r_{i,j} \right] \\
r_{i-1/2,j} &\equiv \frac{1}{2} \left[(1 - \text{sgn}(u_{i-1/2,j}))r_{i,j} + (\text{sgn}(u_{i-1/2,j}) + 1)r_{i-1,j} \right] \\
r_{i,j+1/2} &\equiv \frac{1}{2} \left[(1 - \text{sgn}(v_{i,j+1/2}))r_{i,j+1} + (\text{sgn}(v_{i,j+1/2}) + 1)r_{i,j} \right] \\
r_{i,j-1/2} &\equiv \frac{1}{2} \left[(1 - \text{sgn}(v_{i,j-1/2}))r_{i,j} + (\text{sgn}(v_{i,j-1/2}) + 1)r_{i,j-1} \right]
\end{aligned}$$

If we assume that $u_{i+1/2,j}$, $u_{i-1/2,j}$, $v_{i+1/2,j}$ and $v_{i-1/2,j}$ are all positive, then:

$$\begin{aligned}
r_{i+1/2,j} &= r_{ij} \\
r_{i-1/2,j} &= r_{i-1,j} \\
r_{i,j+1/2} &= r_{ij} \\
r_{i,j-1/2} &= r_{ij-1}
\end{aligned}$$

and the derivative of the discrete mass flux with respect to r_{ij} , i.e. the analogous of (4.40) becomes:

$$\frac{1}{\Delta x} u_{i+1/2,j} + \frac{1}{\Delta y} v_{i+1/2,j}. \quad (4.41)$$

Expression (4.41) is therefore more consistent with the derivative of the continuous mass flux. Similar expressions can be derived when the velocities have different signs. Note that with upwind differencing, the derivative can never be zero as this would imply that mass is not conserved in the control volume.

At first, geometric interpolation of the volume fraction was chosen because it appeared more accurate and consequently, considerable effort had to be expended to determine the cause of the observed instabilities. Isolating interpolation effects as the cause was instrumental in the positive outcome of this study, and required a very significant amount of time. A large number of hypotheses were formulated and tested. The main difficulty in such situations is the need to determine whether the behaviour of the software is due to genuine numerical difficulties or simply to implementation errors.

4.3.3 Interpolation Operators for Multiphase Flows and Implications for Adaption

The extension of the single phase transfer operators to multiphase is actually quite straightforward. The only quantities which require special attention are the volume fractions but since they are cell-centred, their treatment closely resembles that of the pressures.

Restriction of the Residuals The single phase operators which restrict the continuity and momentum residuals are directly applicable to multiphase computations since they are defined at identical locations on grids. The only difference is that M times as many residuals need to be transferred (where M is the number of phases).

As far as the closure relationships (see Section 3.3.1):

$$r_1 + r_2 + \dots + r_M = 1 \quad (4.42)$$

and:

$$p_1 = p_2 = \dots = p_M \quad (4.43)$$

are concerned, no residual transfer is necessary because they essentially act as constraints which are satisfied – to machine precision – for all cells and at any stage of the computations. This is true provided that the closure relationships are satisfied by the initial guess. This fact can be established by the following argument: if on a given cell, (4.42) and (4.43) are satisfied by the current iterate, then they are also satisfied by their Newton corrected values, both before and after line-searching because these processes are linear. Since these conditions are imposed on the initial guess, they are always satisfied to machine precision. Furthermore, we use linear transfer operators for the transfers so if values on a grid satisfy (4.42) and (4.43), their prolonged or restricted values also do so, provided that the interpolation is consistent.

Equations (4.42) and (4.43) therefore essentially act as constraints during the relaxation process and only the residuals of the conservation laws need to be restricted.

Transfer Operators for the Volume Fractions Volume fractions are cell-centred quantities. It is therefore possible to use the same interpolation operators as for the pressures. The restriction does not pose any problems. We use the formula given by equation (4.12). As far as the prolongation of the volume fraction corrections is concerned, it can be defined by equations (4.21) to (4.24) (for first order accuracy) or (4.25) to (4.28) (for second order accuracy).

Boundary conditions for the volume fractions must be supplied. This contrasts with the pressures for which, by the very nature of the quasi Newton coupled solvers (Section 4.2.1) no boundary conditions need to be applied². In order to simplify the transfer procedures for the volume fractions, boundary conditions are applied just after the transfers.

If second order prolongation is selected for the volume fractions, the corrections for boundary cells need careful handling because of the existence of boundary conditions. The interpolation formulae are modified. See Section 5.4.8 for a detailed presentation of the procedure adopted. If first order prolongation is chosen, no special formulation is needed at boundary cells since the fine grids corrections are taken to be equal to the nearest coarse grid correction.

²At outlets, pressure boundary conditions are still needed – see Section 4.2.3.

Prolongation and Physically Admissible Solutions A physically admissible solution is such that every discrete value of the volume fractions belongs to the interval $[0, 1]$. When fine grid volume fractions are restricted to the next coarser grid, it is impossible to generate non-physical values on the coarse grid, provided that the all volume fractions take physically acceptable values on the fine grid. This is due to the existence of a maximum principle which insures that the interpolated values are bounded by the interpolants. It is also impossible for boundary values to take non-physical values. Indeed this is why a low order of accuracy is chosen for the application of the boundary conditions on the volume fractions.

By contrast, in the prolongation stage, it is possible to correct the fine grid volume fractions in such a way that they become non-physical. Obviously, this is more likely when residuals are still high or when the solution field for the volume fractions locally approaches 1 or 0. When non-physical values are generated, the multi-grid algorithm, not surprisingly, quickly diverges. The solution is to test the fine grid values after correction and reset them to physical values if necessary:

$$\begin{aligned} \text{if } r_{ij}^{(f)} > 1, \text{ set } r_{ij}^{(f)} &= 1 - \epsilon \\ \text{if } r_{ij}^{(f)} < 0, \text{ set } r_{ij}^{(f)} &= \epsilon \end{aligned} \quad (4.44)$$

where ϵ is a small number (typically 10^{-6}), added so that correction systems are not made singular by the presence of an exactly zero volume fraction. Alternatively, corrections may be discarded if they lead to non-physical values for the volume fractions:

$$\begin{aligned} \text{if } r_{ij,old}^{(f)} + \Delta r_{ij} > 1, \text{ set } r_{ij,new}^{(f)} &= r_{ij,old}^{(f)} \\ \text{if } r_{ij,old}^{(f)} + \Delta r_{ij} < 0, \text{ set } r_{ij,new}^{(f)} &= r_{ij,old}^{(f)} \end{aligned} \quad (4.45)$$

Procedures (4.44) and (4.45) are referred to as the “cut-off” and “no-change” strategies respectively. The results presented in Chapters 5 and 6 have been obtained using the second procedure, unless otherwise stated.

Conservative Interpolation Procedures For multiphase flows, mass fluxes are not conserved by the transfer operators for the velocities defined above — equations (4.10) to (4.11) and (4.13) to (4.20) — because multiphase mass fluxes depend not only on the velocities but also on the volume fractions.

It is possible to design transfer operators which conserve multiphase mass fluxes exactly. The main idea is to transfer the mass fluxes instead of the velocities. Velocities are then defined in a unique way from the discrete mass flux and volume fraction fields. The definition of the mass fluxes needs to be consistent with the discretisation of the continuity equations: the horizontal component of the mass flux is defined at the same locations as the horizontal velocity, its vertical component at the same grid points as the vertical velocities. In order to define the mass fluxes, we therefore need to interpolate the volume fractions and the same interpolation operators are chosen as in the continuity equations (see Section 4.3.2).

The process can be summarised as follows (see Figure 4.5):

1. On the source grid, compute the mass flux field as used in the continuity equations using formula of the type:

$$q_{i+1/2,j}^u = u_{i+1/2,j} \frac{1}{2} \left((1 + \text{sgn}(u_{i+1/2,j}))r_{i,j} + (1 - \text{sgn}(u_{i+1/2,j}))r_{i+1,j} \right) \quad (4.46)$$

$$q_{i,j+1/2}^v = v_{i,j+1/2} \frac{1}{2} \left((1 + \text{sgn}(v_{i,j+1/2}))r_{i,j} + (1 - \text{sgn}(v_{i,j+1/2}))r_{i,j+1} \right) \quad (4.47)$$

where q^u and q^v are respectively the horizontal and vertical mass fluxes.

2. Transfer the mass flux field to the target grid using the operators (4.10) to (4.11) (for a restriction) or (4.13) to (4.20) (for a prolongation).
3. Transfer the volume fraction field (or corrections) to the target grid using the same operators as for the non-conservative transfers.
4. Compute the velocity field on the target grid using the new mass flux field and the new volume fraction field. This can be done in a unique way, once the interpolation scheme for the volume fraction is specified. Again the interpolated values for the volume fractions are those used for the continuity equation. The following formulae are based on the fact that the volume fraction is a positive quantity.

$$u_{i+1/2,j} = \frac{2q_{i+1/2,j}^u}{\left((1 + \text{sgn}(q_{i+1/2,j}^u))r_{i,j} + (1 - \text{sgn}(q_{i+1/2,j}^u))r_{i+1,j} \right)} \quad (4.48)$$

$$v_{i,j+1/2} = \frac{2q_{i,j+1/2}^v}{\left((1 + \text{sgn}(q_{i,j+1/2}^v))r_{i,j} + (1 - \text{sgn}(q_{i,j+1/2}^v))r_{i,j+1} \right)} \quad (4.49)$$

We have implemented both the conservative and the non-conservative interpolation schemes. Some numerical experiments are reported in Section 6.2. Discrete solutions exist on composite grid for both formulations. Overall, the conservative scheme did not bring significant benefits in terms of accuracy, and was, as could be expected, noticeably less stable in regions where volume fractions take small values.

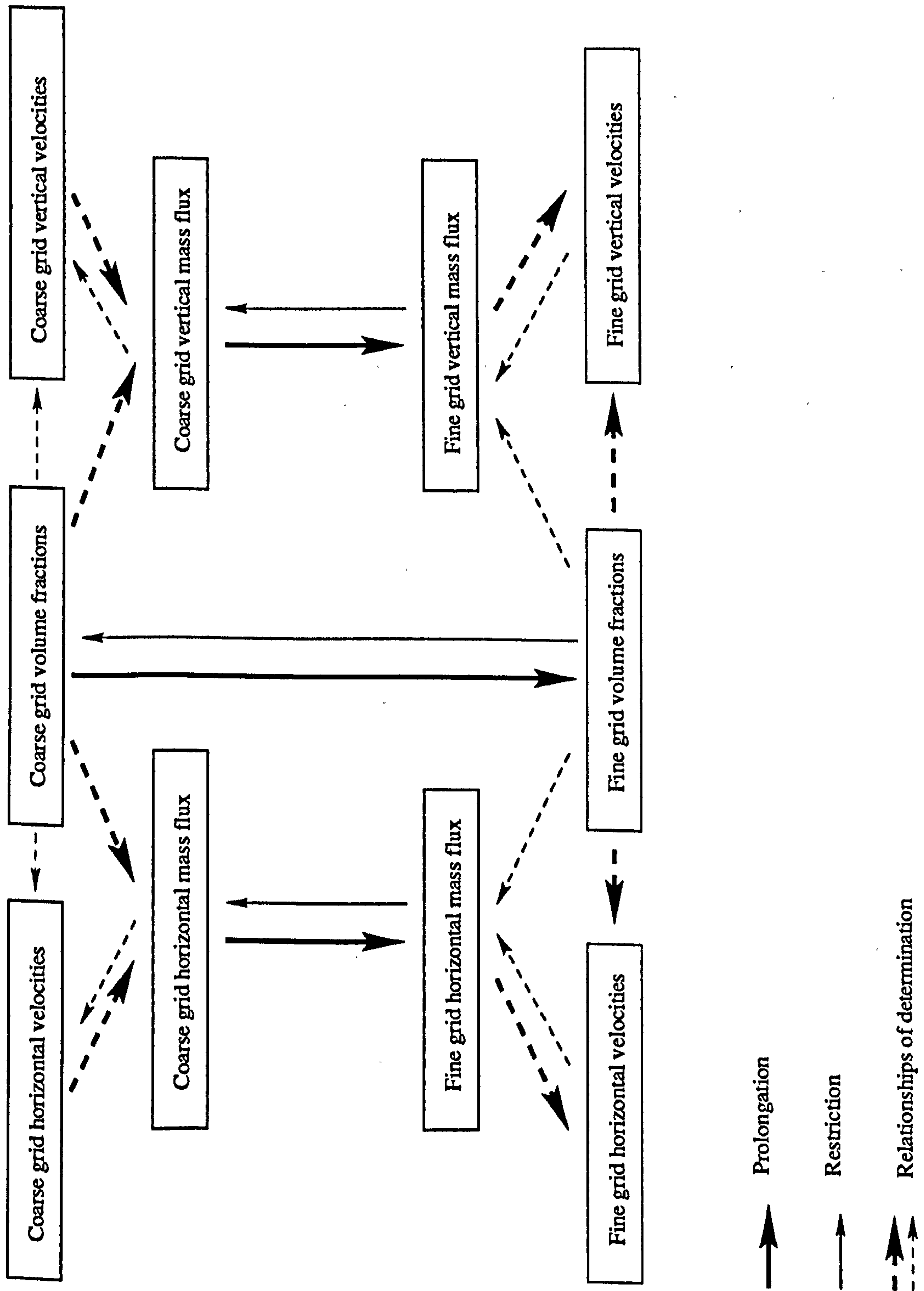


Figure 4.5: Conservative multiphase transfer procedure

4.3.4 Boundary Conditions

The derivation of appropriate boundary conditions for multiphase flows is difficult. In particular, boundary conditions for the volume fractions must be supplied at solid walls. In the present study, we have adopted the following boundary conditions.

1. At solid walls:

- $u = 0, v = 0$ (no slip conditions);
- $\partial_n r = 0$, where ∂_n denotes the derivative in the normal direction;

2. At outlets:

- $\partial_n u = 0$;
- $v = 0$;
- $\partial_n r = 0$;

At solid walls, homogeneous Neumann boundary conditions are imposed on the volume fraction, which could be interpreted as expressing the fact that near walls, the flow is dominated by the structure of the velocity field.

For the sake of simplicity, boundary conditions on the volume fractions are only first order accurate. This prevents the generation of non-physical volume fractions at boundaries. Second order boundary conditions were also tested, and did not have a significant impact on the solutions or on the convergence rates.

For single phase flows, Dirichlet boundary conditions which are exact for a developed flow can be specified at inlets (and outlets): namely, parabolic velocity profiles. We have not been able to find or derive equivalent conditions for multiphase flows. Instead, *entry flow* Dirichlet boundary conditions are specified *for all the multiphase results reported in this thesis*: the normal velocity is constant across inlets, and set to 1 for both phases. The volume fractions are set to 0.5. The benefit of such a simple set of boundary conditions is that the mass fluxes through the inlets are numerically equal for all grids.

4.3.5 Adaption in pamg-multiphase

The refinement algorithm for pamg-multiphase has not changed substantially from the single phase version. The only difference is the definition of $[\tau_{k+1}^k]$ in equation (4.30). In the multiphase case, defects are averaged over quadrants for each phase *separately*, using the same procedure as in the single phase case. $[\tau_{k+1}^k]$ is then defined as the arithmetic average over the phases of these “phasic” defects. Once the “phasic” defects are available, there are many ways of combining them, other than a simple arithmetic average, which could easily be implemented.

4.3.6 Summary of Algorithmic Issues

In this section, we have presented the main algorithmic issues which arose during the development of `pamg-multiphase`. These have required careful analysis and original solutions. In the author's opinion, they testify to the fact that the multi-fluid equations are much more difficult to handle than the Navier-Stokes equations.

The most important obstacle to overcome was the derivation of a consistent discretisation for the governing equations. Geometric interpolation for the volume fractions in the continuity equation leads to unstable discretisations — on staggered grids — which are not consistent with the continuous problem. A less accurate upwind interpolation scheme had to be used instead. Next, the design of a globally convergent quasi-Newton solver was complicated by the high degree of non-linearity of the equations. The design thus required the implementation of line-searching together with very accurate Jacobians which were obtained using automatic differentiation.

By comparison, the extension of the transfer operators was easier — but not trivial. The volume fractions have required special treatment at the boundaries, prolongation operators had to be modified so that only physically realisable solutions (i.e. $0 < r_{ij} < 1$ for all cells) are generated. Furthermore, it is difficult to conserve multiphase mass fluxes across grids because they involve both volume fractions and velocities which are defined at different points on a staggered grid. In view of the implementation of adaption, this is a potentially crucial problem. Consequently, conservative grid transfers based on explicitly transferring the mass fluxes in a conservative way, have also been designed.

Finally, the question of suitable boundary conditions for multiphase flows was also examined. Here again the situation is more complicated than in the single phase case. An obvious issue is the conditions applicable to the volume fractions, particularly at solid walls. Simple cases have been implemented and appear to be adequate (see Chapter 5).

4.4 From `pamg` to `pamg-multiphase`: Computational Issues

The basic solution algorithm chosen for the `pamg-multiphase` solver is efficient but, as Section 4.2 shows, it is also quite complex to implement because it is very complex. In addition, when it is applied to multiphase flows, a number of algorithmic issues arise which further complicate the solver (see Section 4.3). As a result, computational issues have proven important in the development of the code. In order to isolate numerical problems — a pre-requisite to their solution — implementation errors have to be reduced to a minimum. This in turns implies that the software should be well written. The more complex the algorithm, the more effort must be

devoted to the organisation of both code and data structures.

Since the starting point was a well validated single phase solver, the main objective was obviously to minimise code alterations so that reliability would be improved and development time shortened. This has required some careful thought and planning but the outcome was that most of the software components could be re-used. It was possible to restrict significant modifications to a relatively small number of routines, and the validation process was greatly facilitated.

In the author's opinion, the way in which the software was modified was a key factor in the success of the project.

4.4.1 Methodology Adopted

The extension of *pamg* to handle multi-phase problems has implications for both the data structures and the architecture of the code. The key feature is that, due to careful design, *numerical modifications are contained in a very small number of routines*. These are:

- the SCGS relaxation on a patch;
- the computation of the residuals on a patch;
- the computation of the patch refinement criterion.

These routines need information on all phases in order to perform their tasks, not surprisingly, since the equations are coupled. It is actually quite striking that this coupling is only significant in a very limited number of routines when the FAS multigrid method is implemented.

The remaining routines (which deal mainly with grid transfers and application of boundary conditions) treat each phase independently of the others. With appropriate data structures, it is therefore possible to treat a M -phase flow as a collection of M independent single phase flows. The single-phase routines can be re-used without modification and simply called M times, once for each phase. With this strategy, for most routines the extension of *pamg* to *pamg-multiphase* only involved adding DO loops over the number of phases at the relevant places!

Furthermore, *pamg* incorporates a dynamic patch allocation mechanism in which numerical pointers are used to indicate where the data for a patch are stored in memory. By considering the multiphase flow as just a collection of single phase flows, except for routines where the phases are coupled together, it is possible to preserve this mechanism and more importantly, the number of phases is a *dynamic parameter*: *pamg-multiphase* is not a two phase code, it is a real multiphase code (but as yet it is untested for $M > 2$).

Although a multiphase flow may be considered as a collection of single phase flows, an extra unknown is introduced for each phase, namely the volume fraction. Data structures have necessitated careful modification, in particular the COMMONs, but the main tasks which have arisen have been at the level of SUBROUTINES. They include in particular:

- The design of low level routines to handle the volume fractions. This was actually trivial because in most cases, the volume fractions can be handled by the same routine as the pressures because both quantities are cell-centred. In a small number of cases, specific routines had to be written. These were added at a later stage when real multiphase computations (i.e. cases where the volume fractions are allowed to vary) were performed.
- The modification of a significant number of medium level routines on two accounts: (i) argument lists in order to incorporate the volume fraction data, and (ii) calls to low level routines which actually handle the volume fractions

4.4.2 Phased Modification Plan and Testing

The tasks identified above have been carried out in order of increasing complexity:

1. argument lists;
2. handling of the volume fraction in the medium level routines which deal with pointer and storage manipulation;
3. addition of DO loops in intermediate drivers;
4. rewrite of all routines which process the phases in a coupled manner;
5. addition of specific routines for multiphase computations.

At the end of each phase, the code was always in working order in the sense that single phase problems were always successfully computed and the correct solutions were given. Of particular interest is the testing after phases 1 and 2. In order to validate the handling of the volume fractions, single phase flows were computed by storing pressure information in the space reserved for volume fractions. At that stage, pressure and volume fractions were handled in a symmetrical fashion (except during relaxation) and accordingly, the results were not affected by the data structure used to store the pressure information.

4.4.3 Run Times and Optimisation

The design strategy we adopted was to minimise the potential for implementation errors in `pamg-multiphase`: when developing a novel numerical technique, it is frustratingly easy to devote vast amounts of effort to solving a “numerical” problem which turns out to be nothing but an implementation error. The price paid is that the significant level of optimisation achieved in `pamg` has been lost in `pamg-multiphase`. As a result, run times are increased by several orders of magnitude. A systematic study of the performance of the solution therefore became more difficult and costly in terms of computing time. Nevertheless, it remains that `pamg-multiphase` is an intrinsically fast multiphase solver, although it would greatly benefit from optimisation in the following three areas:

- The replacement of the automatic differentiation package by purpose-written FORTRAN routines for the computation of the local Jacobians. `pamg-multiphase` run times were analysed using profiling tools and it was observed that depending on the differentiation method (see [108]), automatic differentiation accounts for 50% to 80% of computing time. The backward method is more efficient in this application.
- At present Jacobians are computed anew for every Newton correction and for every cell. As in `pamg`, significant performance gains may be possible if *frozen* Jacobians were used, assuming that the non-linearity of the equation does not prevent this approximation.
- At present, the linear system given by equation (4.33) is solved by an LU factorisation with partial pivoting and row interchange. A less accurate but less costly direct solver may be equally applicable since the solution process is essentially iterative and equation (4.33) is never solved exactly. Furthermore, the Jacobians are *very* sparse since the coupling between the phases only affects a limited number of entries. The cost of the linear algebra is, in the current version, negligible compared with the automatic differentiation and accounts for less than 1% of execution time. However, as the code is progressively optimised, its weight will increase, unless more specific and cost effective methods are used to solve the Newton correction system (4.33).

4.4.4 Computer Language Issues

While `pamg` is written in FORTRAN77, `pamg-multiphase` is a mixed language code containing both FORTRAN77 and `fortran90`.³ `fortran90` routines are required to make use of the automatic differentiation package AD01 which relies on operator overloading.

³Strictly speaking, `pamg-multiphase` can be considered to be written in `fortran90` only since FORTRAN77 is a subset of the `fortran90` standard.

The choice of language is a trade-off between correctness and ease of compilation since a powerful language permits a high level description of algorithms and a simple language can be easily and efficiently compiled.

Codes like `pang-multiphase`, which are adaptive, multigrid and parallel, have relatively complicated data structures, as well as numerically intensive parts. The handling of the data structure would greatly benefit in flexibility and simplicity from new features of `fortran90` such as pointers, array operators and modules, while the numerical parts are still best implemented using `FORTRAN77` and its proven numerical strength.

4.5 Conclusions

In this chapter, the solution algorithms implemented in both the single-phase and multiphase solvers were described in detail for all their basic components:

- the coupled quasi-Newton solver and the SCGS iterative procedure which define the relaxation procedure;
- the multigrid transfers;
- the grid refinement algorithm.

The basic solvers are very similar but when the approach is applied to multiphase flows, a number of algorithmic issues arise. These are described in Section 4.3, together with the original solutions which have been designed to overcome them. Firstly, the strong degree of non-linearity of the multi-fluid equations has required fine tuning in the Newton method. Next, the presence of the volume fraction has had implications for many parts of the solution algorithm which had to be examined in detail. In particular, the choice of certain standard interpolation schemes for the volume fractions renders the discretised equation inconsistent with the continuous problem.

Finally, the methodology adopted for software development — which is an important issue due to the complexity of the solution algorithm — was also discussed. The main goal was to re-use as many of the software components of the single phase solver as possible, in order to reduce development time and produce a working solution algorithm which can be validated and then relatively easily optimised.

Chapter 5

pamg-multiphase Computations on Uniform Grids: Validation and Performance

5.1 Introduction

So far, the governing equations of the multi-fluid model for multiphase flows have been discretised and a quasi-Newton coupled multigrid solver has been implemented in order to solve the resulting algebraic equations. The aim of this chapter is two-fold: (i) establish the correctness of the implementation and (ii) discuss the performance of the solver. For the moment, we will focus on uniform grids. Adaptation is considered in more detail in Chapter 6. This introduces further complications. However, concomitant gains in error control and efficiency are very significant.

The correctness of the results is established by comparing `pamg-multiphase` with other codes rather than experiments. The first reference code is, quite naturally, the single phase solver `pamg` which has been thoroughly tested [13, 1, 85]. The second choice is a commercial CFD code, `CFX 4.1` [67], which has also been extensively validated and is widely used. `CFX 4.1` solves the same multi-fluid equations as `pamg-multiphase`, using the IPSA solution algorithm (see [69] and Section 2.3.3), and a different discretisation.

The main advantage in using this approach is that the validation of the solution algorithm is greatly facilitated since both inaccuracies in the modelling of the physics and experimental errors can be eliminated. Hence, discrepancies of a numerical and computational origin can be isolated much more easily. In multiphase flows, where the issue of correct models is certainly still an open question, this is very beneficial. A comparison with another code is not a full validation but constitutes an important first step. If positive, it will greatly increase the degree of confidence that one can

have in the software. Comparison with experiments would then validate the physical model used as well as its implementation in `pamg-multiphase`.

First, the accuracy of `pamg-multiphase` for single phase flows is established using three validation cases: (i) a simple channel flow for which analytical solutions are known, (ii) a T-junction with two inlets and (iii) a flow over a backward-facing step with a large recirculation zone. The single phase cases were chosen to mirror the multiphase test cases. They are also representative of the main difficulties faced when solving single phase incompressible flows. The solutions given by the three codes are successfully confronted. Furthermore, in order to fully validate the `pamg-multiphase` implementation, adaptive solutions — for single phase flows only — are compared with those of `pamg`.

Next, we focus on the accuracy of multiphase computations. Three multiphase flow cases, of varying complexity, have been chosen:

- Firstly, a co-current two-phase flow through a channel. This is a simple but non-trivial problem due to the separation of the fluids which takes place;
- Secondly, a two-phase flow through a T-junction with two inlets. This was chosen as a representative example of complex multiphase flow, characterised by strong separation and mixing. The level of difficulty, however, is not as high as if a two-outlet junction had been considered, because the extent of recirculation zones is minimised (but not eliminated);
- Thirdly, a two-phase flow past a backward-facing step. This is another non-trivial case characterised by the presence of a recirculation zone which raises issues of mathematical well-posedness.

The main features of the solutions are discussed and justified. Solutions are again compared with those of `CFX 4.1`. Furthermore, their degree of grid-independence — i.e. the fact that the solution is not significantly dependent on the size of the discretisation grid and can therefore be considered to be an accurate representation of the continuous solution — is investigated and relatively surprising conclusions are drawn from the results. That is, the volume fractions fields are quite sensitive to the grid size. This does not appear to be specific to `pamg-multiphase` but seems to be a characteristic of the equations since the solutions provided by `CFX 4.1` also display this phenomenon.

Having established the accuracy of the `pamg-multiphase` solver, we discuss its performance. Our main measure is the observed convergence factors but we also address the robustness of the solution algorithm. Both the single phase and the multiphase cases are considered. For the former, significant differences between `pamg` and `pamg-multiphase` convergence factors are observed and explained in terms of (i) differences in the quasi-Newton solver and (ii) features of the multi-fluid equations.

For the latter, we first note that multigrid leads to a robust solver for which convergence is not dependent on the quality of the initial guess. Furthermore, it is a *very* successful acceleration technique for multiphase flows. Results are particularly good for the more complicated case of the multiphase T-junction.

Ideally, the convergence rates should be grid independent. This is a property of multigrid methods for linear problems (see Section 2.5.5) which also holds true for some Navier-Stokes solvers such as `pamg`, although the equations become non-linear. For multiphase flows, it has been consistently difficult to obtain grid independent convergence rates with `pamg-multiphase`. Several hypotheses are advanced to explain this phenomenon and the corresponding experiments are reported. In the author's opinion, the root cause for relatively poor convergence rates is that solutions on different grids can be quite different from each other. Again, the lack of grid independent convergence rates stems from the complexity and high degree of non-linearity of the governing equations. Another factor is that the multi-fluid equations involve specific cross derivative terms which are also responsible for grid-dependent convergence rates.

Unless otherwise stated, all the results reported in the following chapters were obtained with zero-valued uniform initial distributions for the velocities and pressures. Initial volume fractions for each phase were set to 0.5. The boundary conditions are those specified in Section 4.3.4: entry flow conditions are applied at the inlet(s) while for multiphase computations, volume fractions are uniformly set to 0.5. In order to facilitate the description of the results, we introduce the concept of grid level. For multigrid computations these are defined in terms of the coarsest grid: a level k grid is a grid whose mesh size is 2^{k-1} times finer than that of the coarsest grid. A level k multigrid computation will often be compared with single grid computations on an equivalent grid. By extension, the same terminology will then be applied to the single grids.

The results presented in this chapter were obtained from converged solutions. A normal criterion for convergence is that the residual is less than the truncation error of the discretisation. See Section 5.3.4 for estimates of the truncations errors. Often computations are performed to a level far below the truncation errors. This allows a thorough discussion of the numerical performance of the solver but is unnecessary for the purpose of obtaining discrete approximations.

5.2 Single Phase Validation Problems

5.2.1 Problem 1: Pseudo-Two-Phase Channel Flow

Problem Definition In this problem, a two-phase channel flow is considered. However, the boundary conditions and the physical properties of the phases are

defined in such a way that the velocity fields are identical for each phase and the volume fractions are constant. In effect, the solution consists of two simple single phase channel flows for which analytical solutions are known. The qualificative *pseudo* reflects the fact that multiphase effects have almost been completely removed from this test case.

The physical properties which characterise the fluids are the viscosity and the density. We consider two identical generic fluids and choose:

Reynolds Number	Viscosity	Density
$Re_1 = 100$	$\mu_1 = 0.01$	$\rho_1 = 1.0$
$Re_2 = 100$	$\mu_2 = 0.01$	$\rho_2 = 1.0$

The computational domain (see Figure 5.1) is a simple rectangle with $0 < x < 3$, $0 < y < 1$. The coarsest grid defined contains 48 cells, and its resolution is $\Delta x_1 = \Delta y_1 = 0.25$. Figure 5.2 shows a grid at level 3. Figure 5.3 specifies the coordinate system and the main sections along which solution profiles are taken.

Boundary conditions are the same for each phase. At the inlet, Dirichlet boundary conditions are specified: (i) a parabolic velocity profile is specified $u_1(0, y) = u_2 = 4y(1 - y)$, $v(0, y) = 0$ and (ii) the volume fractions are set constant: $r_1(0, y) = c$, $0 \leq c \leq 1$ and $r_2(0, y) = 1 - c$. At the outlet, Neumann conditions are applied (see Section 4.3.4).

In this simple case, an analytical solution is available:

$$\begin{aligned}
 u_1(x, y) &= u_2(x, y) = 4y(1 - y) \\
 v_1(x, y) &= v_2(x, y) = 0 \\
 p_1(x, y) &= p_2(x, y) = p_0 - 8\mu x \\
 r_1(x, y) &= c \\
 r_2(x, y) &= 1 - c
 \end{aligned}$$

The starting guess must now be specified. It is an important feature of both the `pamg` and `pamg-multiphase` solvers that convergence is not sensitive to the initial guess. This characteristic, which is not shared by most segregated solvers, makes coupled solvers particularly robust. As mentioned in Section 3.1.3, the robustness is a natural consequence of the implicit continuation method defined by the use of hybrid differencing on a sequence of coarser grids.

In the case of the pseudo-two-phase channel flow, multiphase effects can be completely removed if the initial guesses for the volume fraction fields are the constant c and $1 - c$ for phases 1 and 2 respectively. In this case, the Newton corrections for the volume fractions are always zero and the multiphase solver acts, in effect, as a duplicate single phase flow solver.

We choose the initial volume fraction fields which are different from those of the solution. Alternatively, the initial velocity fields can be taken to be different for each phase. The final solution is not changed (which is fortunate!) but the solution algorithm is then truly multiphase since non-zero volume fraction corrections are generated. Hence, the pseudo-two-phase pipe flow can be considered as a *full* validation of the multiphase sets of equations, admittedly for a simple case.

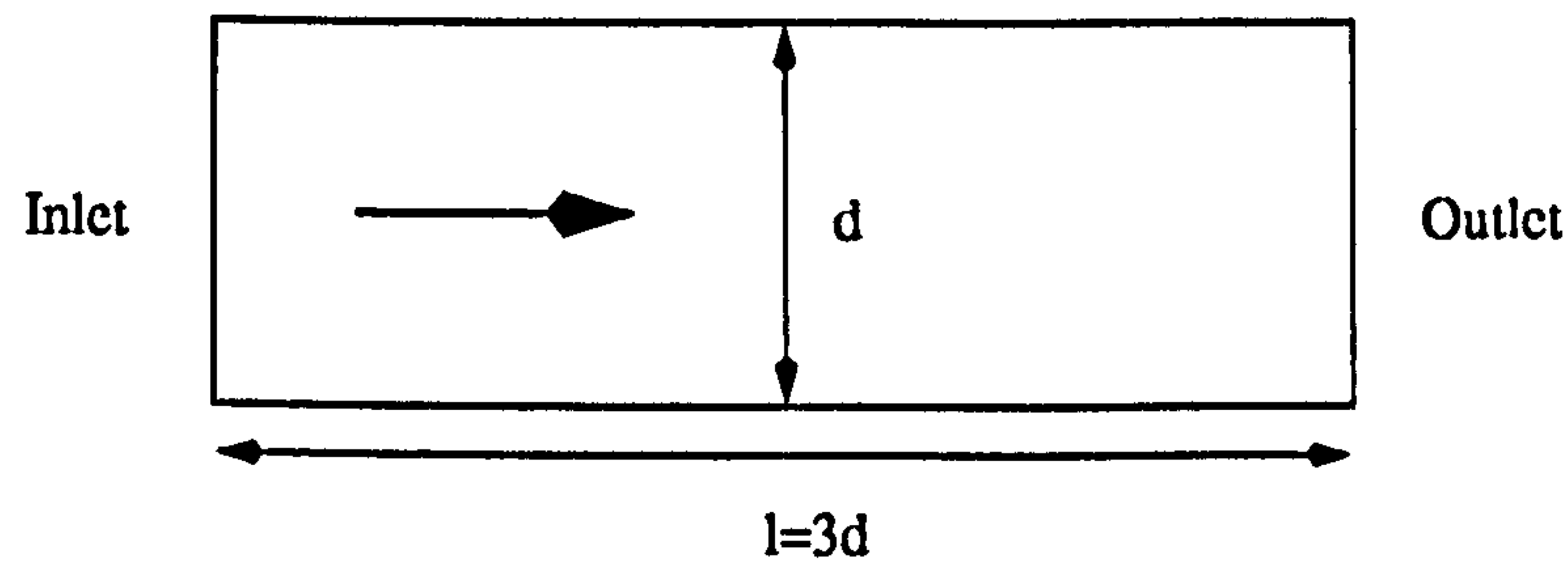


Figure 5.1: Geometrical representation of the channel flow problem

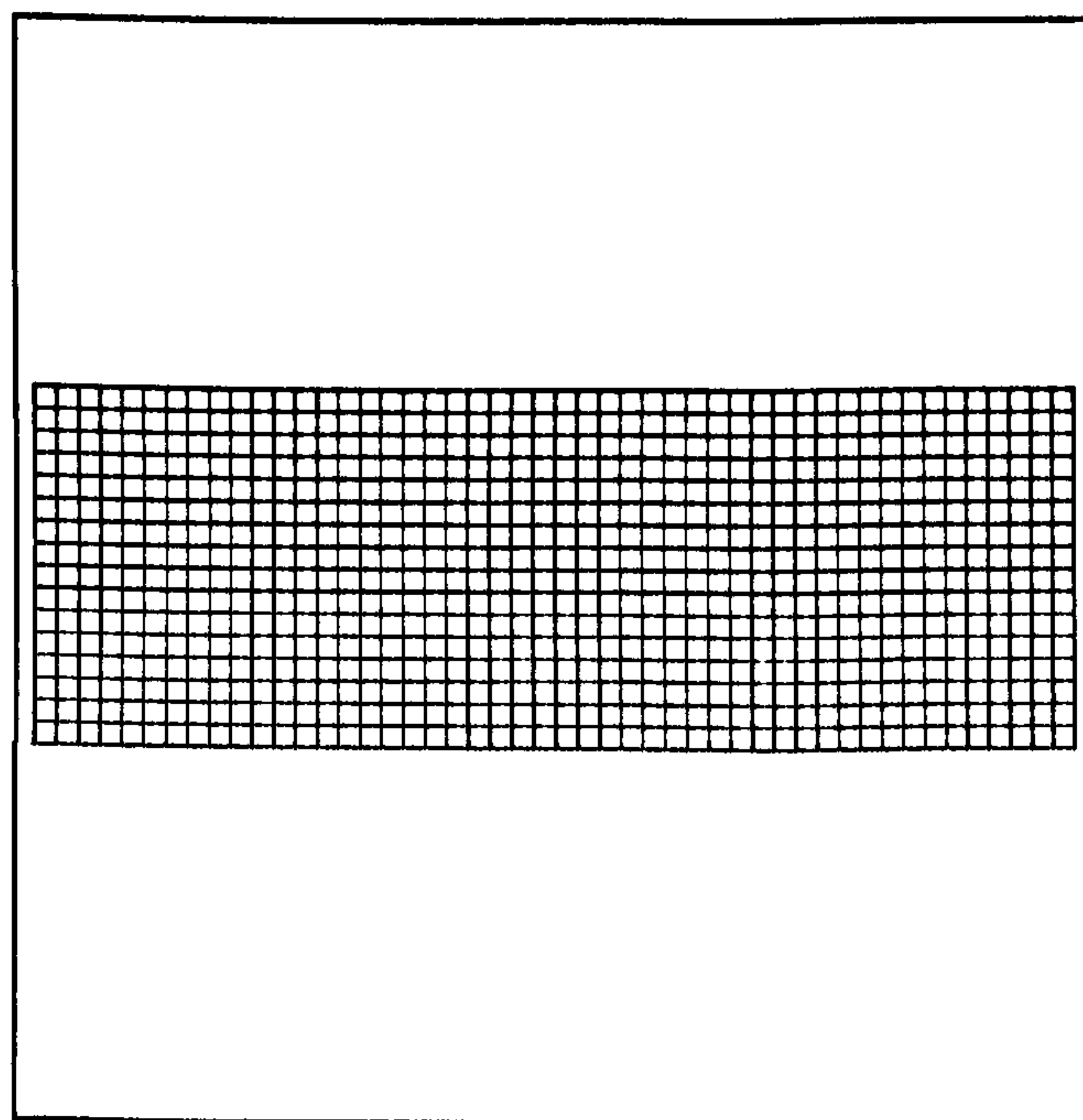


Figure 5.2: Computational grid for the channel flow problem – Uniform level 3 grid (768 cells, $\Delta x = \Delta y = 0.0625$)

Results and Conclusions Figure 5.4 shows some streamlines computed from the pang-multiphase solution. Streamlines for this simple flow are straight horizontal lines. This is not a surprising result since vertical velocities are identically zero everywhere for this geometry and the chosen set of boundary conditions.

The solution obtained is very close to that of the differential equations. In particular, the parabolic velocity profile across the pipe is observed (Figure 5.5) and the pressure drop across the pipe is correctly modelled (Figure 5.6). The computed pressure gradient is exactly equal to its theoretical value $8\mu = 0.08$.

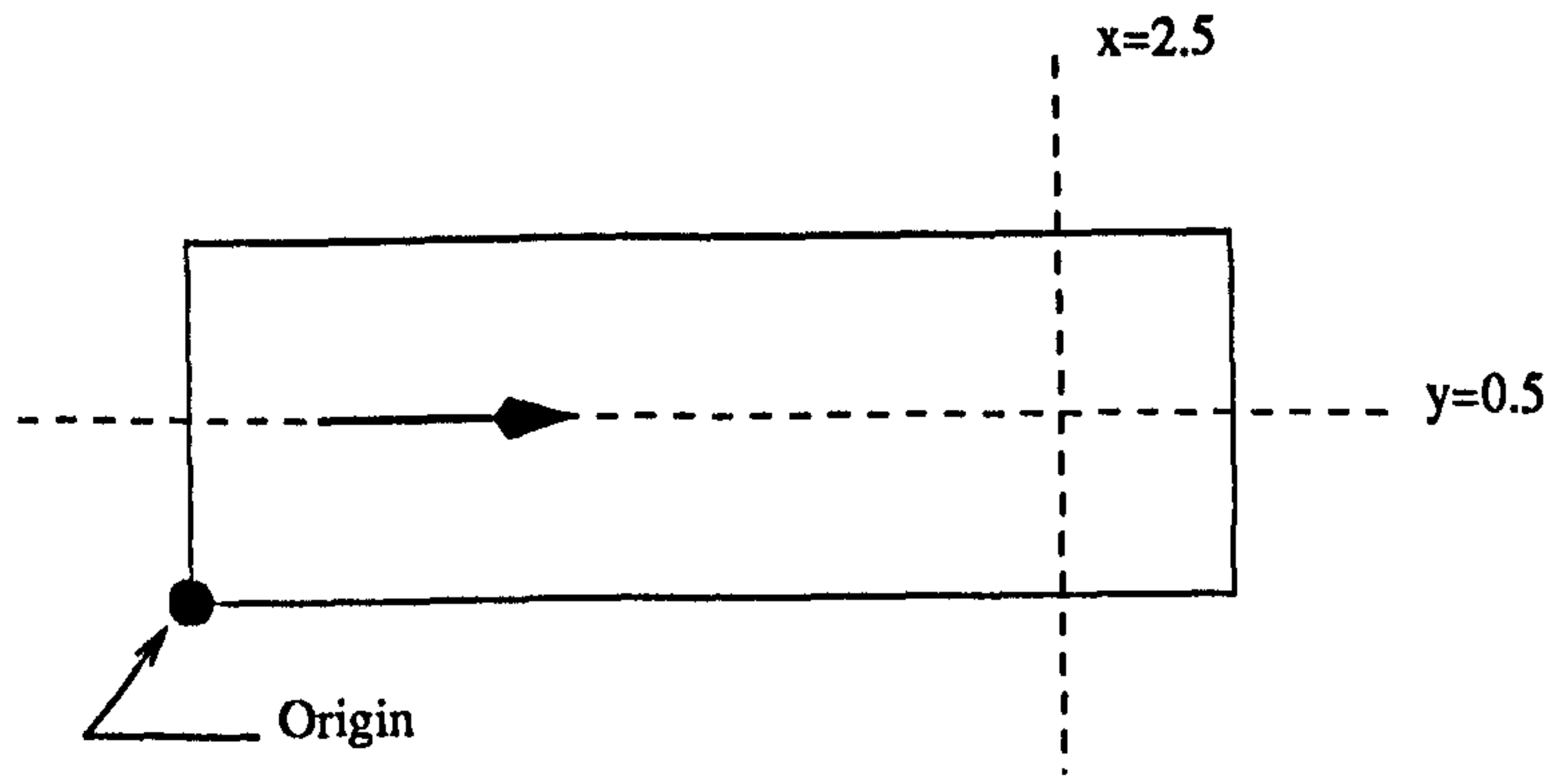


Figure 5.3: Channel flow problem – Coordinate system and main sections used to give the solution profiles

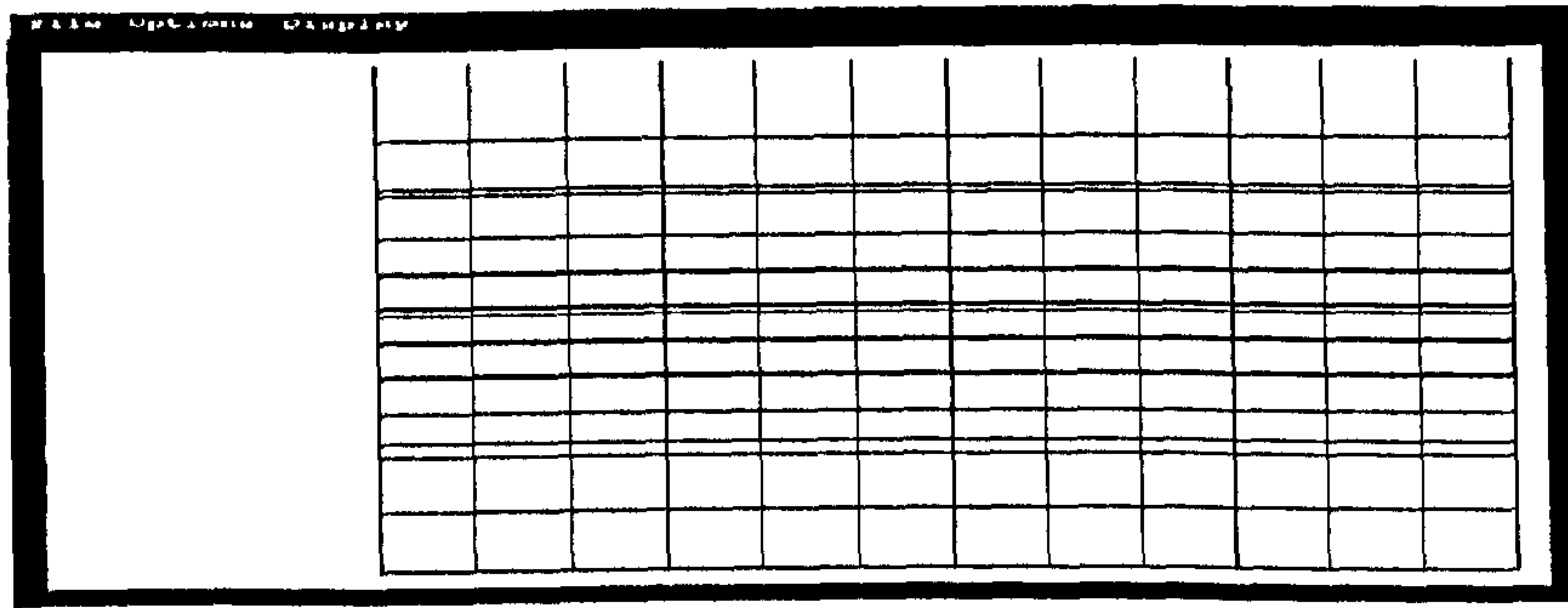


Figure 5.4: Pseudo multiphase channel flow – Streamlines for the pamg-multiphase solution

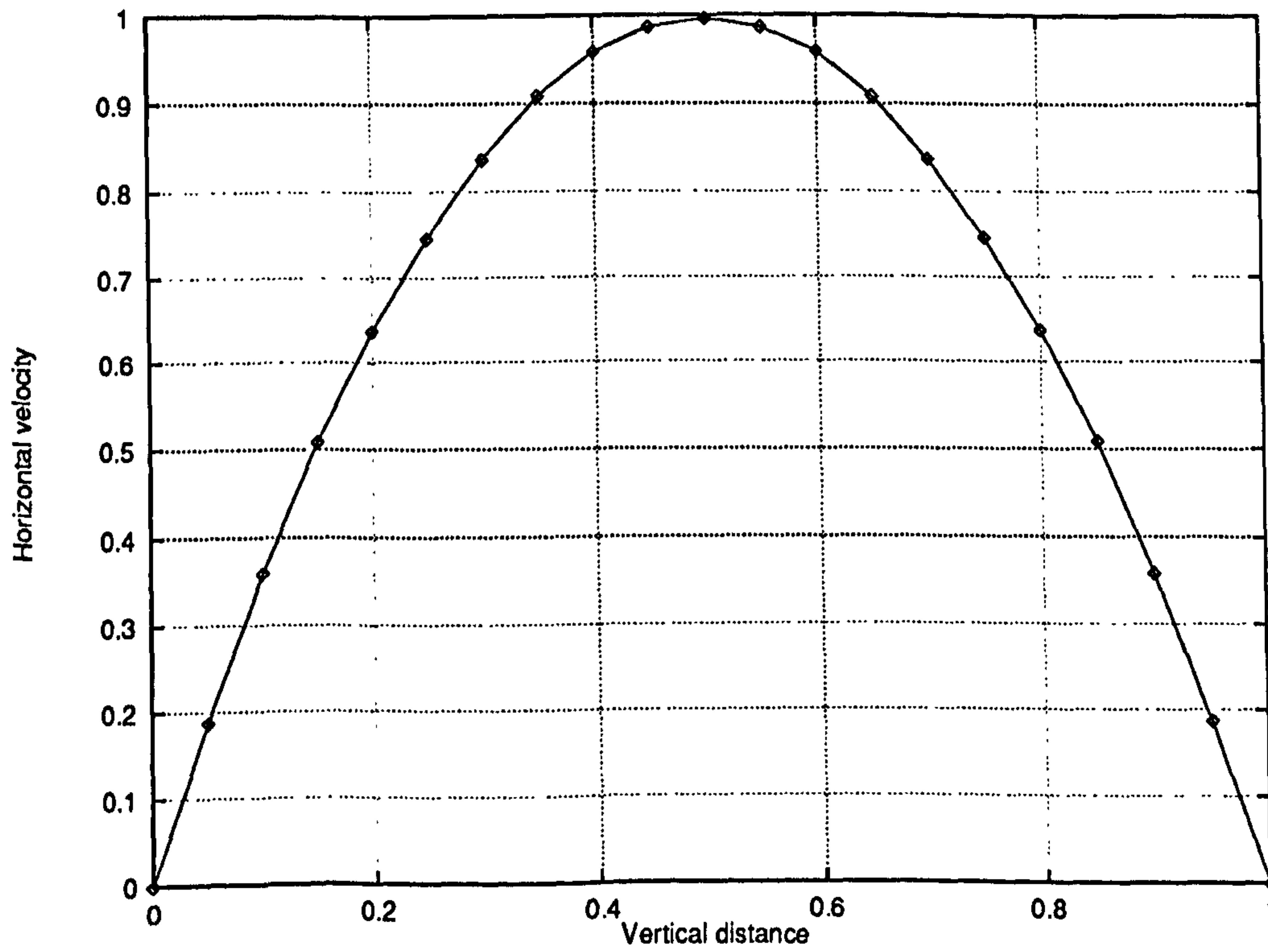


Figure 5.5: Pseudo multiphase channel flow – Velocity profile along the line $x = 2.5$

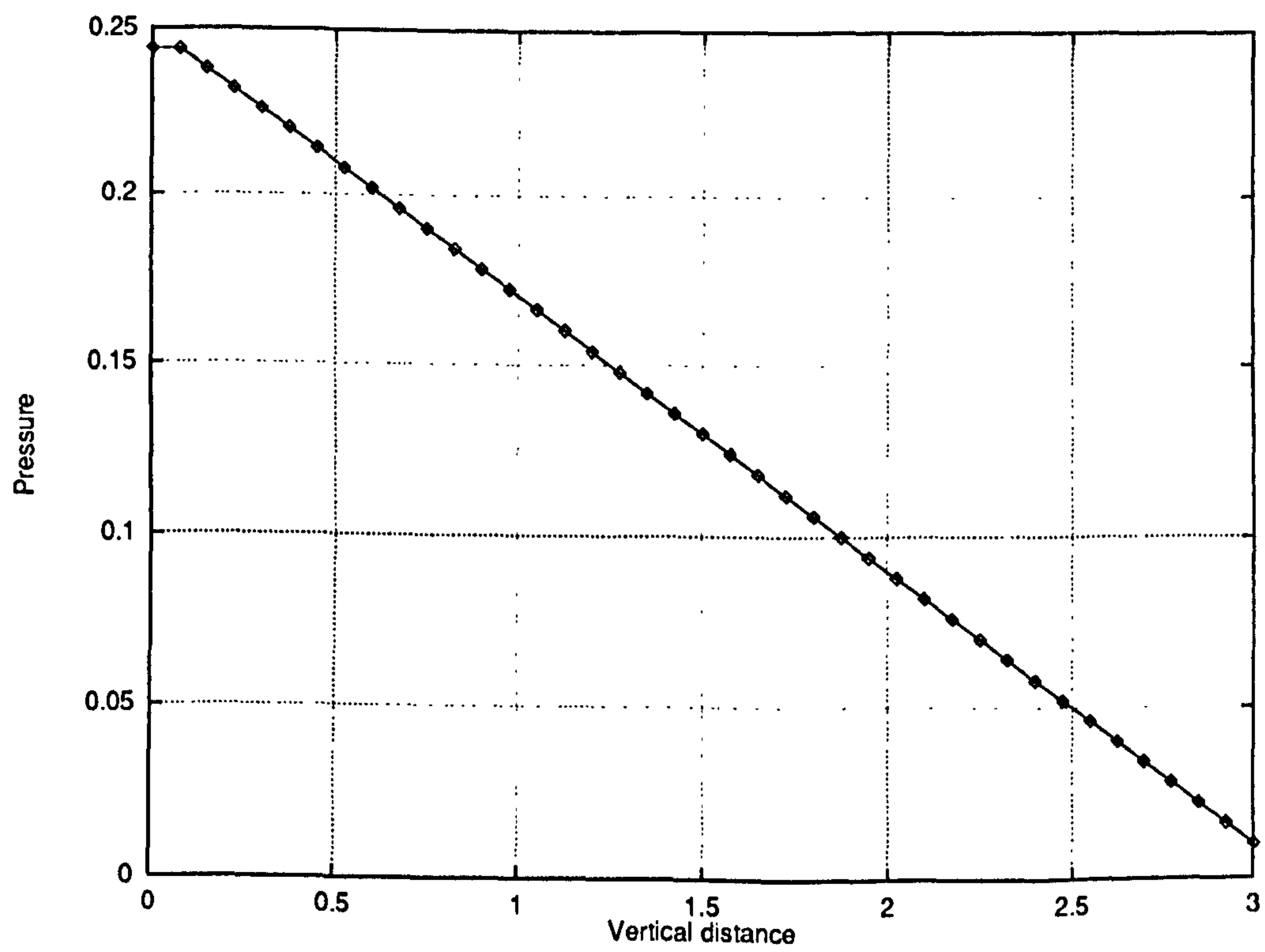


Figure 5.6: Pseudo multiphase channel flow – Pressure drop along the line $y = 0.5$

5.2.2 Problem 2: Single Phase T-Junction with 2 Inlets

Problem definition In this case, a true *single phase* flow through a T-junction with two inlets is considered. This aims to avoid recirculation zones and simplify the complexity of the two-phase computations (Section 5.3.2). Note that the two-phase computation still has difficult features, particularly mixing and separation. In the single phase case, we observe strong curvature in the streamlines (Figure 5.10) and, on sufficiently fine grids, there is even a small recirculation zone just downstream of the junction.

The geometry of the problem is shown in Figure 5.7, while Figure 5.8 specifies the coordinate system and the main sections along with solution profiles are taken.

The fluid has the following physical properties:

- $\mu = 0.01$ ($Re = 100$, based on the channel width),
- $\rho = 1.0$

Finally, entry flow conditions are imposed at both inlets using Dirichlet boundary conditions. This choice is made for consistency with the multiphase computations. At the outlet, Neumann conditions are imposed.

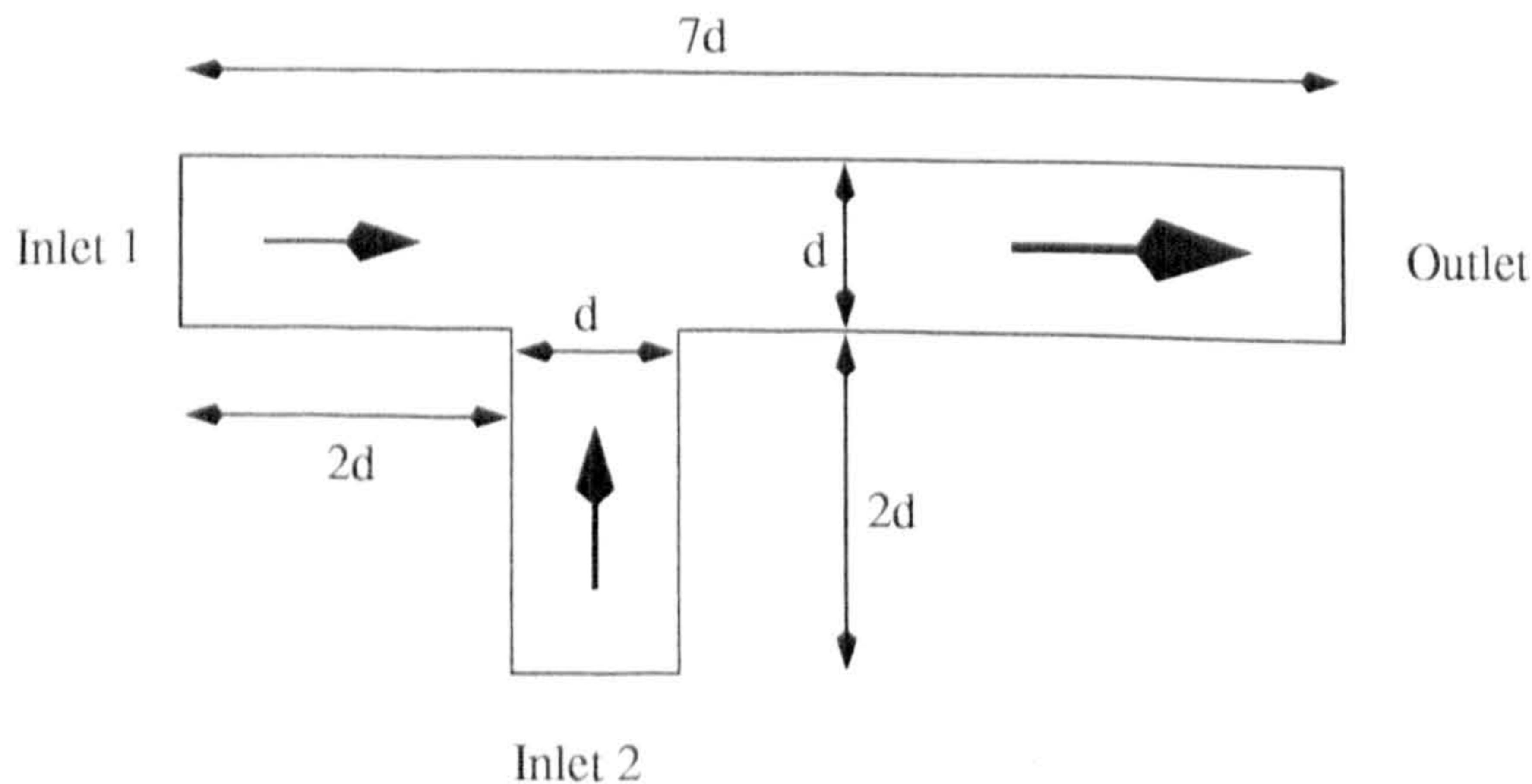


Figure 5.7: Geometrical representation of the T-junction problem

The coarsest grid contains 146 cells and its resolution is $\Delta x_1 = \Delta y_1 = 0.25$. Figure 5.9 shows the level 3 grid. Figure 5.16 shows that the length of the exit section is adequate. The flow is close to a fully developed state at the exit. Nevertheless, it may have been better to model a longer section of the exit section of the T-junction.

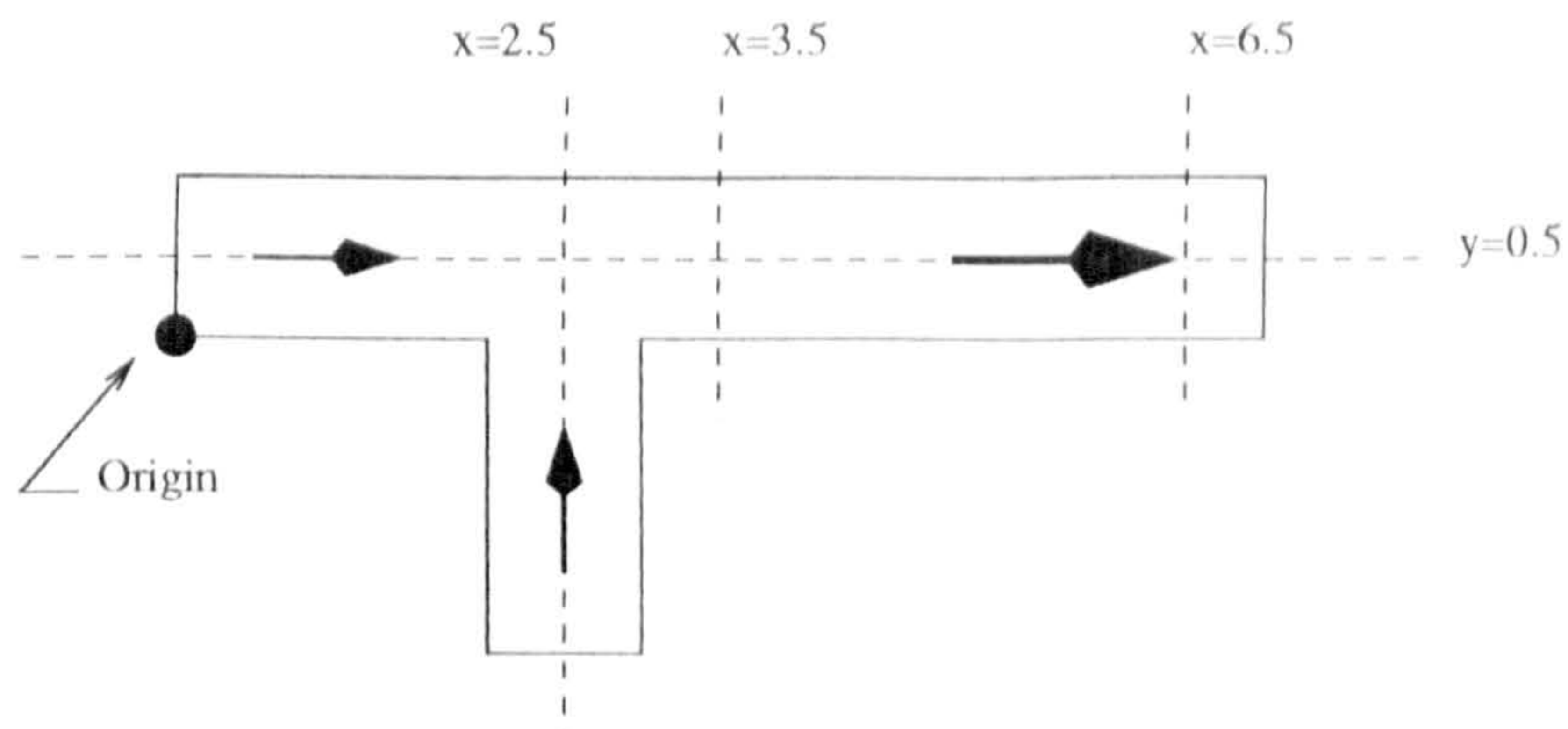


Figure 5.8: T-junction Problem – Coordinate system and main sections used to give the solution profiles

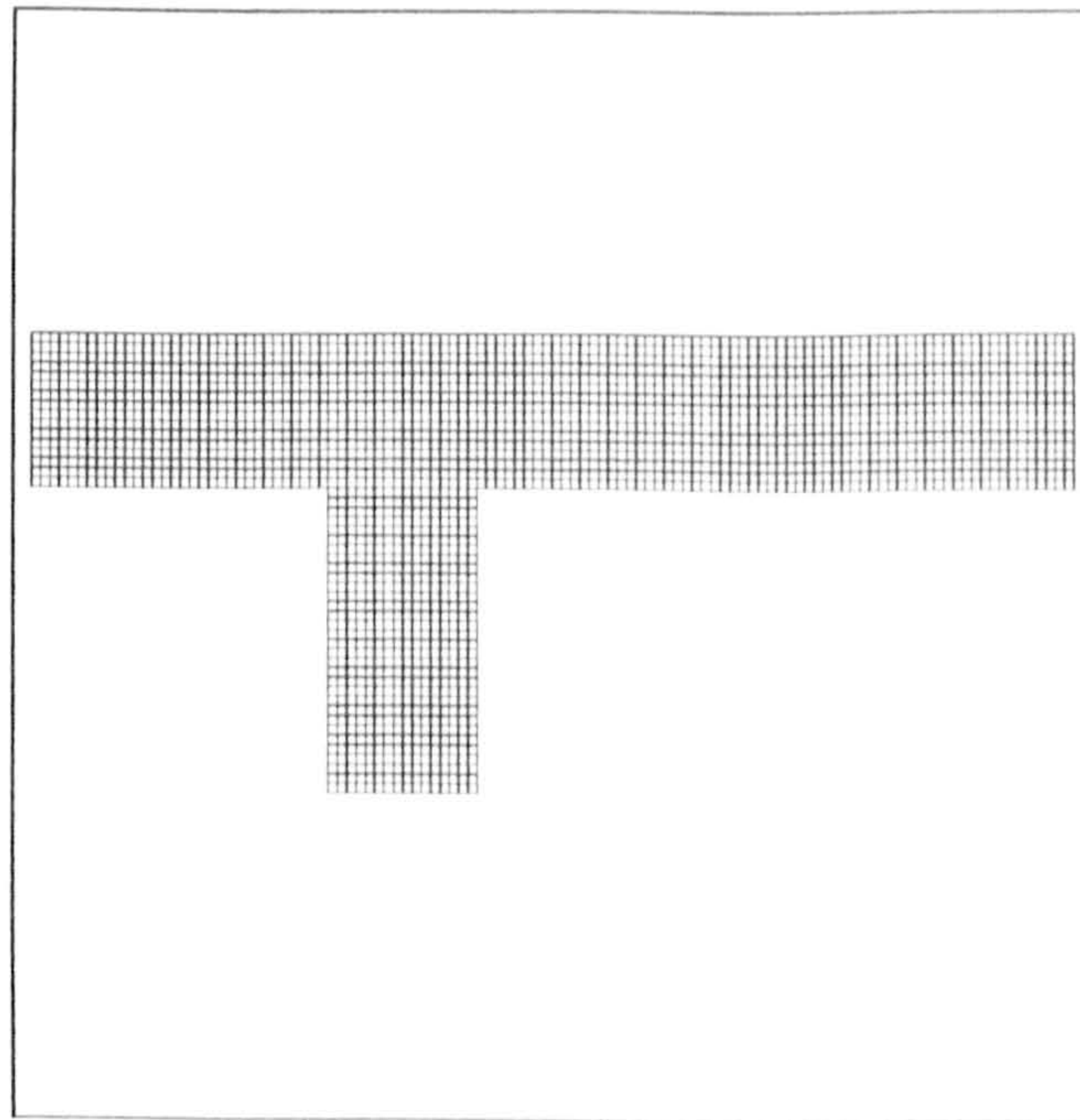


Figure 5.9: T-junction problem – Uniform grid at level 3 (2336 cells, $\Delta x = \Delta y = 0.0625$)

Results and Conclusions Figure 5.10 shows some streamlines computed from the `pamg-multiphase` solution and illustrates the mixing process which occurs in this type of flow. Note the small recirculation zone just downstream of the junction.

The solutions obtained with `pamg-multiphase` are consistent with both those of the original `pamg` code and the CFX 4.1 solutions (see Figures 5.11 and 5.12). It was also confirmed that `pamg-multiphase` behaves like the original `pamg` in the case of *single phase* adaptive computations (see Figures 5.13 to 5.16). In particular, the grids produced by refinement were identical for both versions (see Figure 6.60).

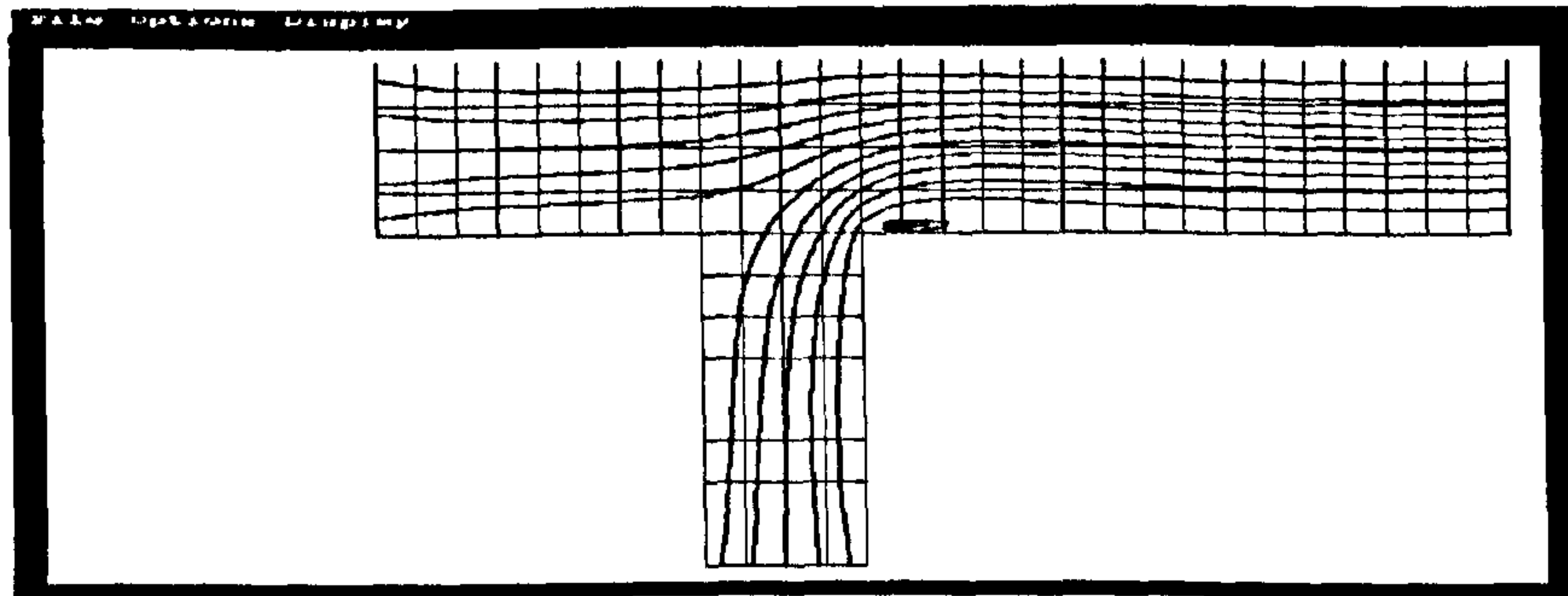


Figure 5.10: Single phase T-junction problem – Streamlines for the `pamg-multiphase` solution

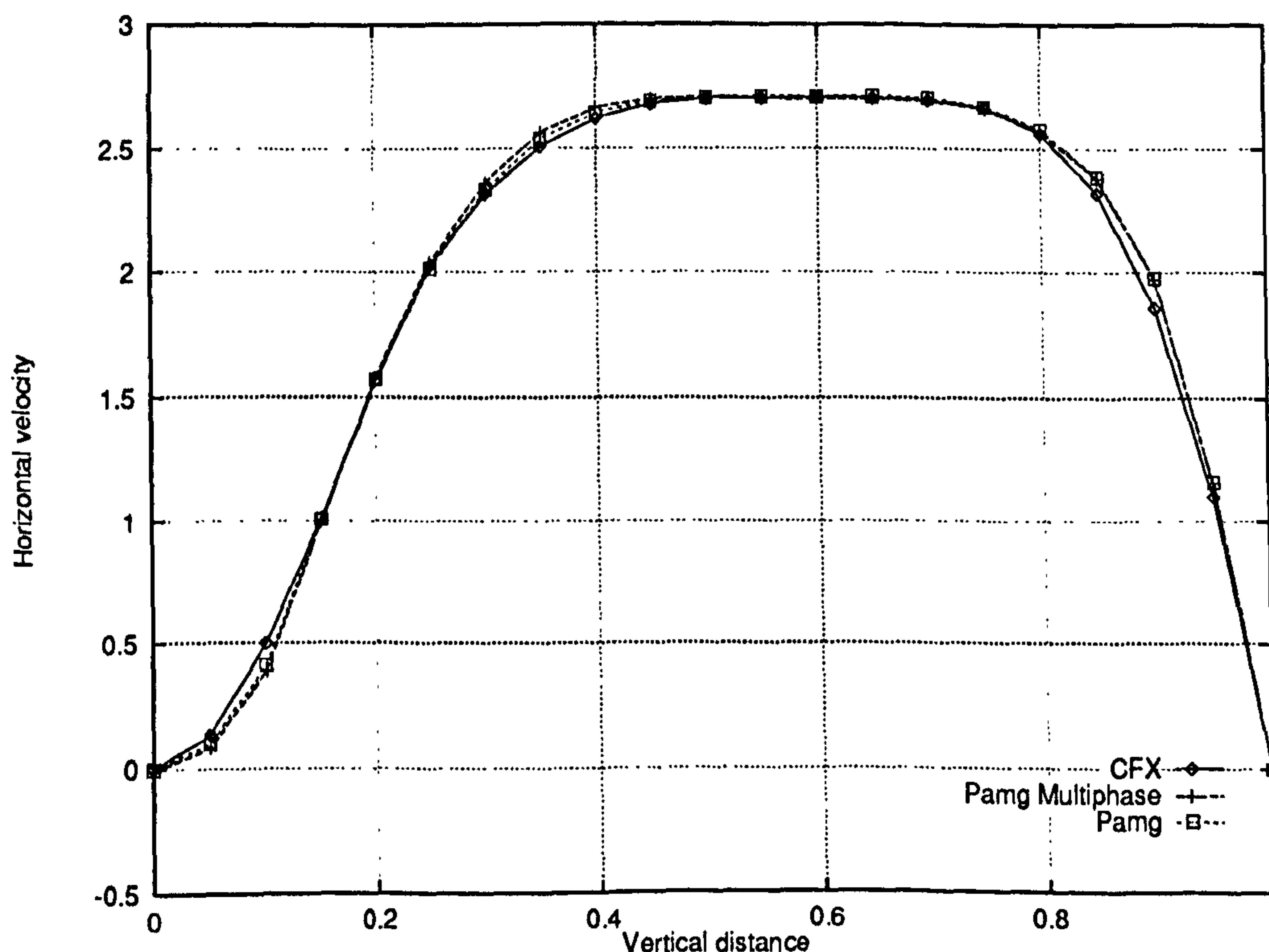


Figure 5.11: Single phase flow through a T-junction – Horizontal velocity profiles along the line $x = 3.5$ on level 3 uniform grids for different solvers

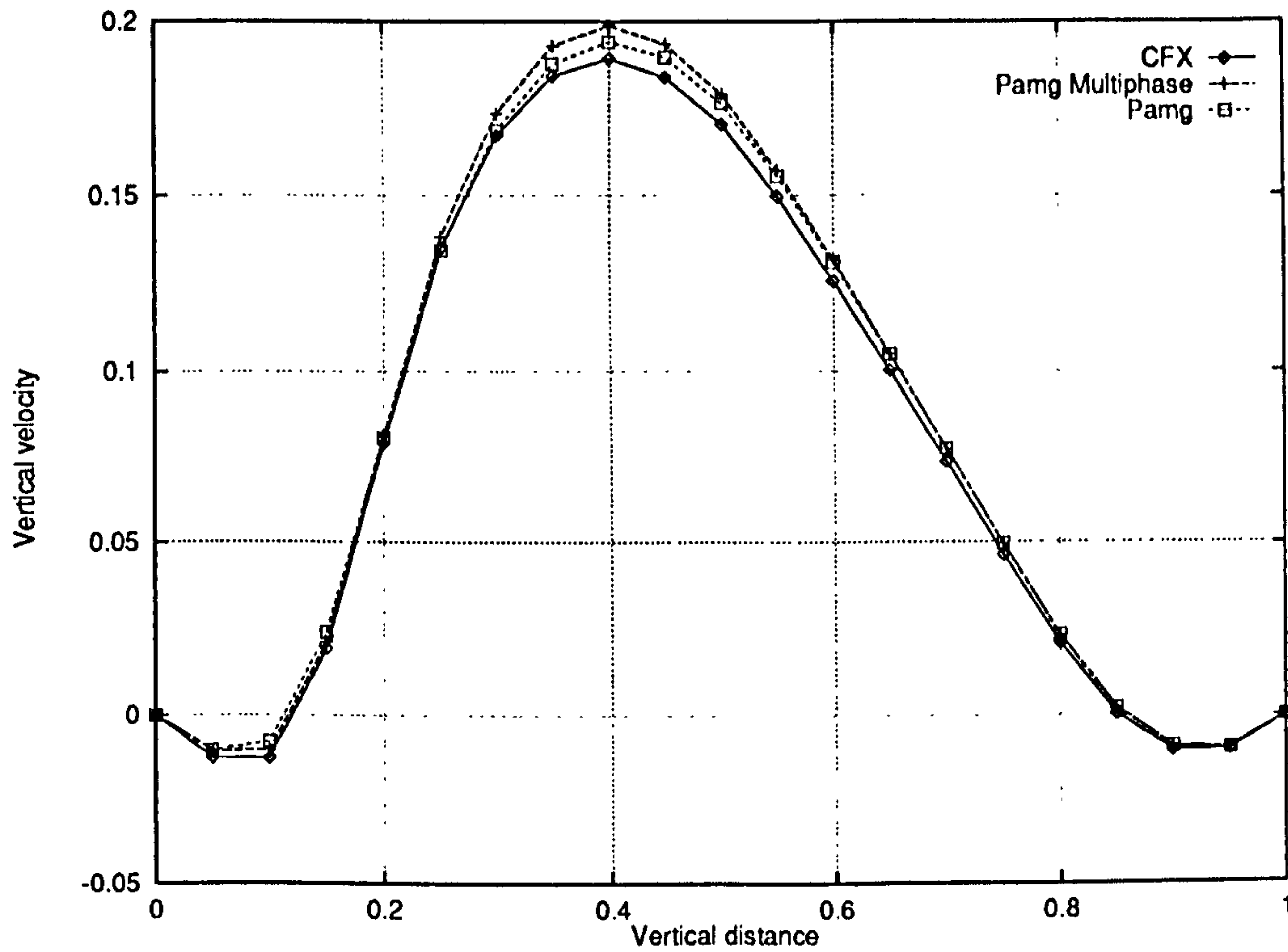


Figure 5.12: Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.5$ on uniform grids for different solvers

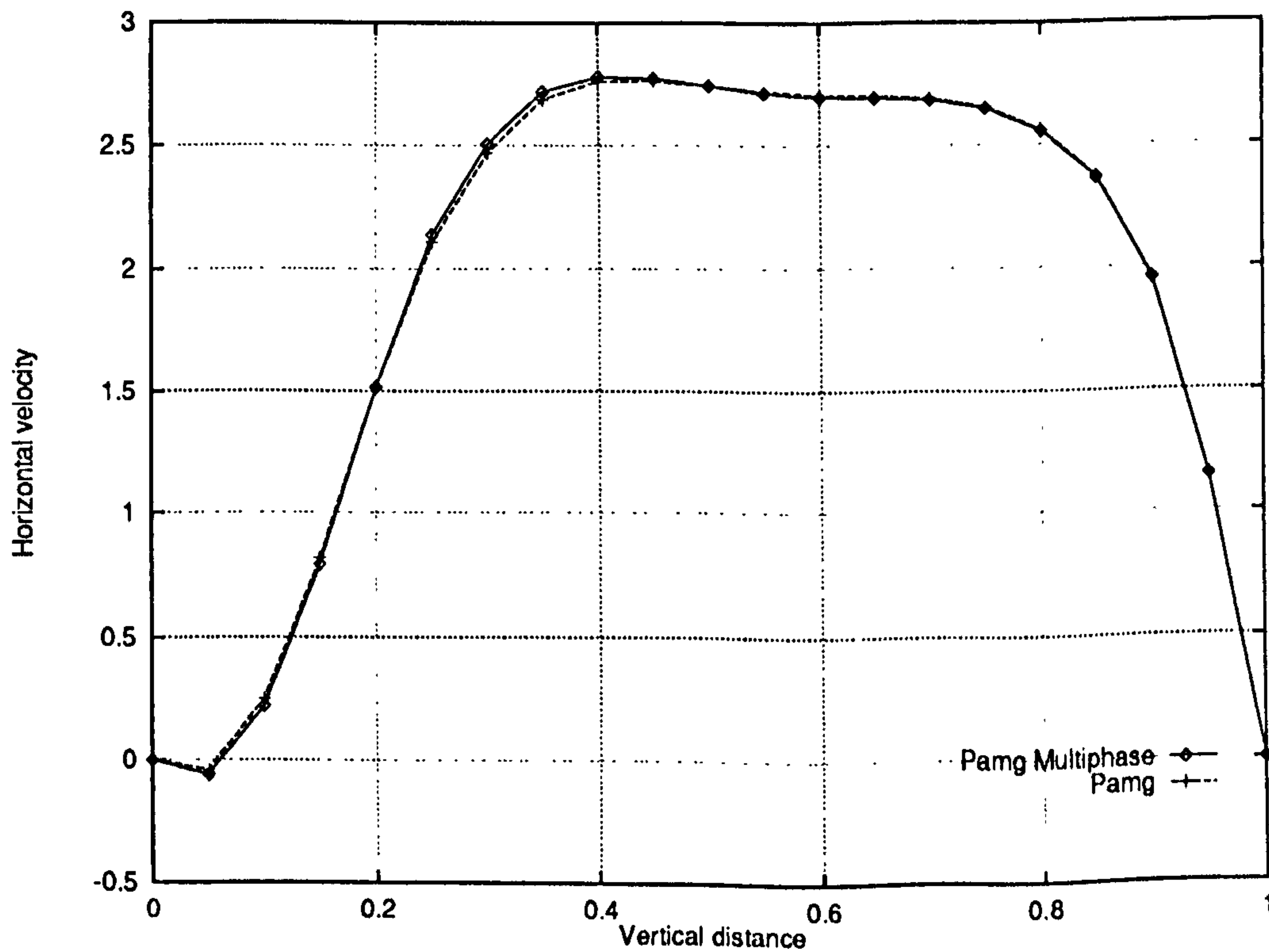


Figure 5.13: Single phase flow through a T-junction – Horizontal velocity profiles along the line $x = 3.5$ on adaptive (3,4) grids

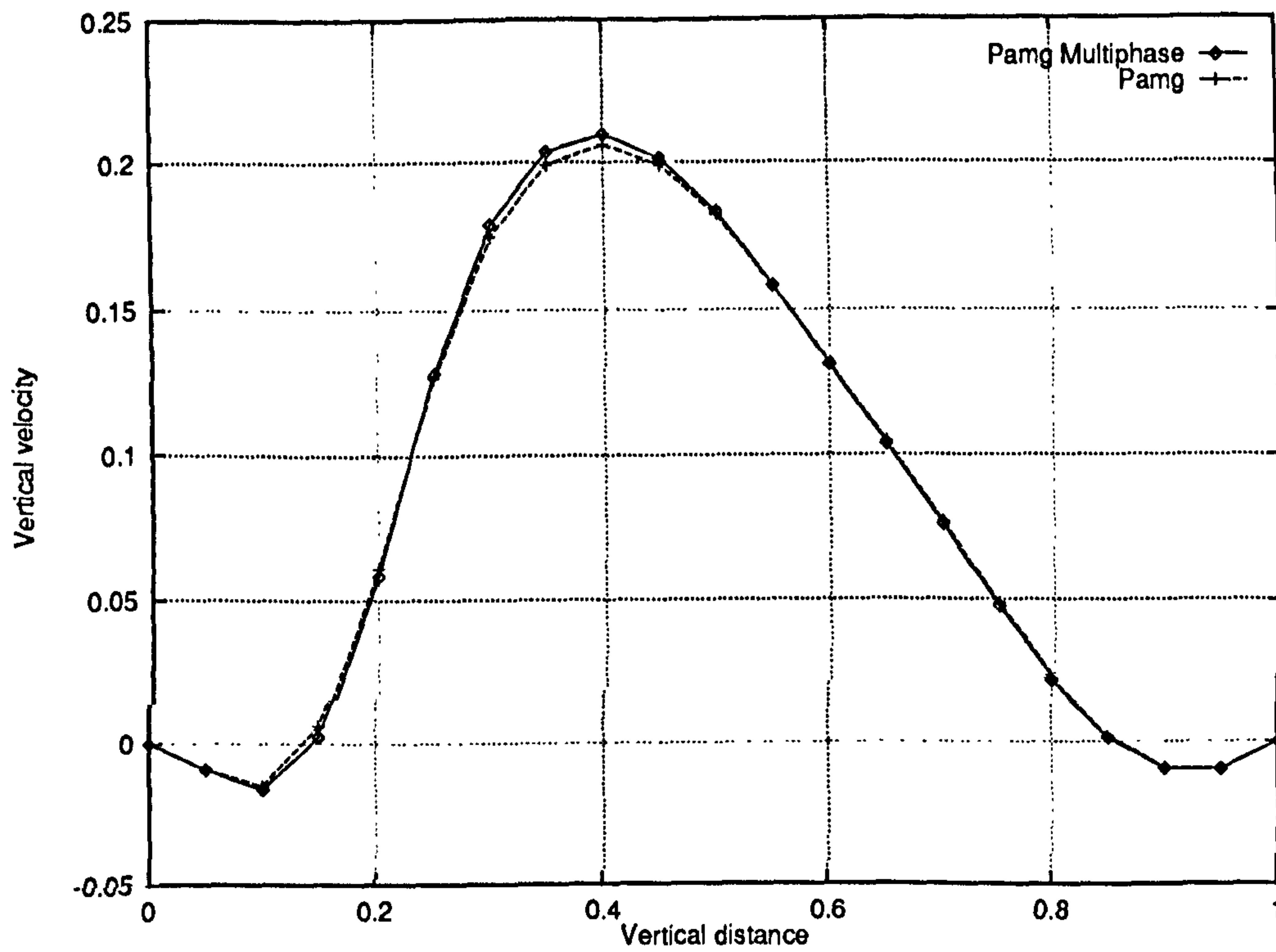


Figure 5.14: Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.5$ on adaptive (3,4) grids

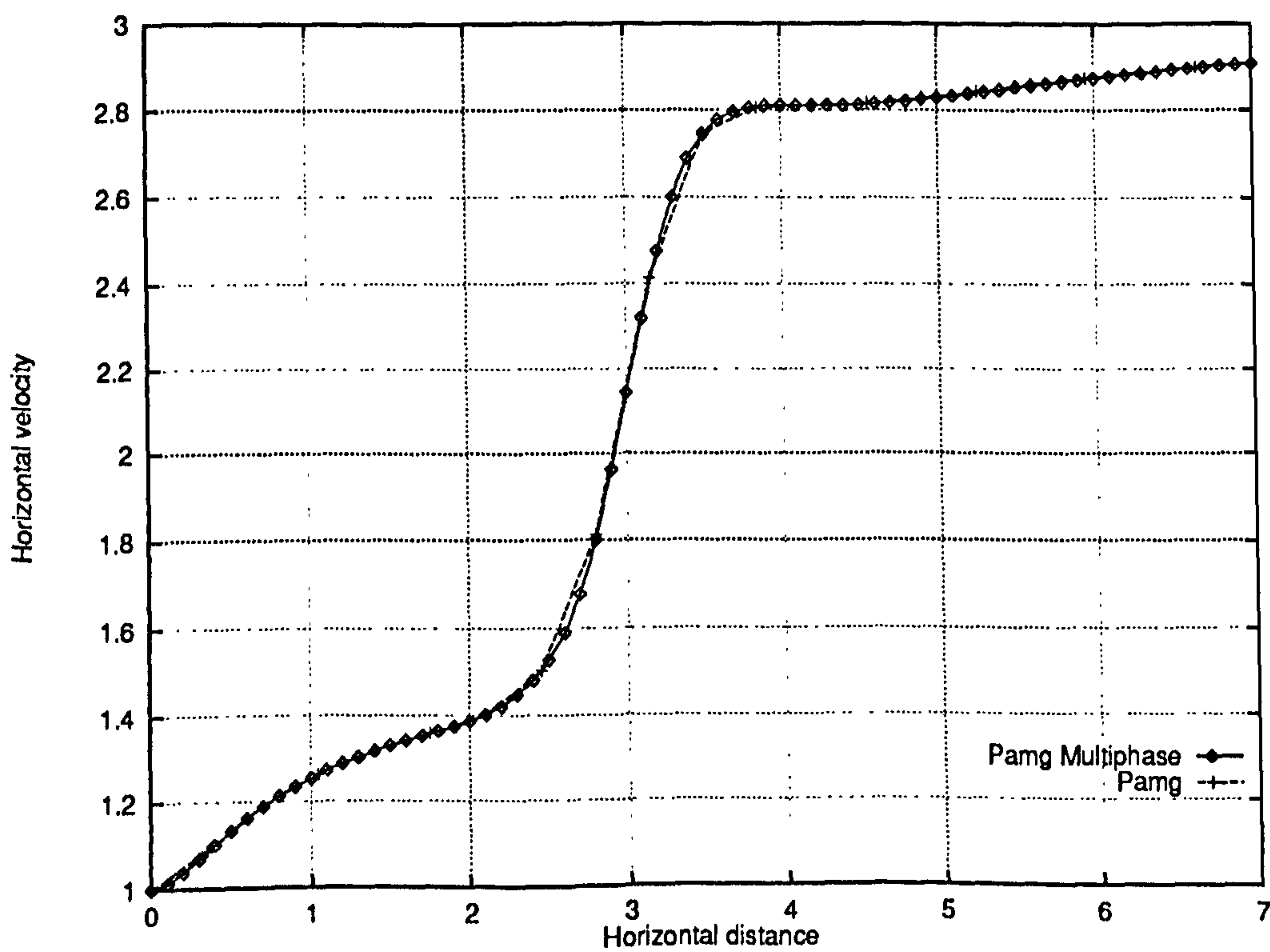


Figure 5.15: Single phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.5$ on adaptive (3,4) grids

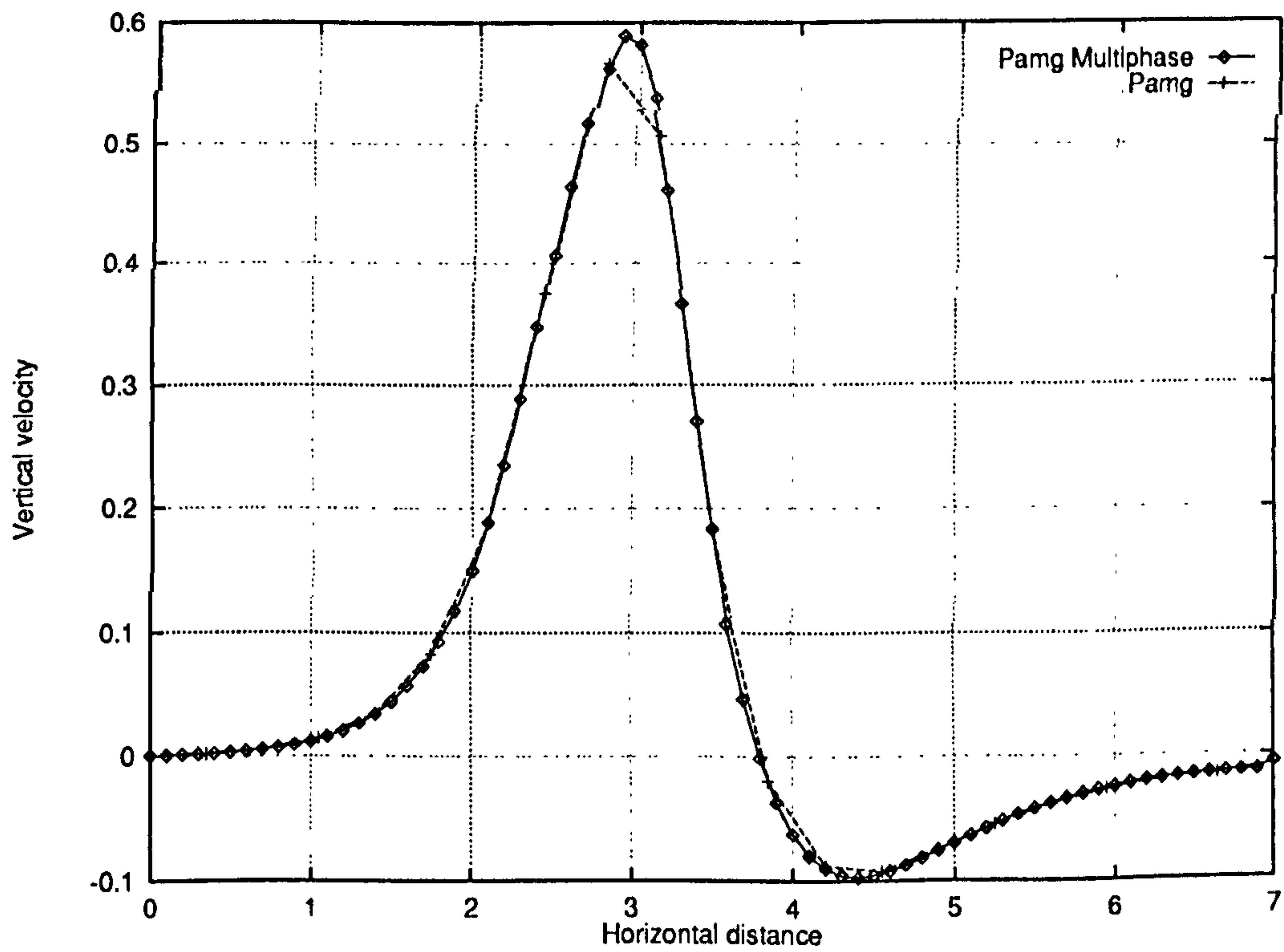


Figure 5.16: Single phase flow through a T-junction – Velocity profile along the line $y = 0.5$ on adaptive (3,4) grids – Note that the computational grids are identical even though the number of interpolated point differ.

5.2.3 Problem 3: Single Phase flow through a Backward-Facing Step

Problem Definition The third validation problem is a single phase flow through a backward-facing step. The main feature of the flow is the presence just downstream of the step of a recirculation zone whose length increases with the Reynolds number. A diagrammatic representation of the flow is provided by Figure 5.17. Figure 5.18 specifies the coordinate system and the main sections along with solution profiles are taken.

The physical properties of the fluids are: $\rho = 1$ and $\mu = 0.0033$ so that the Reynolds number, based on the height of the step is $Re = 150$. The geometry is as follows:

- Size of the domain: $0 < x < 30, 0 < y < 1$
- Position of the step: $x = 3.0$
- Height of the step: $h = 0.5$

The coarsest grid contains 576 cells and its resolution is $\Delta x = \Delta y = 0.125$. Figure 5.19 shows the computational grid at level 3. It is important that the computational domain is long compared with the height of the step. Otherwise the boundary conditions interfere with the recirculation zone.

At the inlet, a parabolic velocity profile is imposed $u(y) = 16y(0.5 - y)$, $v(0, y) = 0$ and $r(0, y) = 0.5$ for $0 \leq y \leq 0.5$ while Neumann boundary conditions for developed flows are applied at the outlet.

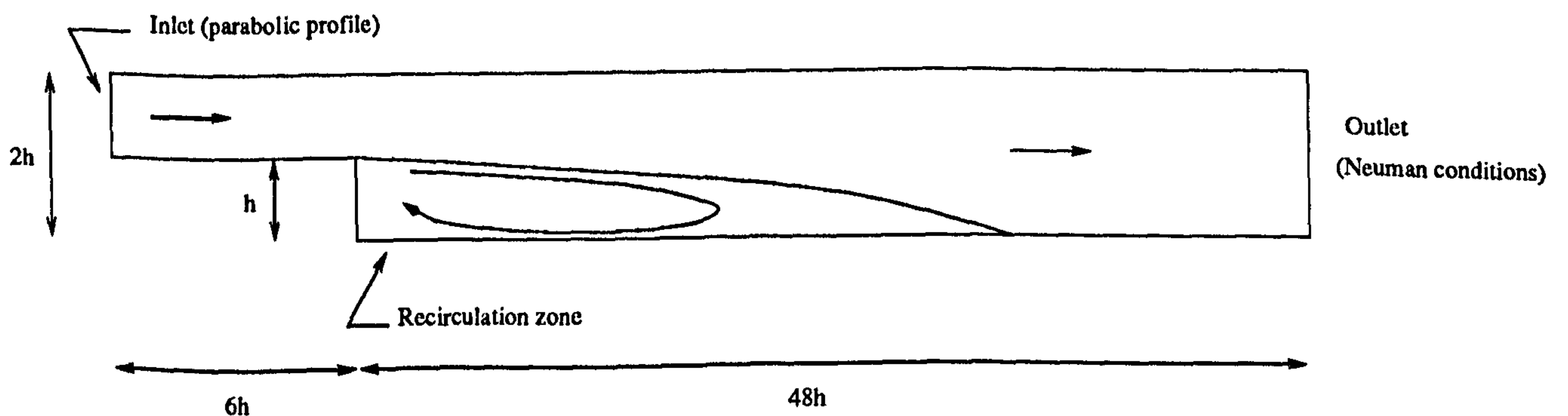


Figure 5.17: Diagrammatic description of the problem of a two-phase flow through a backward facing step

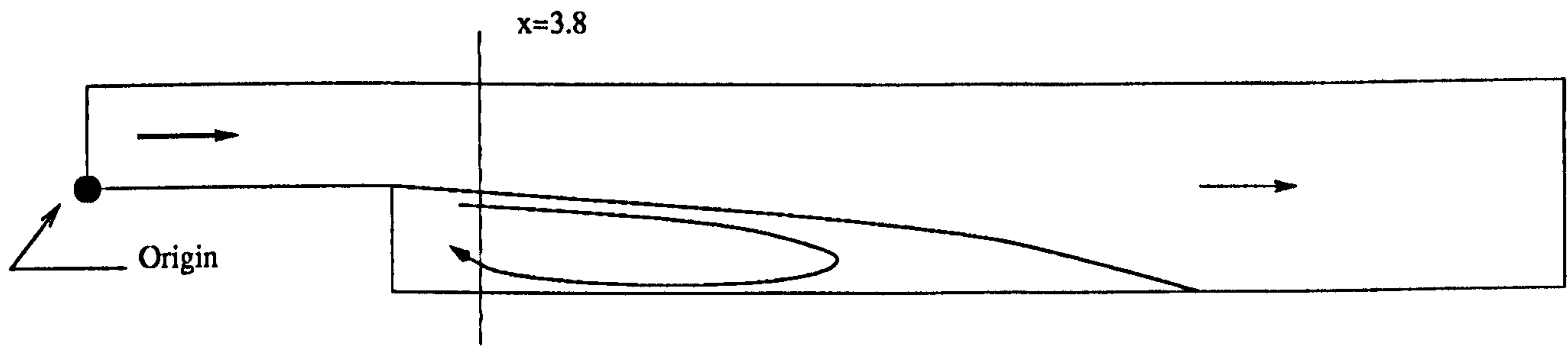


Figure 5.18: Single phase backward-facing step problem – Coordinate system and main sections used to give solution profiles

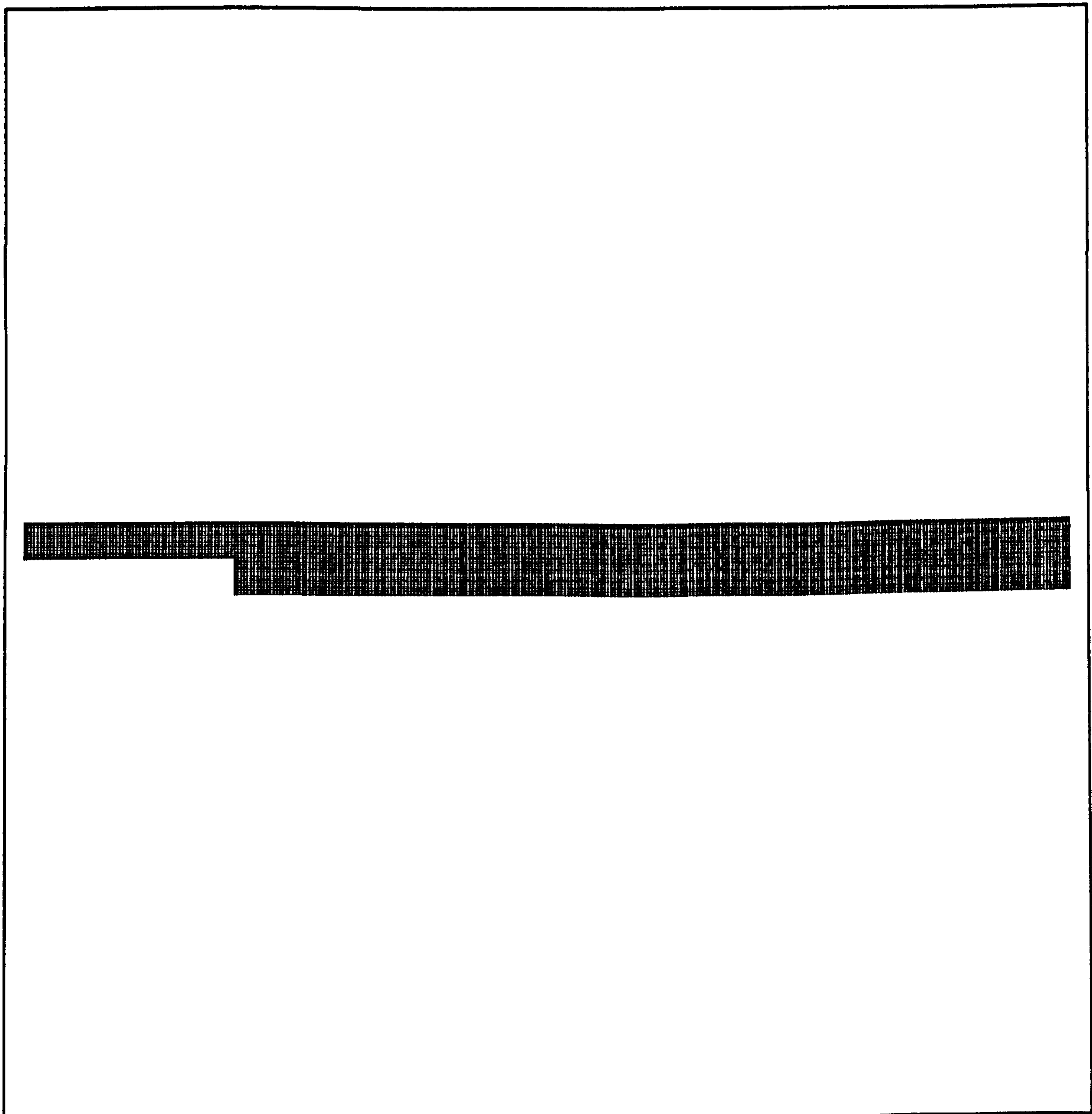


Figure 5.19: Single phase backward-facing problem – Uniform grid at level 3 (9216 cells, $\Delta x = 0.03125$, $\Delta y = 0.046875$)

Results and Conclusions Figure 5.20 shows some streamlines computed from the `pamg-multiphase` solution. The recirculation zone, which characterises this type of flow, appears clearly just downstream of the step. Note that the aspect ratio of the vertical and horizontal distances is not in scale on this figure.

Here again, we compare the original and multiphase versions of `pamg` for adaptive computations up to a level 4 grid (see Figure 6.63). As before, the results are in very good agreement (Figures 5.21 to 5.23). In particular, the velocity profiles taken along the line $x = 3.8$ (Figures 5.21 and 5.22) reveal a change of sign which confirms the existence of a recirculation zone downstream of the step. Furthermore, the solution on a level 3 adaptive grid were compared with the results provided by `CFX 4.1` for the uniform grid shown in Figure 5.19, which is equivalent to a `pamg` uniform grid at level 3. Here again, the measure of agreement is very good (Figures 5.24 and 5.25)

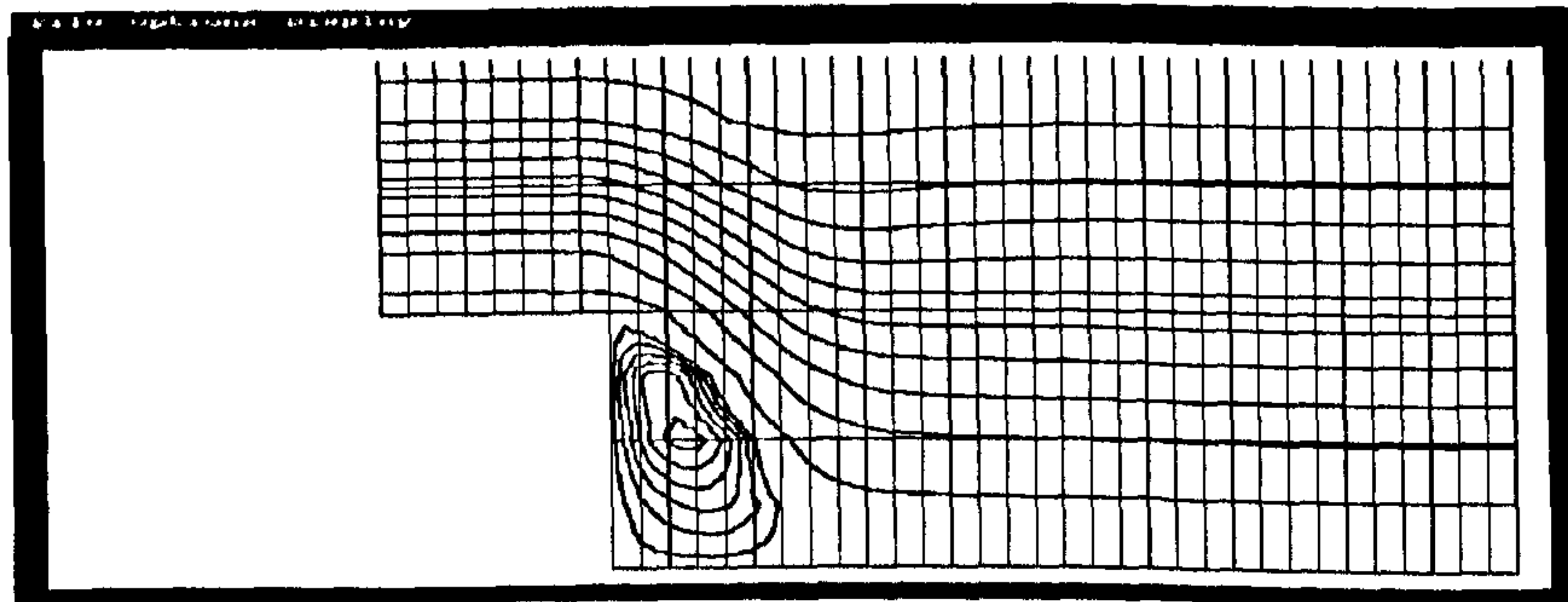


Figure 5.20: Single phase T-junction problem – Streamlines for the `pamg-multiphase` solution (shortened in the horizontal direction)

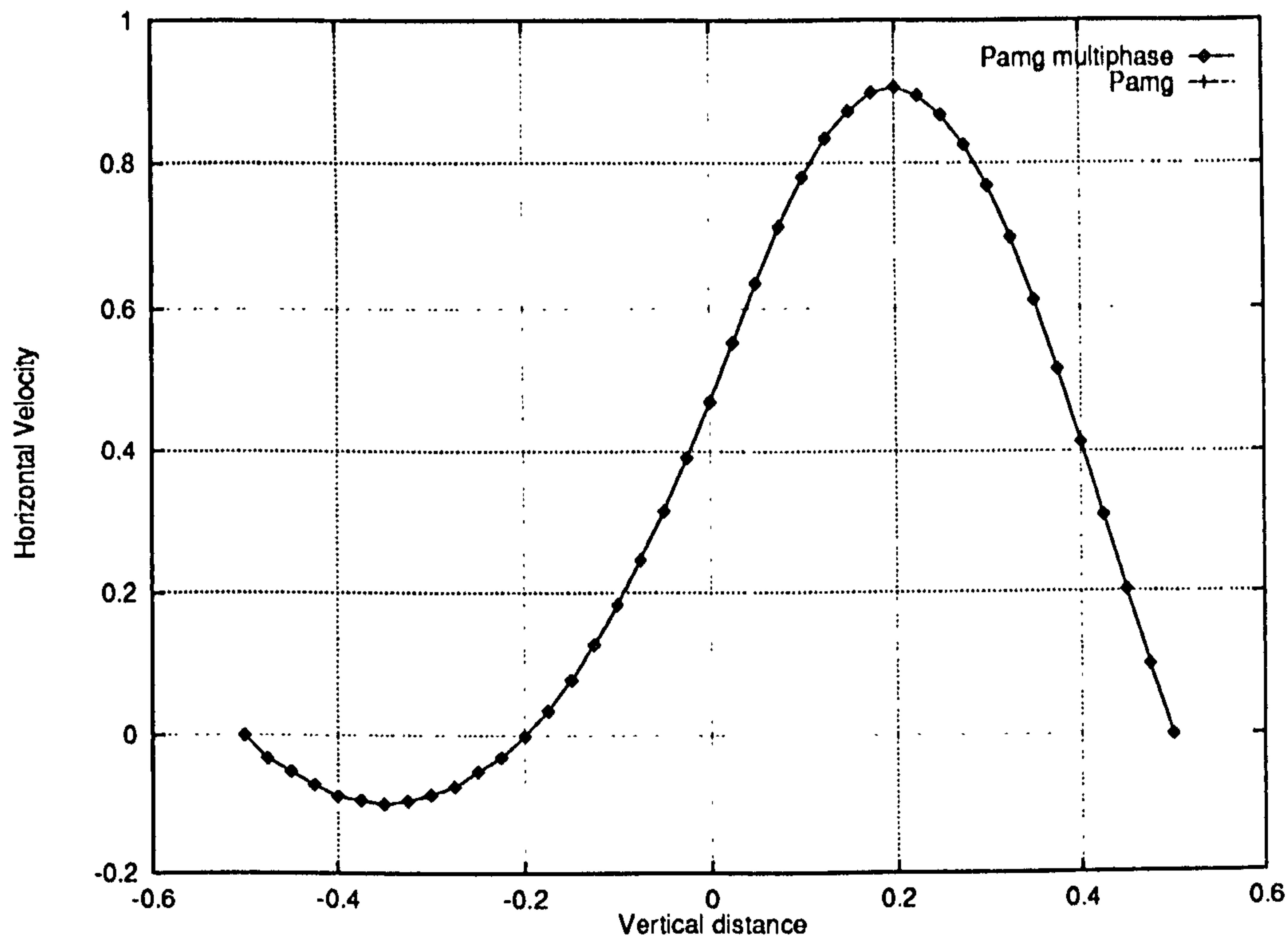


Figure 5.21: Single phase backward-facing problem – Horizontal velocity profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4

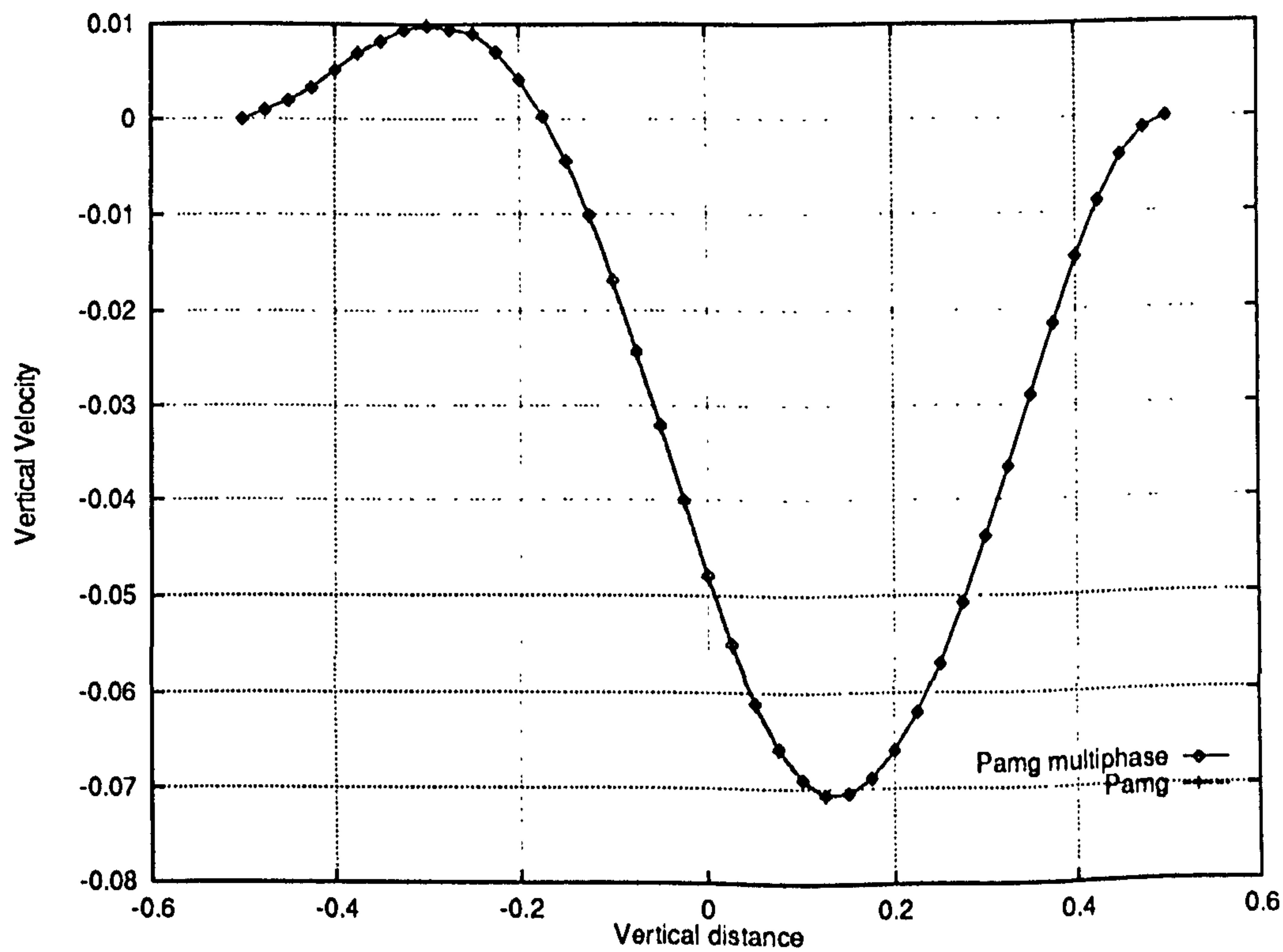


Figure 5.22: Single phase backward-facing problem – Vertical velocity profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4

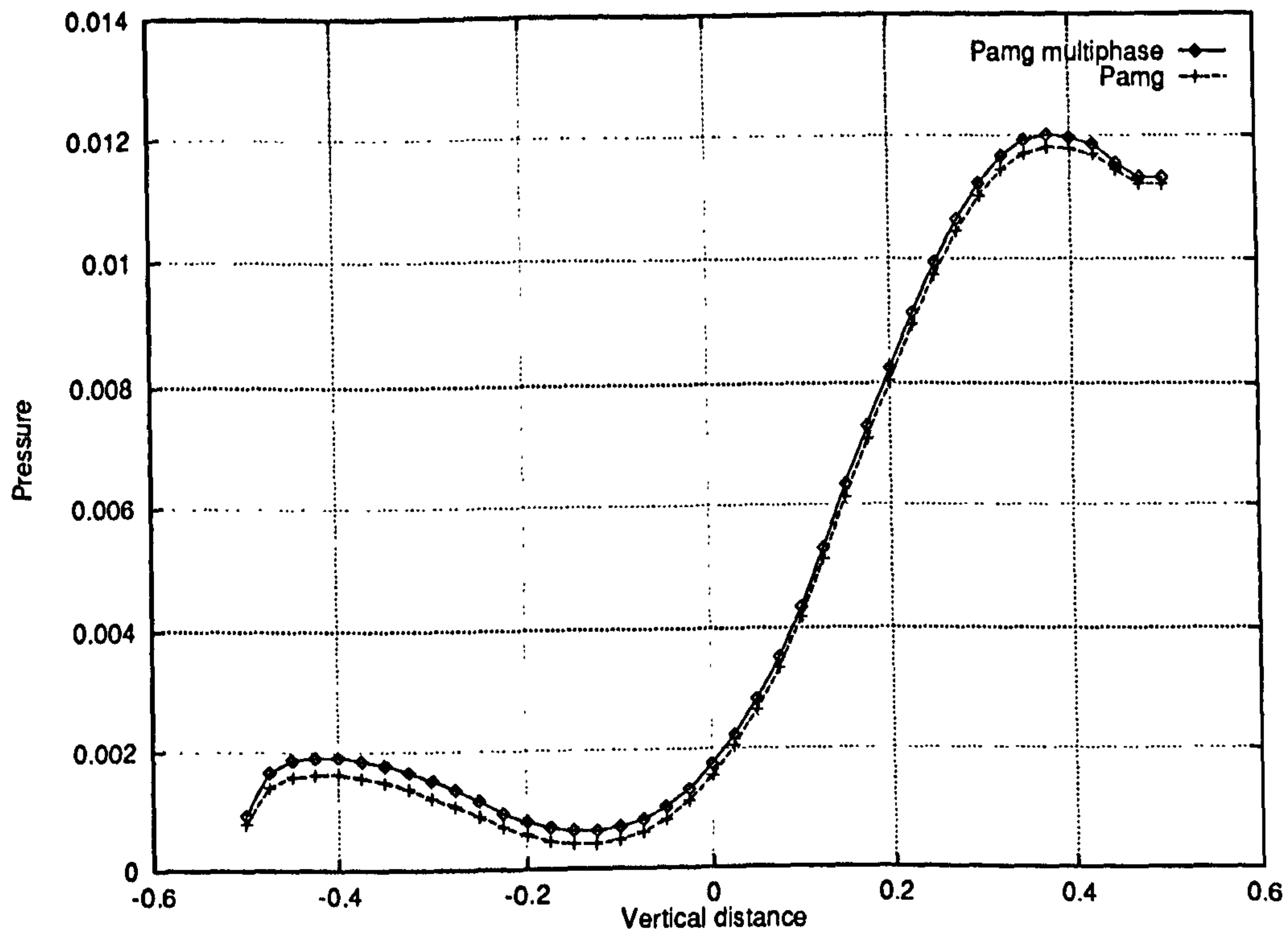


Figure 5.23: Single phase backward-facing problem – Pressure profile along the line $x = 3.8$ – Comparison of the pamg and pamg-multiphase solution on an adaptive grid up to level 4

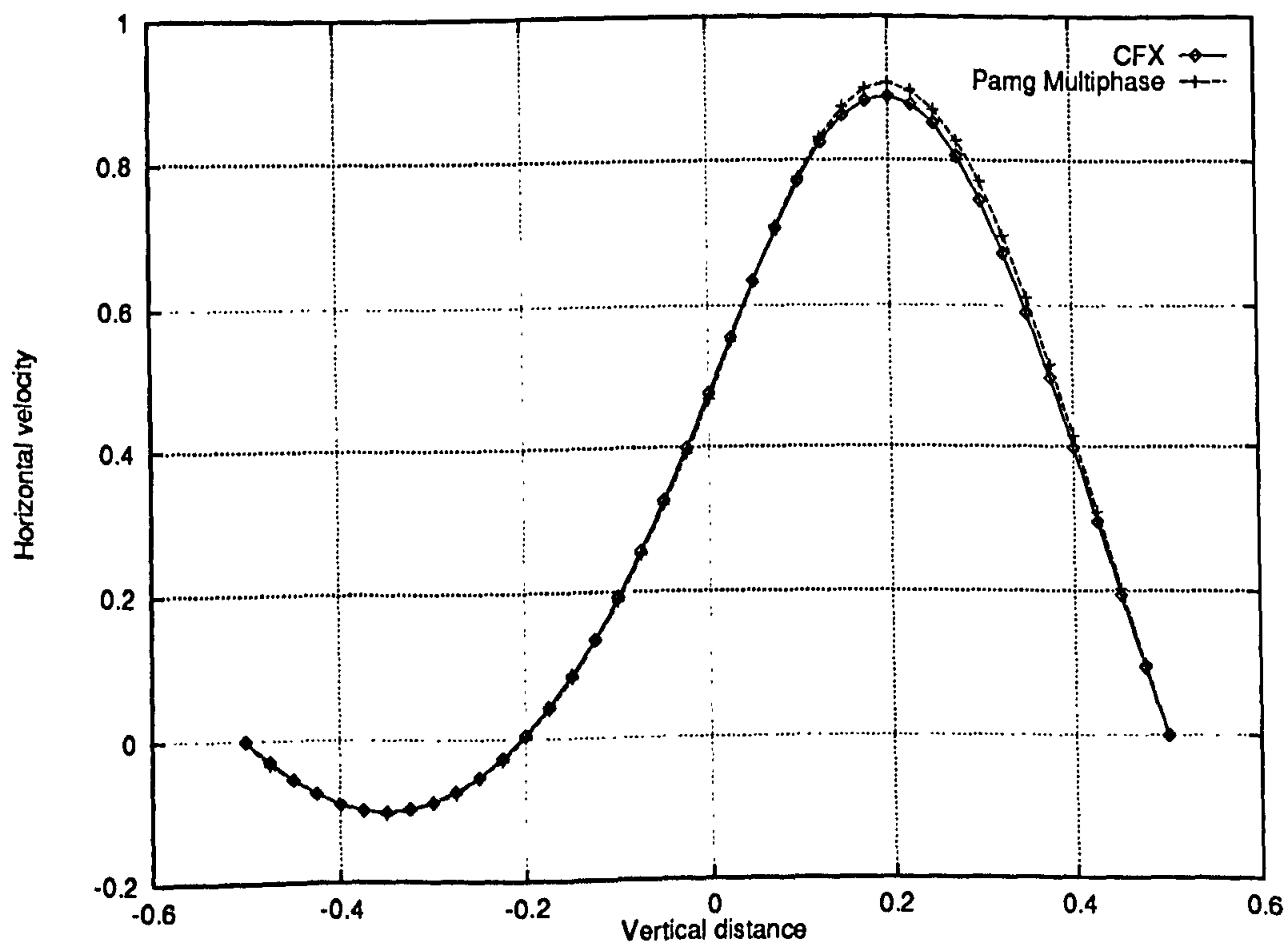


Figure 5.24: Single phase backward-facing problem – Horizontal velocity profile along the line $x = 3.8$ – Comparison of the pamg-multiphase and CFX 4.1 solutions on a uniform level 3 grid

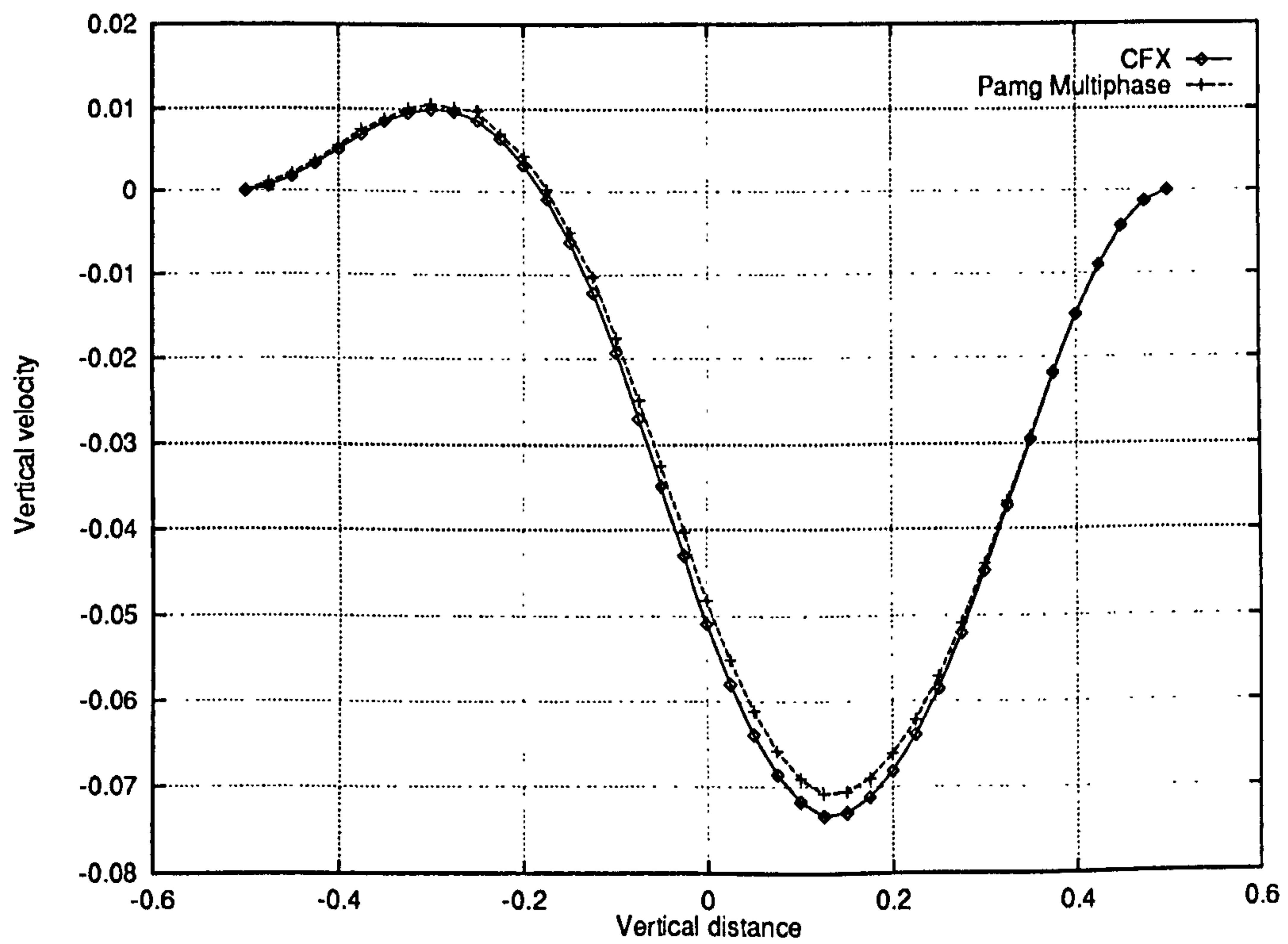


Figure 5.25: Single phase backward-facing problem – Vertical velocity profile along the line $x = 3.8$ – Comparison of the pamg-multiphase and CFX 4.1 solutions on a uniform level 3 grid

5.3 Multiphase Test Problems

The correctness of the `pang-multiphase` code has been established for single phase flows. We now focus on multiphase flows. Here, only comparisons with the `CFX 4.1` solutions are available. Three test cases of varying complexity are examined.

- Firstly, we consider a two-phase channel flow, which is geometrically simple but has significant phase separation when the fluids have different properties.
- Secondly, a two-phase flow through a T-junction with two inlets, chosen as representative of a complex two-phase flow with mixing and separation.
- Thirdly, a two-phase flow through a backward-facing step which is characterised by potentially large recirculation zones. This test case raises novel issues of mathematical well-posedness.

5.3.1 Problem 1: Two-Phase Channel Flow

Description and Specifications This first test problem is relatively easy in its geometry since we simply consider the flow of two *different* fluids in a channel. The geometry is the same as for Section 5.2.1. but the two phases have different densities and viscosities:

Reynolds Number	viscosity	density
$Re_1 = 100$	$\mu_1 = 0.01$	$\rho_1 = 1.0$
$Re_2 = 100$	$\mu_2 = 0.005$	$\rho_2 = 0.5$

For multiphase flows, the Reynolds number has been defined as a simple extension of the usual single phase definition:

$$Re_\alpha \equiv \frac{\rho_\alpha u_\alpha d}{\mu_\alpha}.$$

The characteristic length d is the channel width at the inlet.

The computational grids are also the same as for the single phase channel problem of Section 5.2.1. At first glance (Figures 5.26, 5.27 and 5.35), it may seem that the length of channel being modelled is sufficient for the flow to reach an almost developed state. Subsequent results (Section 5.4.11) show that this is actually far from the truth. However, it also established that modelling the flow over such a small length does not alter the accuracy of the results.

Main Features of the Solution Figures 5.26 and 5.27 give streamlines for each phase respectively while Figure 5.114 shows the volume fraction distribution throughout the computational domain. The combination of these figures gives a good overall view of the flow. Figures 5.28 to 5.33, for their part, are solution profiles taken at various locations and contains very precise information which can be used for comparisons.

Since both phases share the same pressure field, the phase with less inertia is accelerated relative to the other (Figure 5.28). As a result, there is significant flow separation as shown by the volume fraction field (Figure 5.29). Along the main direction of the flow as well as across the channel width, it can clearly be seen that each phase has a very distinct velocity field (Figures 5.28 and 5.30). In addition, the multiphase velocity profiles are quite different from the single phase parabolic profiles (Figure 5.30).

The result of the simulations also indicates that the flow evolves toward a separated layered pattern with the denser phase concentrated toward the sides of the channel (Figures 5.32 and 5.33). This is correct as it corresponds to a minimal energy configuration. Figure 5.31 shows that there is some relative motion between the phases in the transverse direction as the flow separates. The minimal energy configuration is obtained when the more viscous fluid occupies the regions near the walls. In order to confirm this, a two-phase channel flow where the fluids have the following properties was computed:

Reynolds Number	viscosity	density
$Re_1 = 100$	$\mu_1 = 0.01$	$\rho_1 = 1.0$
$Re_2 = 200$	$\mu_2 = 0.005$	$\rho_2 = 1.0$

Figure 5.34 shows the volume fraction profile near the outlet for the most viscous phase and demonstrates that the less viscous phase is almost absent near solid walls.

Figures 5.32 and 5.33 suggest that the boundary conditions chosen for the volume fractions at solid walls, $(\partial r/\partial y = 0)$ are not adequate. Further experiments with a second order alternative, $\partial^2 r/\partial y^2 = 0$, have shown that this does not affect the convergence pattern of the solver.

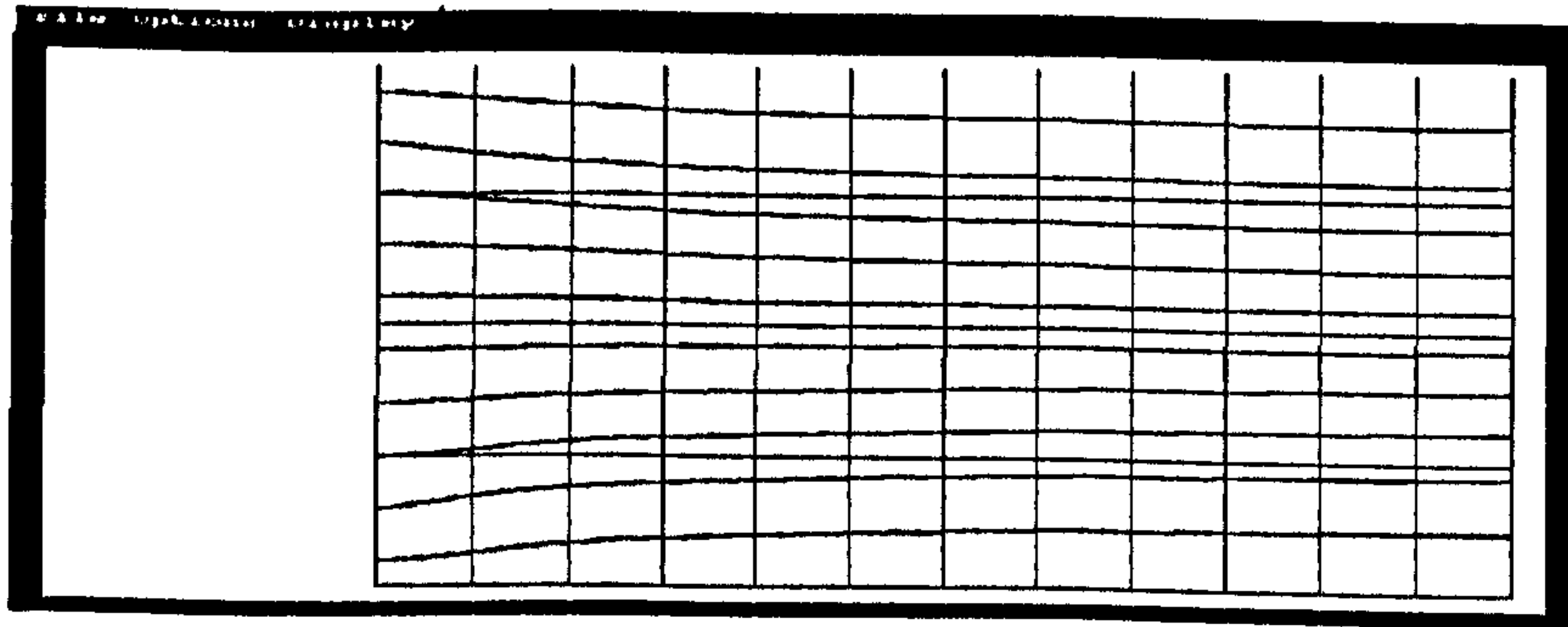


Figure 5.26: Two-phase channel flow – Streamlines for phase 1

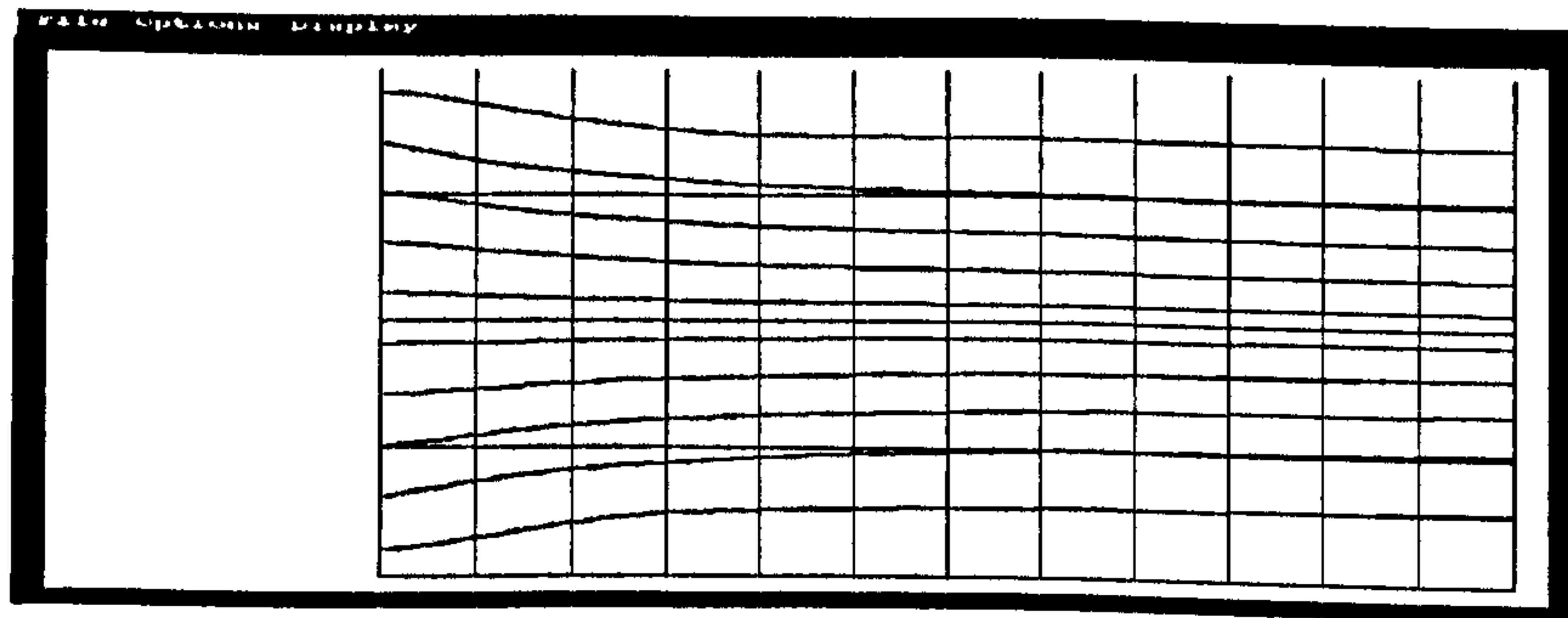


Figure 5.27: Two-phase channel flow – Streamlines for phase 2

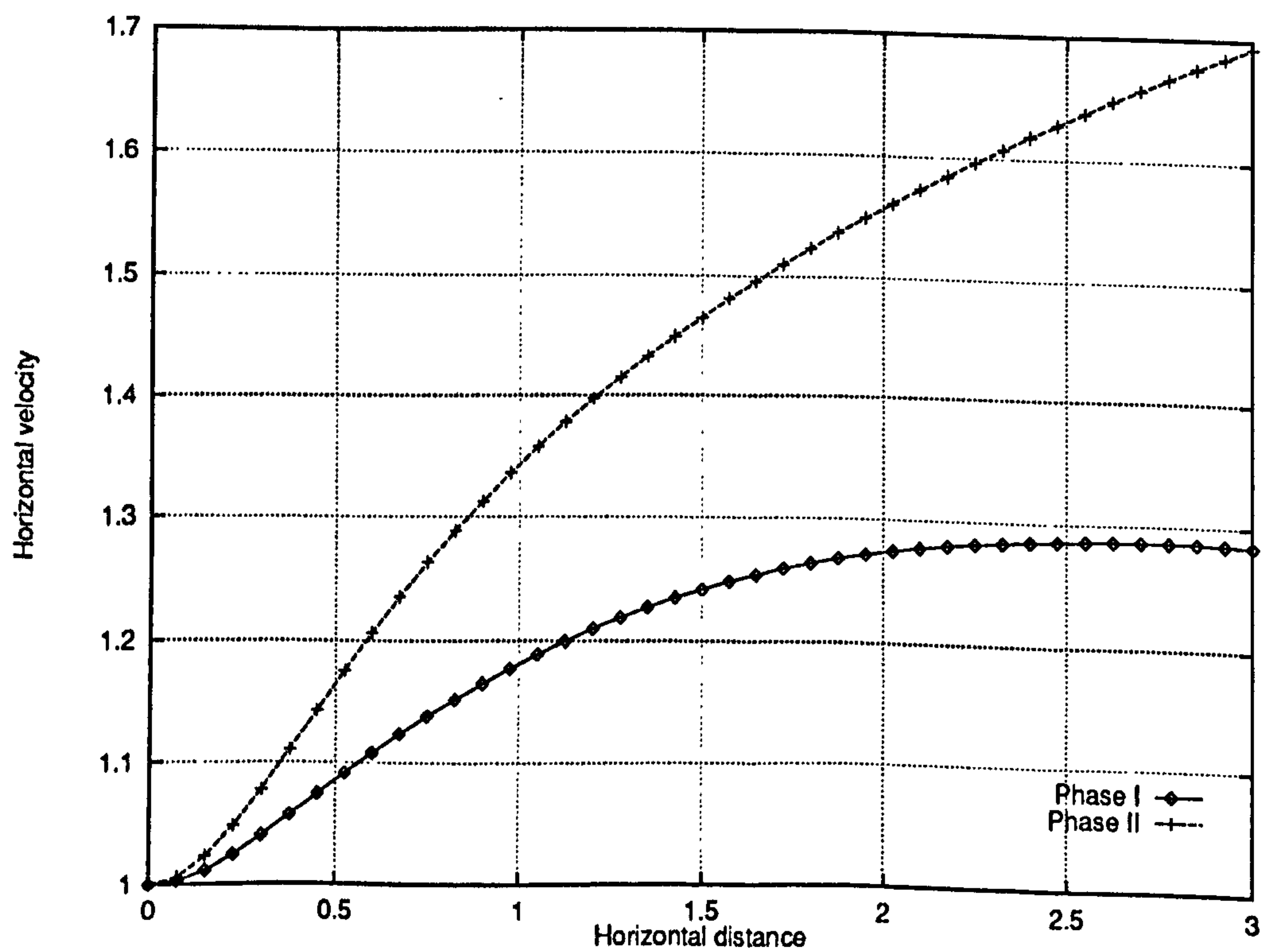


Figure 5.28: Two-phase channel flow – pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.5$

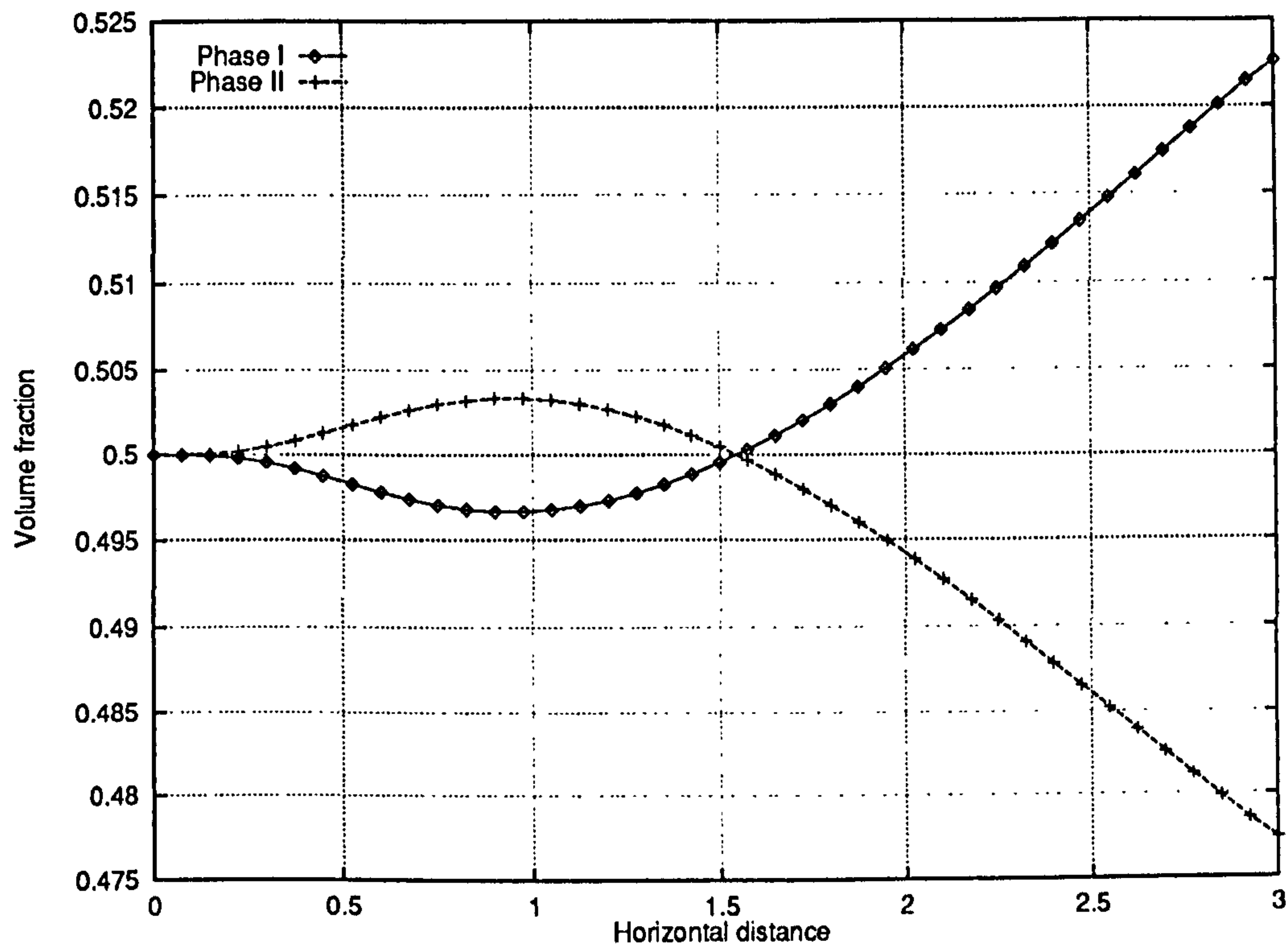


Figure 5.29: Two-phase channel flow – pang-multiphase results – Volume fraction profiles along the line $y = 0.5$

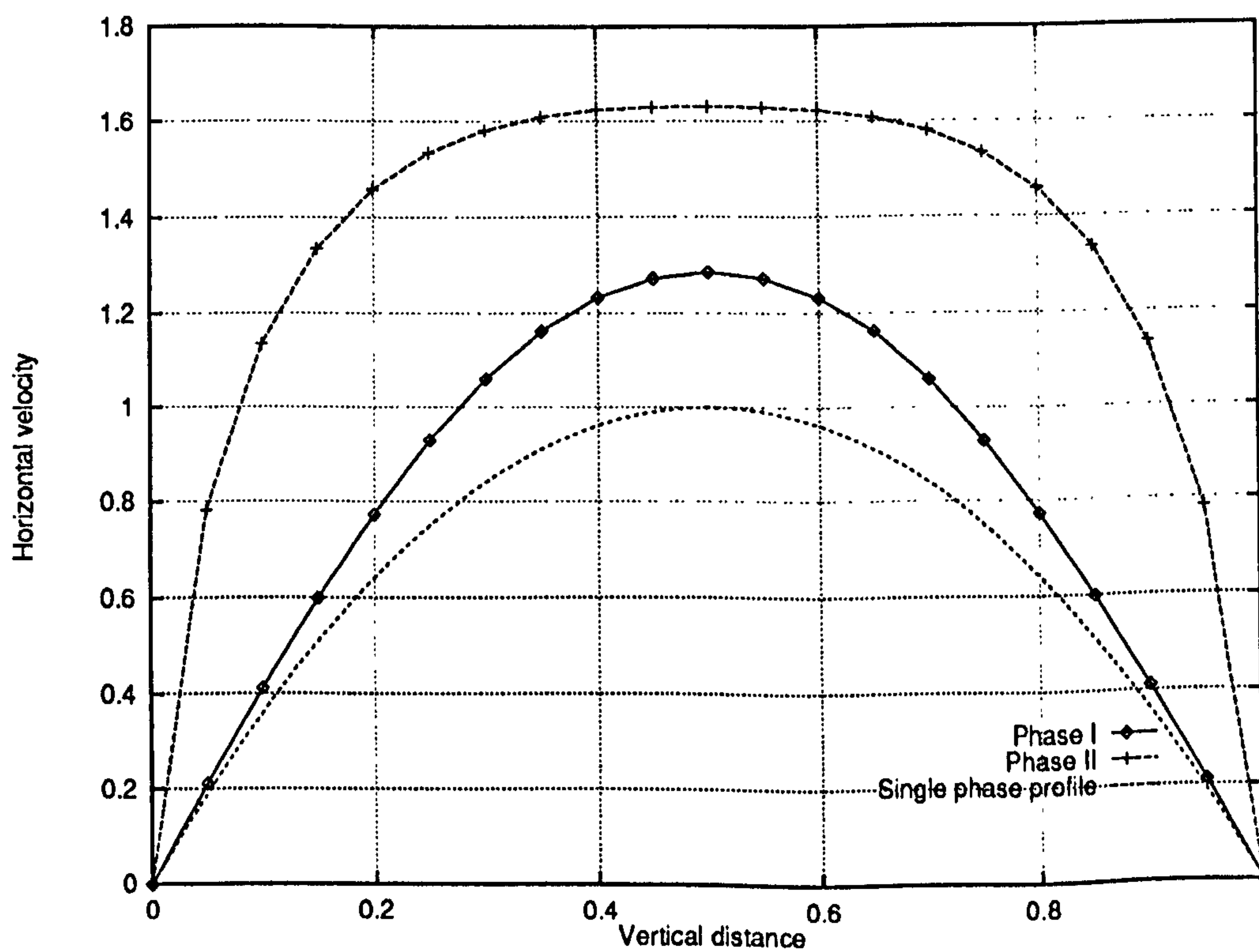


Figure 5.30: Two-phase channel flow – pang-multiphase results – Horizontal velocity profiles along the line $x = 2.5$

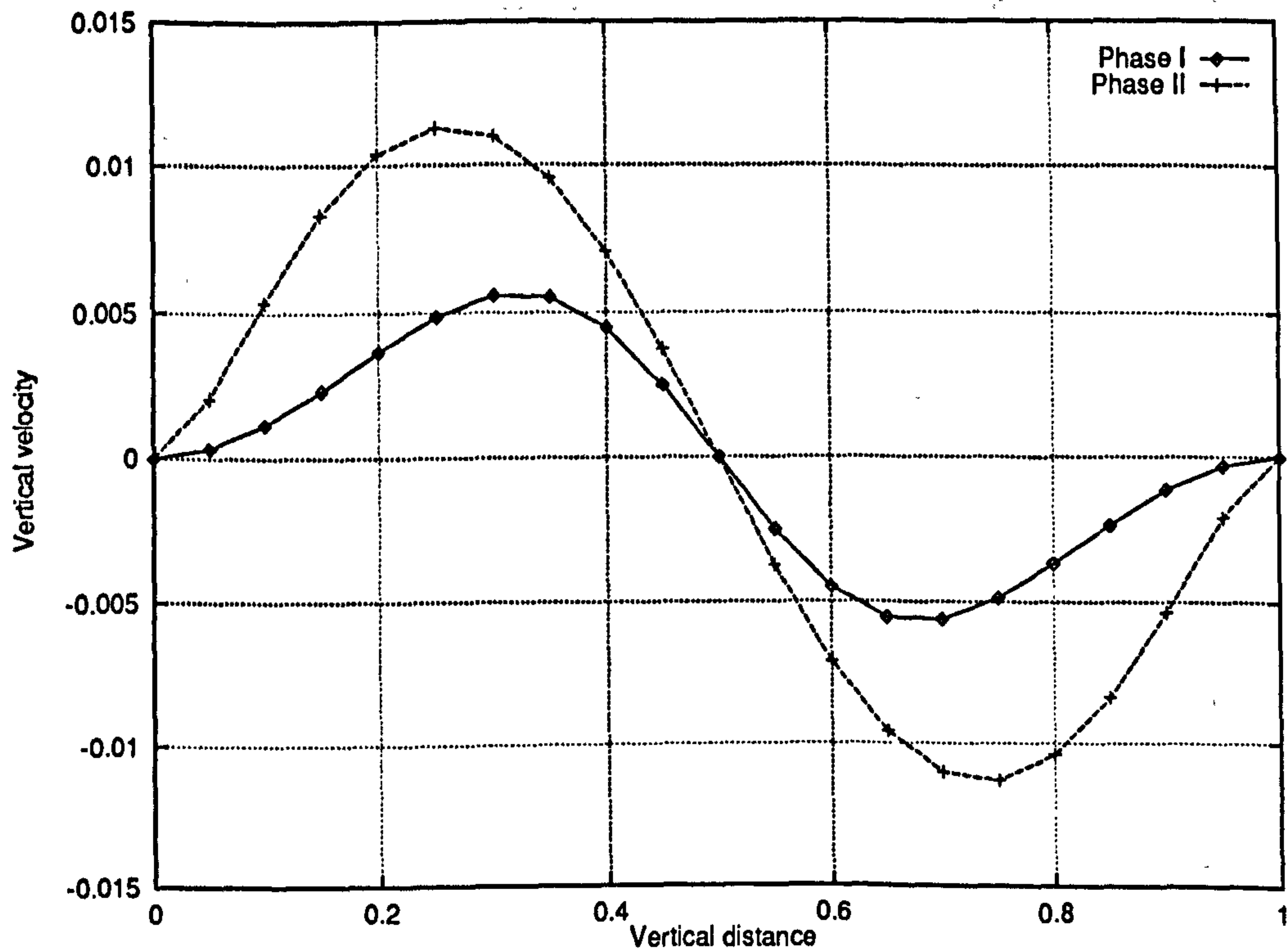


Figure 5.31: Two-phase channel flow - pamg-multiphase results - Vertical velocity profiles along the line $x = 2.5$

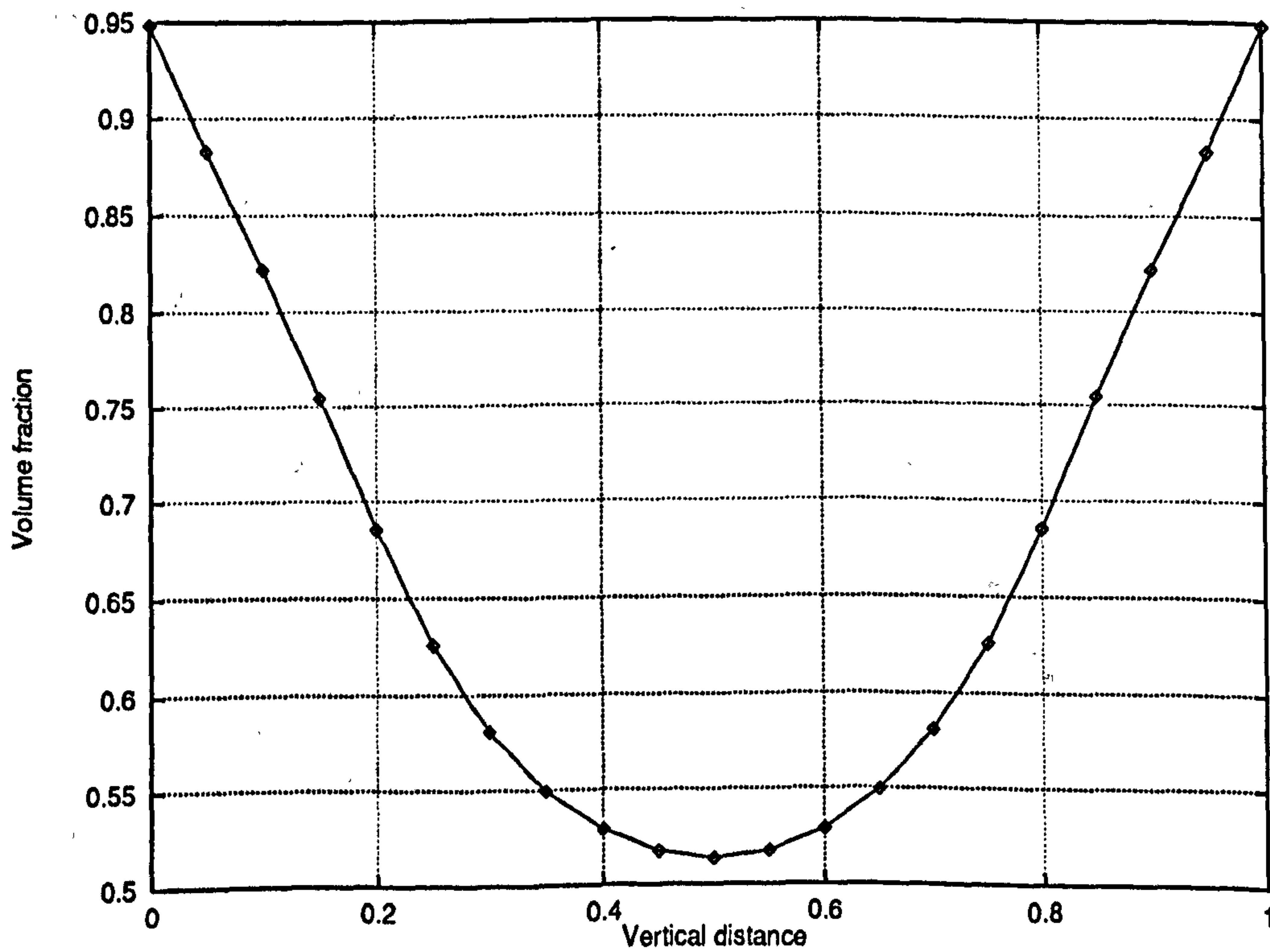


Figure 5.32: Two-phase channel flow - pamg-multiphase results - Volume fraction profile for phase 1 along the line $x = 2.5$

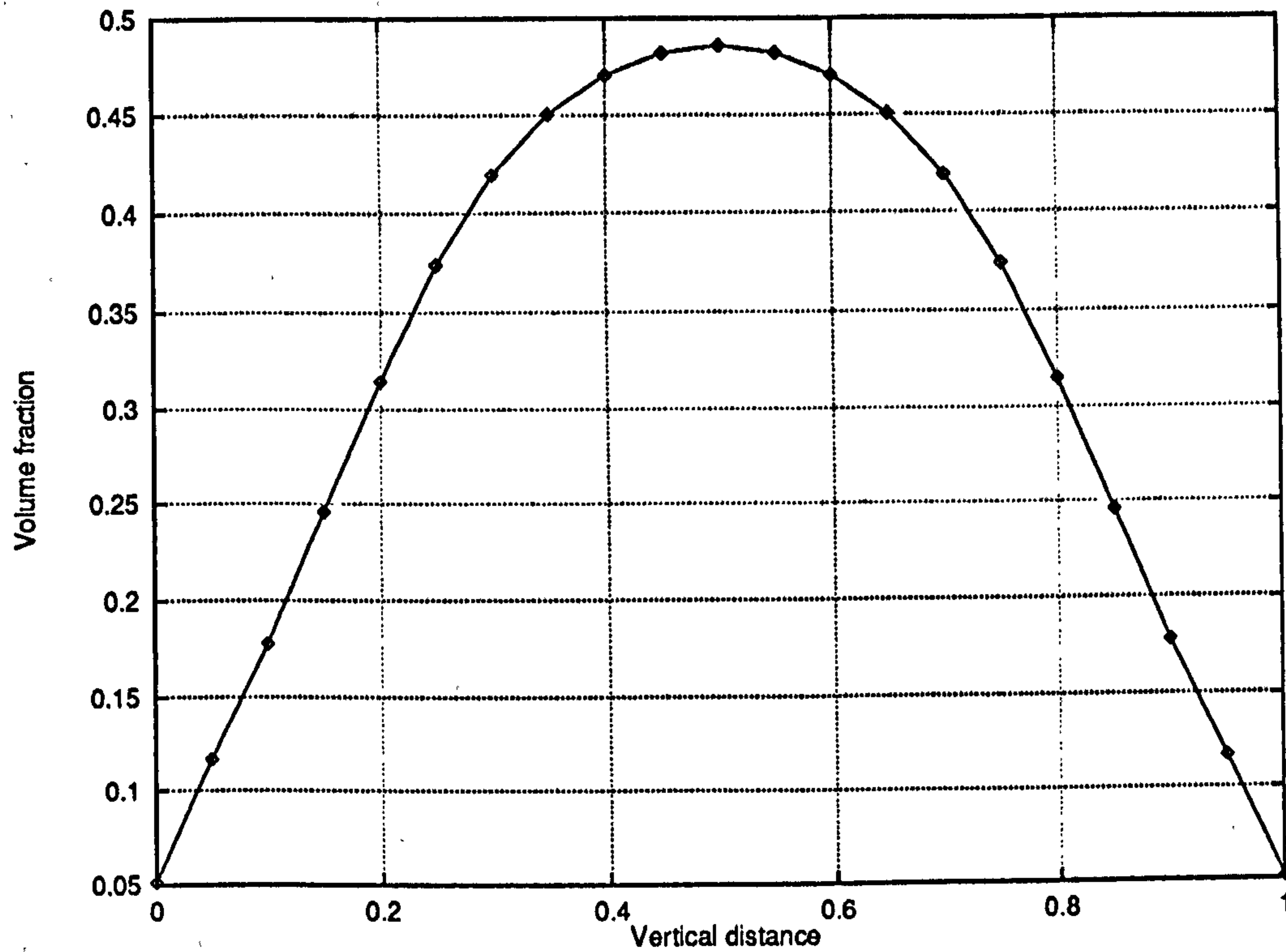


Figure 5.33: Two-phase channel flow – pamg-multiphase results – Volume fraction profile for phase 2 along the line $x = 2.5$

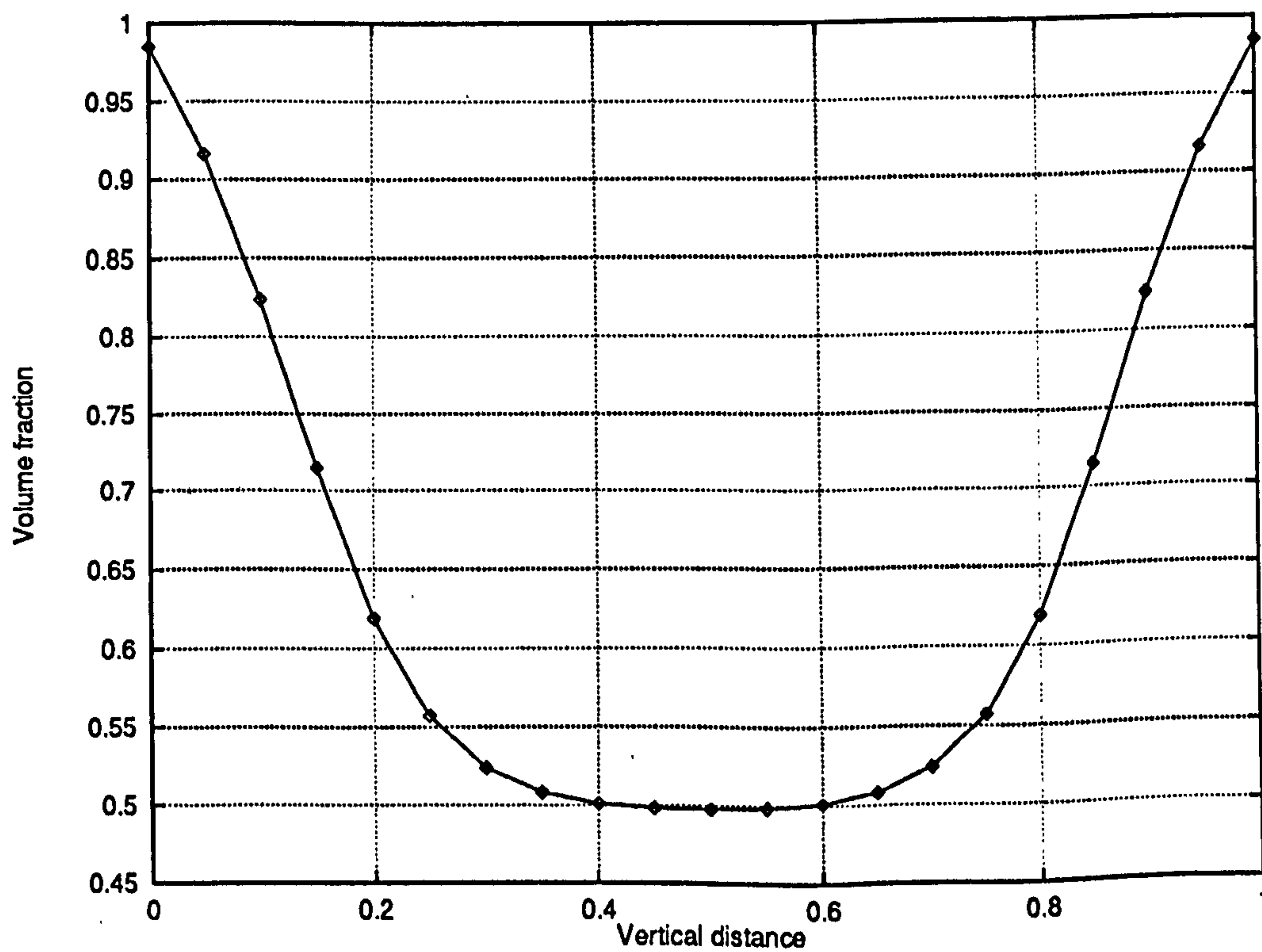


Figure 5.34: Two-phase channel flow – pamg-multiphase results – Volume fraction profile for phase 1 along the line $x = 2.5$ for the following set of physical properties: $(\mu_1 = 0.01, \rho_1 = 1)$ and $(\mu_2 = 0.005, \rho_2 = 1)$

Comparisons with CFX 4.1 Solutions The solutions given by `pang-multiphase` have been compared with those provided by the `CFX 4.1` code which uses the IPISA algorithm [67, 69], see Section 2.3.3, to solve the multi-fluid equations. Figures 5.35 to 5.41 show that good, but not perfect, agreement is obtained. In particular, as the outlet is approached, the horizontal velocity in the flow direction differs more noticeably (Figure 5.35). This may be due to different treatment of the boundary conditions — see Section 5.3.2. Other factors which can explain the relatively small discrepancies are:

- different discretisations of the same governing equations;
- different convergence criteria;
- different gridding (`CFX 4.1` uses collocated grids).

Furthermore, it is well known that the Navier-Stokes equations for incompressible flows are singular: the pressure is not uniquely defined as it admits an arbitrary additive constant. This can be easily seen in the equations since they only involve pressure gradients or in discrete terms, pressure differences. For compressible flows, this issue does not arise since the absolute pressure is fixed by an equation of state. The pressure singularity for incompressible flows explains the results of Figure 5.36: the `pang-multiphase` and `CFX 4.1` pressure fields are different and yet almost equivalent since they differ only by an additive constant, as shifting by a suitable amount demonstrates (Figure 5.37)

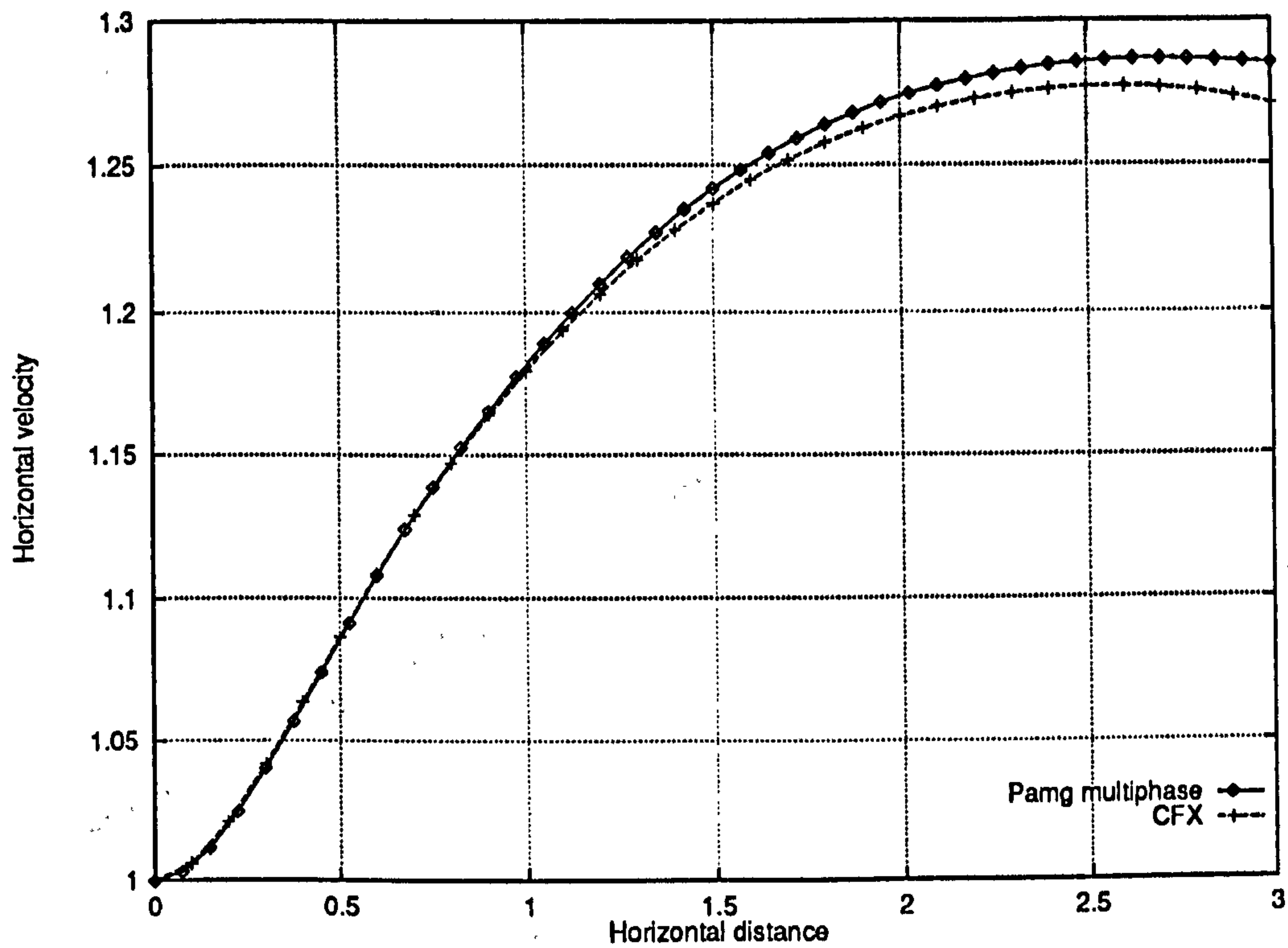


Figure 5.35: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)

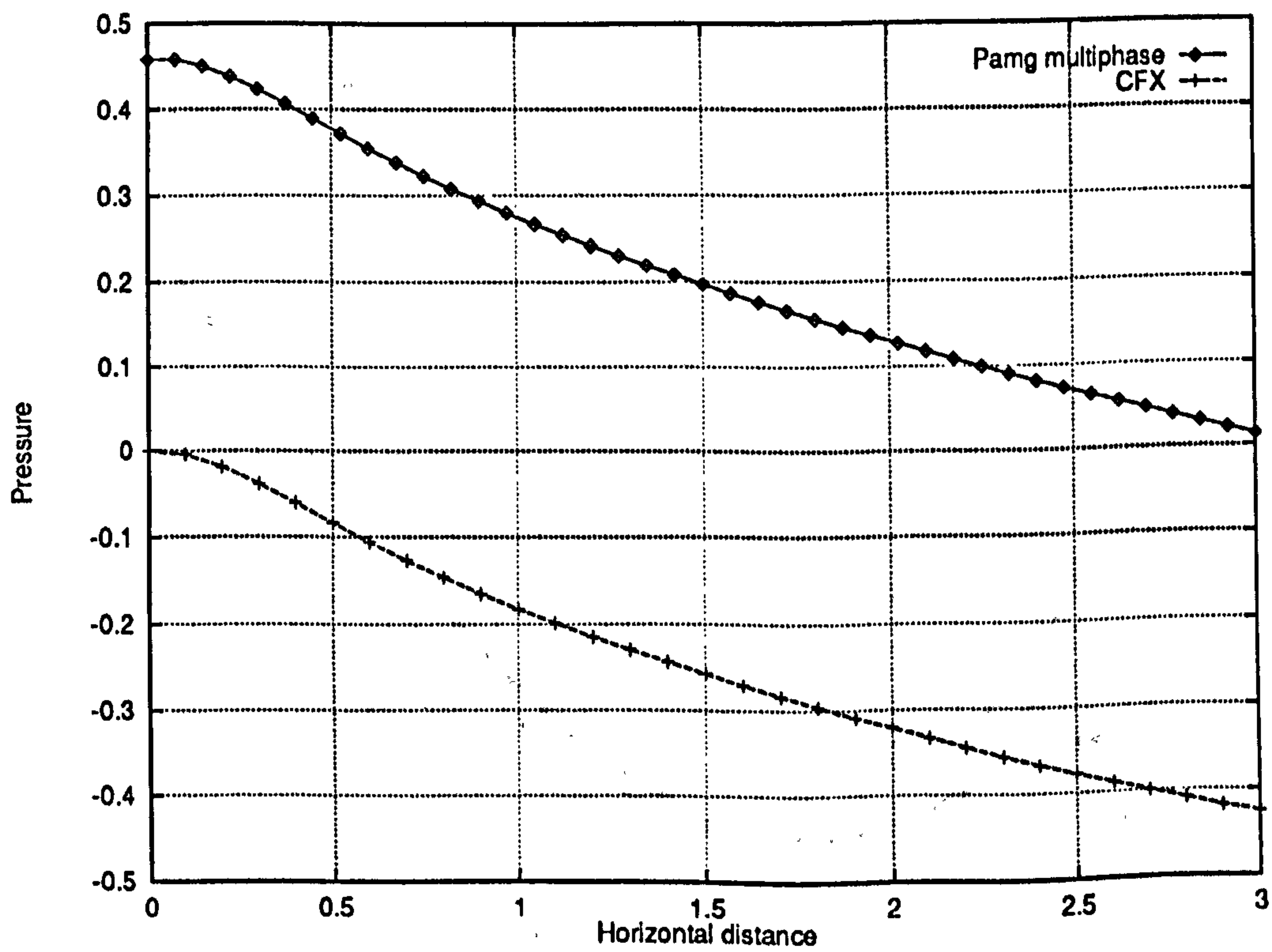


Figure 5.36: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ (Phase 1) – See Figure 5.37

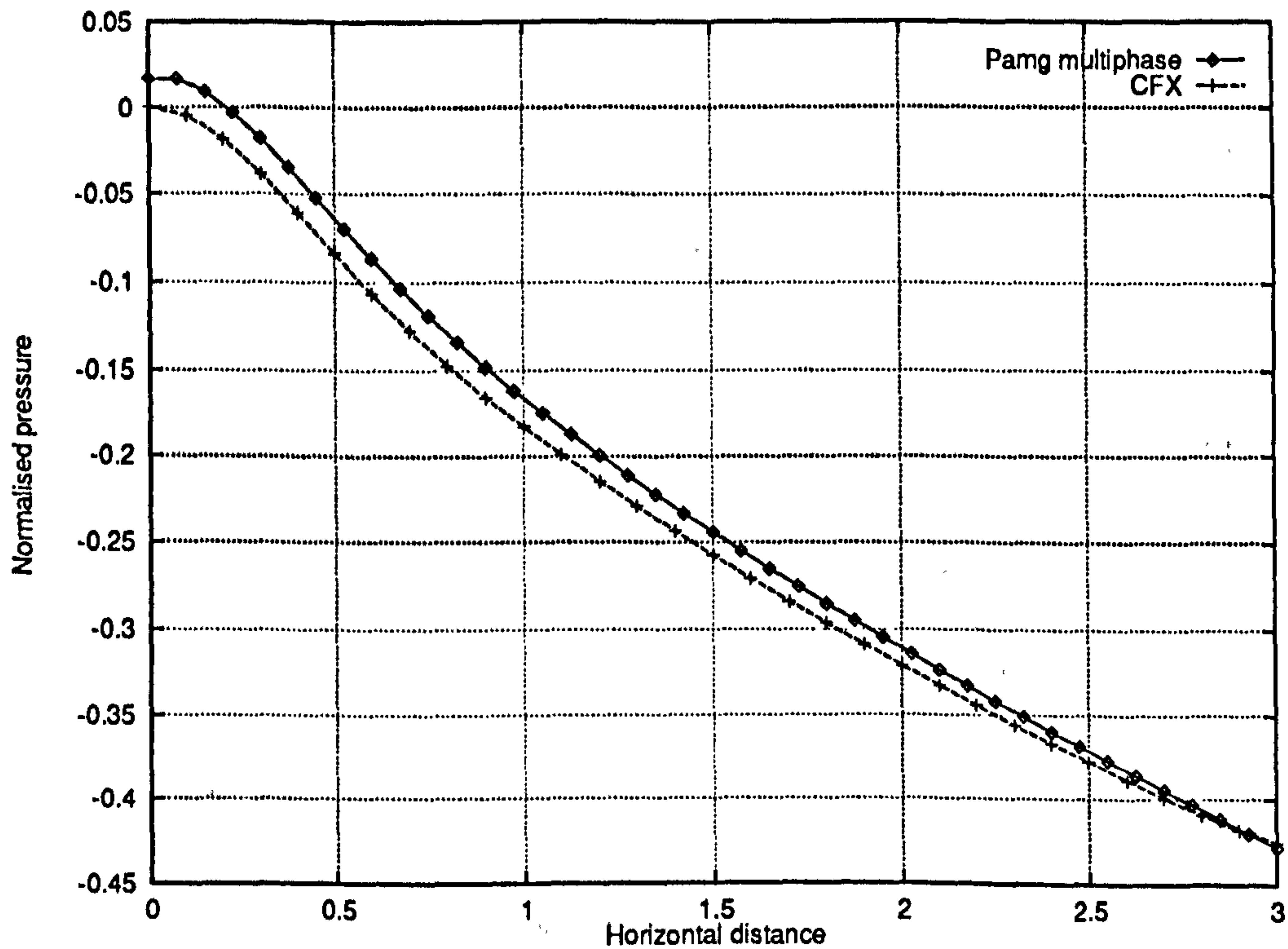


Figure 5.37: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ (Phase 1) after shifting

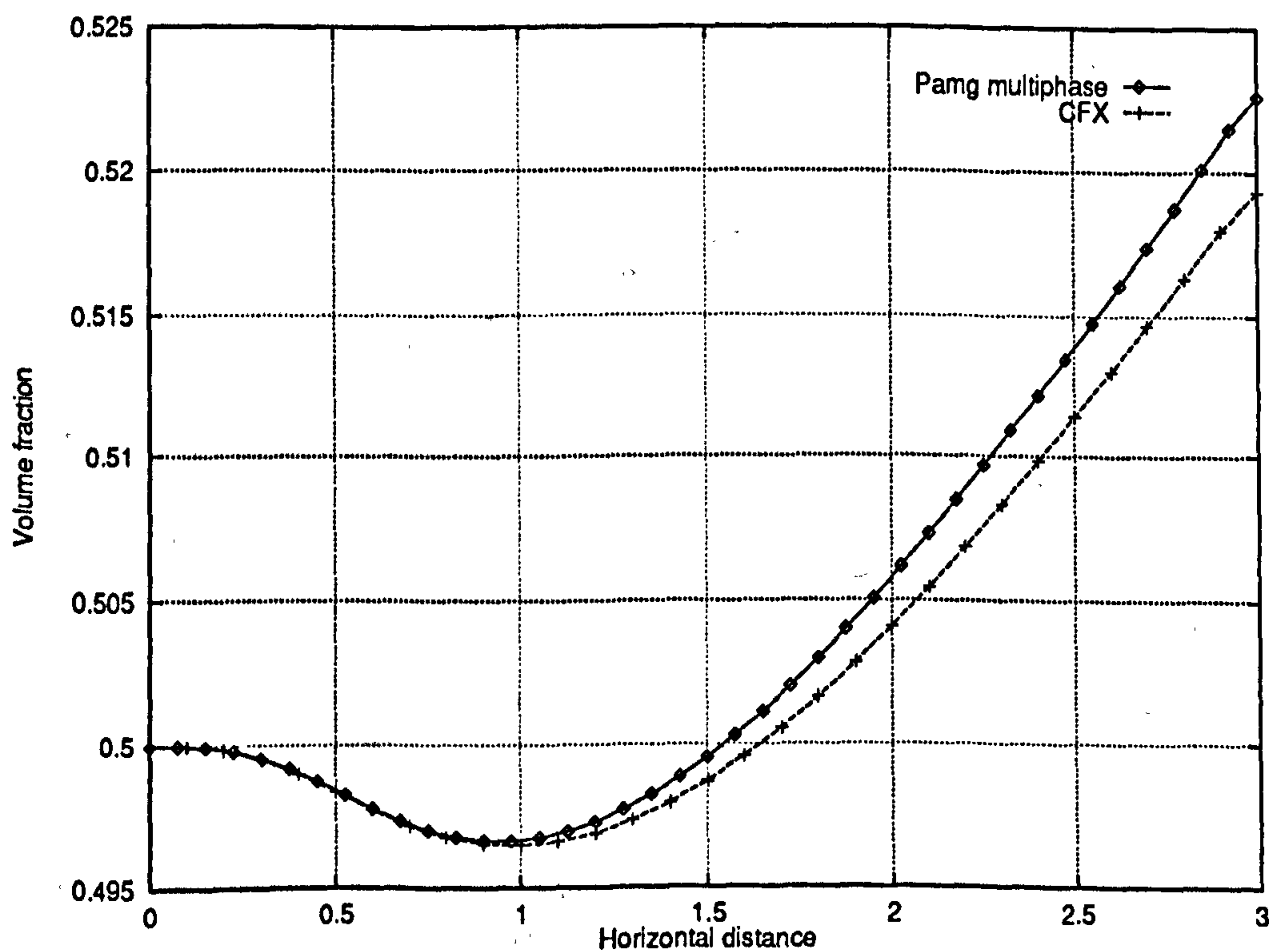


Figure 5.38: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Volume fraction profiles along the line $y = 0.5$ (Phase 1)

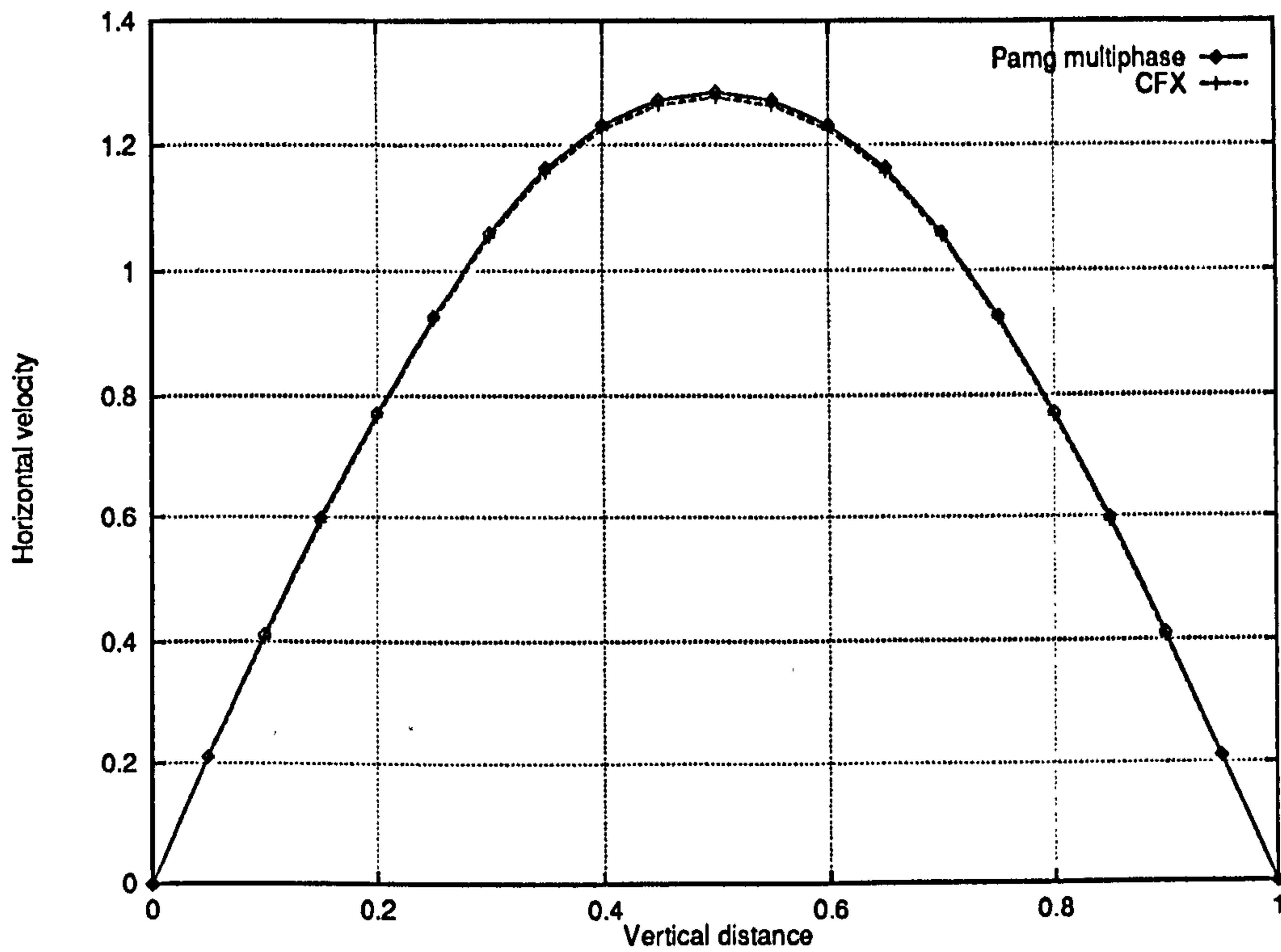


Figure 5.39: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $x = 2.5$ (Phase 1)

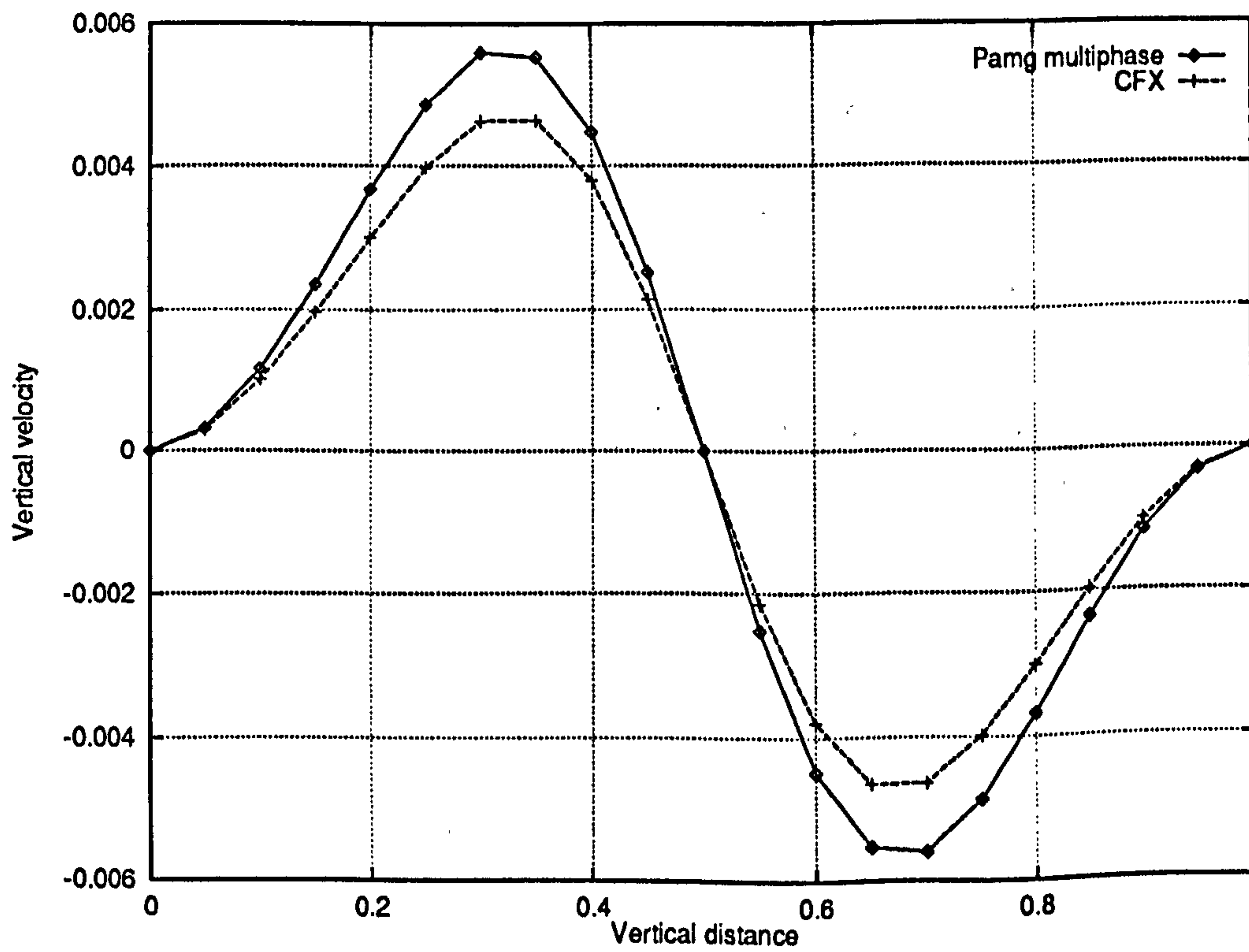


Figure 5.40: Two-phase channel flow – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $x = 2.5$ (Phase 1) – Note the scale

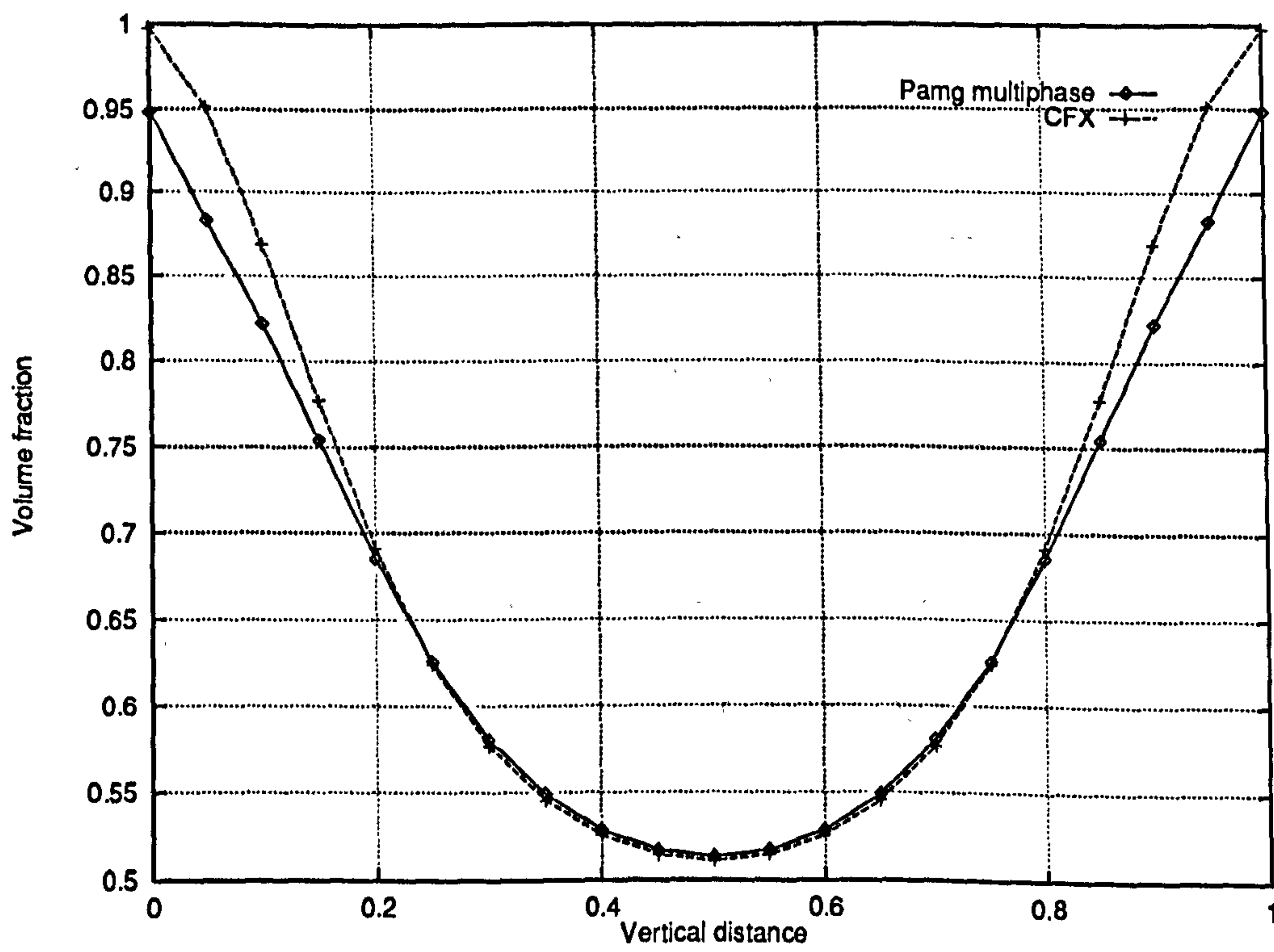


Figure 5.41: Two-phase channel flow - Comparison of pamg-multiphase and CFX 4.1 results - Volume fraction profiles along the line $x = 2.5$ (Phase 1)

Grid Independence of the Solutions Grid independence is a very important characteristic of discrete solutions. It is a good indication — but not an absolute one — that the solution is reasonably close to the solution of the continuous problem, since grid independence implies that the truncation error is limited. The failure of a code to produce grid independent solutions often indicates an implementation or discretisation error. It should be noted that some quantities may not be grid independent, for instance pressures, and that some may be more sensitive than others. The choice of a non-dimensionalisation may be important in that last respect.

Studies of the grid independence of the solution have been carried out – see Figures 5.42 to 5.46 – which support the conclusion that provided the finest grid was fine enough, the results provided by `pamg-multiphase` have a significant degree of grid independence.

There is one very noteworthy exception: the volume fractions. Their evolution along the pipe is strongly dependent on the grid size – see Figure 5.46. Figures 5.47 and 5.48 show that the CFX 4.1 solutions share this feature to some extent. It is therefore reasonable to conclude that the root cause is in the governing equations and that the `pamg-multiphase` implementation is not to blame. It is also possible that the boundary conditions play an important role.

Finally, comparison of Figures 5.46 and 5.48 shows that `pamg-multiphase` tends to diffuse the volume fractions more than CFX 4.1 for a given grid size. This may be a direct consequence of the use in `pamg-multiphase` of an upwind interpolation for the volume fractions in the continuity equations (Section 4.3.2).¹

¹It should be noted however, that CFX 4.1 uses upwind interpolation for the volume fractions in the convective terms.

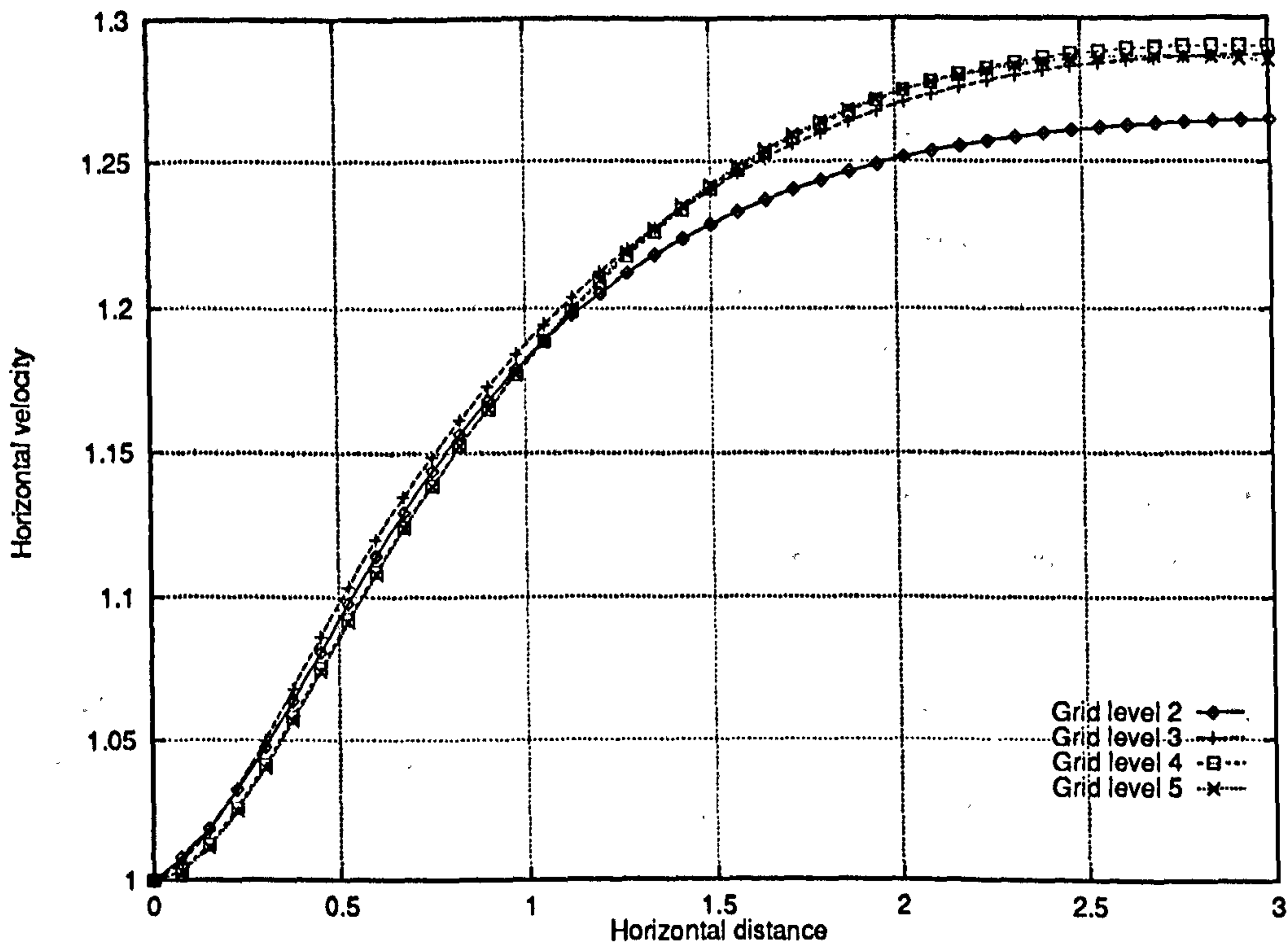


Figure 5.42: Two-phase channel flow – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)

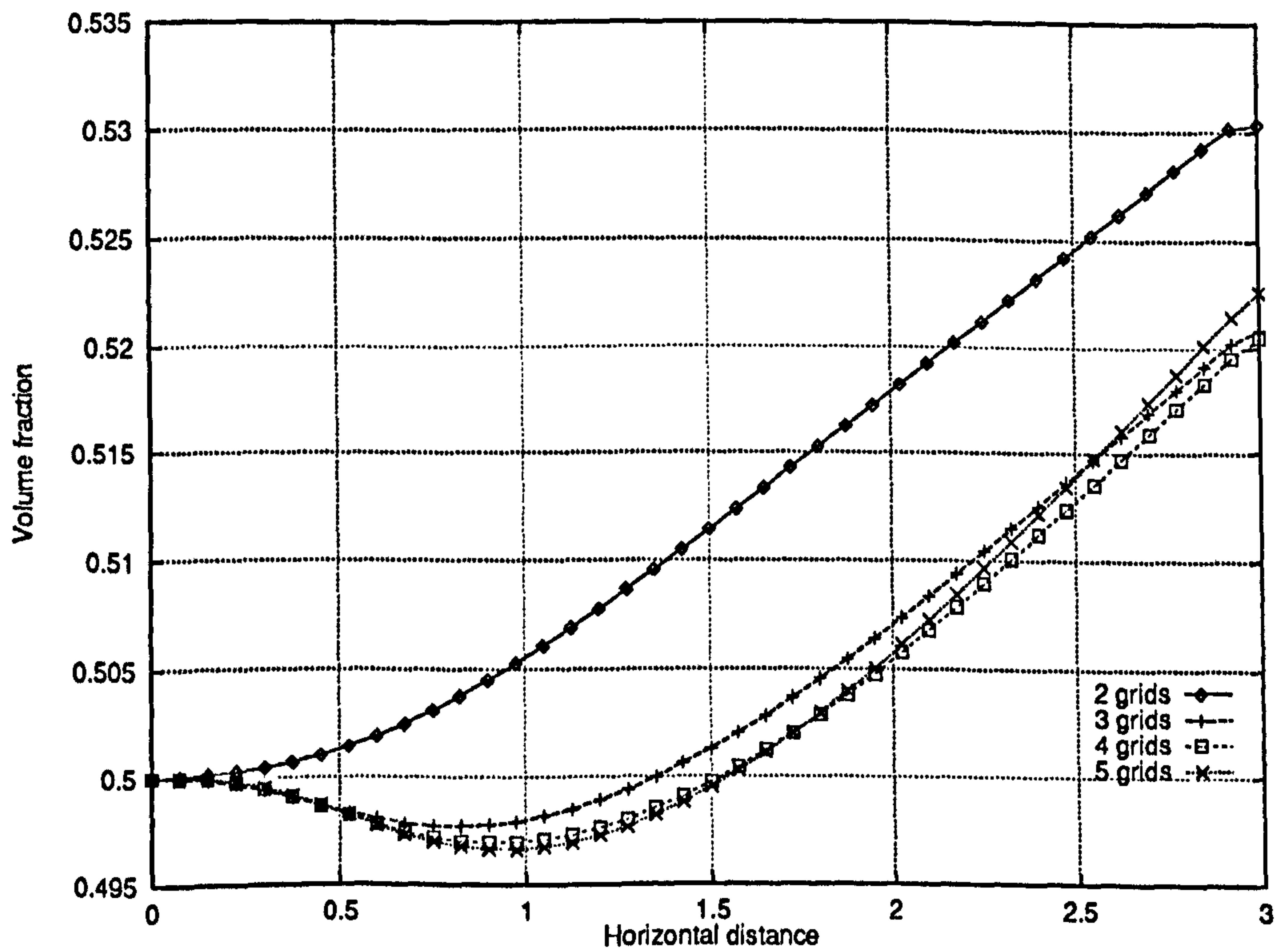


Figure 5.43: Two-phase channel flow – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $y = 0.5$ (Phase 1)

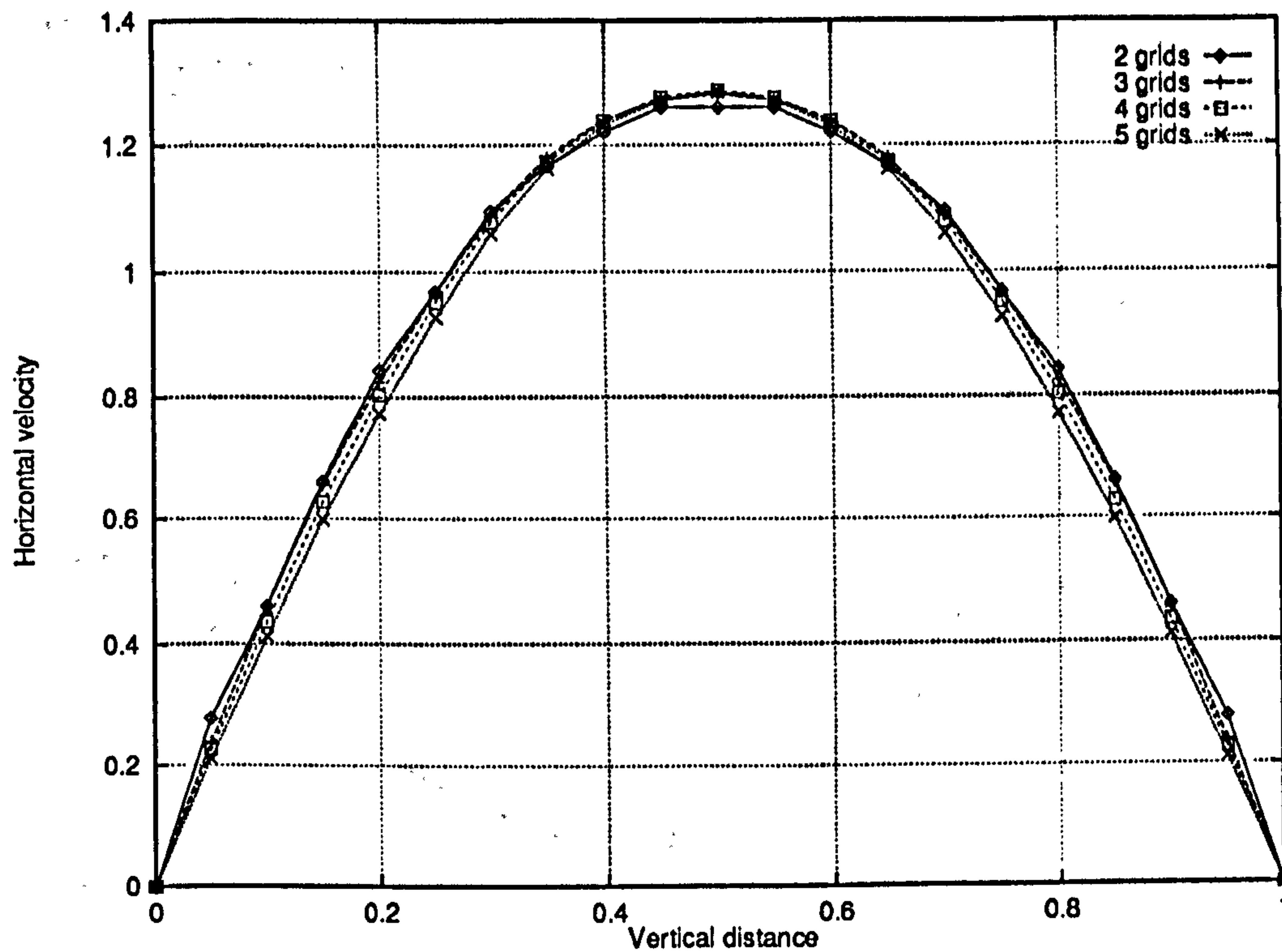


Figure 5.44: Two-phase channel flow – Grid independence for pang-multiphase results – Horizontal velocity profiles along the line $x = 2.5$ (Phase 1)

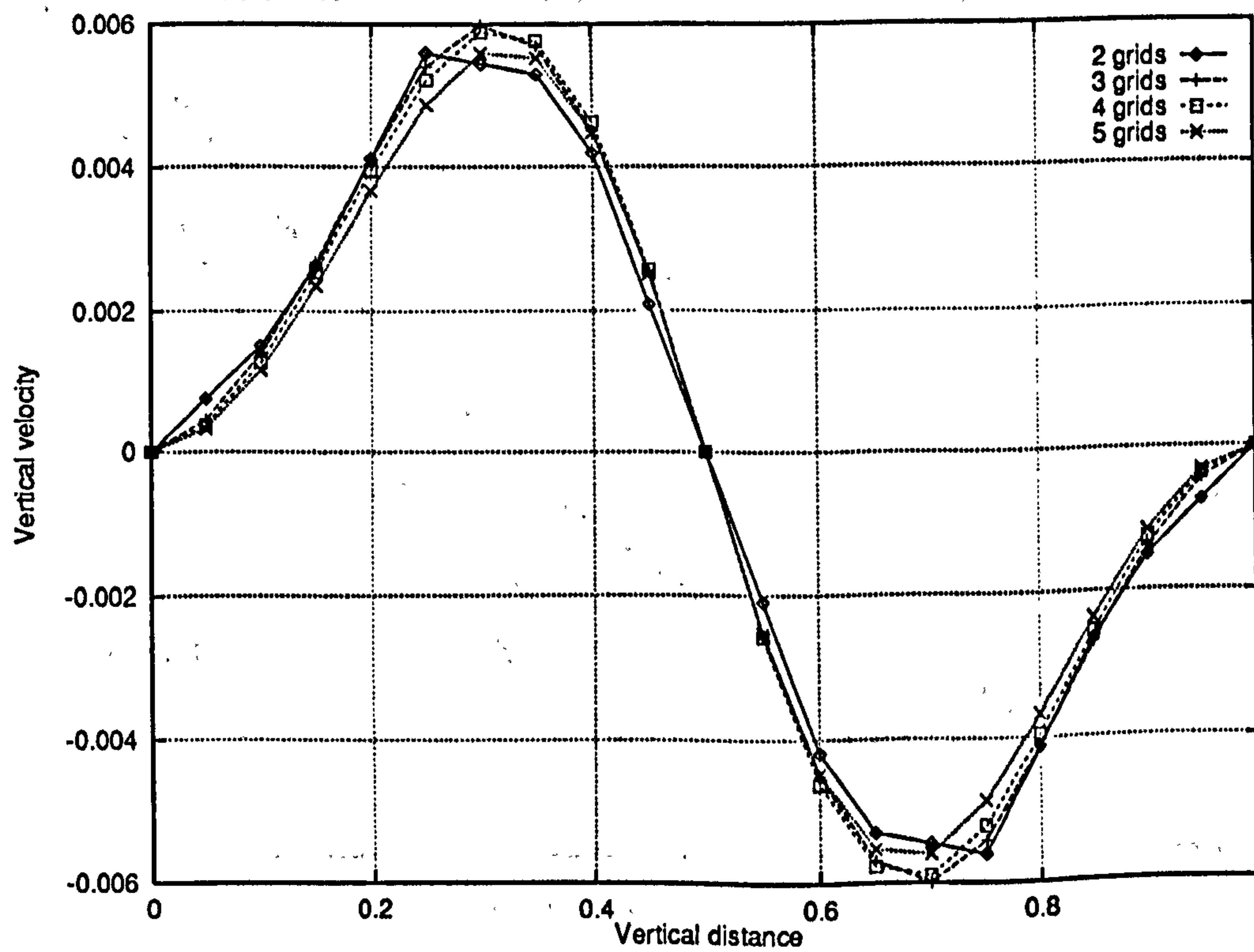


Figure 5.45: Two-phase channel flow – Grid independence for pang-multiphase results – Vertical velocity profiles along the line $x = 2.5$ (Phase 1)

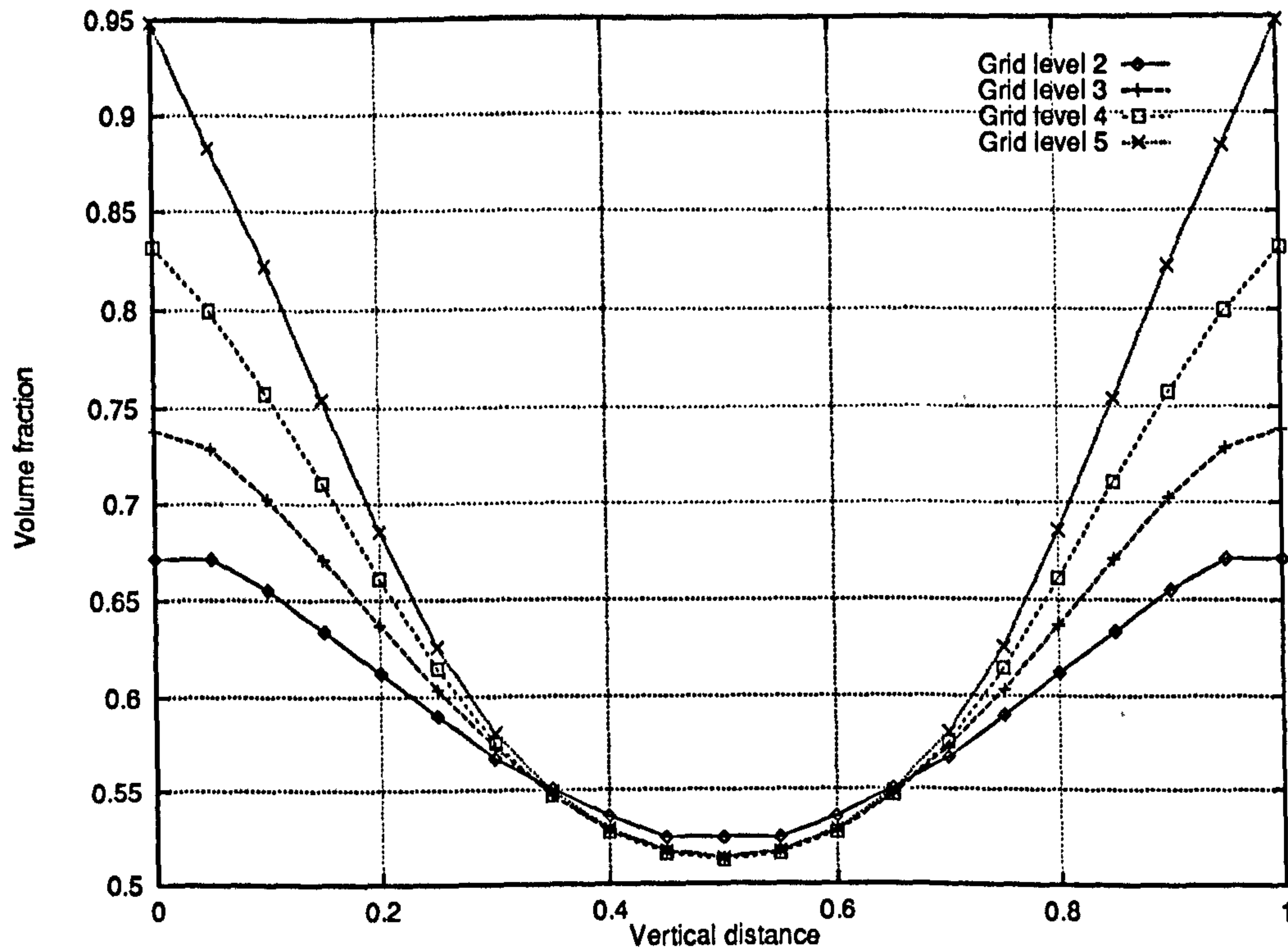


Figure 5.46: Two-phase channel flow – Grid independence for pang-multiphase results – Volume fraction profiles along the line $x = 2.5$ (Phase 1)

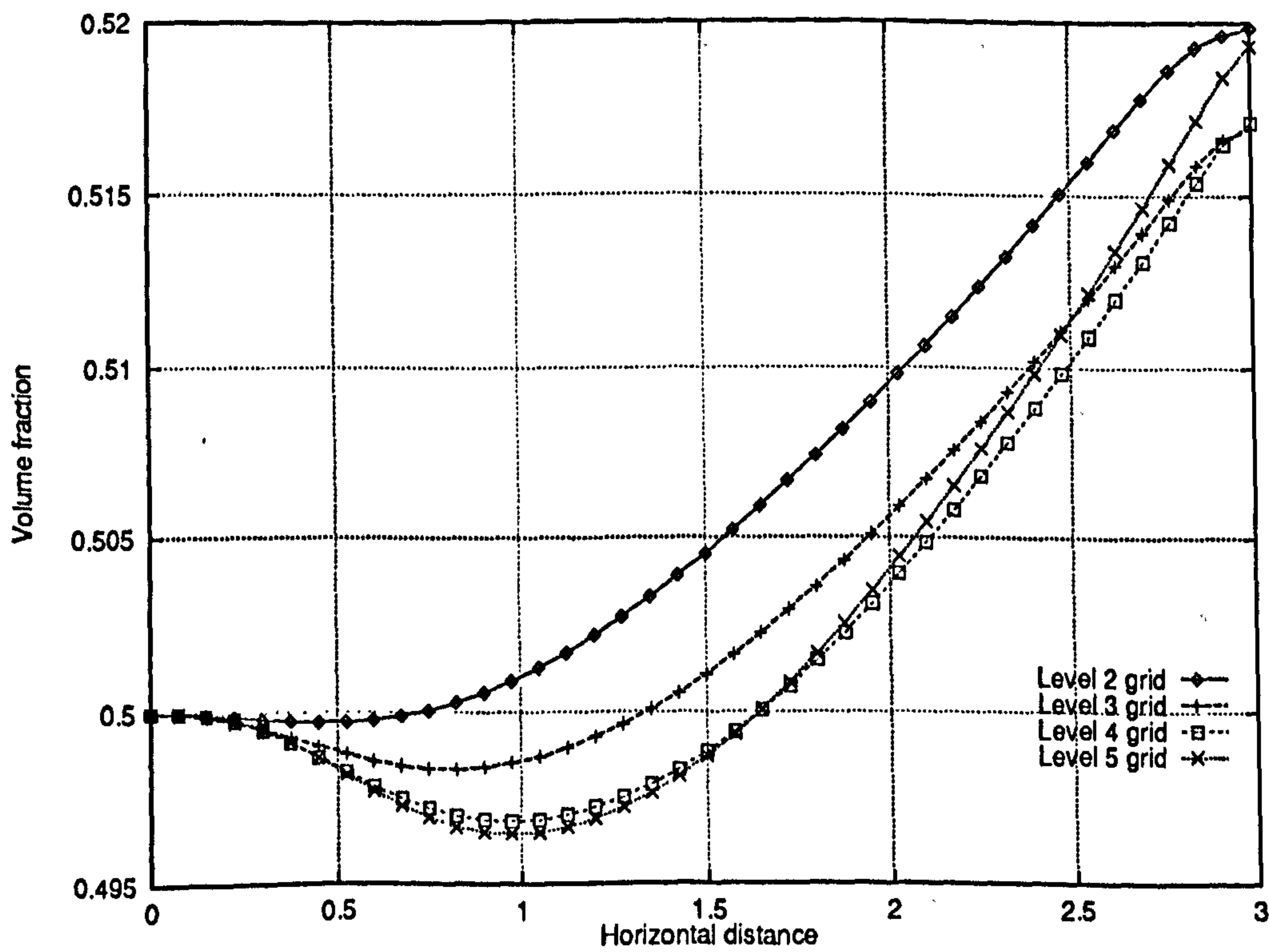


Figure 5.47: Two-phase channel flow – Grid independence for CFX 4.1 results – Volume fraction profiles along the line $x = 0.5$ (Phase 1) – Compare with Figure 5.43

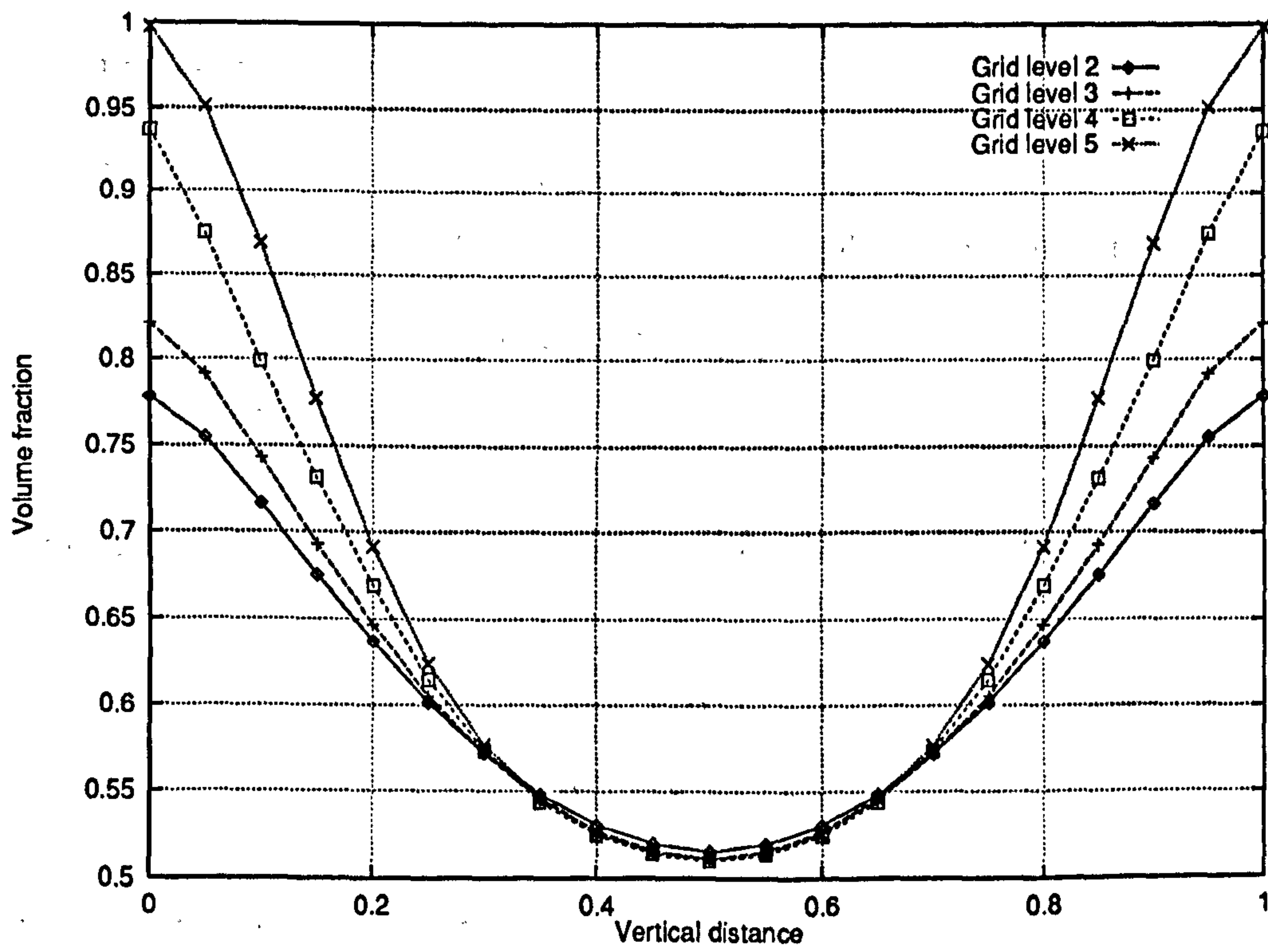


Figure 5.48: Two-phase channel flow – Grid independence for CFX 4.1 results – Volume fraction profiles along the line $x = 2.5$ (Phase 1) – Compare with Figure 5.46

Addition of Inter-Phase Momentum Transfer Terms In most multiphase flows, momentum and energy are transferred between phases as they interact. In some cases, mass can also be transferred, but here for the sake of simplicity this is ignored, and we only consider momentum transfers. Mathematically, inter-phase transfers are often modelled by *algebraic* source terms. These are preferred because they do not modify the nature of the differential operators on the left-hand-side of equation (3.63), a feature which would complicate the system even further.

A simple, but widely used, inter-phase momentum transfer model has been chosen: the *mixture model* (see [67, page 316]), so that comparisons with CFX 4.1 are easier. The mixture model is typical of more complex scenarios and other models could easily be implemented.

In the momentum equation 3.63, inter-phase transfers are modelled by terms of the form: ²

$$c_{\alpha\beta}(\mathbf{v}_\beta - \mathbf{v}_\alpha).$$

For the mixture model, $c_{\alpha\beta}$ is defined by equation (3.68):

$$c_{\alpha\beta} = c_{\beta\alpha} = \begin{cases} \frac{C_D}{d_{\alpha\beta}} \rho_{\alpha\beta} r_\alpha r_\beta |\mathbf{v}_\beta - \mathbf{v}_\alpha| & \text{if } \alpha \neq \beta \\ 0 & \text{otherwise} \end{cases}$$

For the results presented below, the following values for the drag coefficient and the inter-facial length have been chosen:

$$C_D = 1.0$$

$$d_{12} = d_{21} = 0.1.$$

Interphase momentum transfers have the effect of equalising phase velocities. Due to these transfers, the phase with less momentum is accelerated by the one with more momentum. Numerically, the addition of momentum transfer terms often has the effect of stabilising the computations. This is particularly true for more complex flows (see the multiphase T-junction problem, Section 5.3.2)

In the limit of large transfers coefficients $c_{\alpha\beta}$, we can consider that the phases share the same velocity field. The pang-multiphase solutions display this pattern clearly as Figures 5.49 to 5.52 show. Both the horizontal and vertical velocities fields differ far less than when momentum transfers are neglected (compare Figures 5.49, 5.50 and 5.51 with Figures 5.30 5.31 and 5.28 respectively). Furthermore, the observation of the volume fraction field (Figure 5.52) reveals that phase separation is not as rapid with inter-phase momentum transfers as without. This is not surprising since there is less relative motion between the phases.

²See equation (3.67).

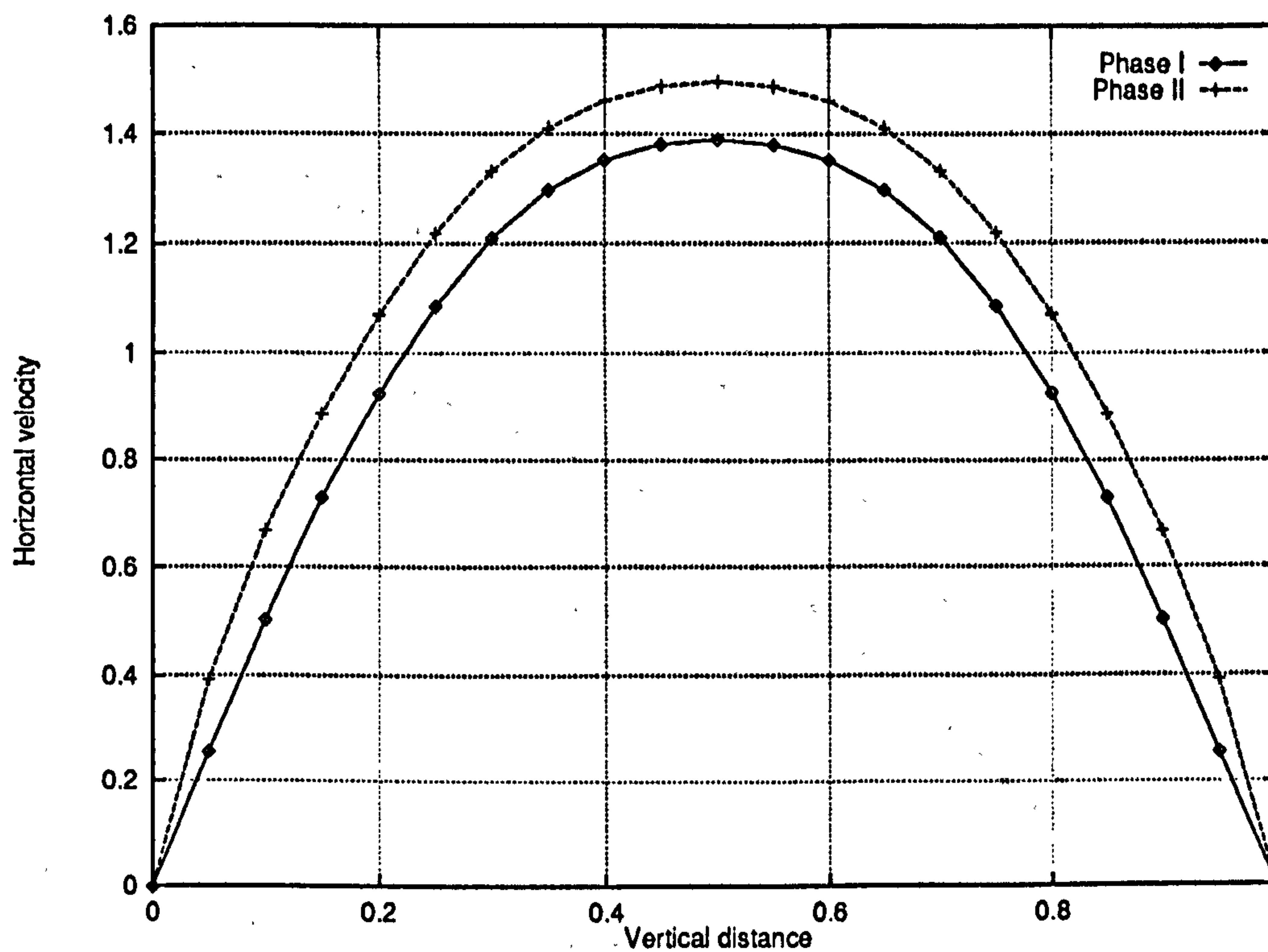


Figure 5.49: Horizontal velocity profile when inter-phase momentum transfer are modelled, along the line $x = 2.5$ - Compare with Figure 5.30

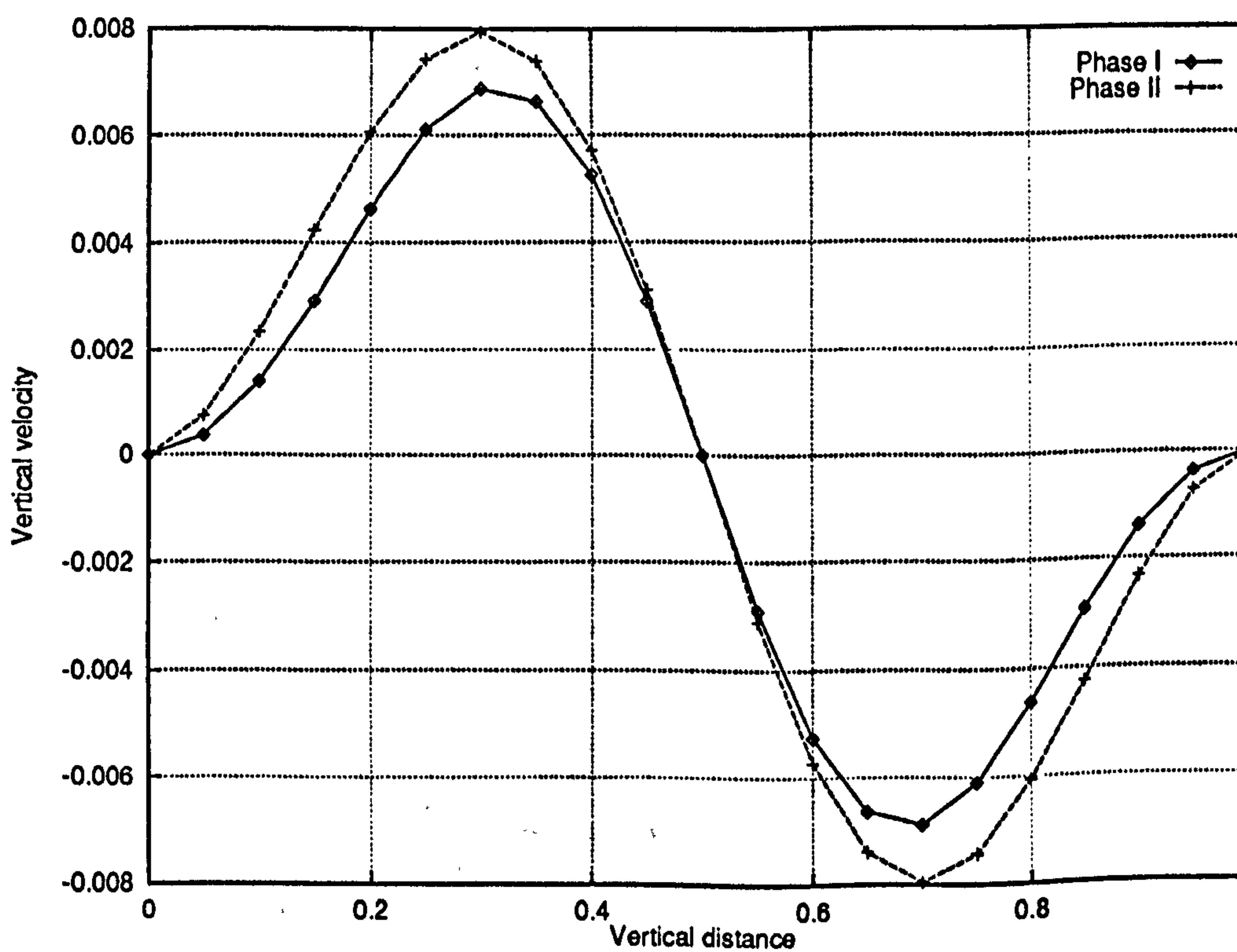


Figure 5.50: Vertical velocity profile when inter-phase momentum transfer are modelled, along the line $x = 2.5$ - Compare with Figure 5.31

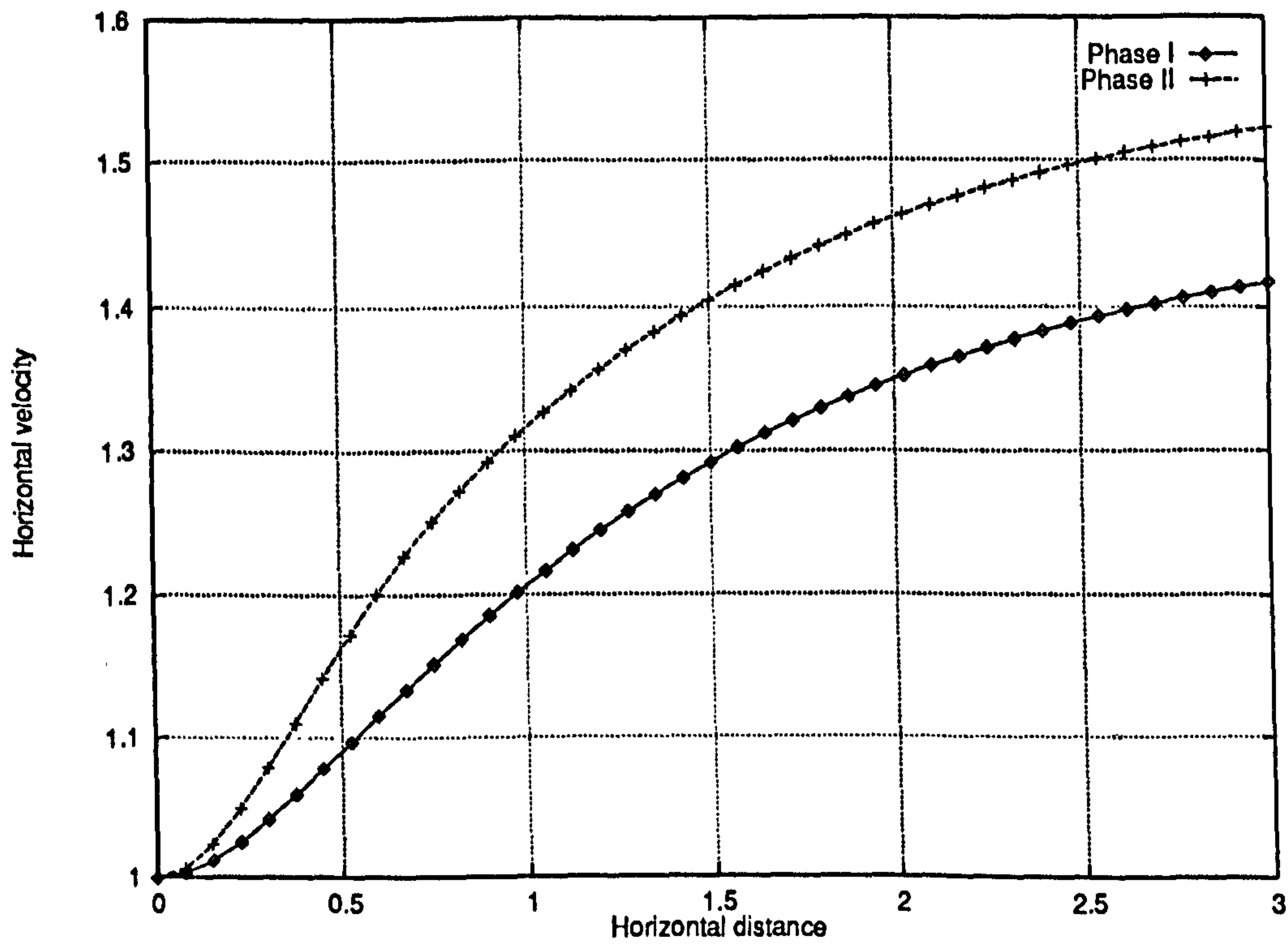


Figure 5.51: Horizontal velocity profile when inter-phase momentum transfer are modelled, along the line $y = 0.5$ - Compare with Figure 5.28

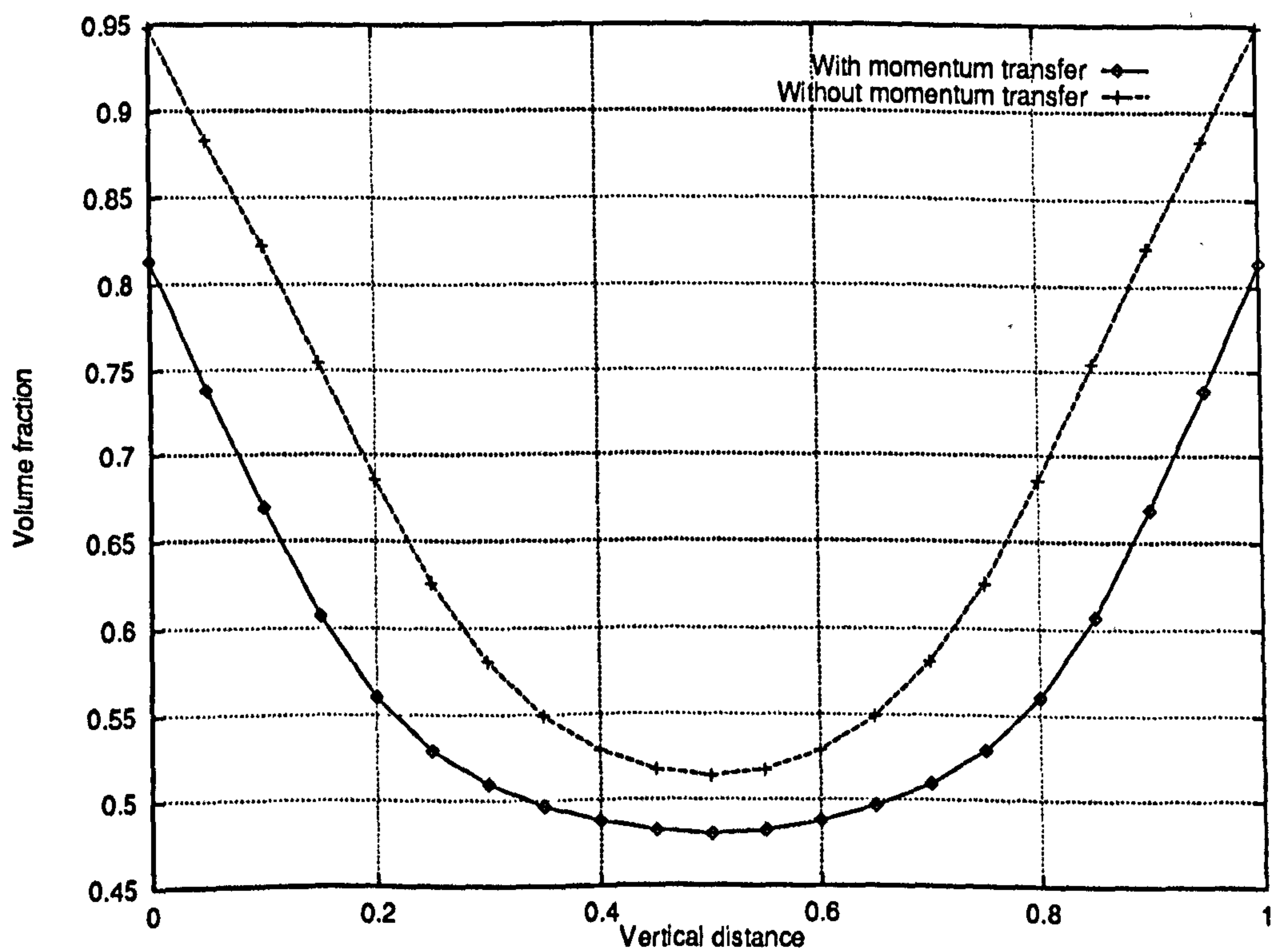


Figure 5.52: Comparison of the volume fraction profiles with and without inter-phase momentum transfer terms, along the line $x = 2.5$

5.3.2 Problem 2: Two-phase Flow Through a T-Junction with Two Inlets

Problem Description In this second multiphase test problem, a two-phase flow through a T-junction with two inlets is considered. The aim is to demonstrate the correctness of the code for relatively complex flow patterns. When convergence rates are considered in Section 5.4, this test case will also prove that by using a quasi-Newton coupled multigrid solver, it is possible to *efficiently* simulate complex two-phase flows. The simulation of multi-phase flows through T-junctions is challenging due to the large amount of phase separation which usually occurs. The two inlets case is simpler than the two outlets configuration because the extent of recirculation zone is minimised. Recirculation zones are specifically addressed in the final test case (Section 5.3.3), which concerns a two-phase flow through a backward-facing step.

The geometry is identical to the single phase case (see Section 5.2.2). The fluids, which are different, have the following physical properties³ :

Reynolds Number	viscosity	density
$Re_1 = 100$	$\mu_1 = 0.01$	$\rho_1 = 1.0$
$Re_2 = 75$	$\mu_2 = 0.0066$	$\rho_2 = 0.5$

For this test case, we allow inter-phase momentum transfers and use the mixture model. As before, and for the sake of simplicity, we let both the drag coefficient and the inter-facial length be constants rather than dependent on the state of the flow. The values are:

$$C_D = 1 \quad d_{\alpha\beta} = 0.1$$

If transfer momentum terms are neglected, it has been observed that convergence is not achieved. This fact again confirms that the source terms have a stabilising effect on the computations.

The coarsest grid is the same as the one used for the single phase flow case (see Figure 5.9). As in the single phase case, the length of the exit section is adequate (Figure 5.57) but modelling a longer section would have been advisable to ensure that the boundary conditions did not affect the results.

Flow Solution and Discussion The features of the solution agree with what could be expected intuitively. Figures 5.53, 5.54 and 5.115 show the streamlines for each phase and the volume fraction map throughout the computational domain, based on the pang-multiphase solution. Note that it is only the phase with less

³The Reynolds numbers are based on the channel width

inertia which recirculates downstream of the junction, so that the problem is still well posed in a mathematical sense.

Some solution profiles are shown in Figures 5.55 to 5.63. As expected, the flow pattern is much more complicated than in the case of the two-phase channel flow. In each of the incoming branches, the phases separate toward a layered flow pattern consistent with the solutions observed for the two-phase channel flow: the denser phase tends to flow in the outer part of the channel. See Figure 5.115.

However, when the junction is reached, the phase with less inertia is forced by the other phase toward the bottom of the exit section of the junction and the flow becomes stratified, as Figure 5.63 clearly shows. Note that gravity is *not* included in the model. As Figure 5.58 shows, there is a large pressure drop at the junction which results from the fact that a lot of energy is lost in the turning and mixing process.

When the solutions are compared with those provided by CFX 4.1, a very good measure of agreement is observed, although this test case apparently highlighted a shortcoming of CFX 4.1 in the application of mass flow boundary condition at the outlet [111] (see [67, pages 145–146]). That is, spurious patterns are generated at the outlet (see Figures 5.55, 5.56 and 5.57). When mass flow boundary conditions at the outlet are replaced by pressure boundary conditions (this is made possible by the absence of body forces), the flow at the outlet is correctly modelled (Figures 5.64 to 5.66).

The pressure drop along the horizontal section of the channel is shown in Figure 5.58. Due to the pressure singularity which exists for incompressible flows, the `pamg-multiphase` and CFX 4.1 pressure fields are different but equivalent, as shifting by a suitable amount reveals (Figure 5.59).

Figure 5.67 shows contours of the volume fraction field for phase 1 generated from the CFX 4.1 solutions with pressure boundary conditions. These agree with the `pamg-multiphase` solutions (see Figure 5.115) and allow a good visualisation of the flow structure.

The `pamg-multiphase` solutions obtained on grids at level 3 and 4 have been compared and in sharp contrast to the solutions obtained for the two-phase channel flow (Section 5.3.1), they are significantly more grid independent (see Figures 5.68 to 5.81). In regions where the phases are almost completely separated, the velocity field of the minority phase may take different values on different grids. See Figure 5.61 for an example. This is acceptable since the volume fractions are close to zero and consequently, the amount of transported momentum is very small (see Figure 5.62).

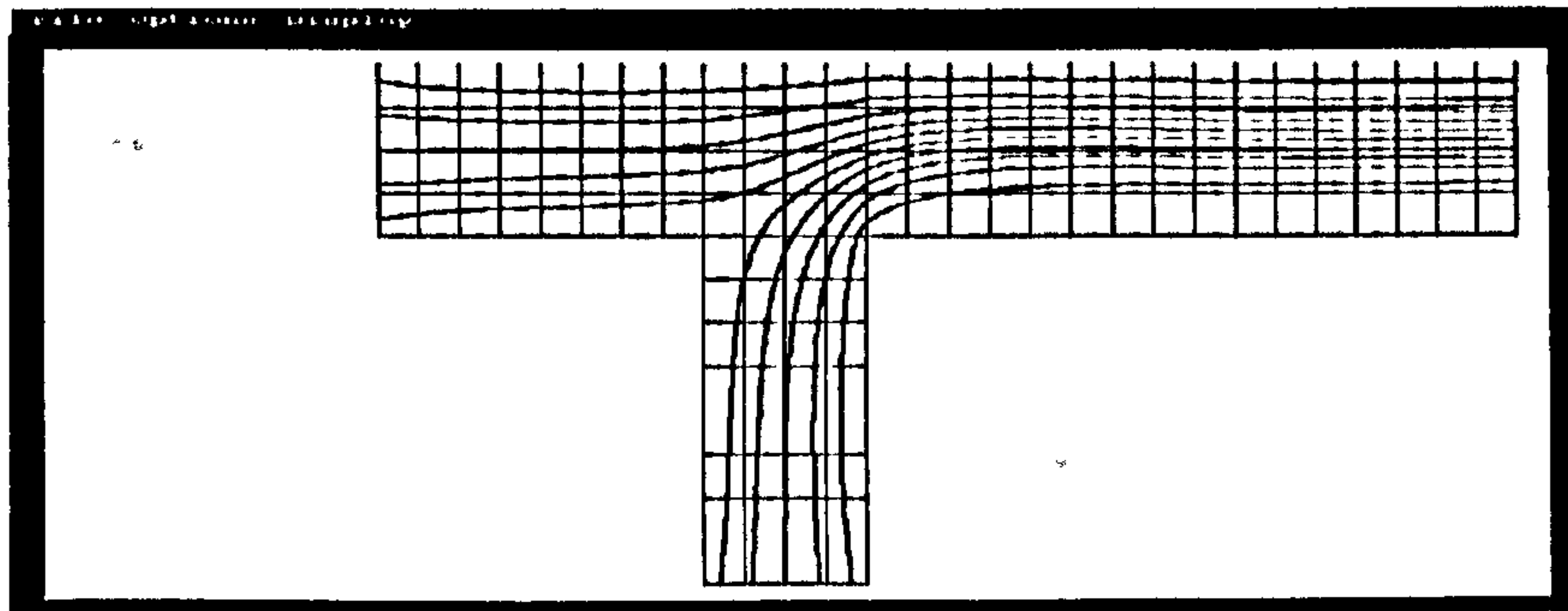


Figure 5.53: Multiphase T-junction problem – Streamlines for phase 1

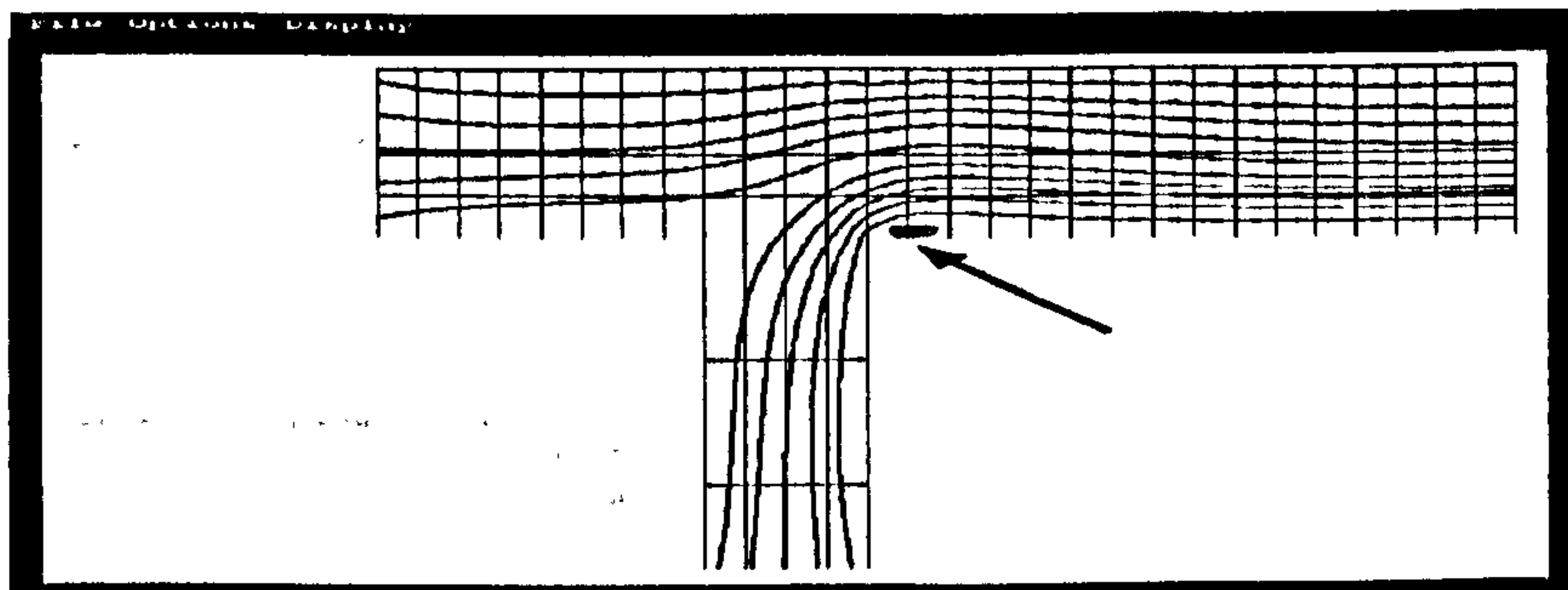


Figure 5.54: Multiphase T-junction problem – Streamlines for phase 2 – Note the recirculation region

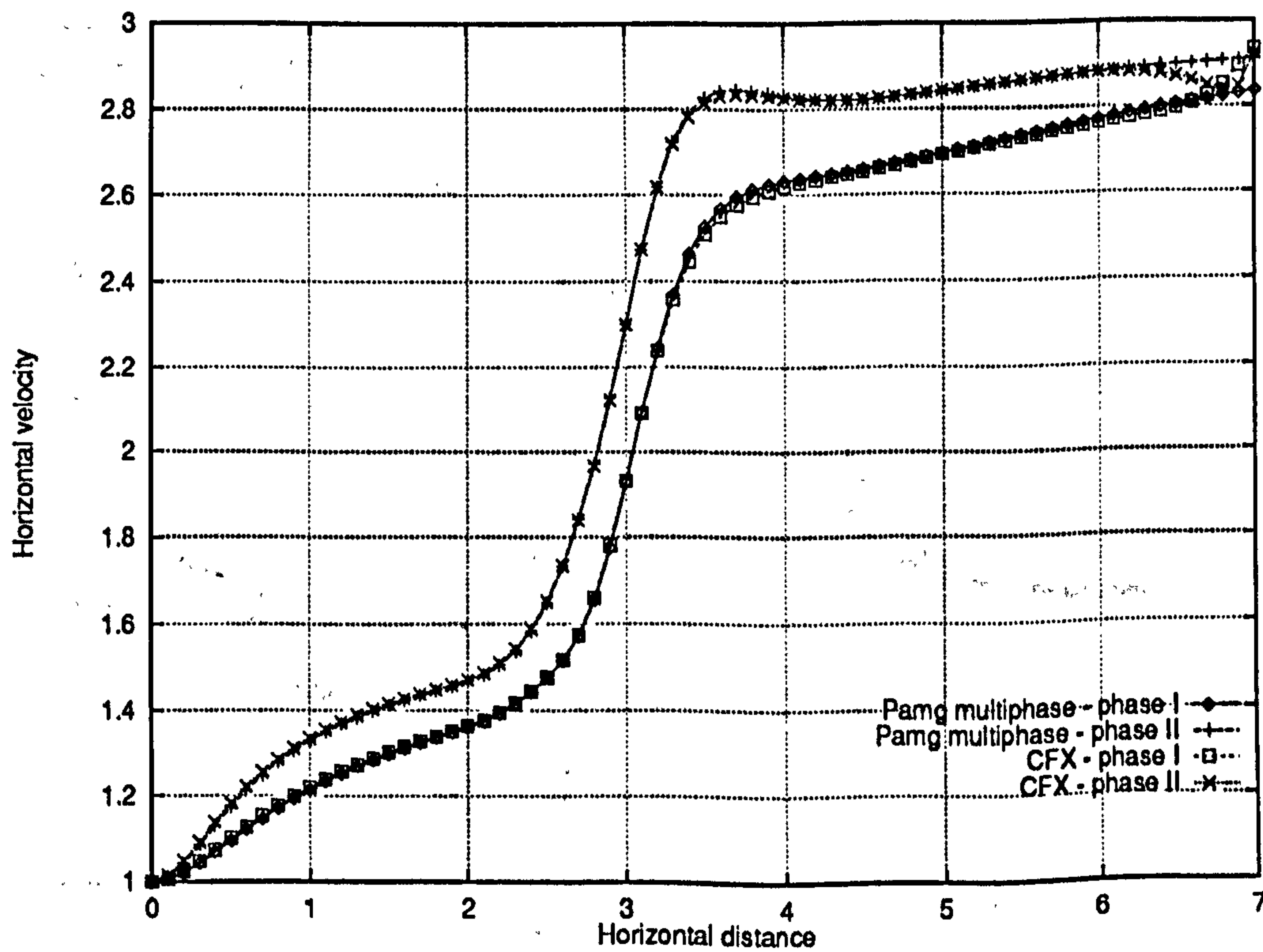


Figure 5.55: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ – Note the problems experienced by CFX 4.1 at the outlet, see below Figure 5.56

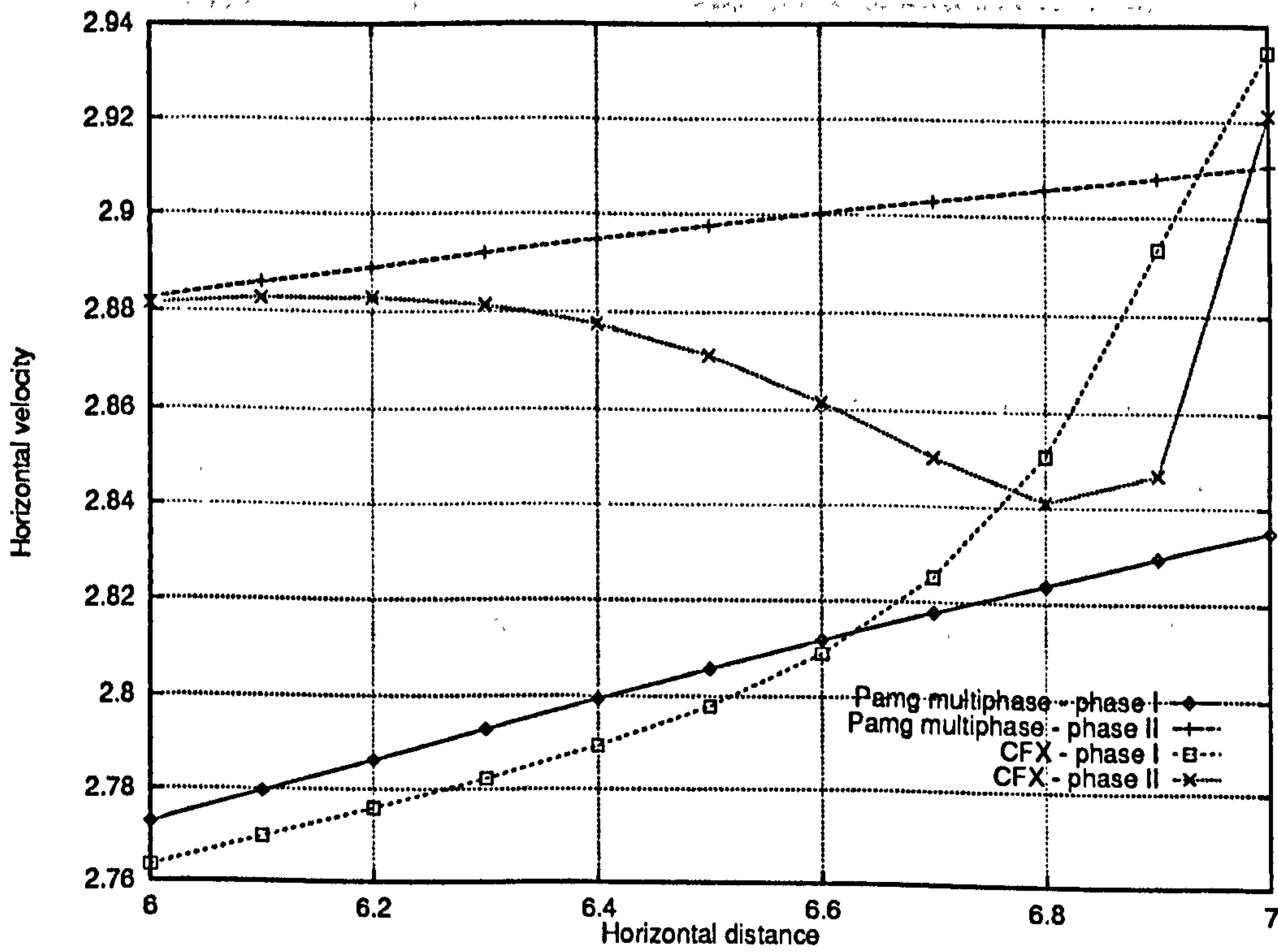


Figure 5.56: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $y = 0.5$ (near the outlet – $6 \leq x \leq 7$) showing the inaccurate solution generated by CFX 4.1

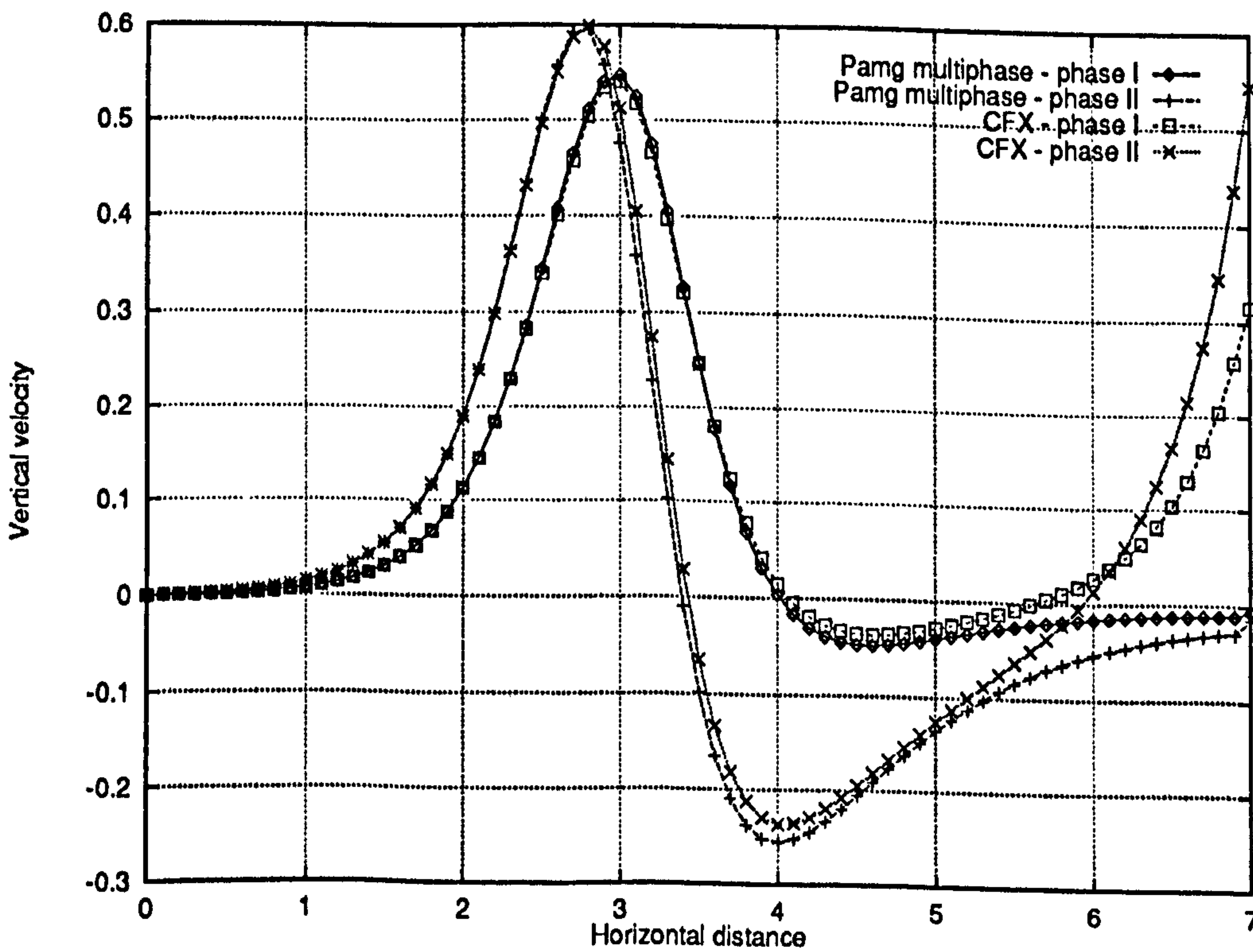


Figure 5.57: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $y = 0.5$ – Note again the poor quality of the CFX 4.1 solution at the outlet

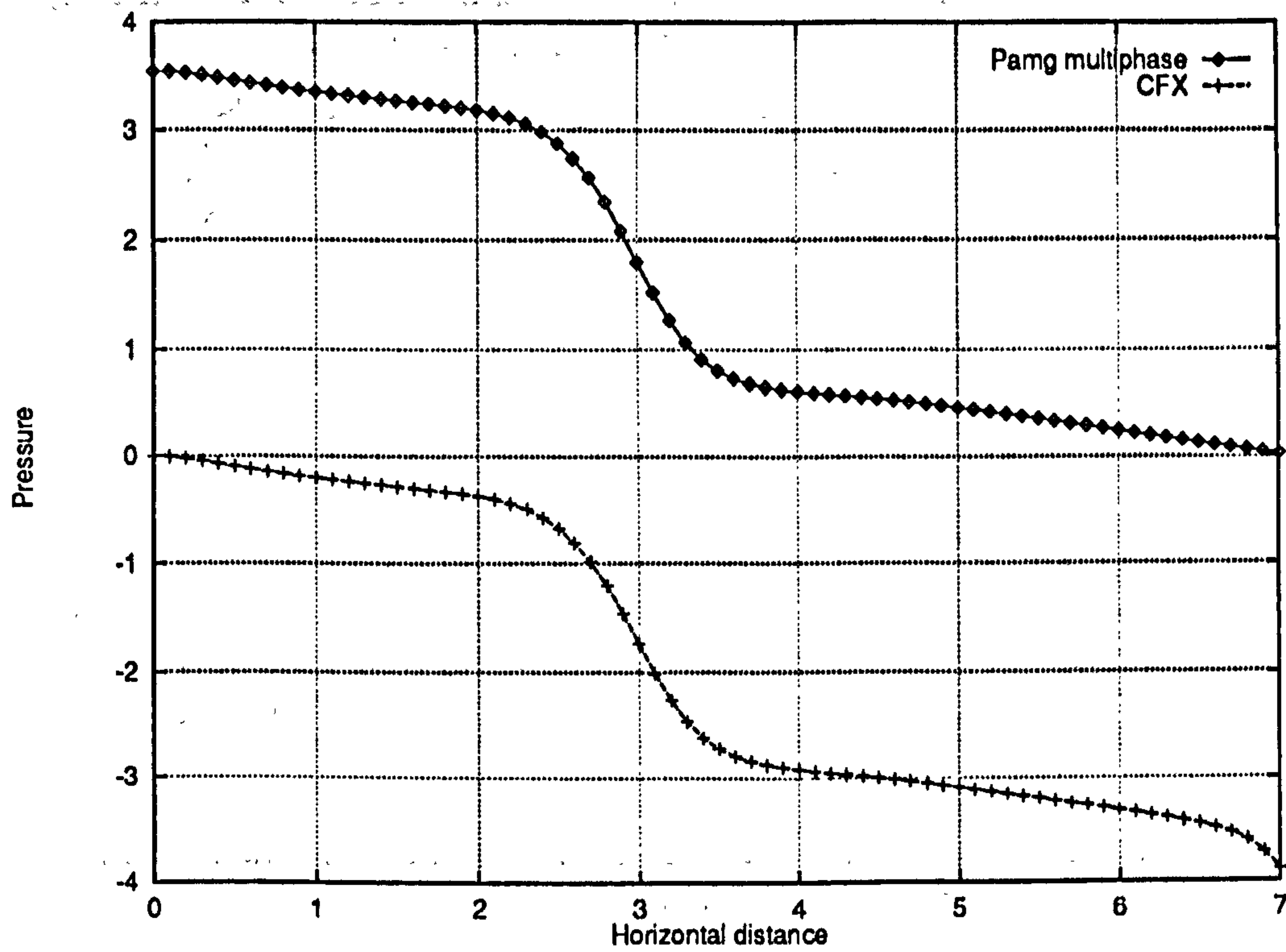


Figure 5.58: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$

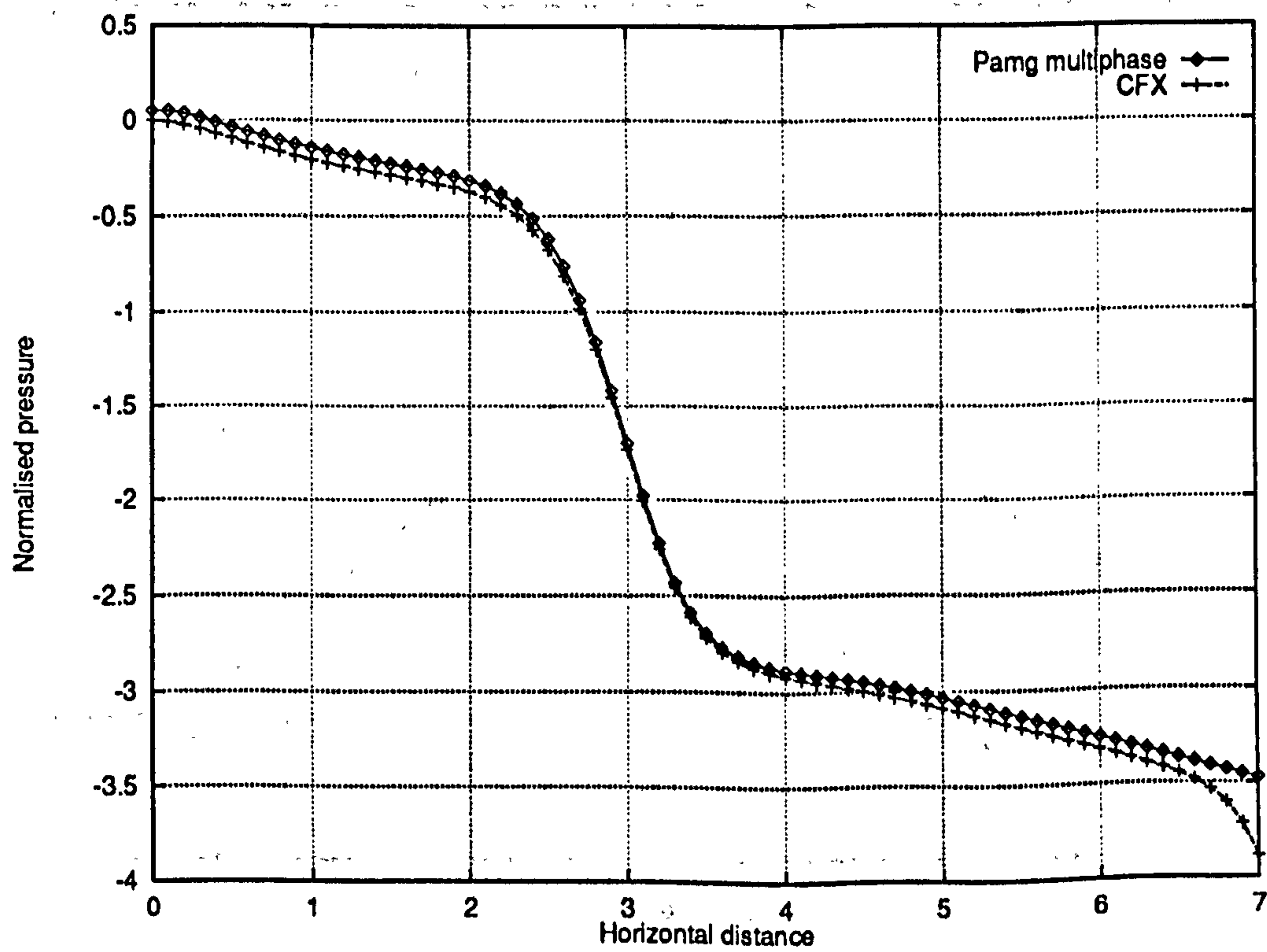


Figure 5.59: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Pressure profiles along the line $y = 0.5$ after shifting

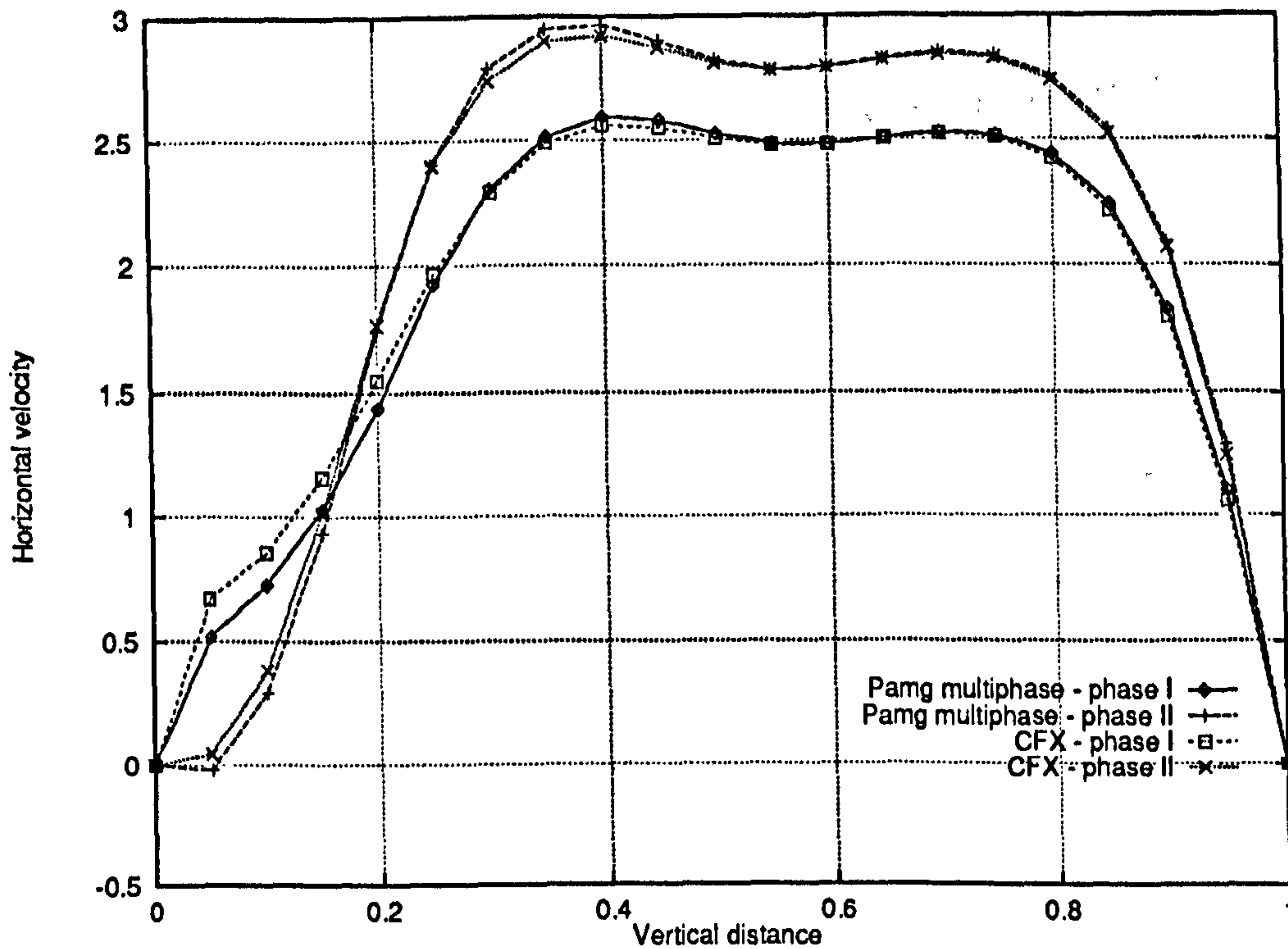


Figure 5.60: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Horizontal velocity profiles along the line $x = 3.5$, just downstream of the recirculation region

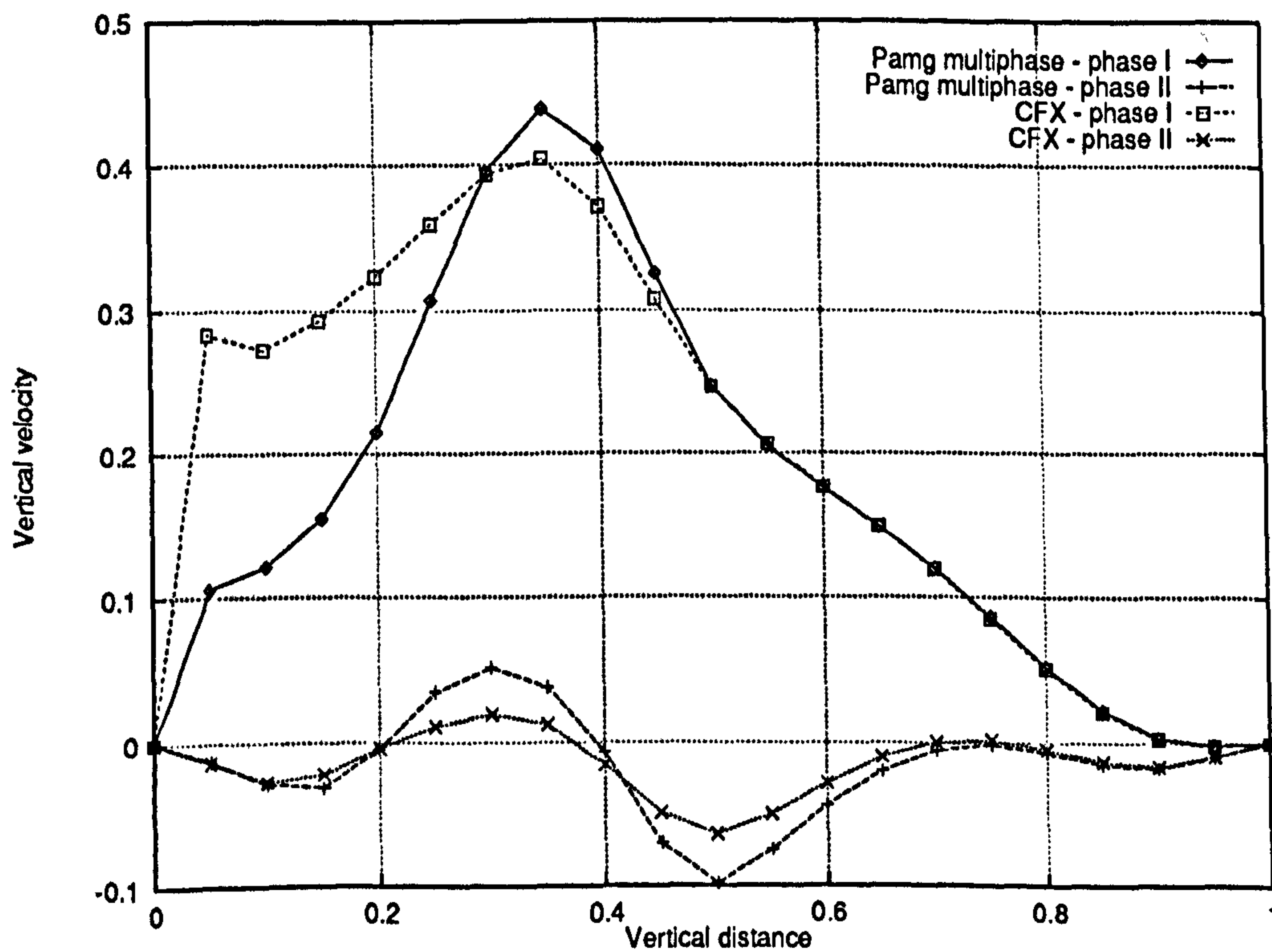


Figure 5.61: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical velocity profiles along the line $x = 3.5$, just downstream of the recirculation region – See Figure 5.62

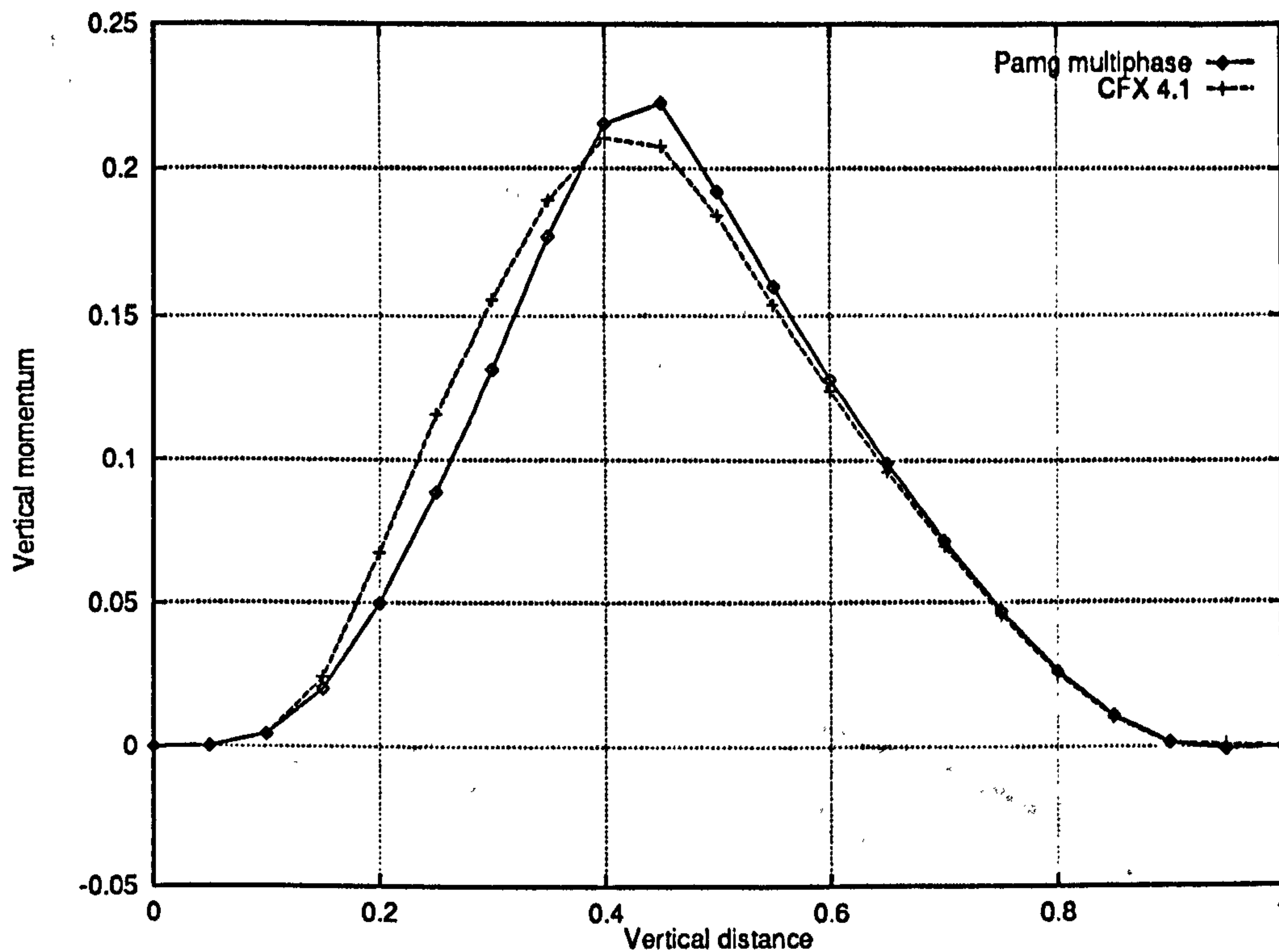


Figure 5.62: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Vertical momentum profiles along the line $x = 3.5$, just downstream of the recirculation region – See Figure 5.61

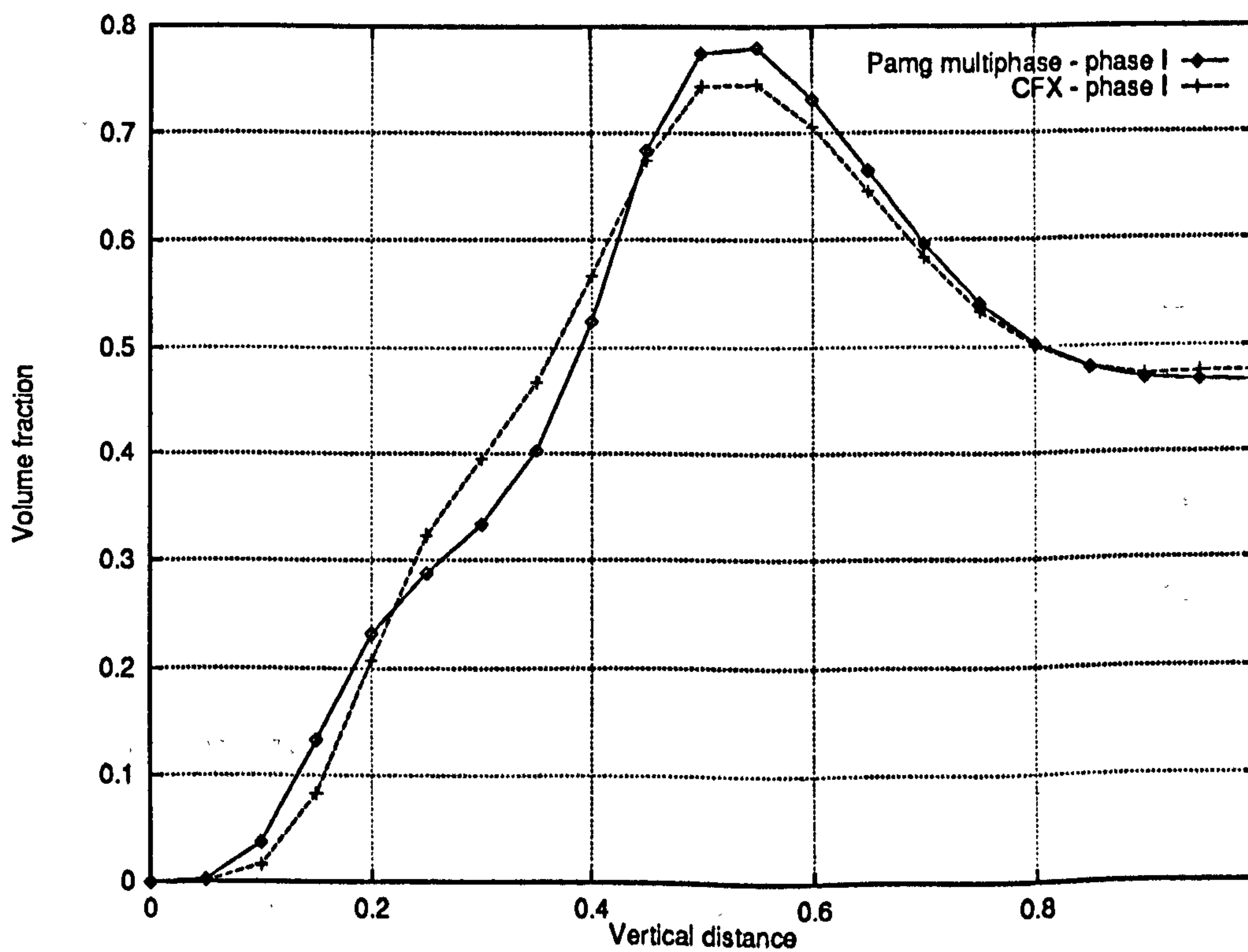


Figure 5.63: Multiphase T-junction problem – Comparison of pamg-multiphase and CFX 4.1 results – Volume fraction profiles along the line $x = 3.5$, just downstream of the recirculation region for phase I (i.e. the denser phase)

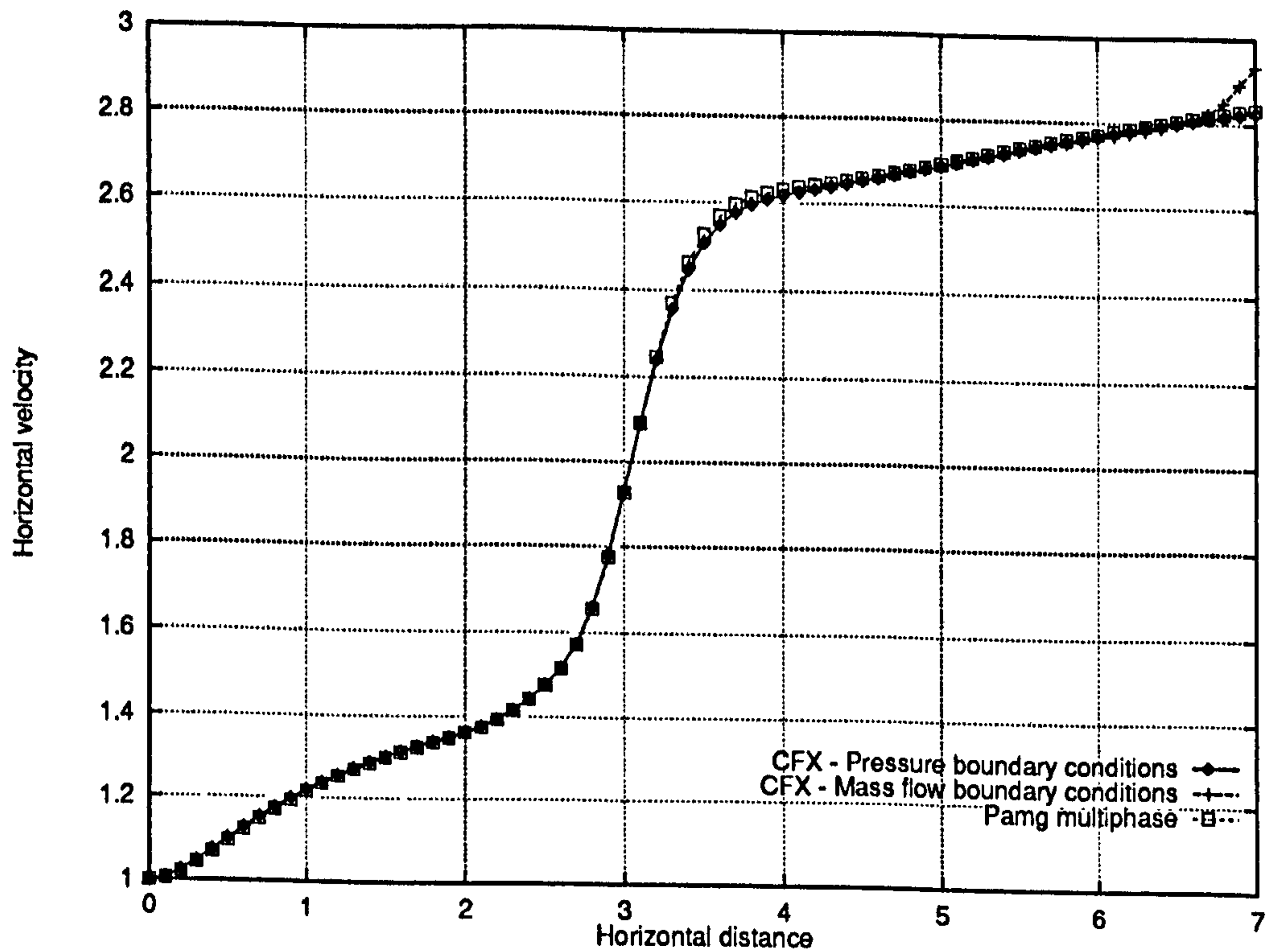


Figure 5.64: Multiphase T-junction problem – CFX 4.1 solution – Horizontal velocity profiles along the line $y = 0.5$ for different outlet boundary conditions, showing their effect on the quality of the solution

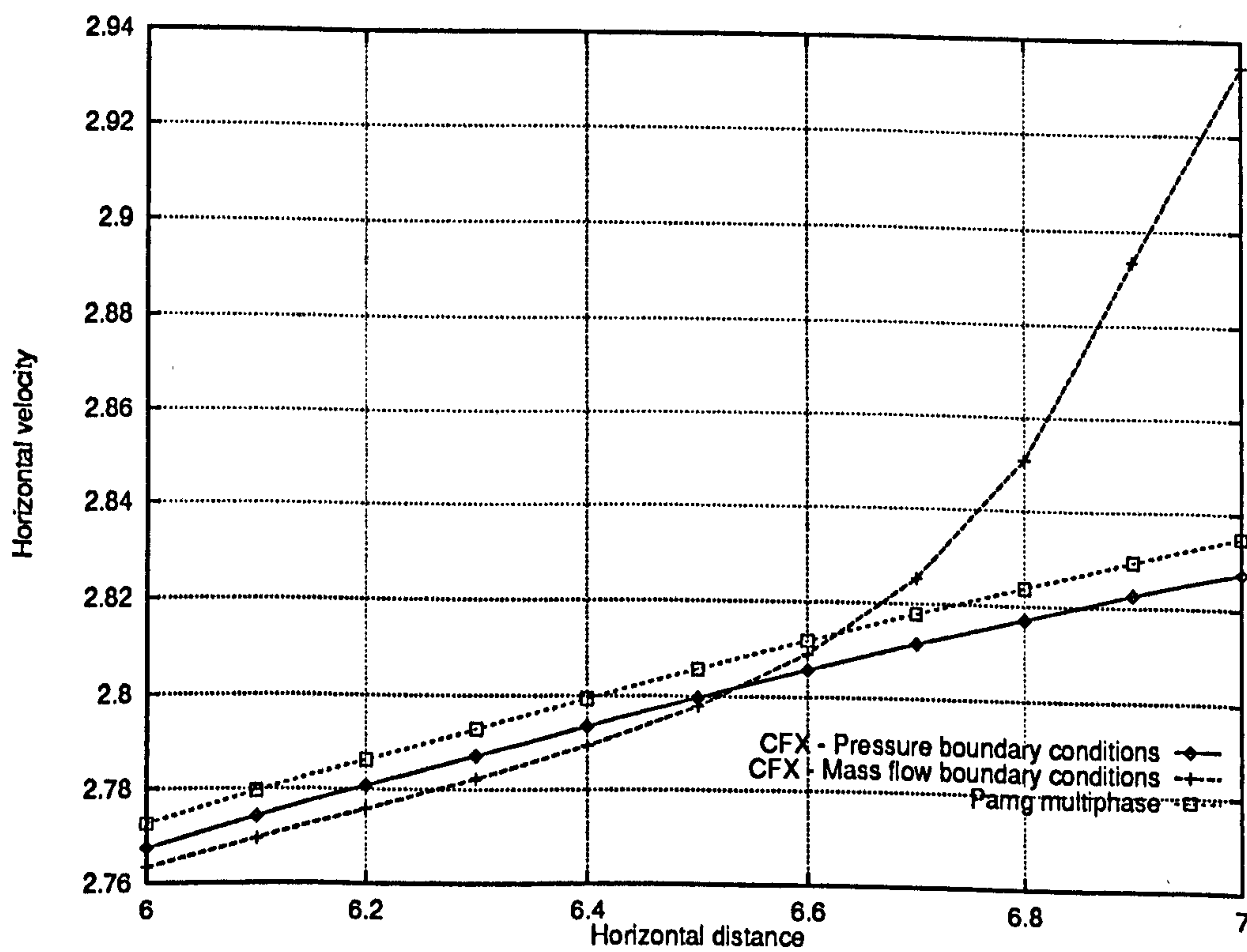


Figure 5.65: Multiphase T-junction problem – CFX 4.1 solution – Horizontal velocity profiles along the line $y = 0.5$ for different outlet boundary conditions – Detail of the solutions near the outlet showing the effect of the outlet boundary conditions

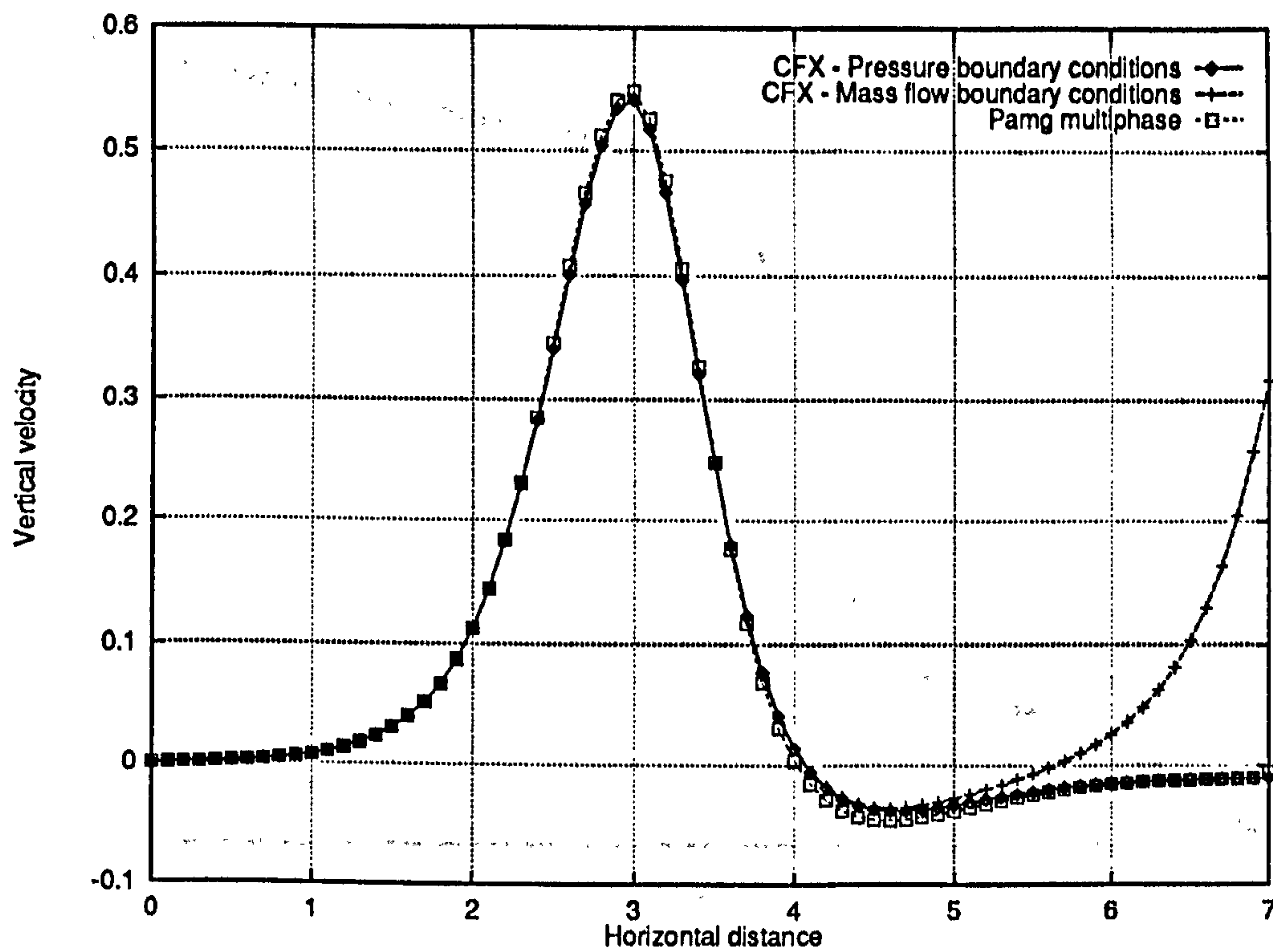


Figure 5.66: Multiphase T-junction problem – CFX 4.1 solution – Vertical velocity profile along the line $y = 0.5$ for different outlet boundary conditions showing their effect on the quality of the solution

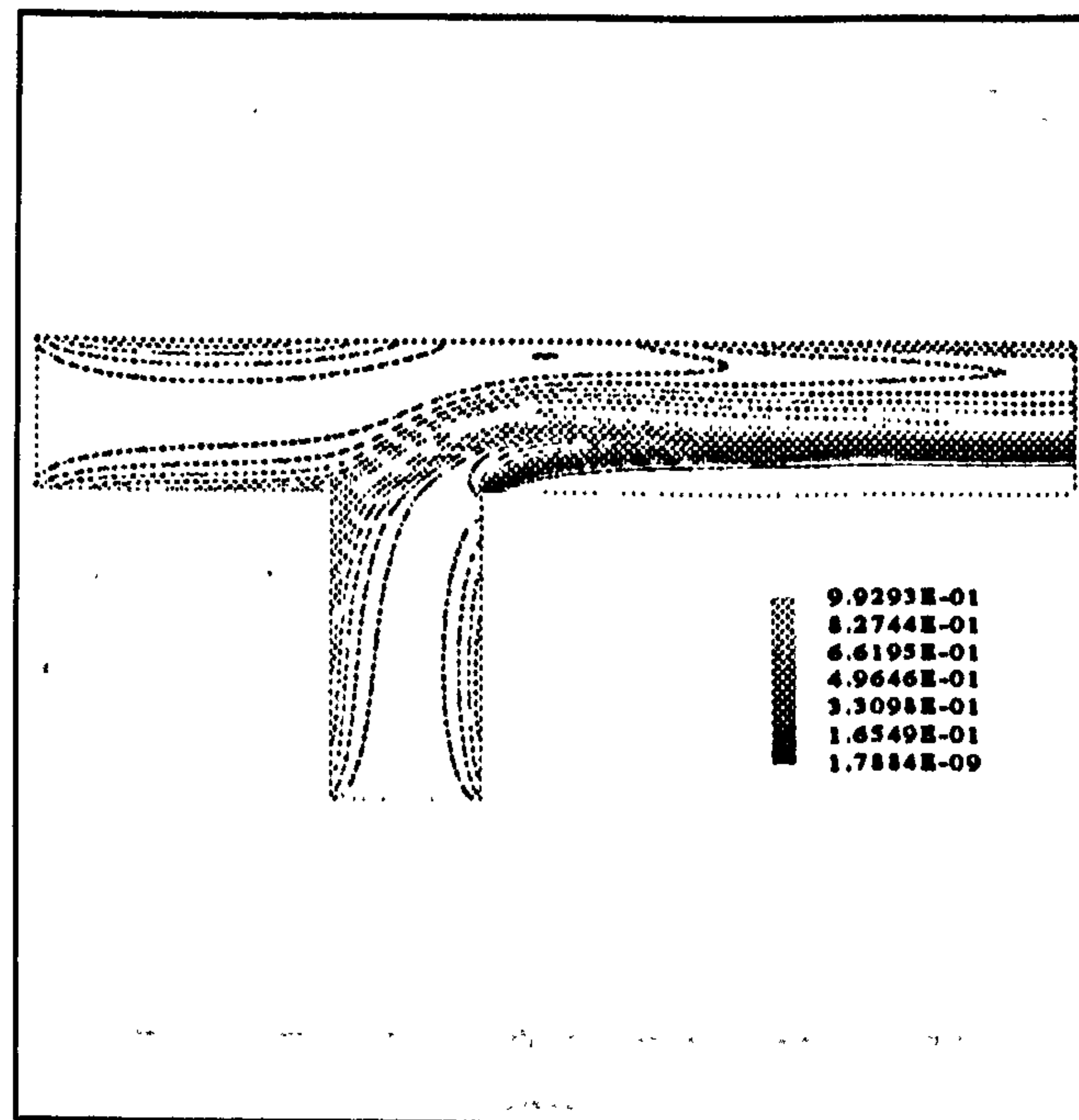


Figure 5.67: Multiphase T-junction problem – CFX 4.1 solution – Volume fraction contours for phase 1

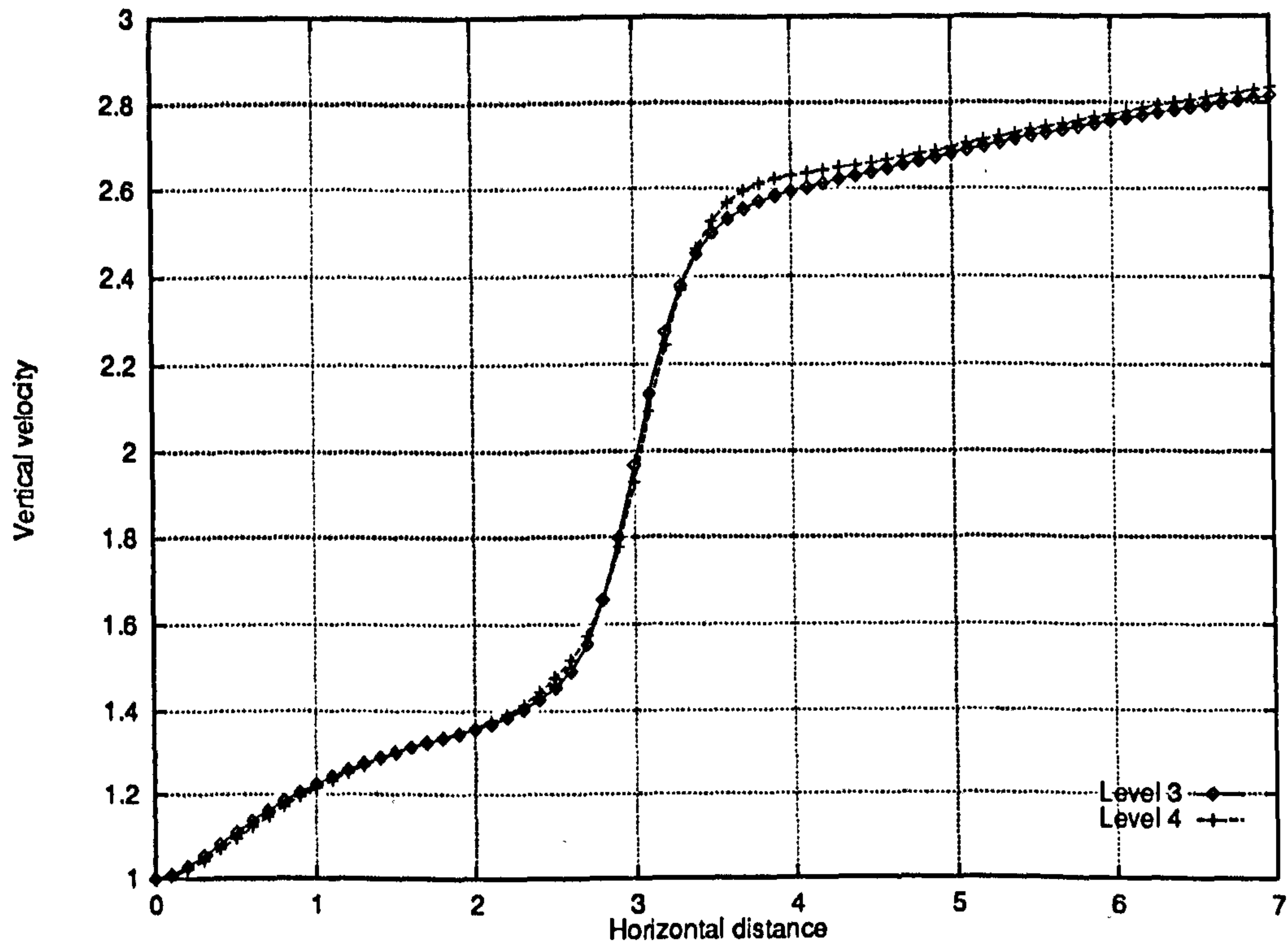


Figure 5.68: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Horizontal velocity profiles along the line $y = 0.5$ (Phase 1)

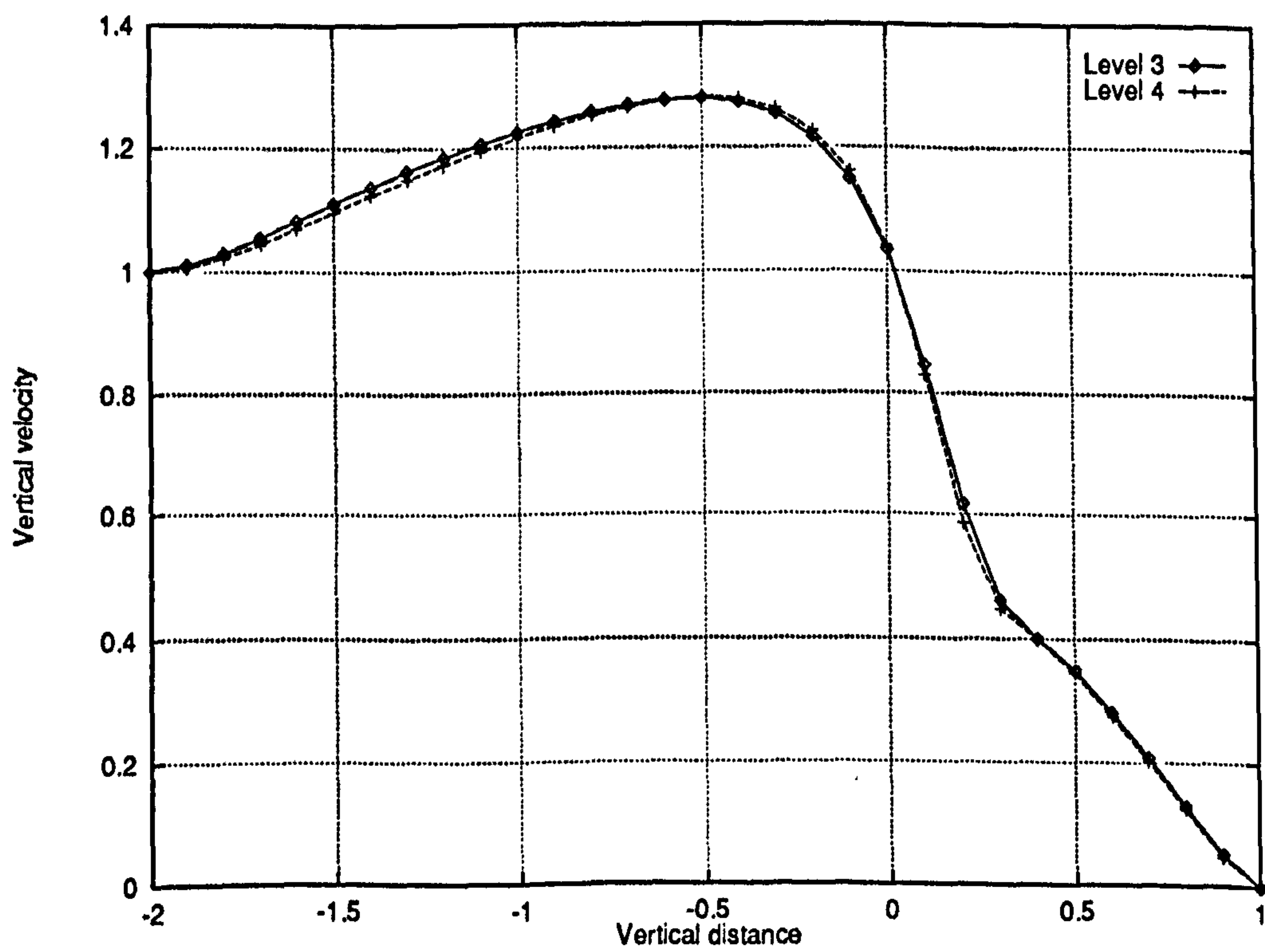


Figure 5.69: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Vertical velocity profiles along the line $x = 2.5$ (Phase I)

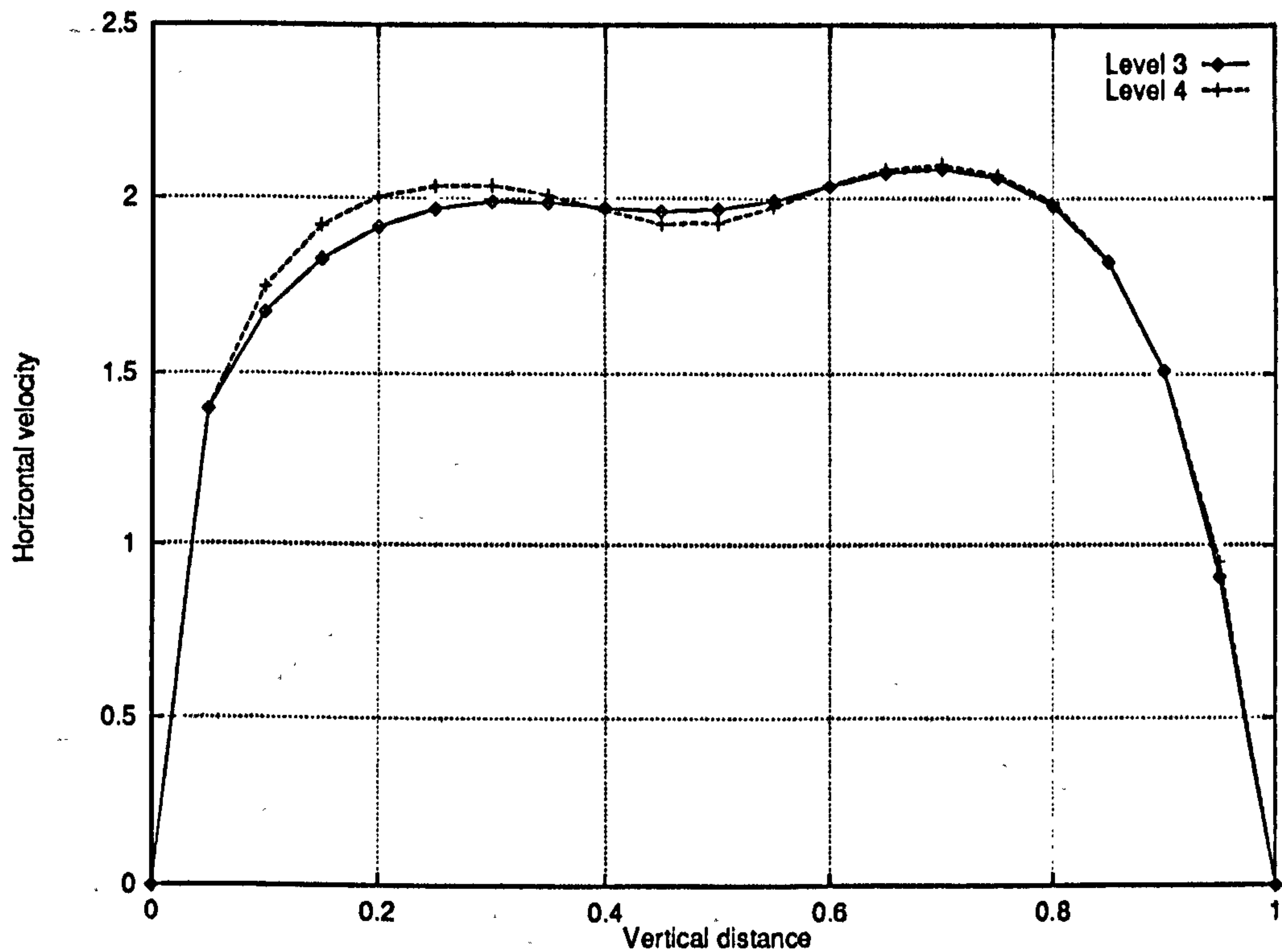


Figure 5.70: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Horizontal velocity profiles along the line $x = 3.0$ (Phase I)

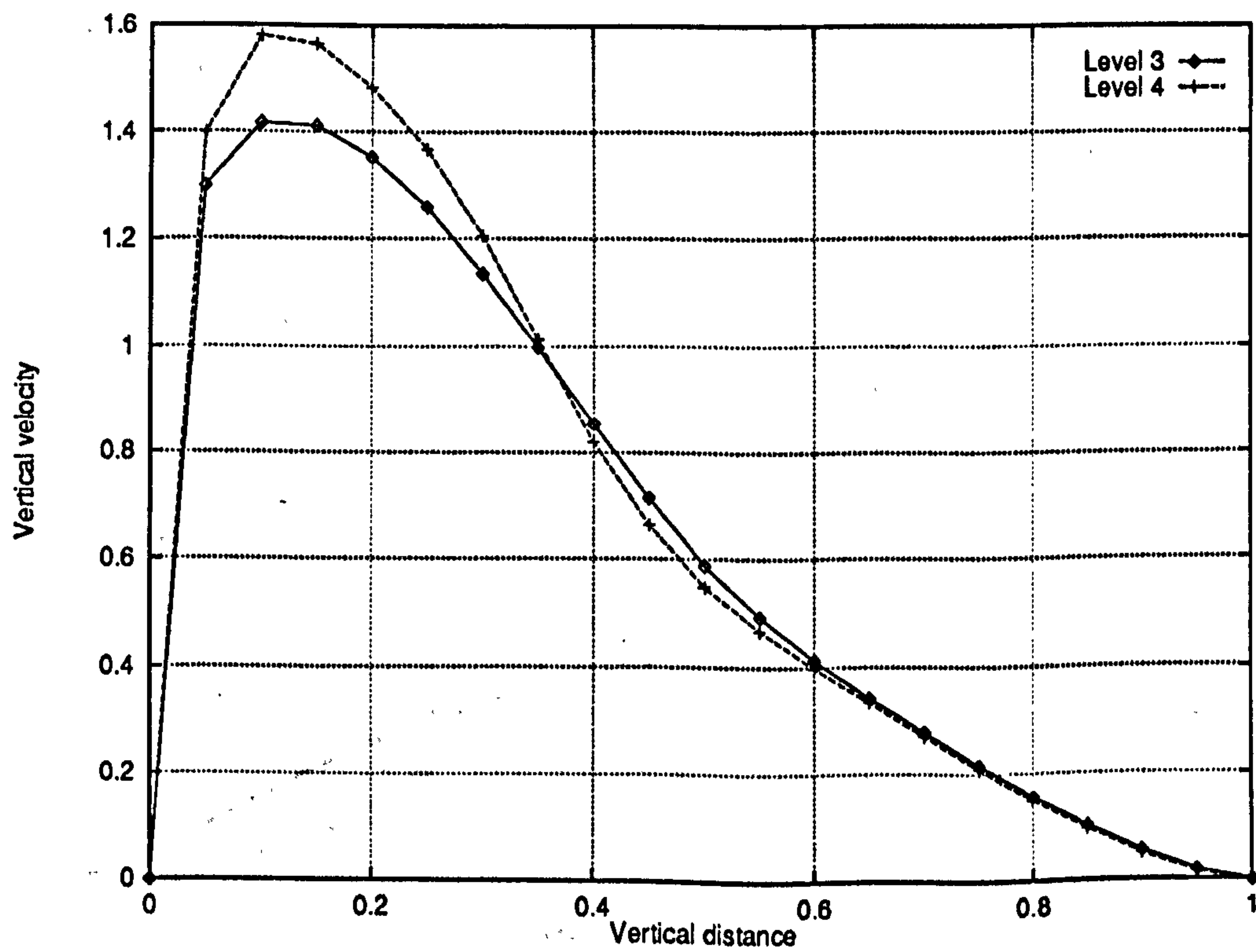


Figure 5.71: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Vertical velocity profiles along the line $x = 3.0$ (Phase I)

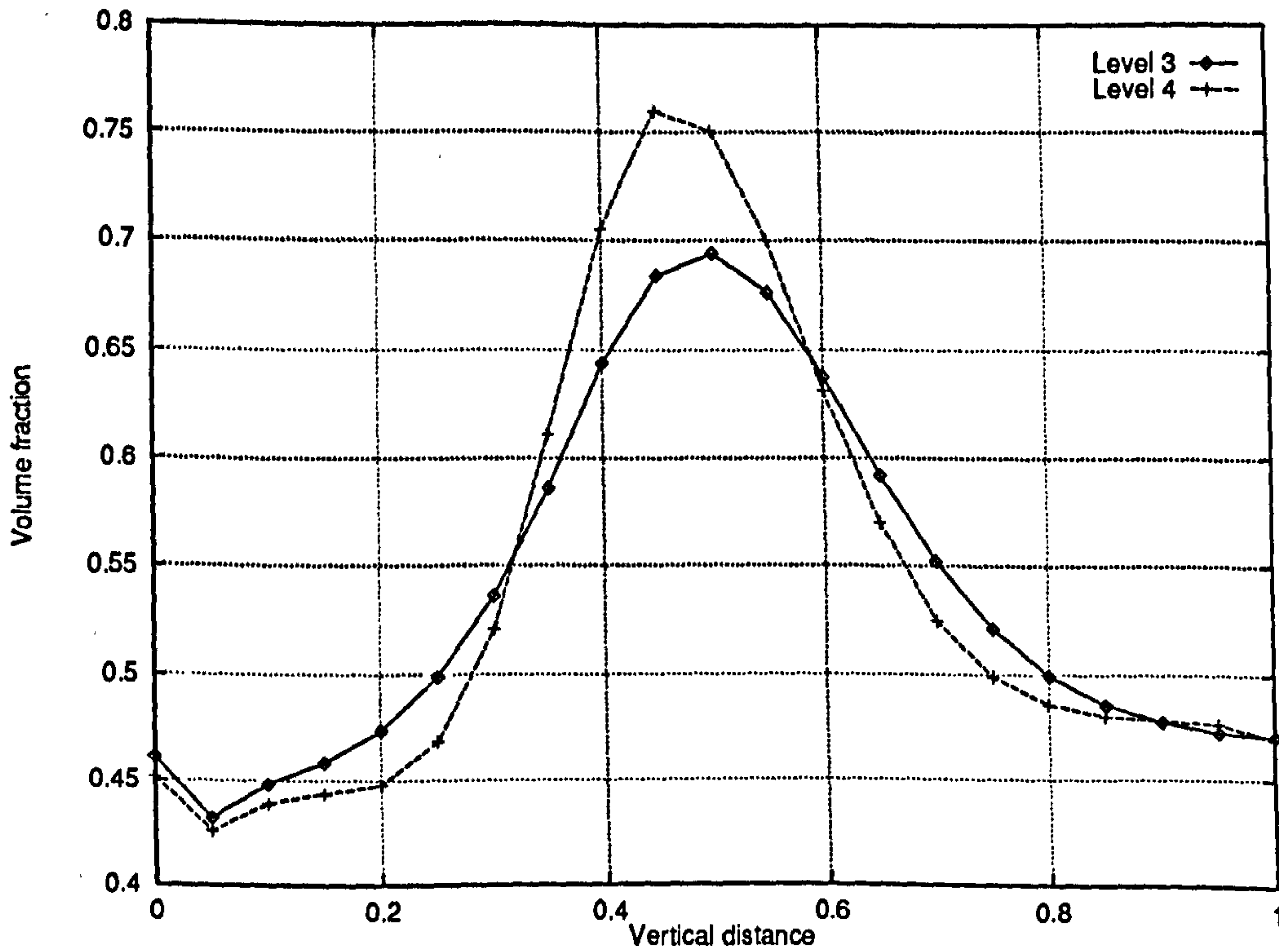


Figure 5.72: Multiphase T-junction problem - Grid independence for pang-multiphase results - Volume fraction profiles along the line $x = 3.0$ (Phase I)

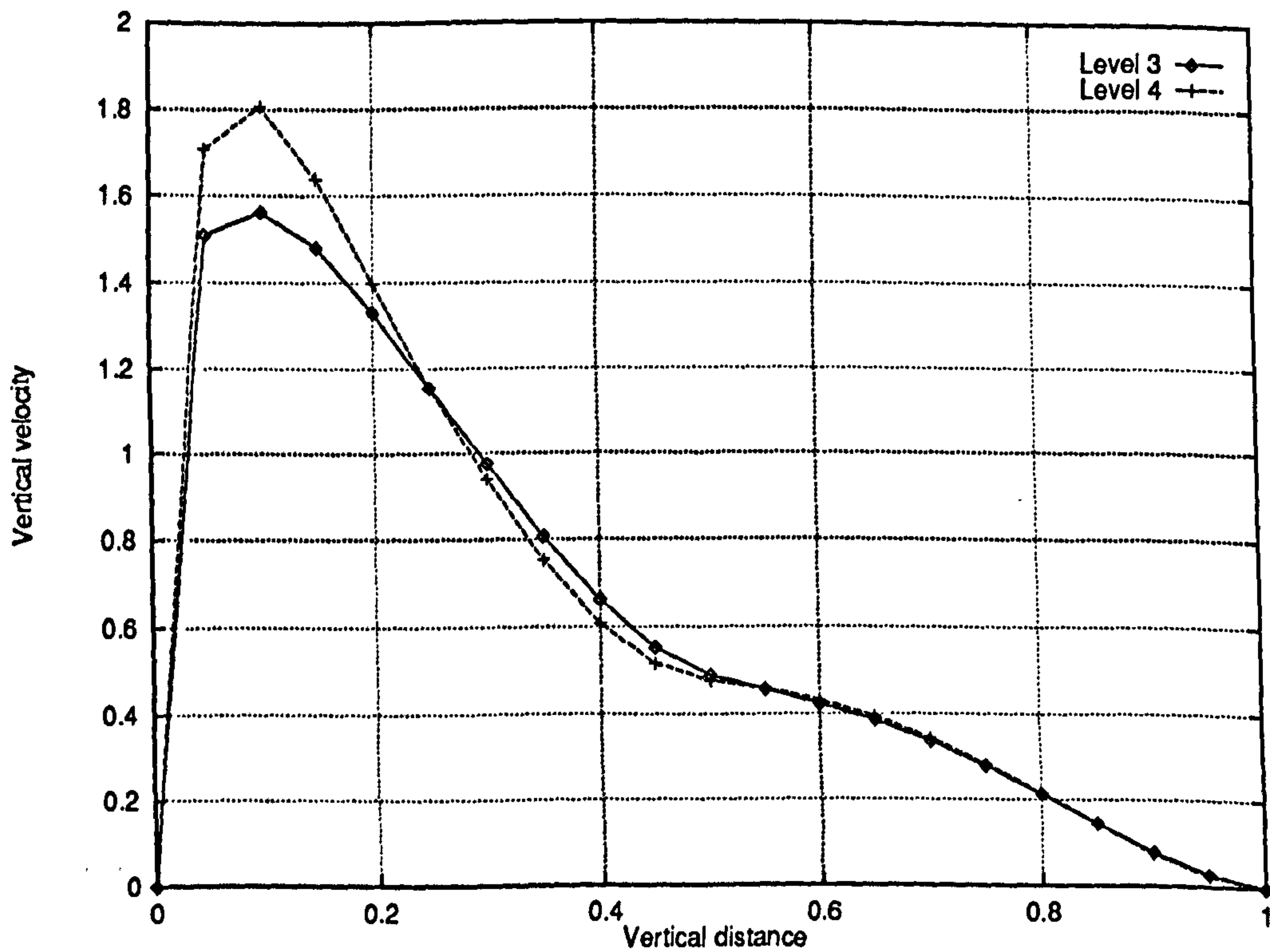


Figure 5.73: Multiphase T-junction problem - Grid independence for pang-multiphase results - Vertical velocity profiles along the line $x = 3.0$ (Phase II)

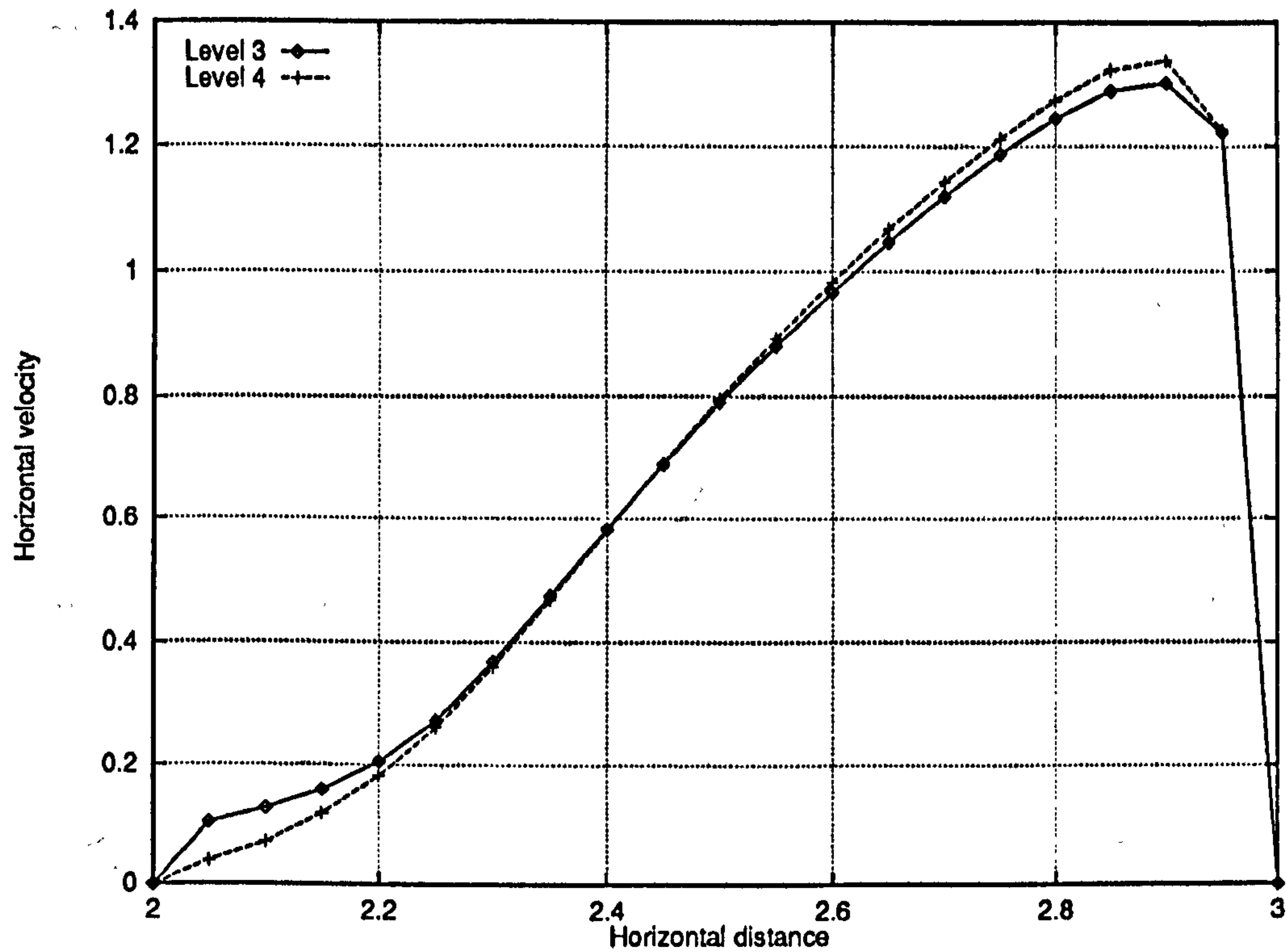


Figure 5.74: Multiphase T-junction problem - Grid independence for pang-multiphase results - Horizontal velocity profiles along the line $y = 0.0$ (Phase II)

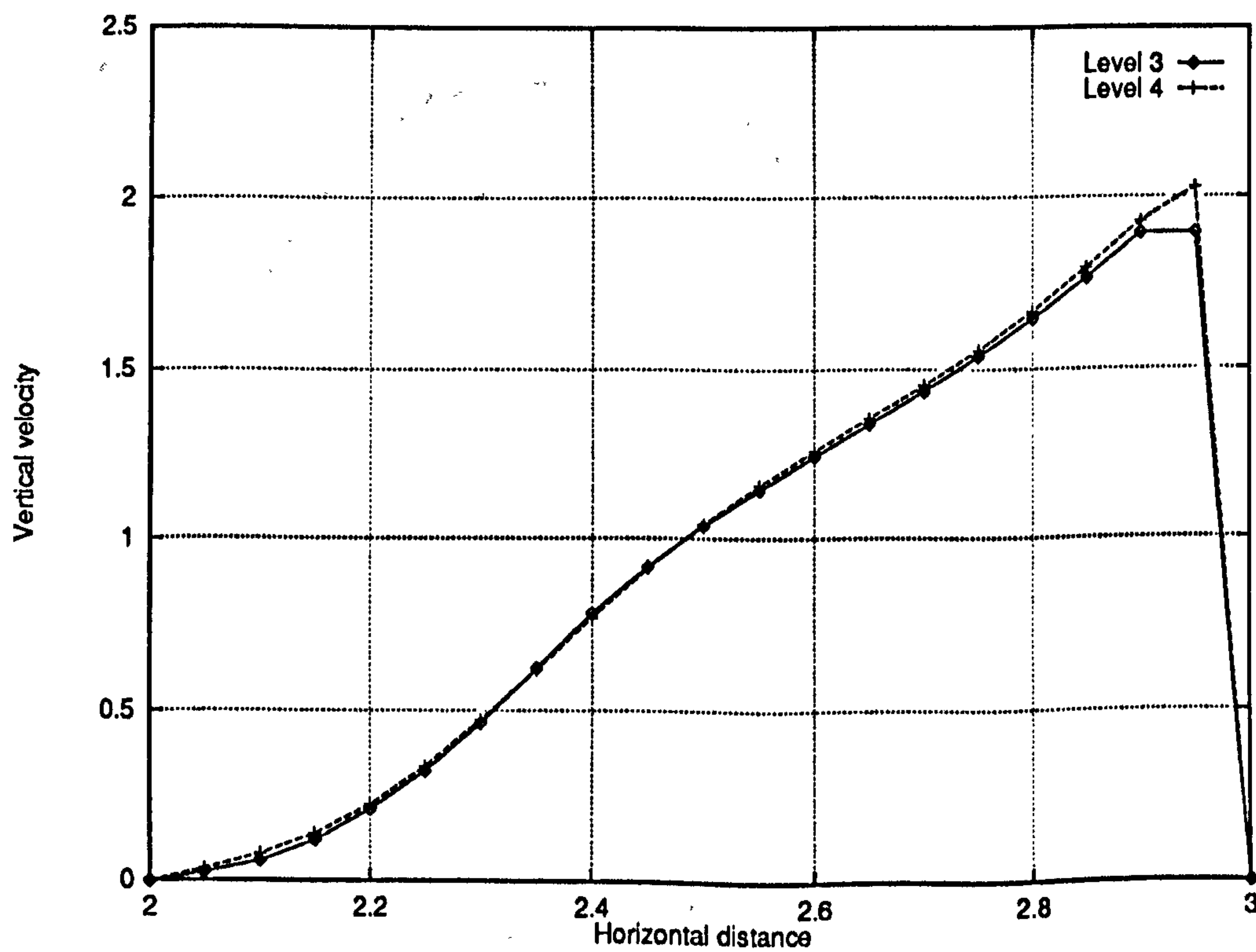


Figure 5.75: Multiphase T-junction problem - Grid independence for pang-multiphase results - Vertical velocity profiles along the line $y = 0.0$ (Phase I)

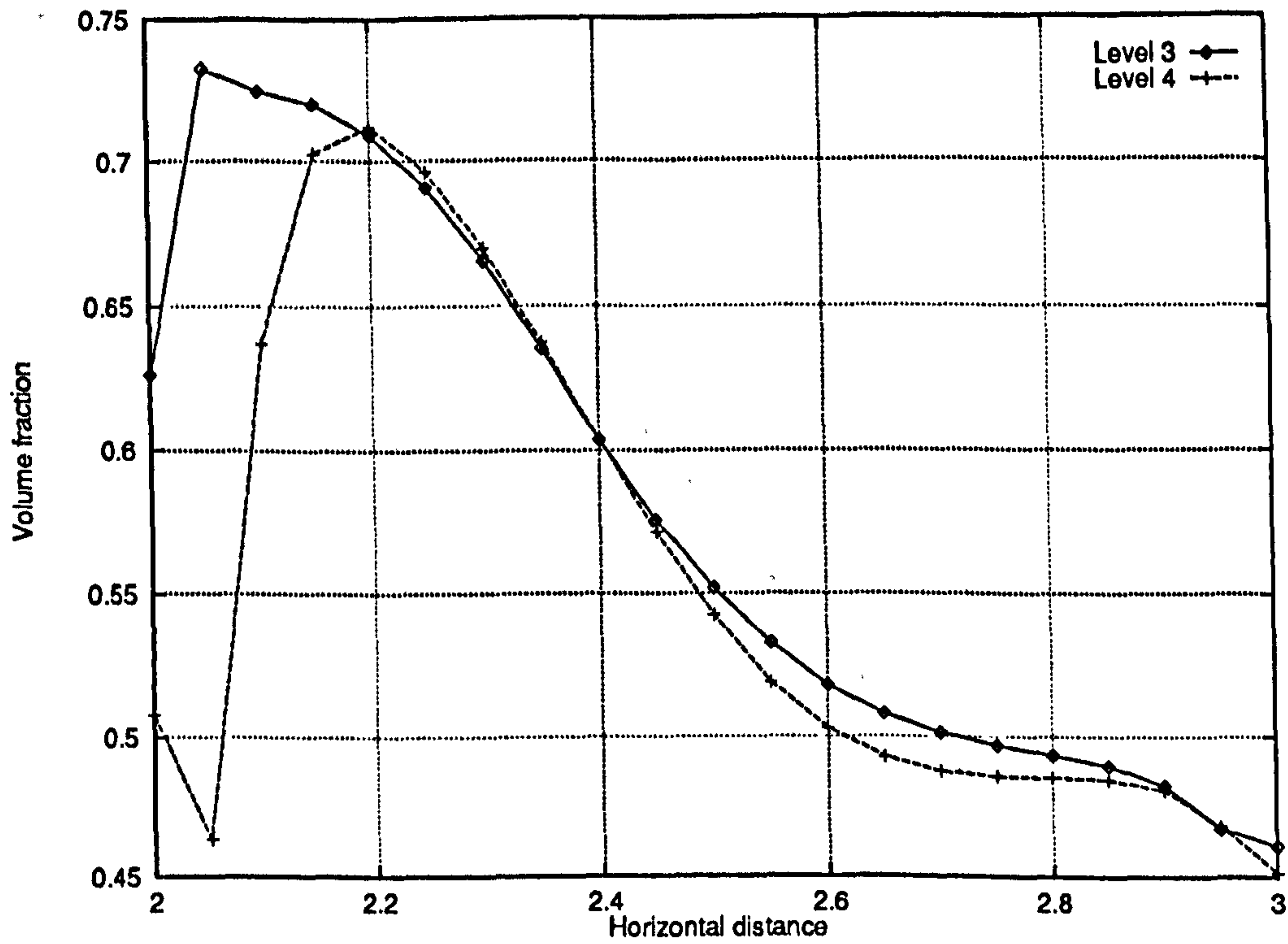


Figure 5.76: Multiphase T-junction problem – Grid independence for pamg-multiphase results – Volume fraction profiles along the line $y = 0.0$ (Phase I) showing that the volume fractions can be grid-dependent in certain regions

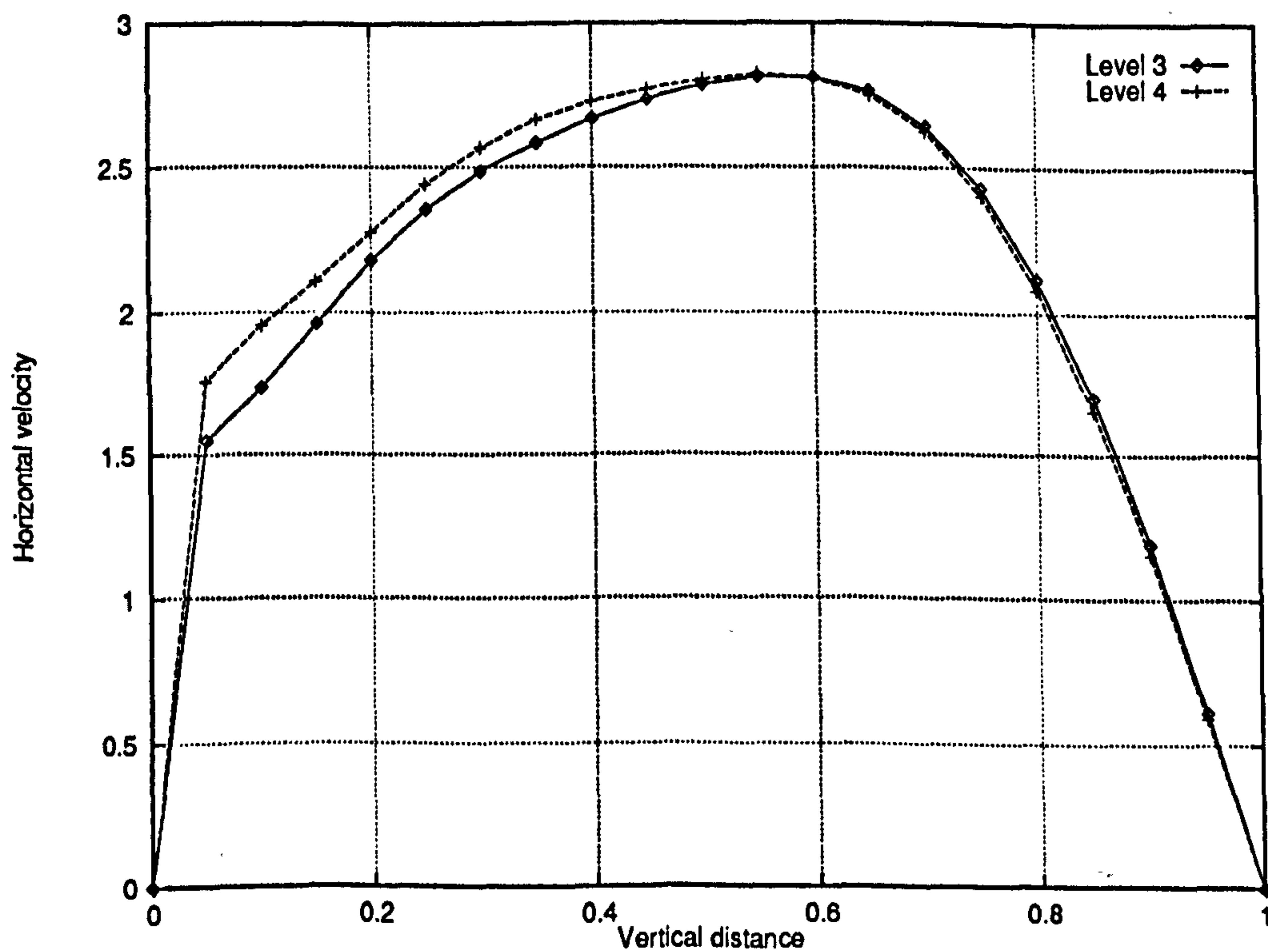


Figure 5.77: Multiphase T-junction problem – Grid independence for pamg-multiphase results – Horizontal velocity profiles along the line $x = 6.5$ (Phase I)

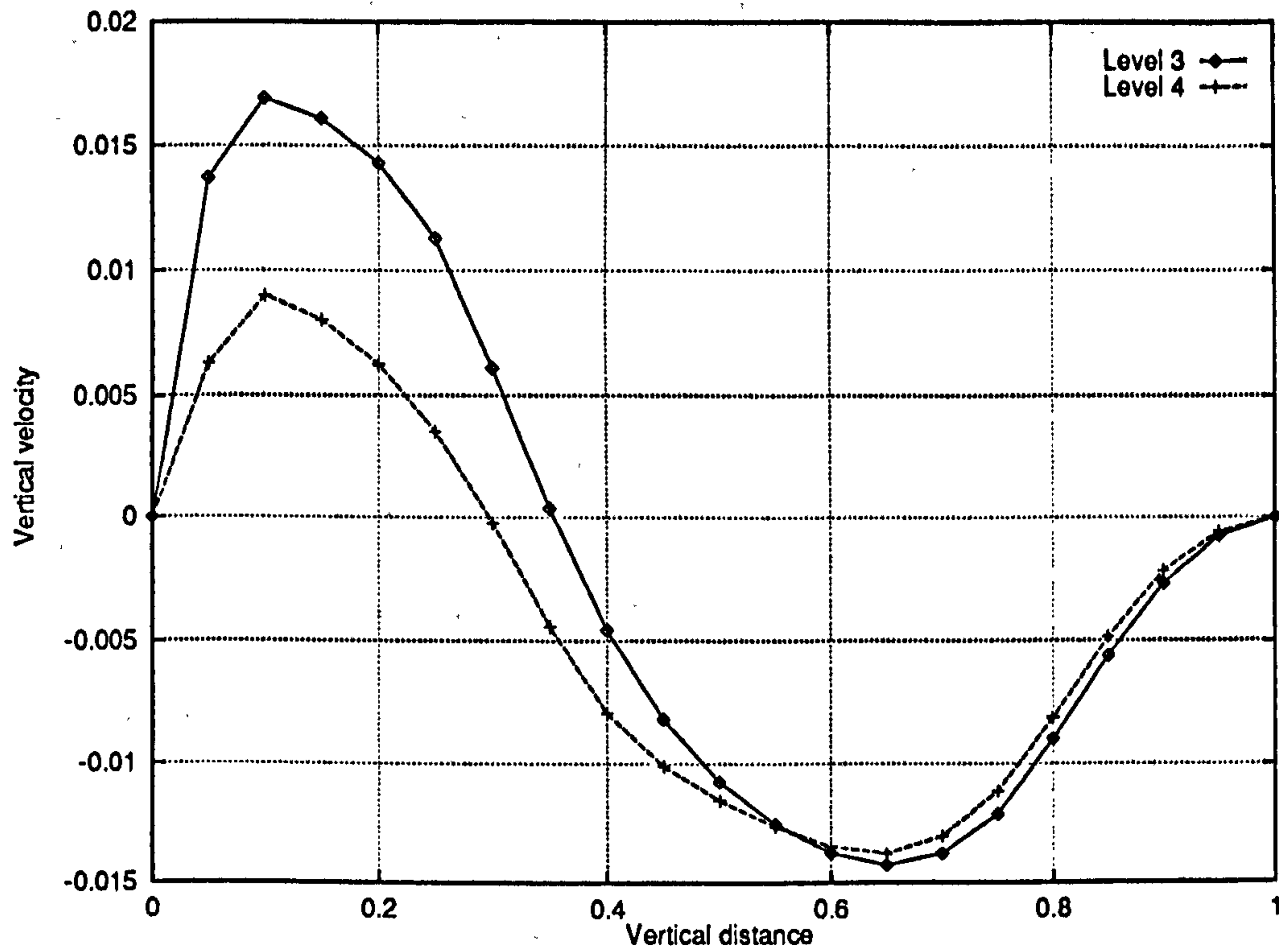


Figure 5.78: Multiphase T-junction problem - Grid independence for pang-multiphase results - Horizontal velocity profiles along the line $x = 6.5$ (Phase I)

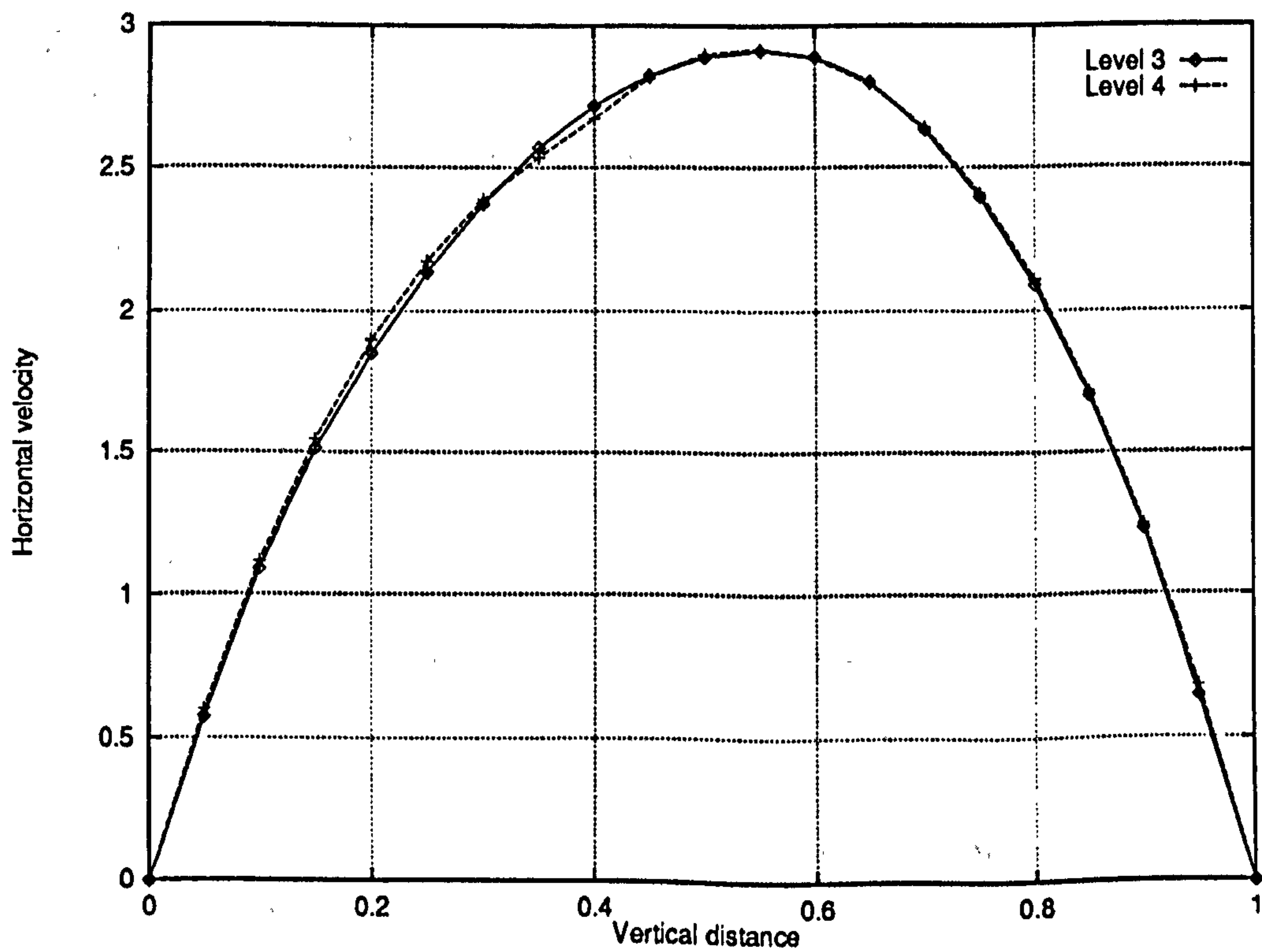


Figure 5.79: Multiphase T-junction problem - Grid independence for pang-multiphase results - Horizontal velocity profiles along the line $x = 6.5$ (Phase II)

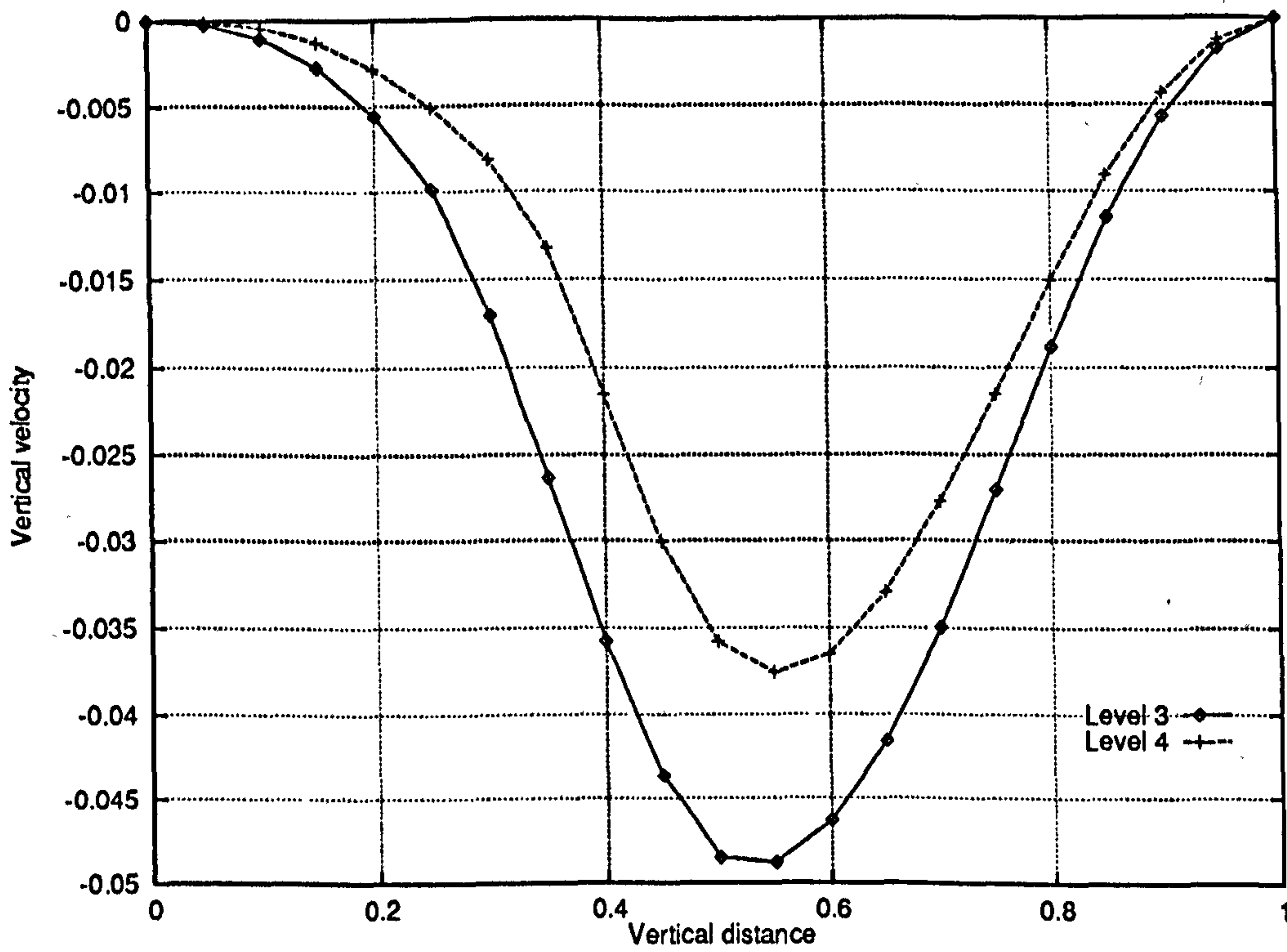


Figure 5.80: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Vertical velocity profiles along the line $x = 6.5$ (Phase II)

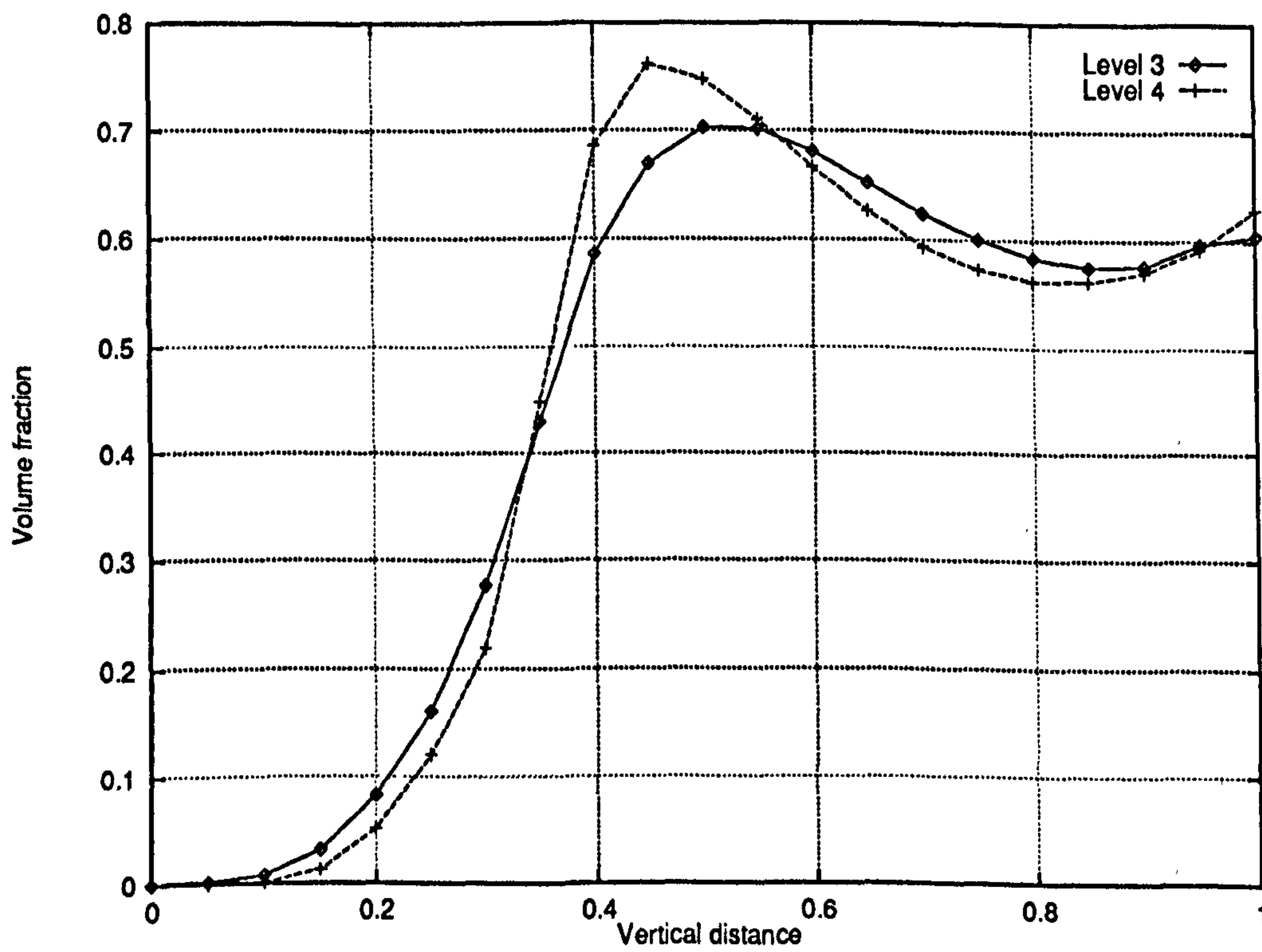


Figure 5.81: Multiphase T-junction problem - Grid independence for pamg-multiphase results - Volume fraction profiles along the line $x = 6.5$ (Phase I)

5.3.3 Test Problem 3: Two-phase Backward-Facing Step

Problem Definition This third multiphase test problem is designed to test the accuracy of `pang-multiphase` in the presence of recirculation zones i.e. regions where the streamlines are closed curves. This implies that there is no mass transfer between the main flow and recirculation zones. The latter are driven by shear. They are kept in movement by momentum transfers only from the main flow. Recirculation zones in single phase flows are a difficult pattern to simulate. In particular, their length can be very sensitive (see Section 5.2.3).

Once multiphase flows are considered, there is a further difficulty. Since recirculation zones are characterised by the absence of mass transfers with the main flow, it is impossible in a steady state calculation to specify the relative volume occupied by each phase if both phases are recirculating in any given region of the flow. Consequently, the steady problem may not be well-posed in a mathematical sense because several solutions may exist which depend on the initial guess. This is a novel type of singularity, specific to multiphase flows. As a direct consequence, obtaining solutions for the two-phase flows over a backward-facing step has proved relatively difficult. This question is examined in greater depth in Section 5.4.10.

For now, we are only concerned with the accuracy of the computations. Comparison with the solutions given by `CFX 4.1` can be performed when we have at least *one* test case for which the quasi-Newton coupled multigrid algorithm converged. One such test case is summarised below.

The problem is summarised by Figure 5.82. Figure 5.83, for its part, specifies the coordinate system and the main sections along which solution profiles are taken. The fluid properties are as follows ⁴:

Reynolds Number	Viscosity	Density
$Re_1 = 10$	$\mu_1 = 0.1$	$\rho_1 = 1.0$
$Re_2 = 7.5$	$\mu_2 = 0.066$	$\rho_2 = 0.5$

The geometry of the problem is:

- $0 \leq x \leq 10.5$,
- $0 \leq y \leq 1$ before the step and $-1 \leq y \leq 1$ after the step
- Position of the step : $x = 3$
- Height of the step: $h = 1.0$

⁴The Reynolds numbers are based on the height of the step.

The coarsest grid contains 192 cells. The grid sizes are $\Delta x = 0.375$ and $\Delta y = 0.25$. Figure 5.84 shows the grid at level 3. Figures 5.91 and 5.92, which show velocity profiles along an horizontal line in the bottom half of the computational domain, indicates that the length of the computational domain is adequate here again. The recirculation zone is very small, which is to be expected at lower Reynolds numbers.

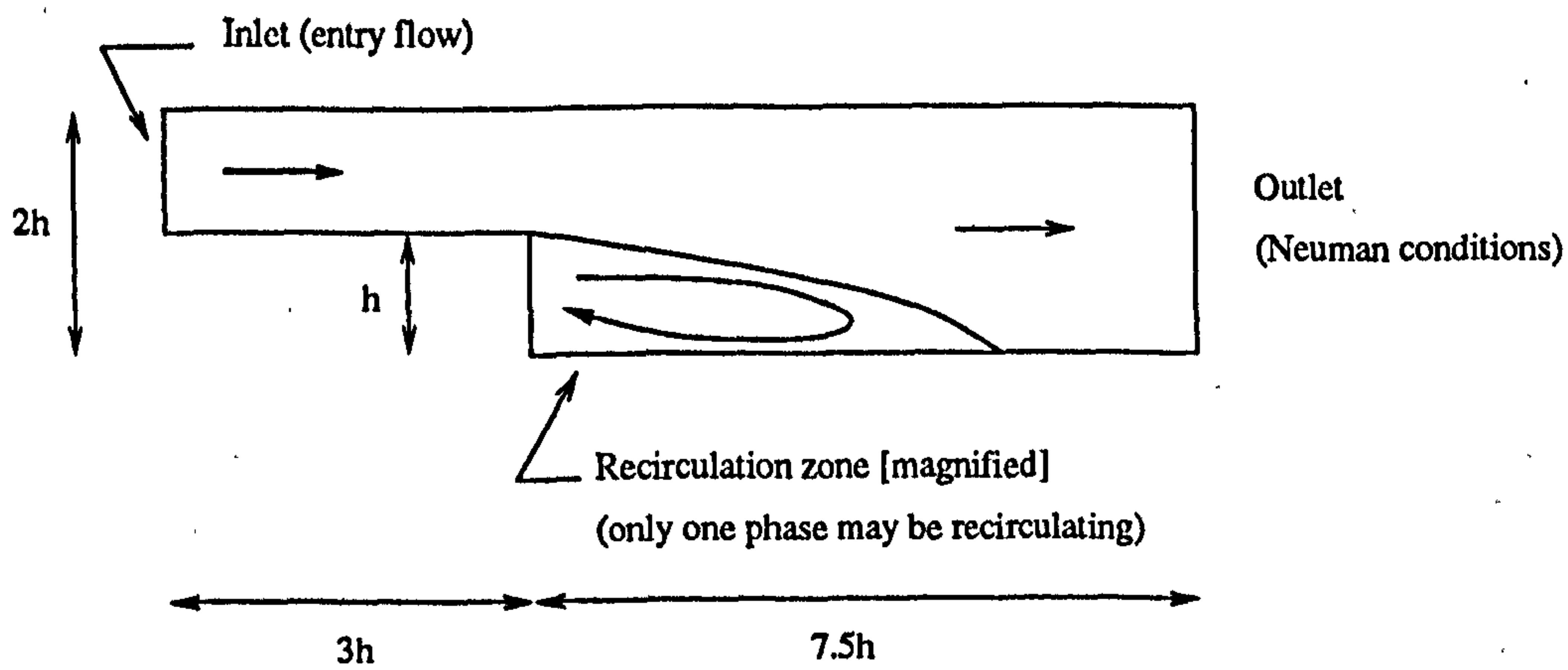


Figure 5.82: Diagrammatic description of the problem of a two-phase flow through a backward facing step

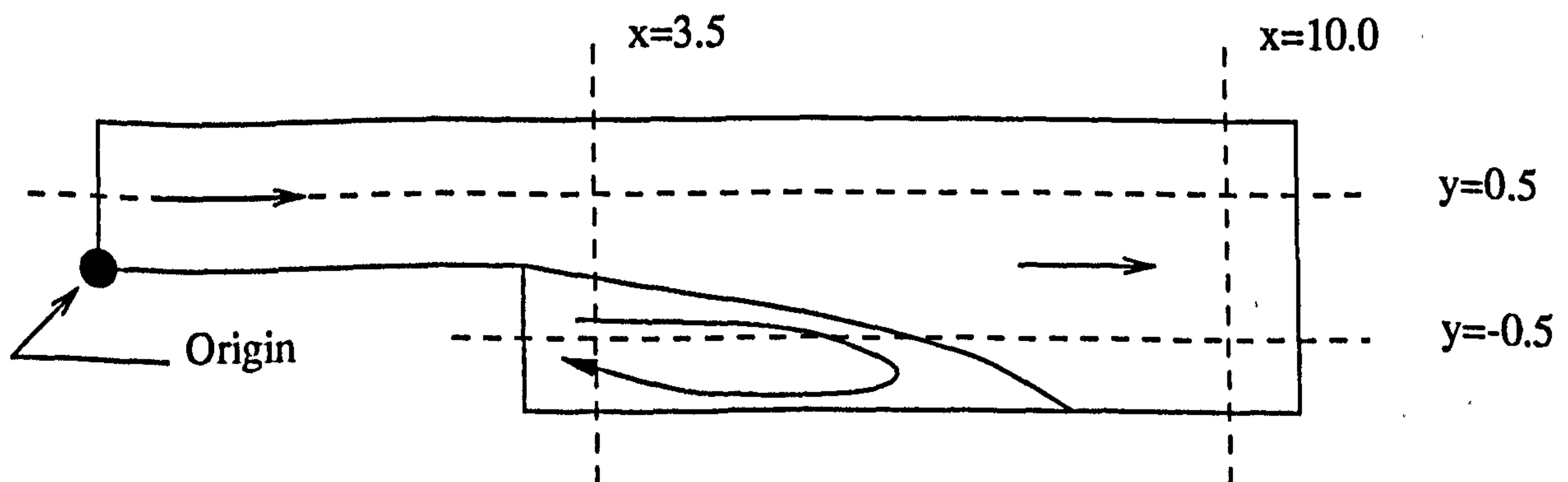


Figure 5.83: Multiphase backward-facing step problem – Coordinate system and main sections

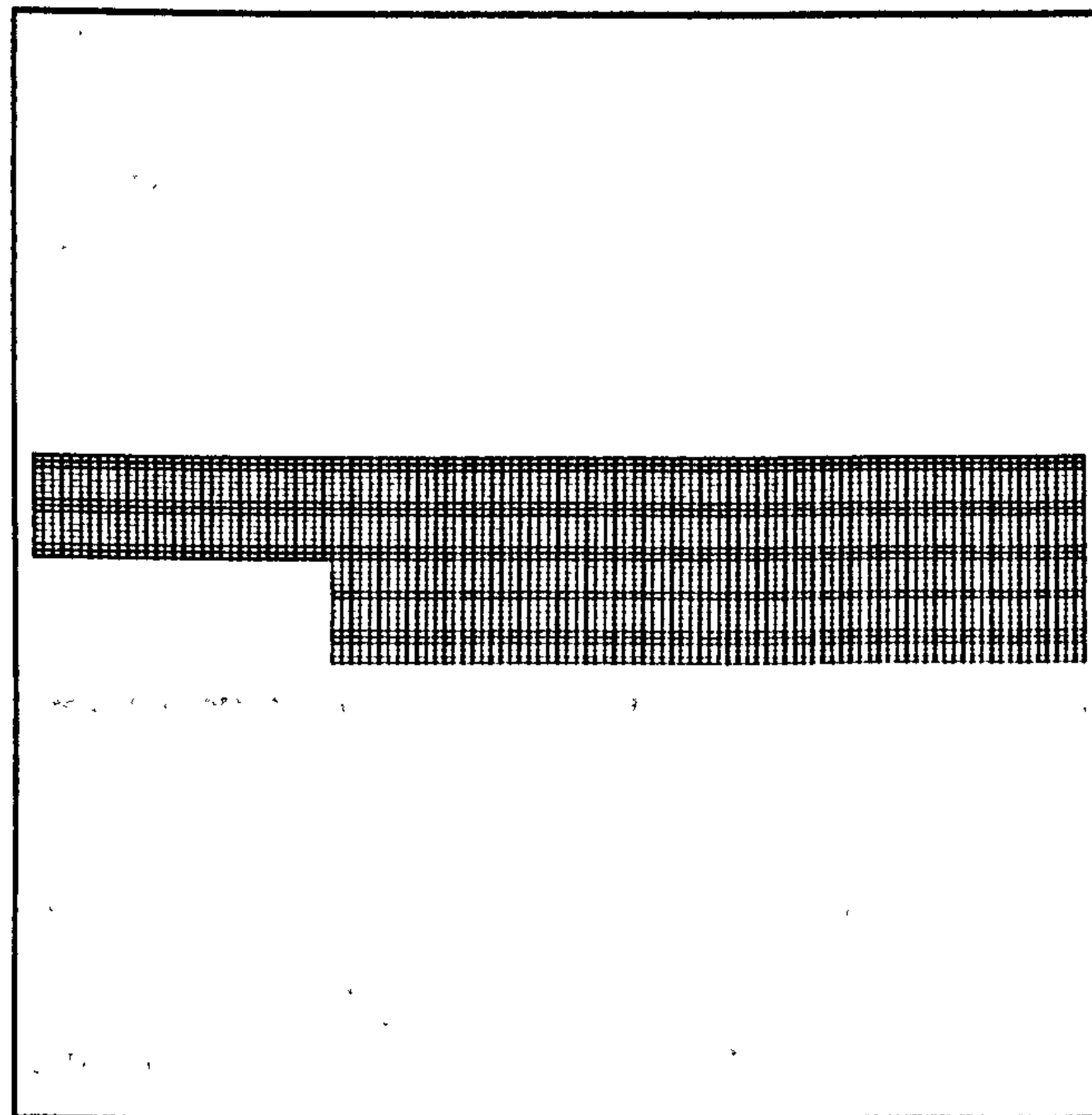


Figure 5.84: Multiphase backward-facing step problem – Uniform grid at level 3 (3072 cells, $\Delta x = 0.09375$, $\Delta y = 0.0625$)

Results and Discussion The features of the solution agree with independent calculations (and experiments). Figures 5.85, 5.86 and 5.116 show the streamlines for each phase and the volume fraction map throughout the computational domain, based on the `pamg-multiphase` solution.

Figures 5.87 to 5.101, for their part, present solution profiles obtained for two different grid levels. The main conclusions are:

- (i) We expect that the fluids will be greatly decelerated because of the step since a greater cross-section is available to them. Consequently the pressure gradient will become less steep downstream of the step. The change in the pressure gradients at the step readily appears (see Figure 5.90). Deceleration of the phases is also observed (Figure 5.87).
- (ii) A small recirculation zone can be observed next to the step. It is very slight and therefore does not appear very clearly on the solution profiles. However, in Figure 5.94 which shows the horizontal velocity profile for phase 1 along the line $x = 3.5$, it can be seen that the velocity is negative at the bottom of the channel.

Examination of the solution reveals that recirculation zone tends to be occupied by one phase (Figure 5.98). Phase separation appears to be almost complete. Crucially, only one phase is recirculating. When the Reynolds numbers are increased, the phases still tend to separate but recirculation occurs for both of them. This is physically acceptable but causes numerical difficulties (see Section 5.4.10).

Comparisons between the `pamg-multiphase` solutions at levels 2 and 3 reveal a reasonable degree of grid independence. It should be noted that when results appear not to be grid independent, the volume fraction is usually close to zero. This indicates that the momentum is very small and one would expect the velocity field to be badly conditioned. Examples of this behaviour are shown in Figures 5.94 and 5.95. Compare with Figures 5.96 and 5.97 respectively which show the evolution of the momentum .

As before, the `pamg-multiphase` solutions were compared with those of CFX 4.1 and the results agree quite closely, as Figures 5.102 to 5.112 demonstrate. As previously noted for the channel flow problem (Section 5.3.1), `pamg-multiphase` diffuses the volume fraction slightly more than the CFX 4.1 (Figures 5.104 and 5.106). The treatment of the volume fraction boundary conditions is another factor which could explain the variations.

Furthermore, due to the pressure singularity which exists for incompressible flows, the `pamg-multiphase` and CFX 4.1 pressure fields are different but equivalent, as shifting by a suitable amount reveals (see Figures 5.108 and 5.109).

Similarly to the T-junction problem, the CFX 4.1 solution is oscillatory near the outlet. See Figures 5.107 and 5.112. See also Figure 5.113 which shows contours for

the volume fraction field for phase 1. The reason for the presence of oscillations is as yet unexplained. Some evidence suggests that it is not connected with the type of boundary conditions imposed at the outlet.

The main value of the test case presented here is that it validates the software implementation. The fact that the `pang-multiphase` solver converged for at least one recirculation problem indicates that the main reason for lack of convergence in other cases is likely to be either mathematical or numerical, as opposed to merely computational. In the author's opinion, the reason why this test case converged is that only one phase is recirculating. Hence the volume fractions are fully determined by the constraint of mass conservation on the non-recirculating phase. This problem is therefore well-posed. See Section 5.4.10 for further details.

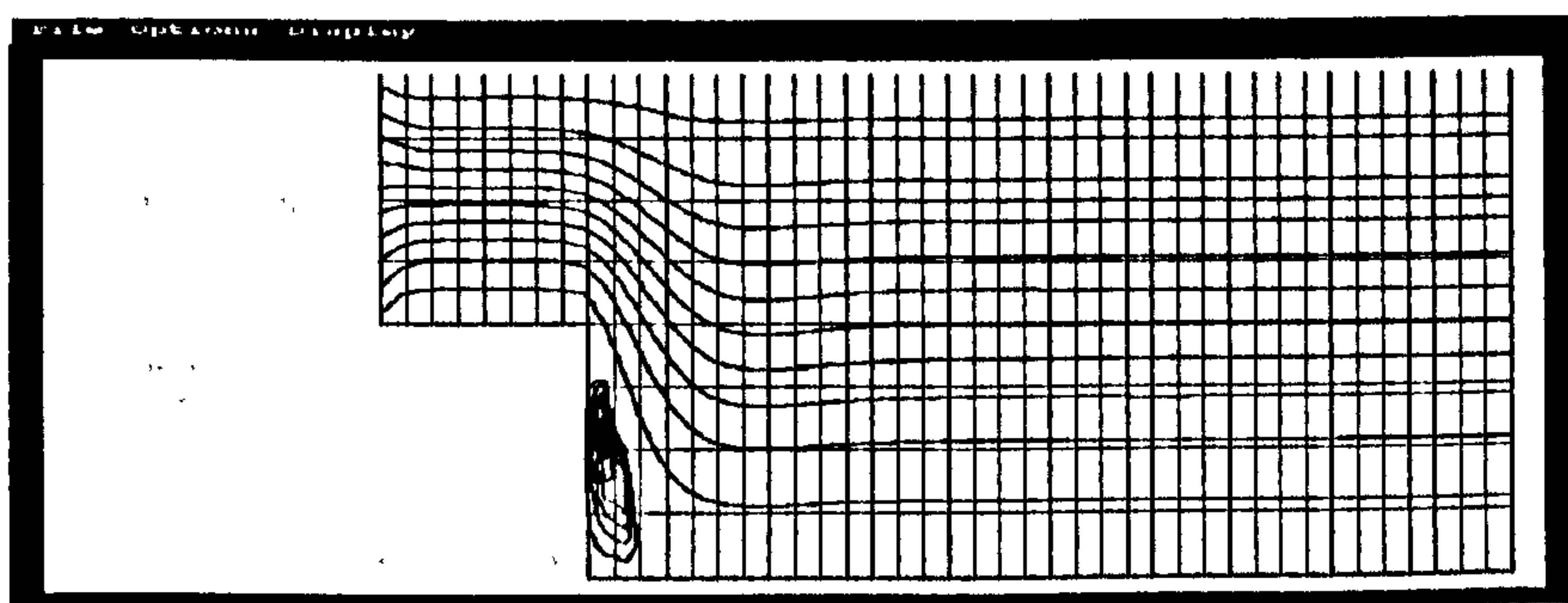


Figure 5.85: Multiphase backward-facing step problem – Streamlines for phase 1

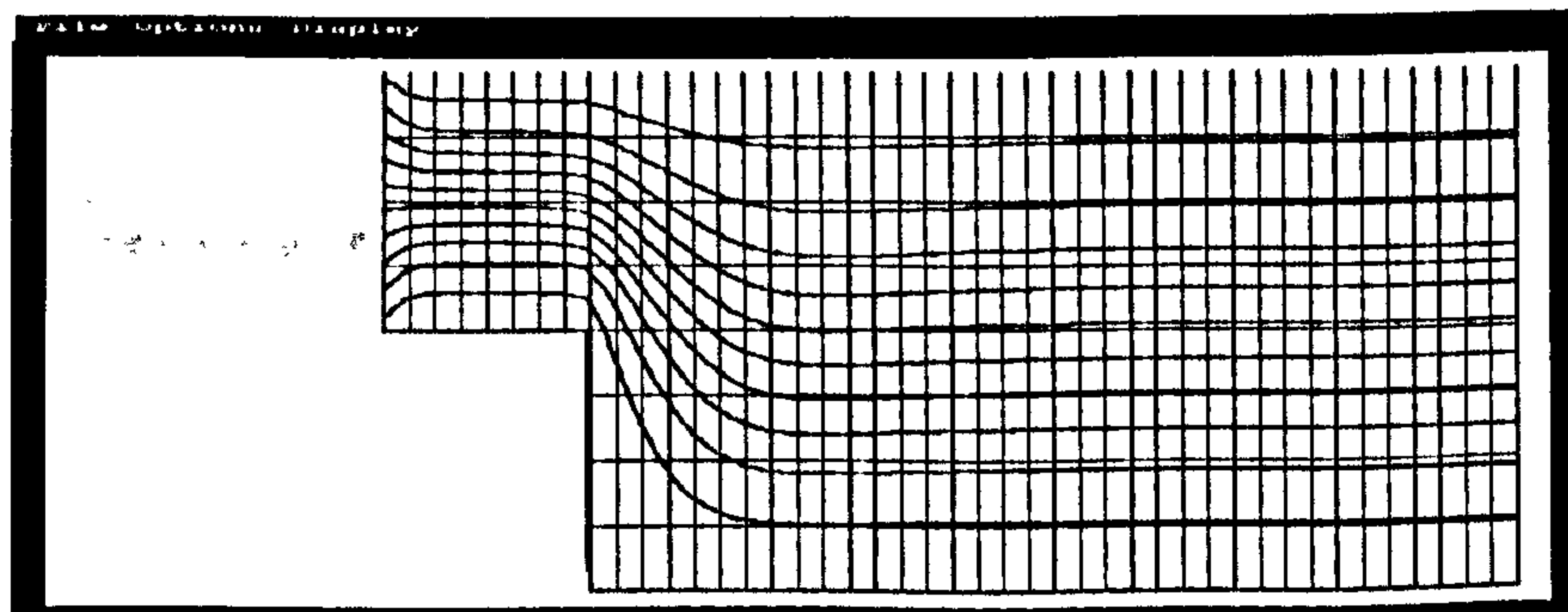


Figure 5.86: Multiphase backward-facing step problem – Streamlines for phase 2

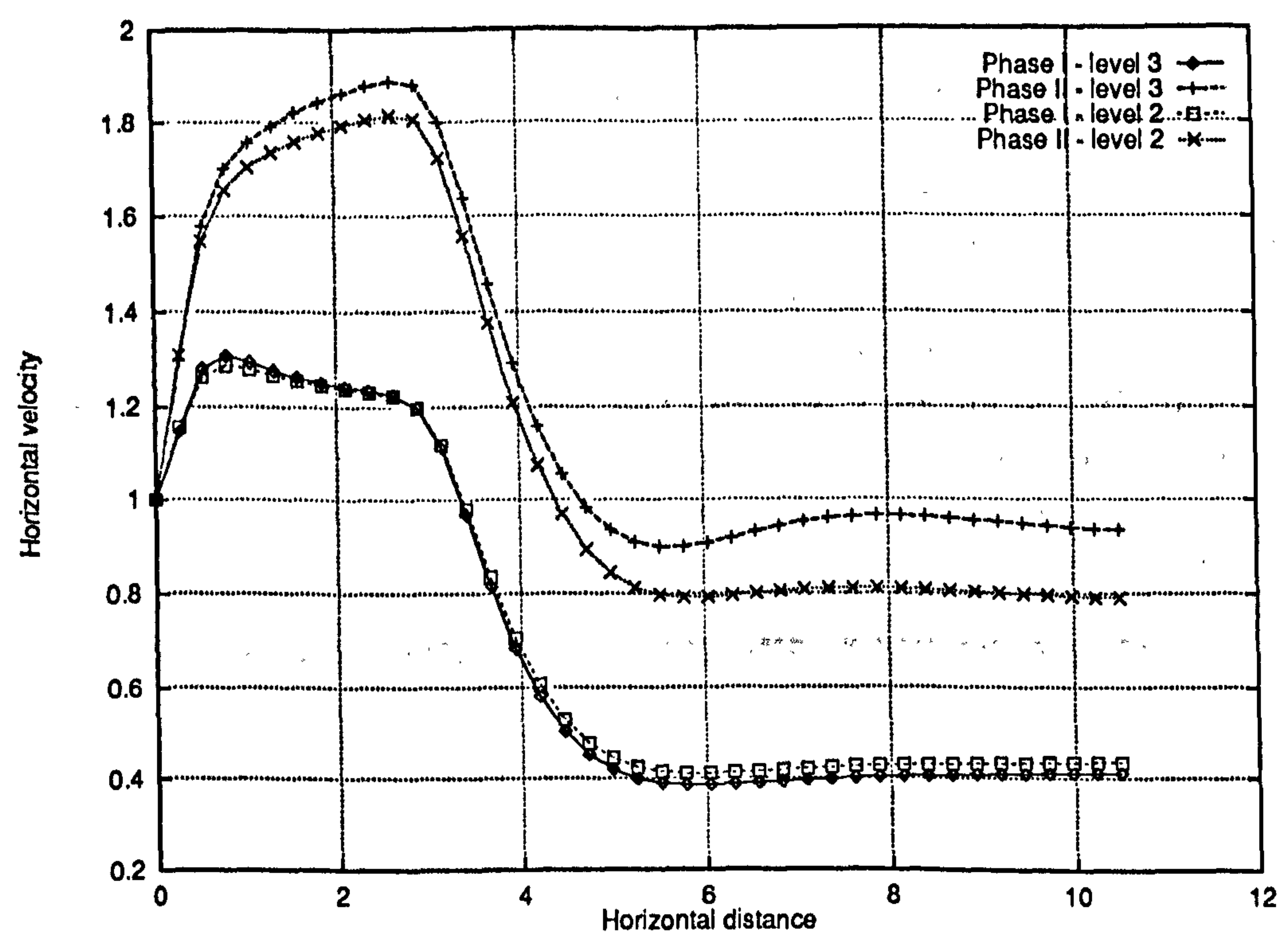


Figure 5.87: Multiphase backward-facing step problem – Horizontal velocity profiles along the line $y = 0.5$ showing the fluid deceleration through the step

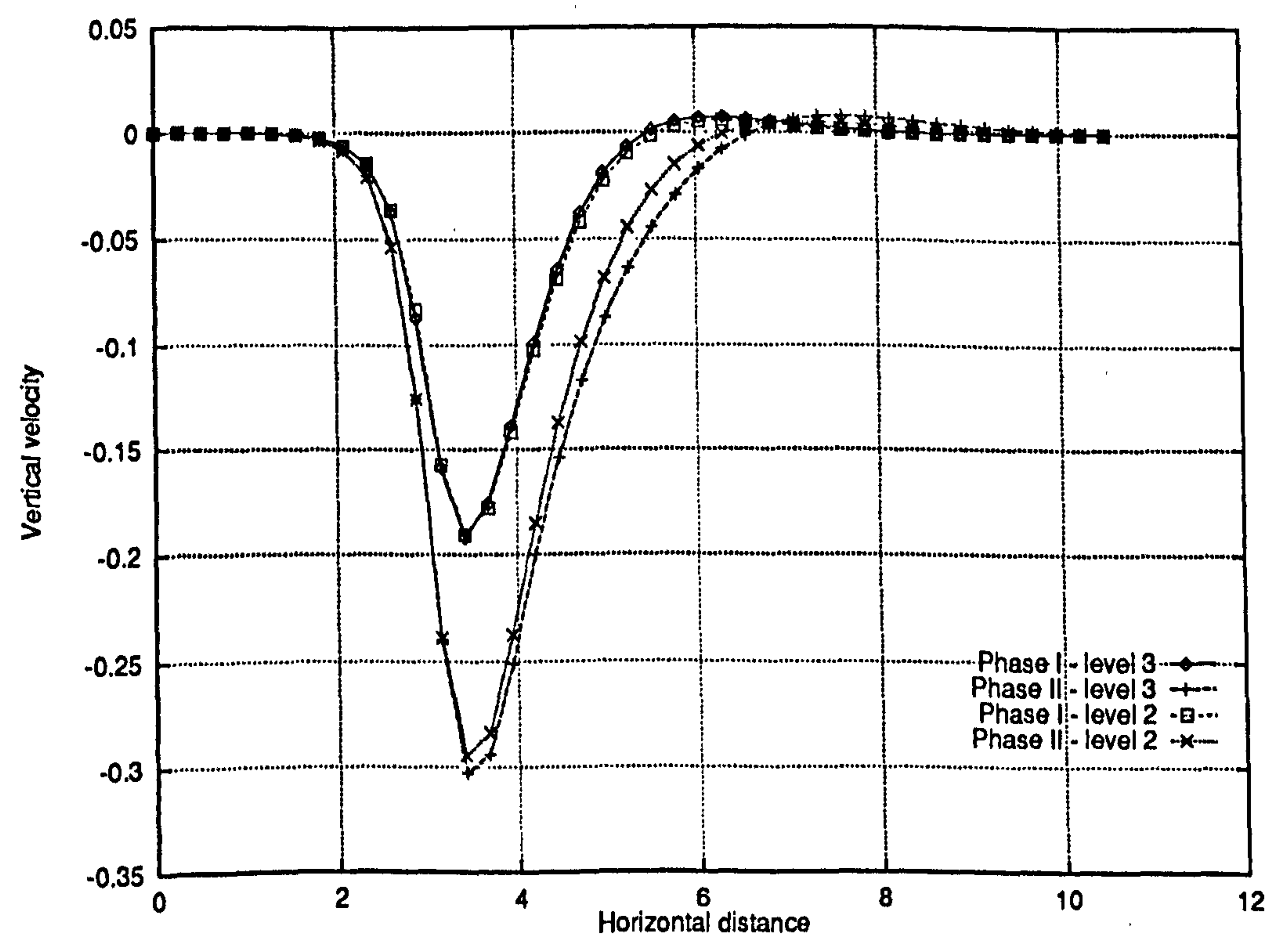


Figure 5.88: Multiphase backward-facing step problem – Vertical velocity profiles along the line $y = 0.5$ – Comparison of pamg-multiphase solutions at levels 2 and 3

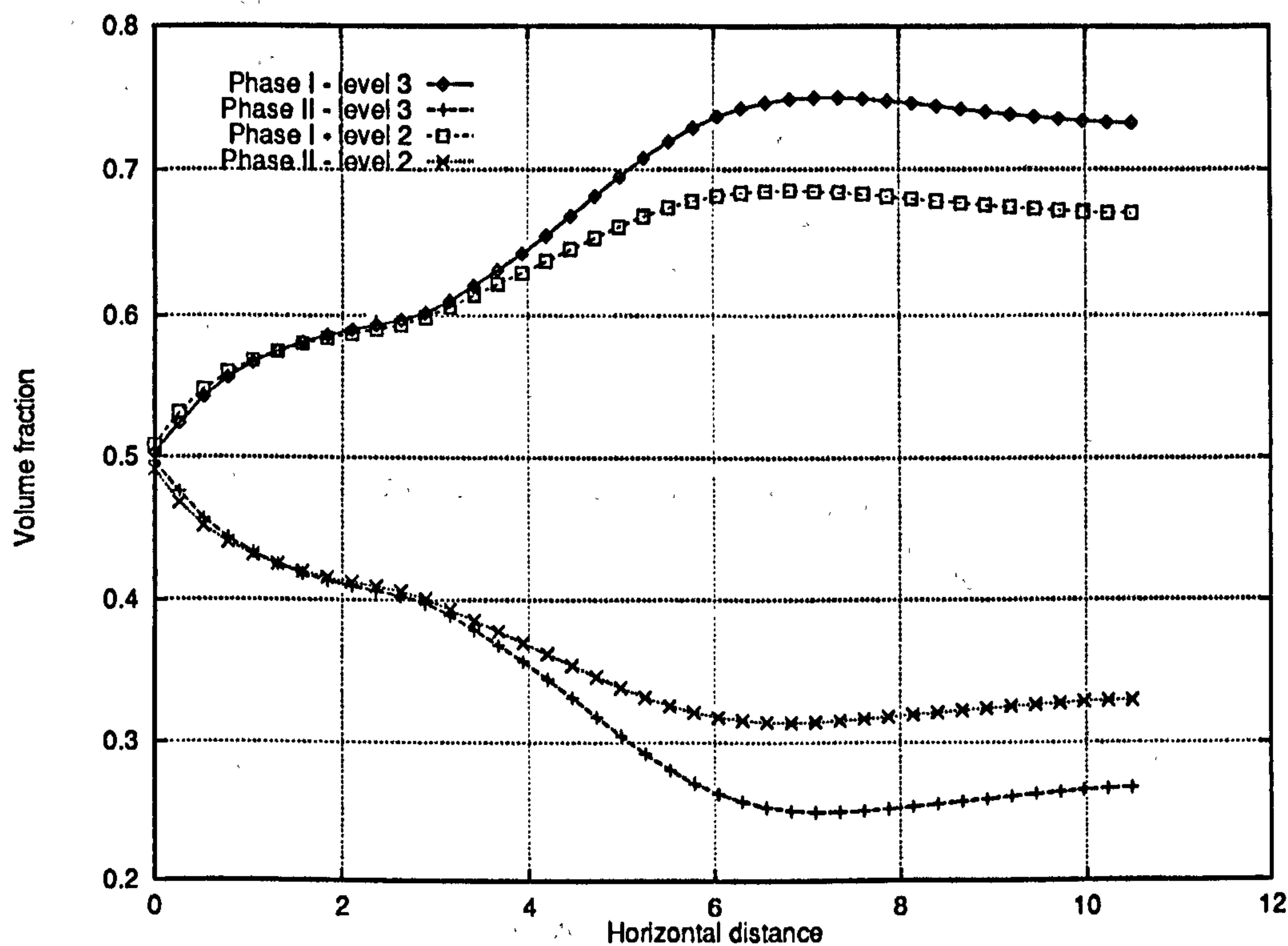


Figure 5.89: Multiphase backward-facing step problem - Volume fraction profiles along the line $y = 0.5$ - Comparison of pang-multiphase solutions at levels 2 and 3

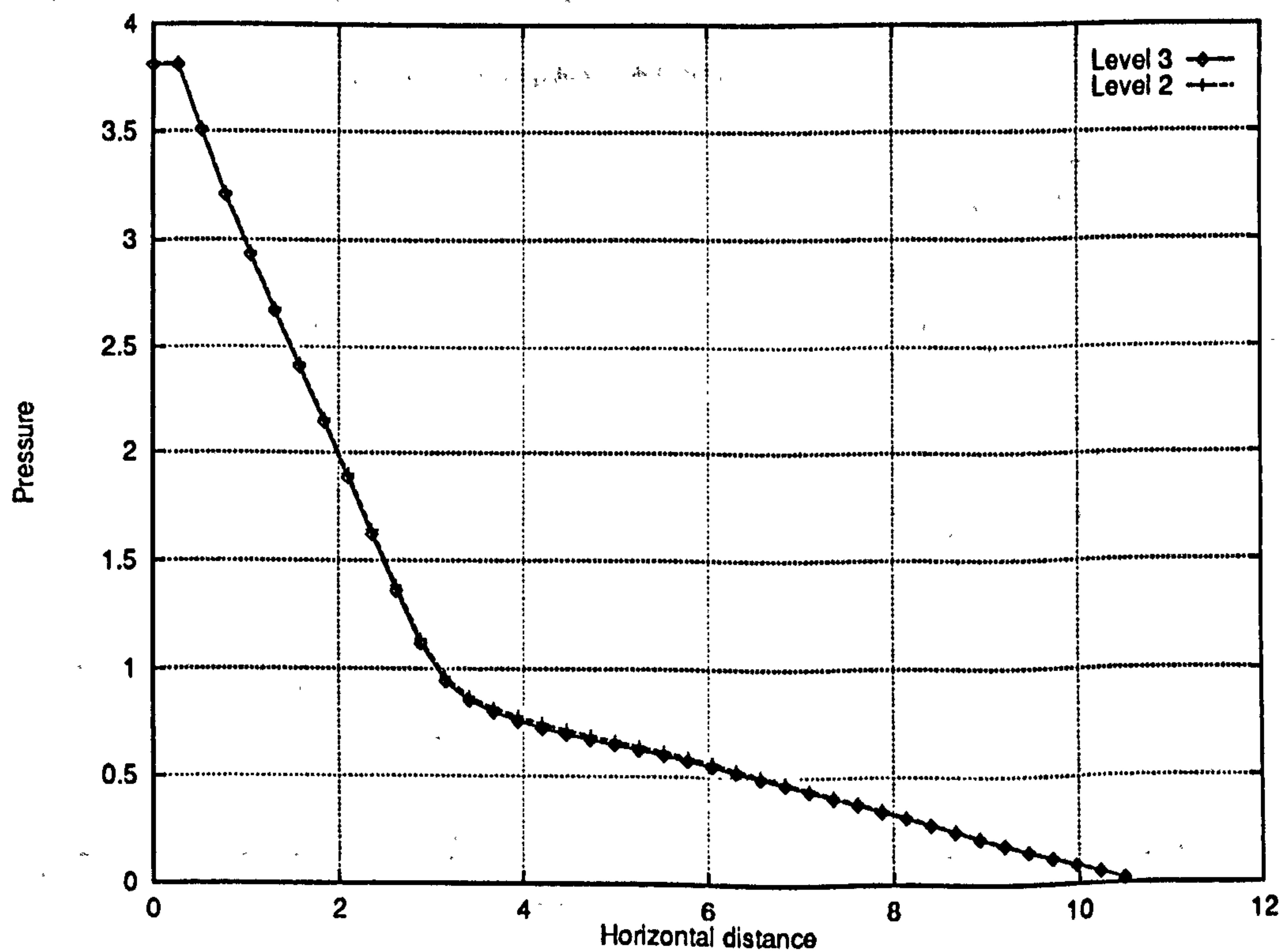


Figure 5.90: Multiphase backward-facing step problem - Pressure profiles along the line $y = 0.5$ - Comparison of pang-multiphase solutions at levels 2 and 3

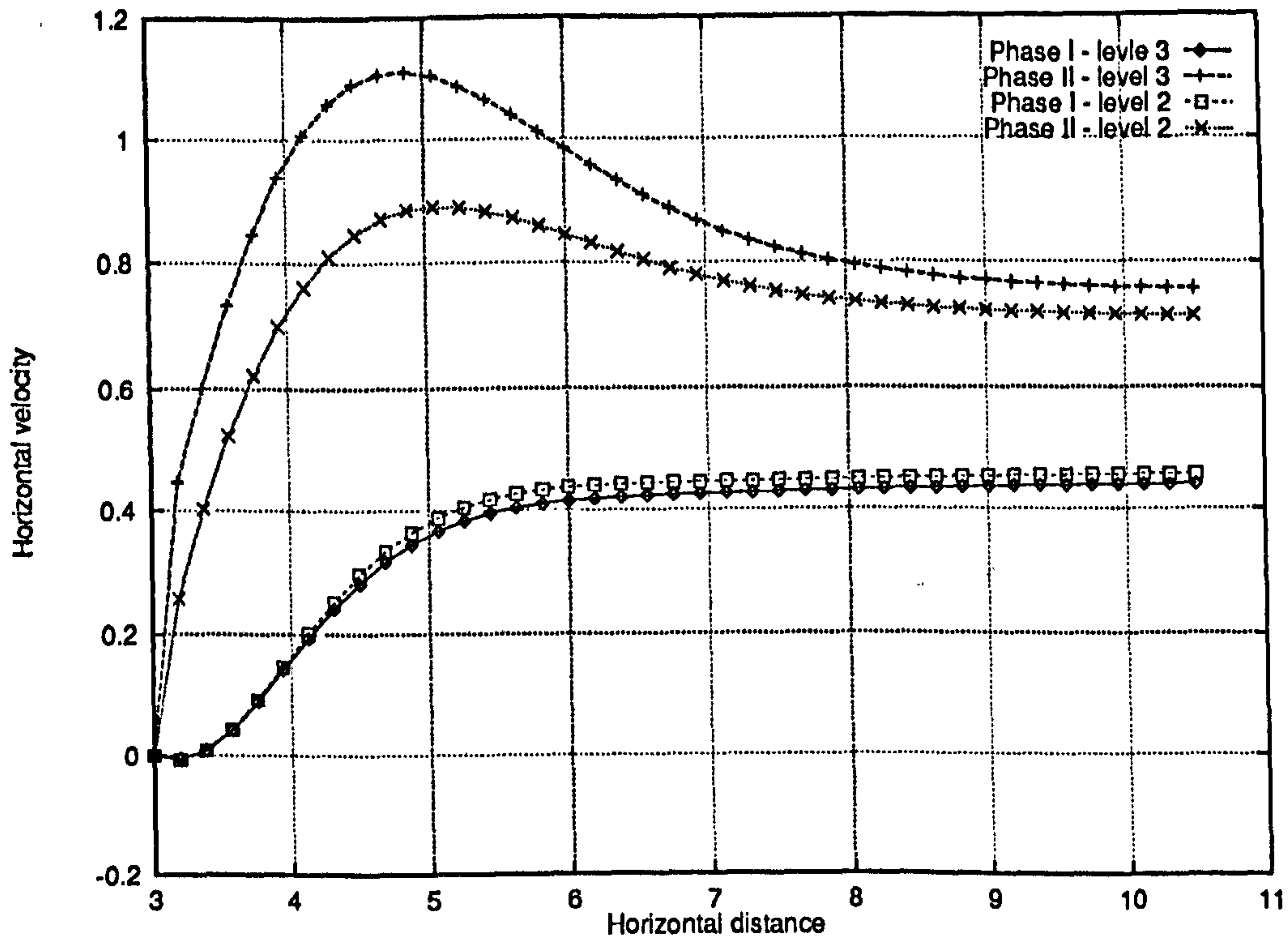


Figure 5.91: Multiphase backward-facing step problem – Horizontal velocity profiles along the line $y = -0.5$ – Comparison of pang-multiphase solutions at levels 2 and 3

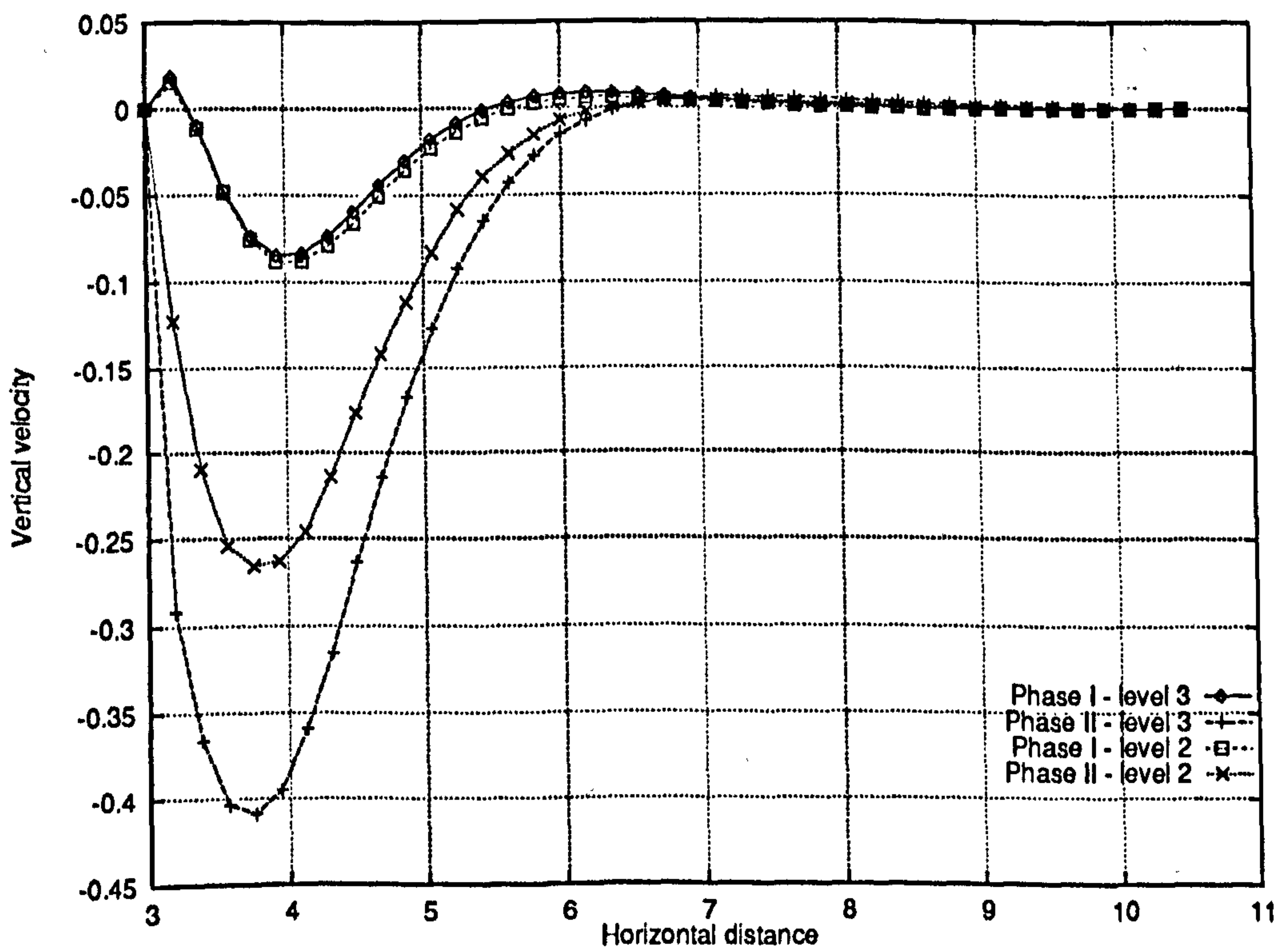


Figure 5.92: Multiphase backward-facing step problem – Vertical velocity profiles along the line $y = -0.5$ – Comparison of pang-multiphase solutions at levels 2 and 3

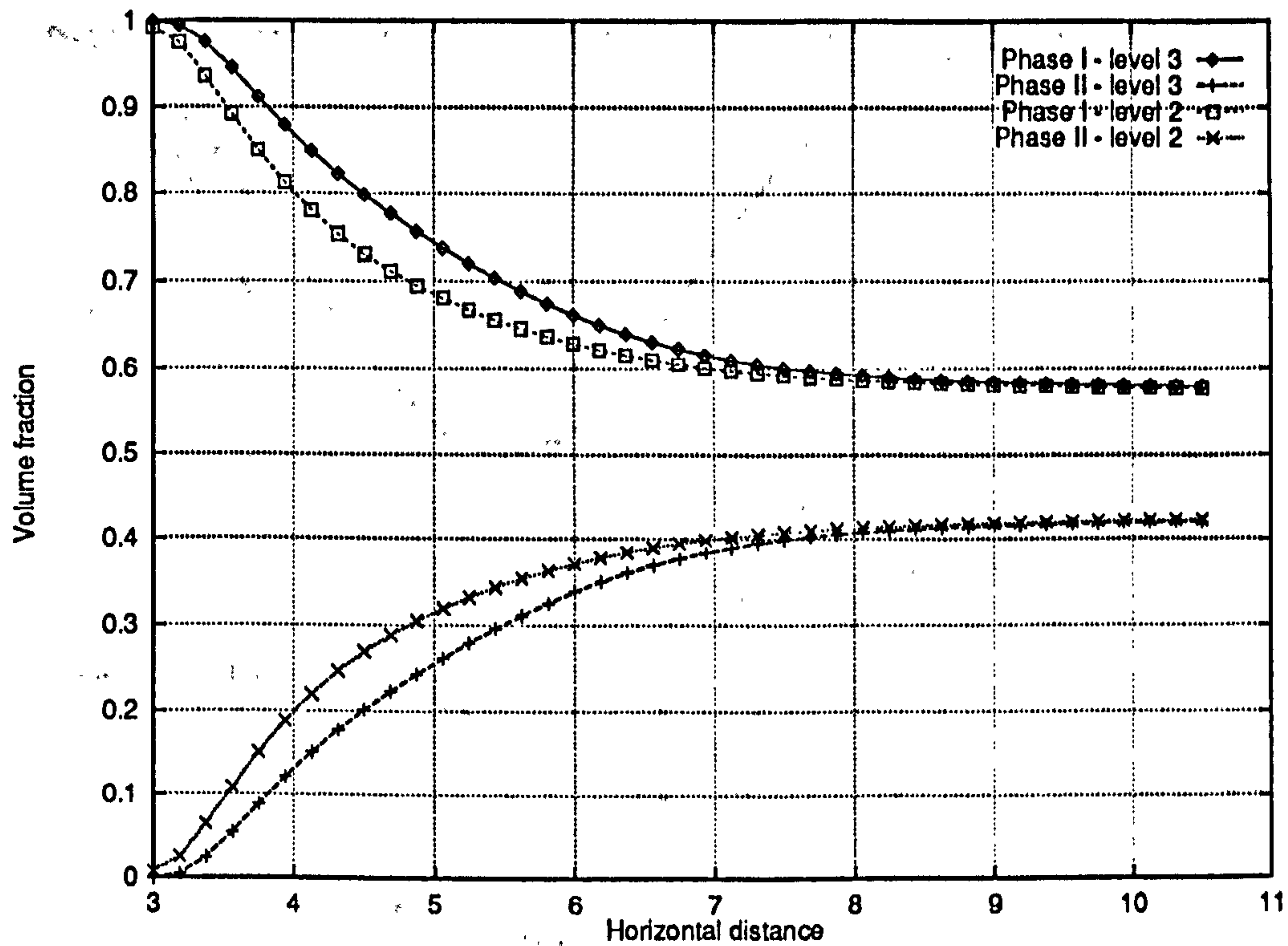


Figure 5.93: Multiphase backward-facing step problem – Volume fraction profiles along the line $y = -0.5$ – Comparison of pang-multiphase solutions at levels 2 and 3, showing the separation of the phases just downstream of the step

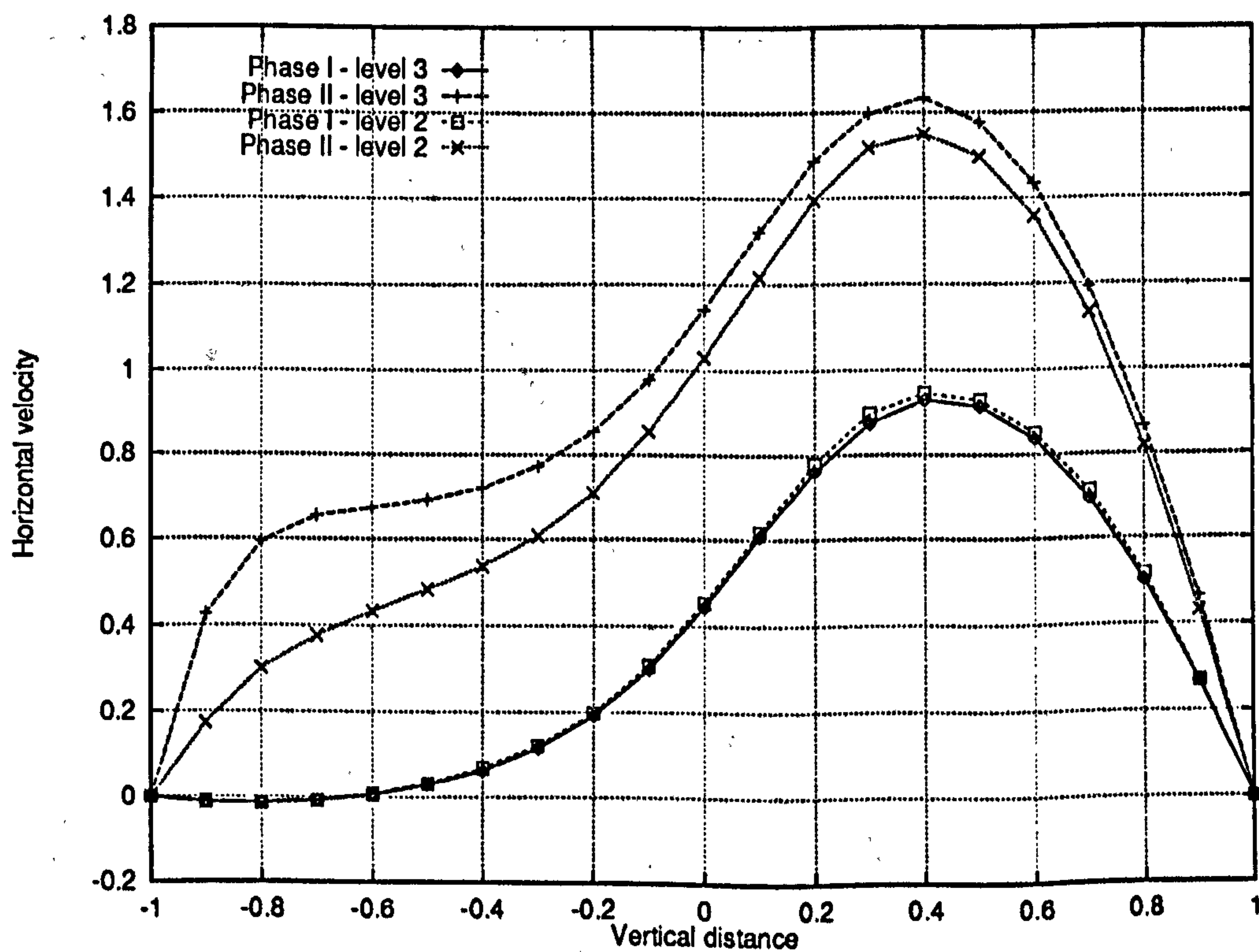


Figure 5.94: Multiphase backward-facing step problem – Horizontal velocity profiles along the line $x = 3.5$ – Comparison of pang-multiphase solutions at levels 2 and 3 showing the effect of the recirculation zone on the values of primitive variables

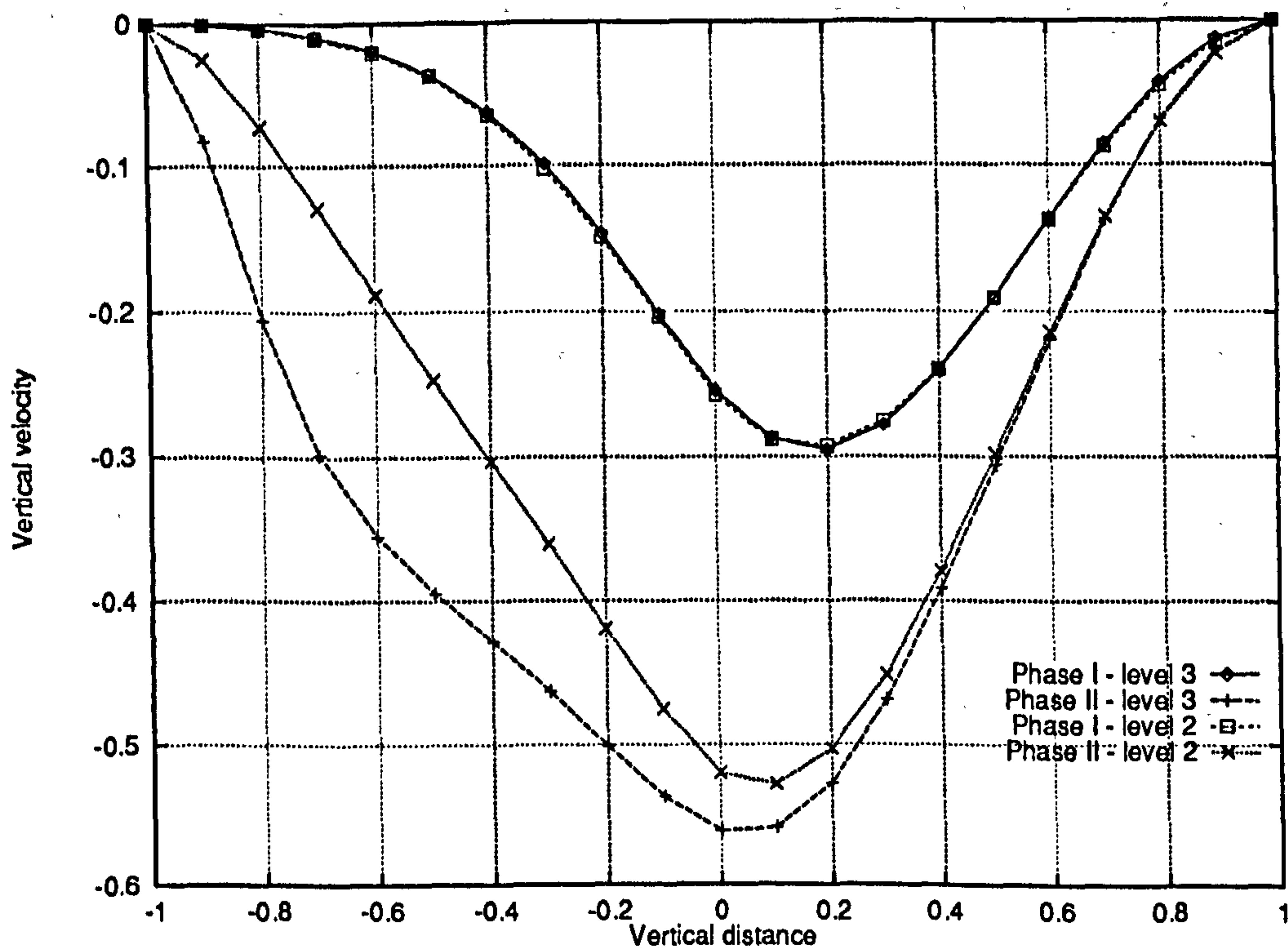


Figure 5.95: Multiphase backward-facing step problem – Vertical velocity profiles along the line $x = 3.5$ – Comparison of pang-multiphase solutions at levels 2 and 3 showing the effect of the recirculation zone on the values of primitive variables

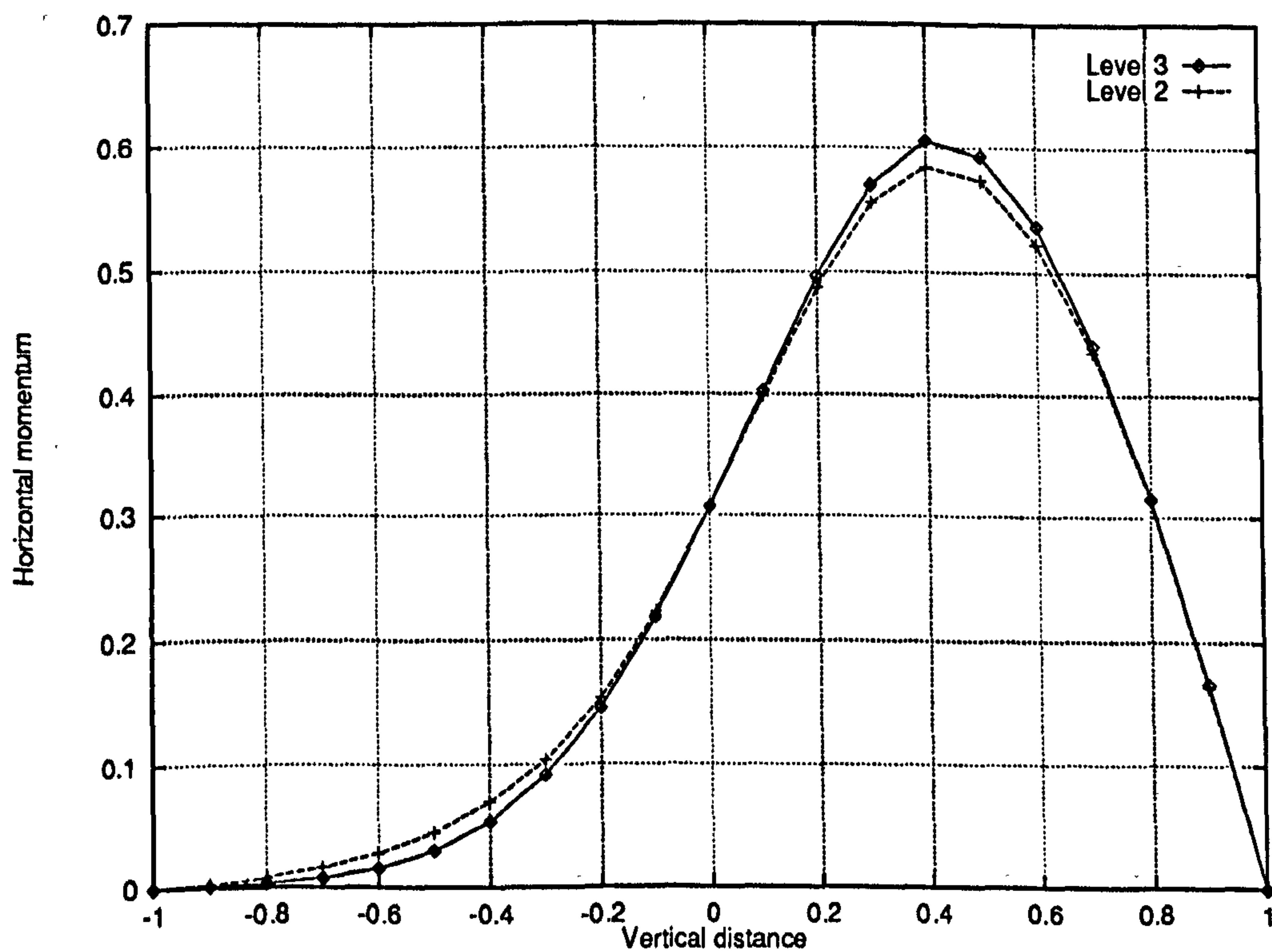


Figure 5.96: Multiphase backward-facing step problem – Horizontal momentum profiles along the line $x = 3.5$ for phase II – Comparison of pang-multiphase solutions at levels 2 and 3

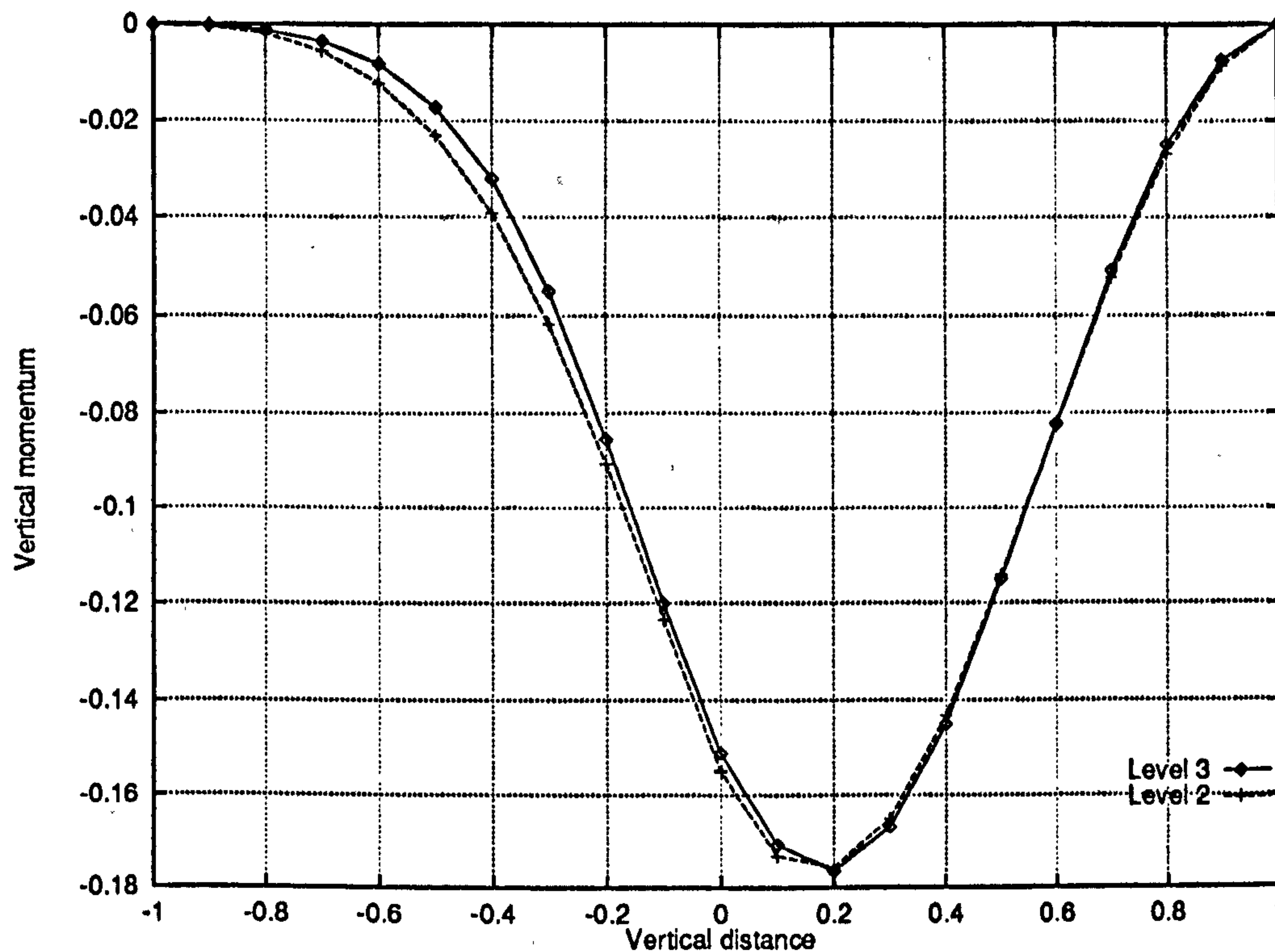


Figure 5.97: Multiphase backward-facing step problem – Vertical momentum profiles along the line $x = 3.5$ for phase II – Comparison of pang-multiphase solutions at levels 2 and 3

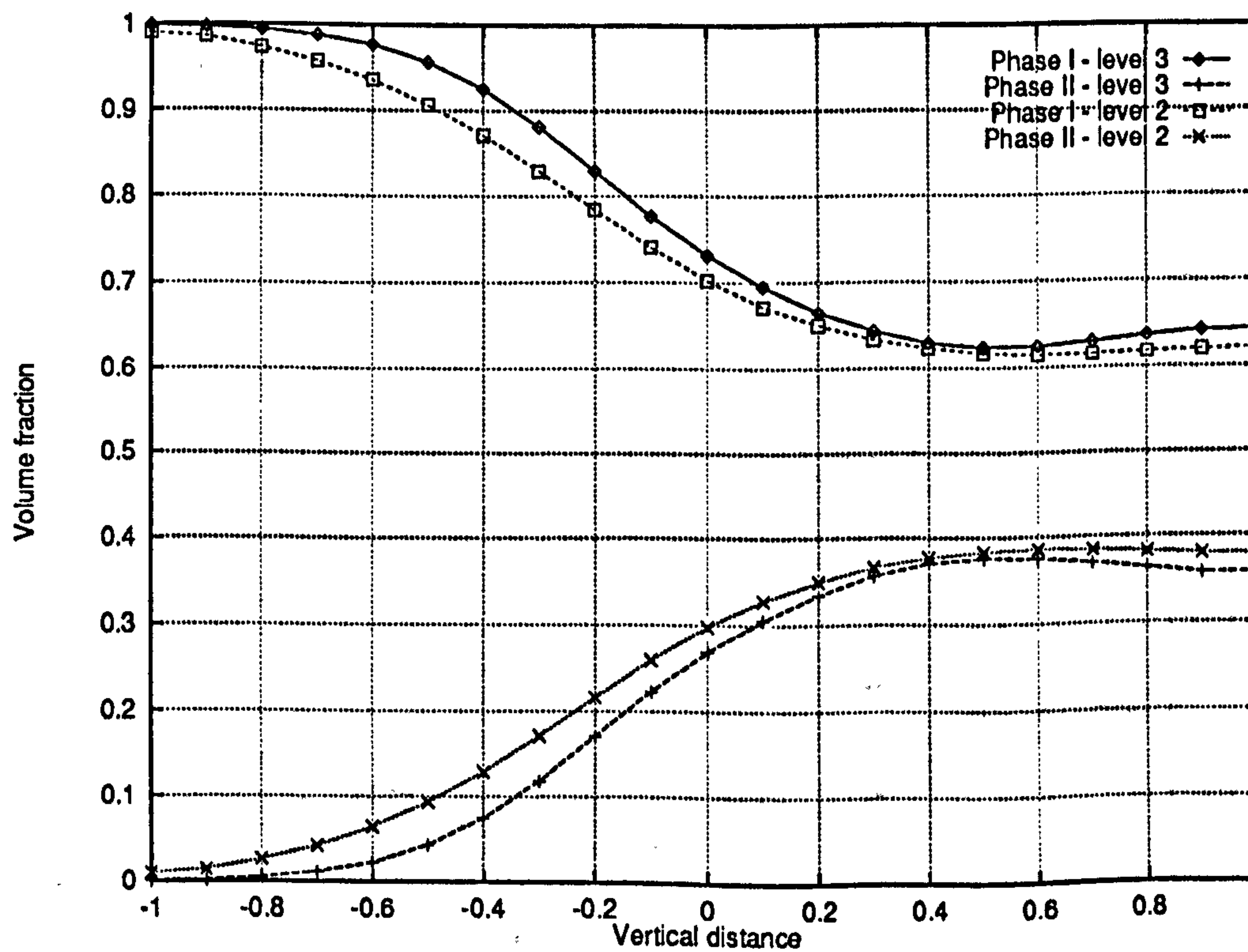


Figure 5.98: Multiphase backward-facing step problem – Volume fraction profiles along the line $x = 3.5$ – Comparison of pang-multiphase solutions at levels 2 and 3

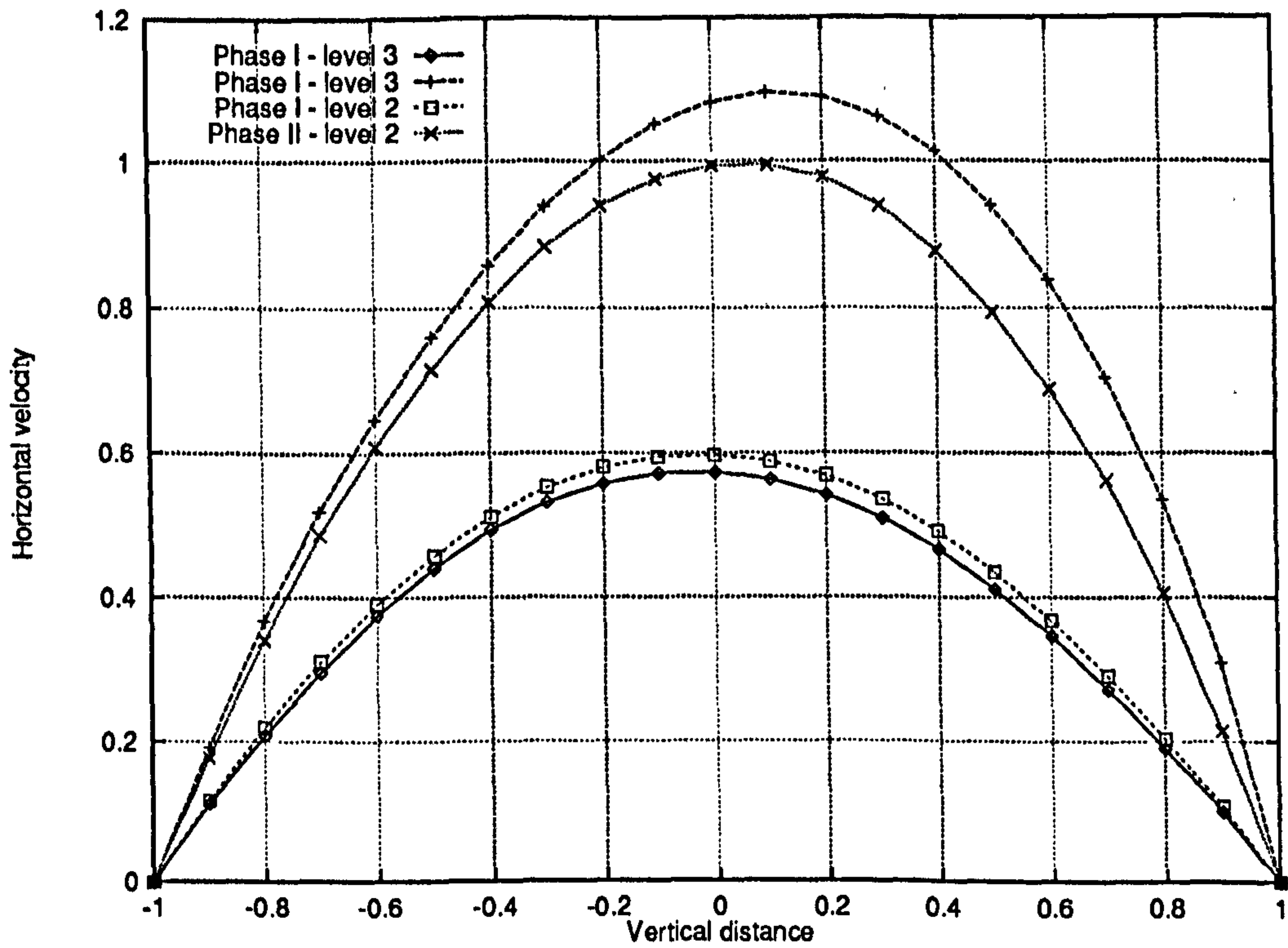


Figure 5.99: Multiphase backward-facing step problem – Horizontal velocity profiles along the line $x = 10.0$ – Comparison of pang-multiphase solutions at levels 2 and 3

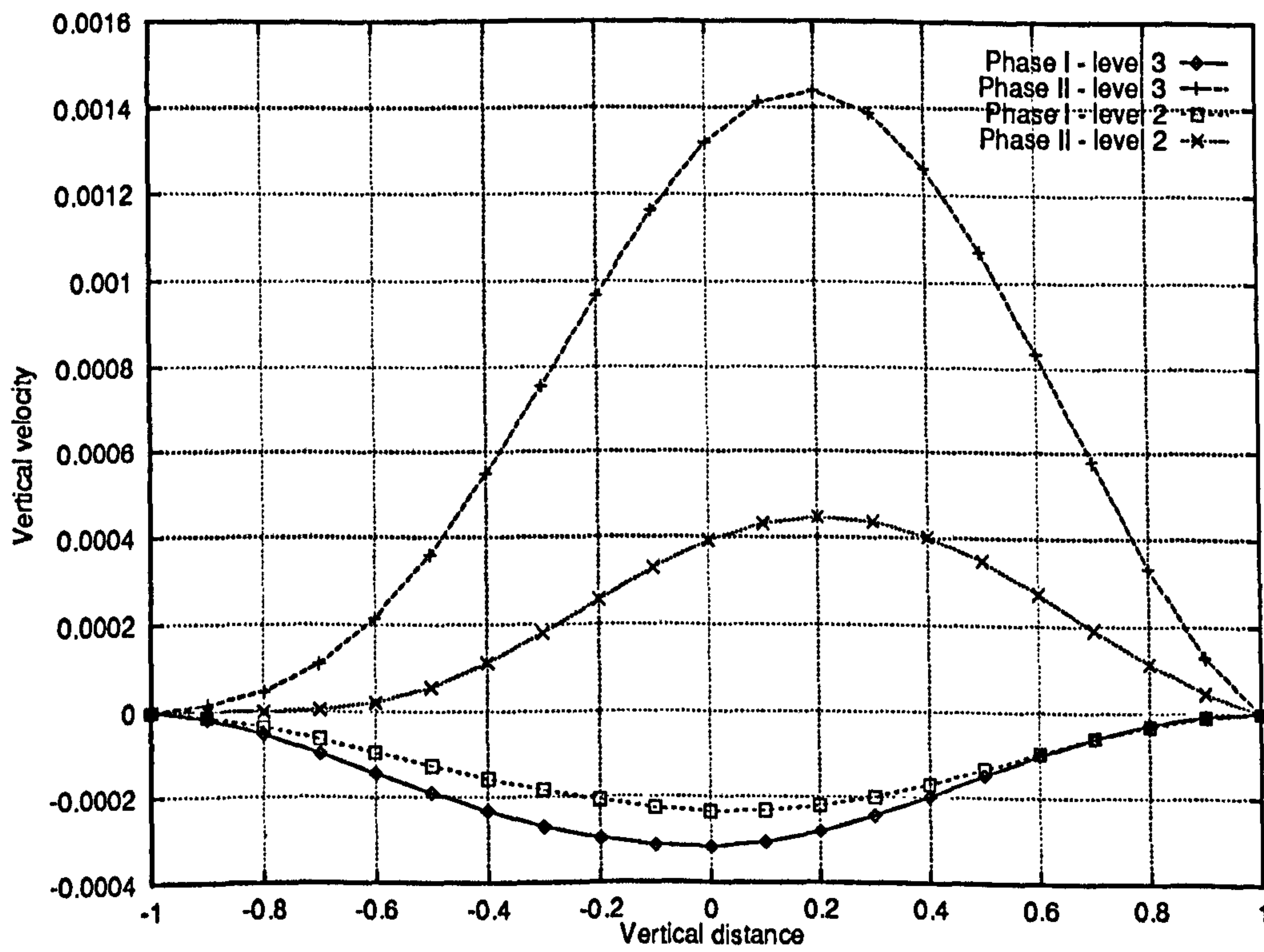


Figure 5.100: Multiphase backward-facing step problem – Vertical velocity profiles along the line $x = 10.0$ – Comparison of pang-multiphase solutions at levels 2 and 3 – Note the scale

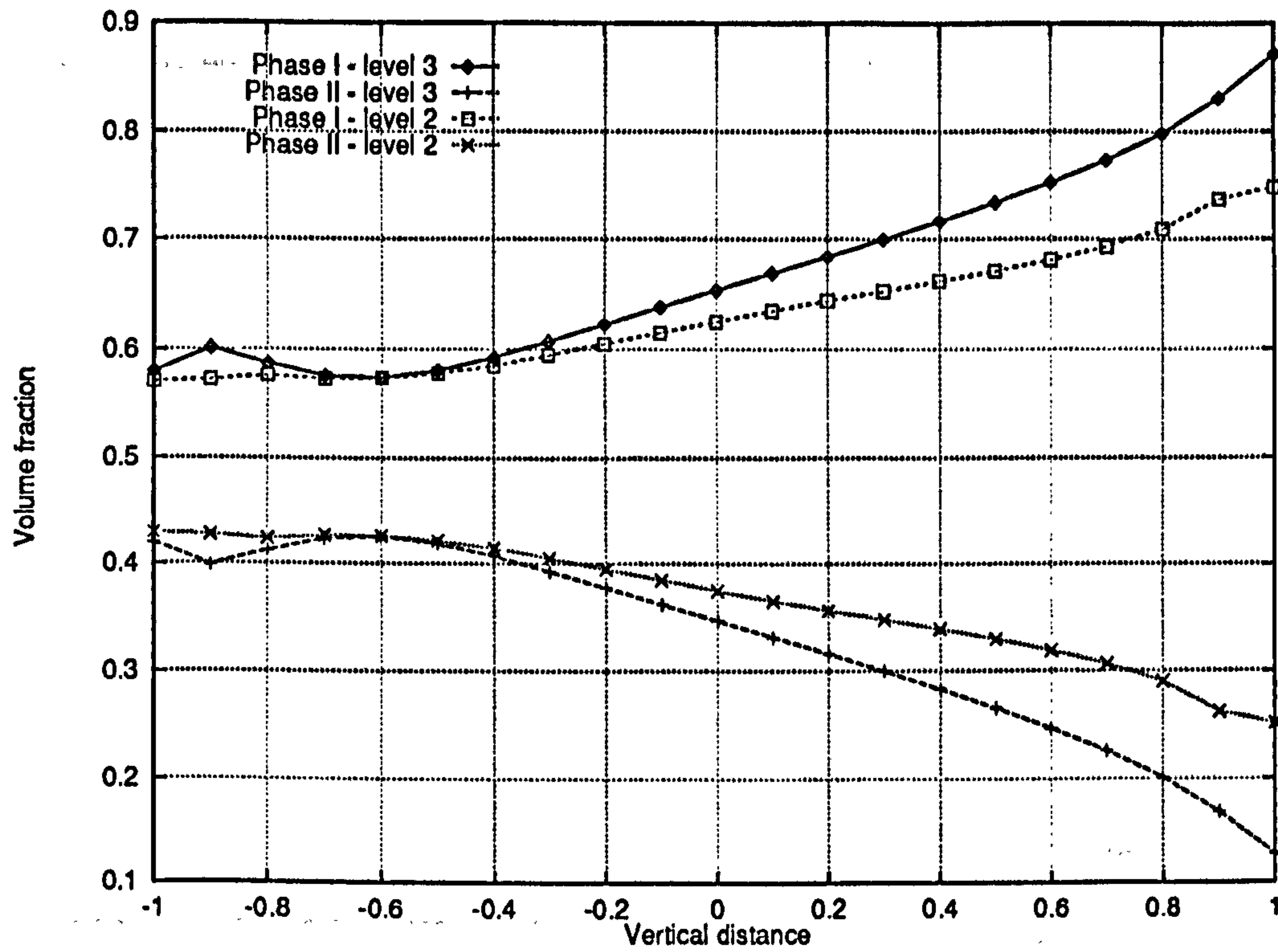


Figure 5.101: Multiphase backward-facing step problem – Volume fraction profiles along the line $x = 10.0$ – Comparison of pamg-multiphase solutions at levels 2 and 3

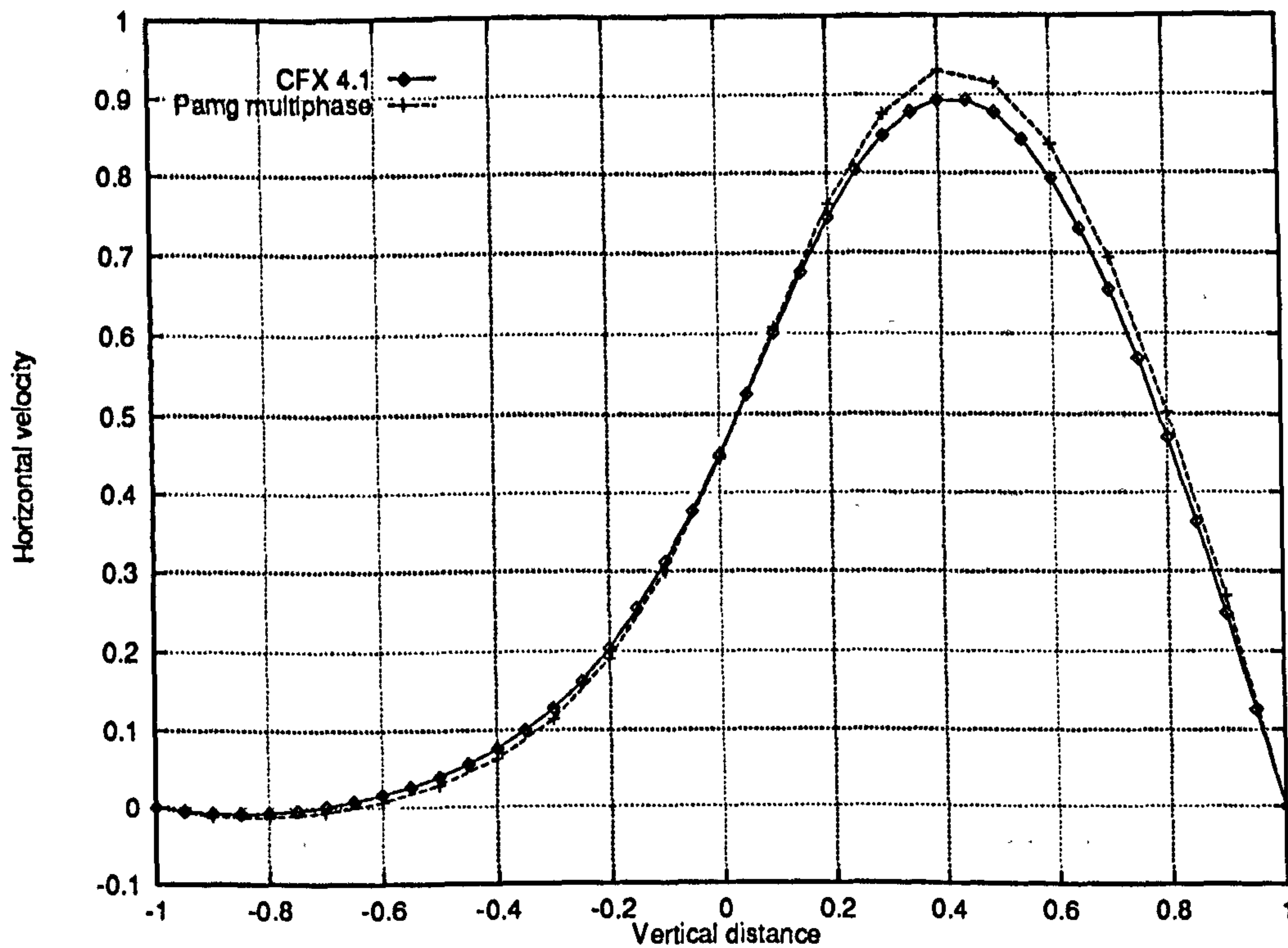


Figure 5.102: Two-phase backward-facing problem – Horizontal velocity profile along the line $x = 3.5$ (phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

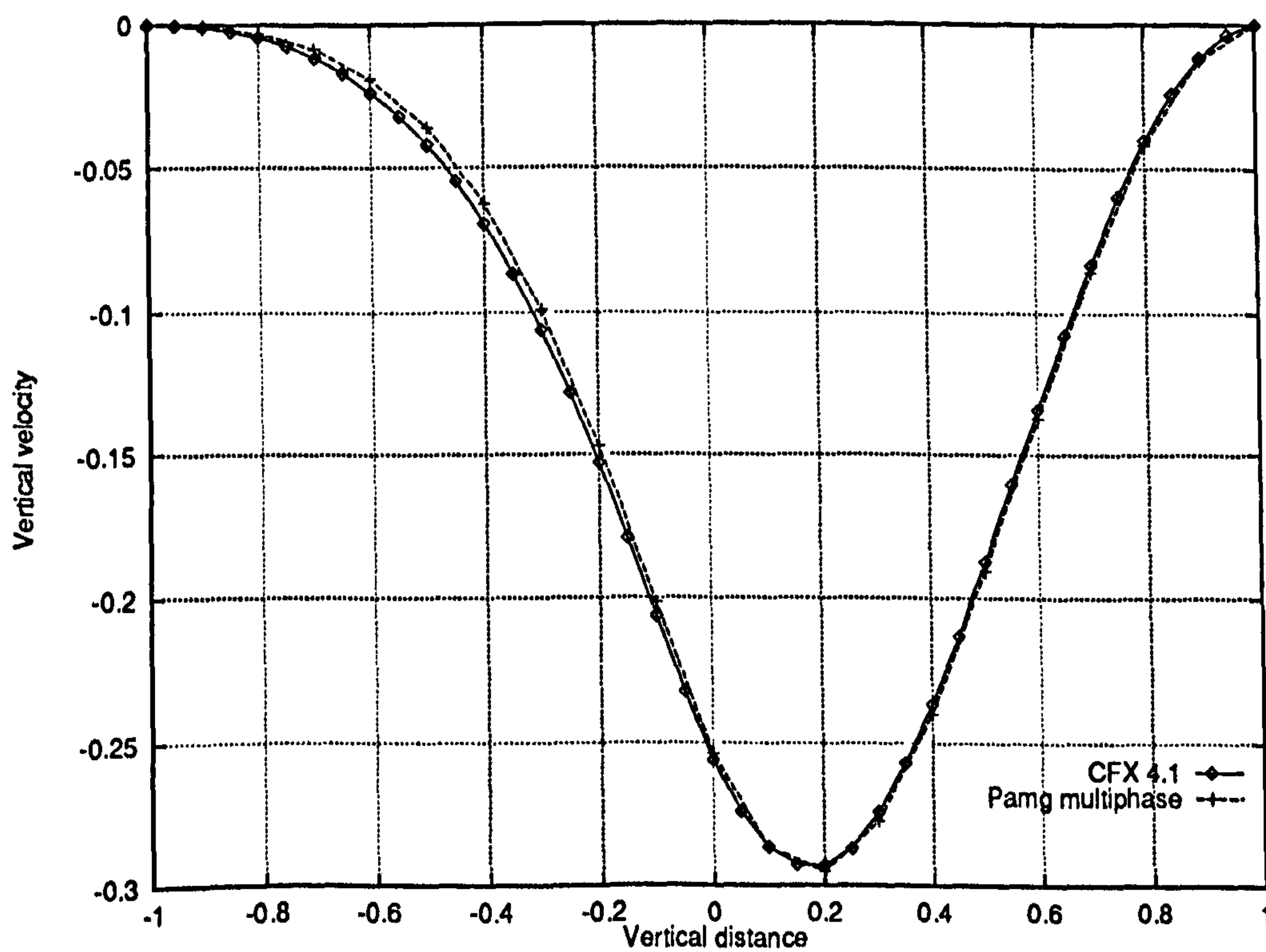


Figure 5.103: Two-phase backward-facing problem – Vertical velocity profile along the line $x = 3.5$ (phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

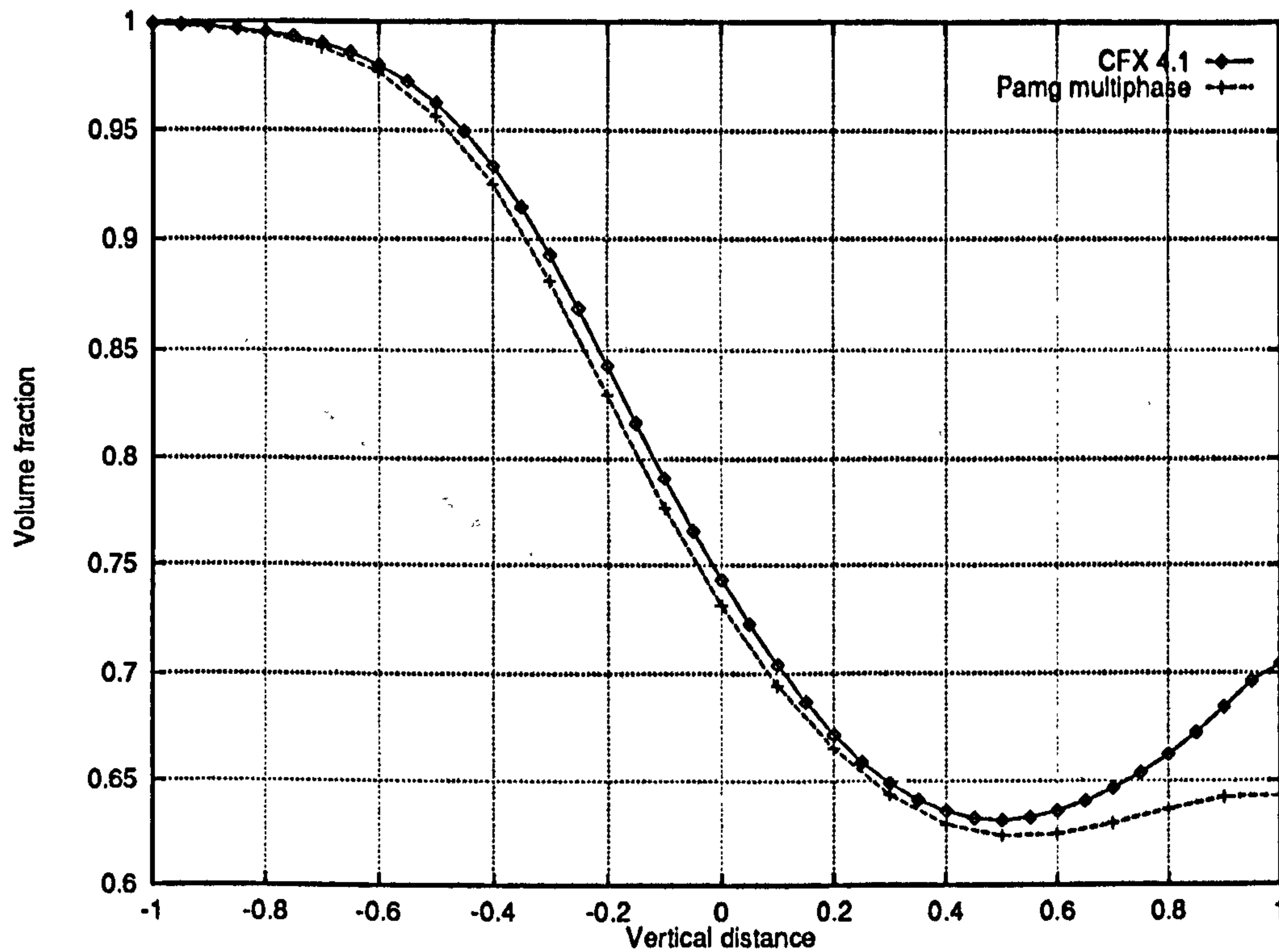


Figure 5.104: Two-phase backward-facing problem – Volume fraction profile along the line $x = 3.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids)

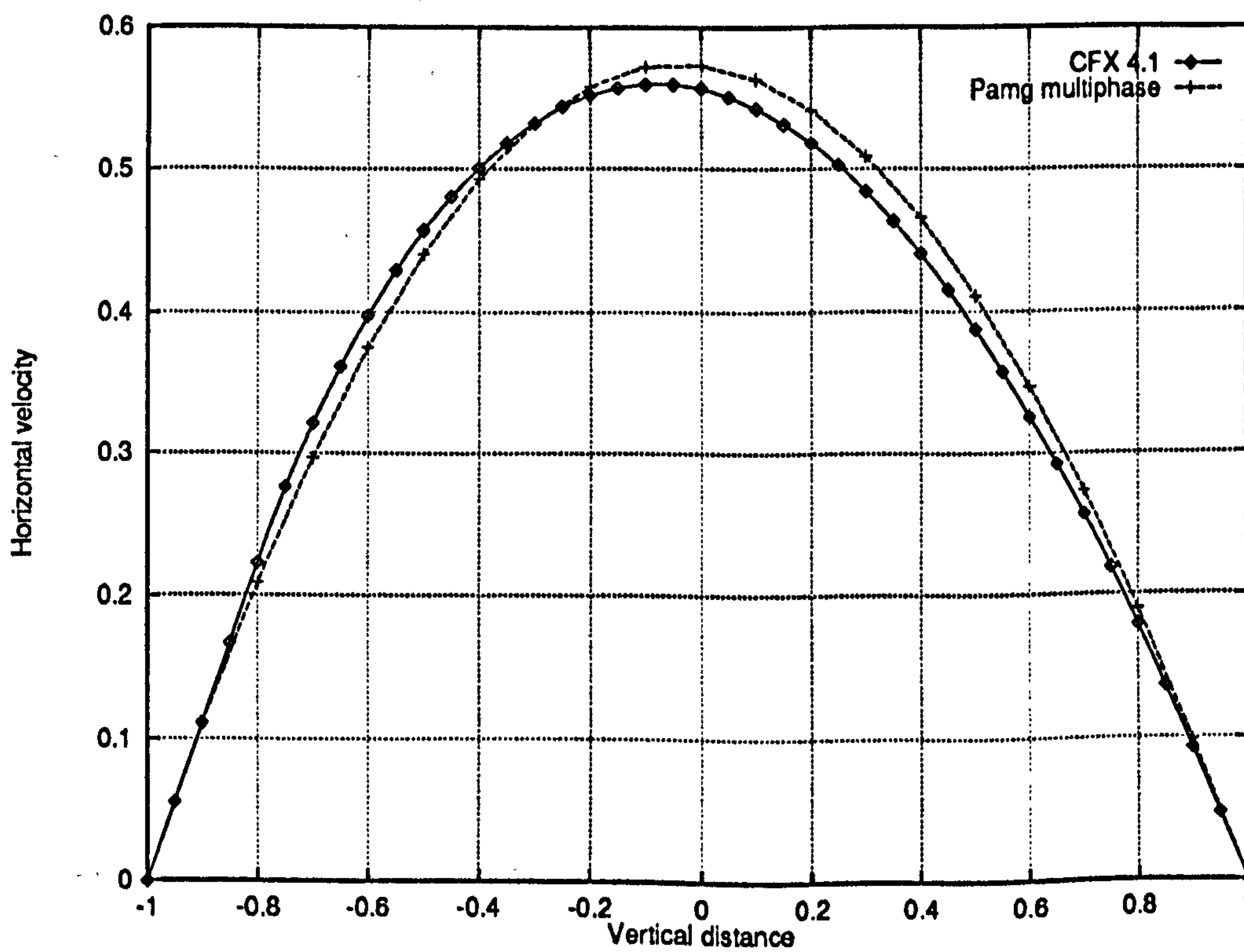


Figure 5.105: Two-phase backward-facing problem – Horizontal velocity profile along the line $x = 10.0$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

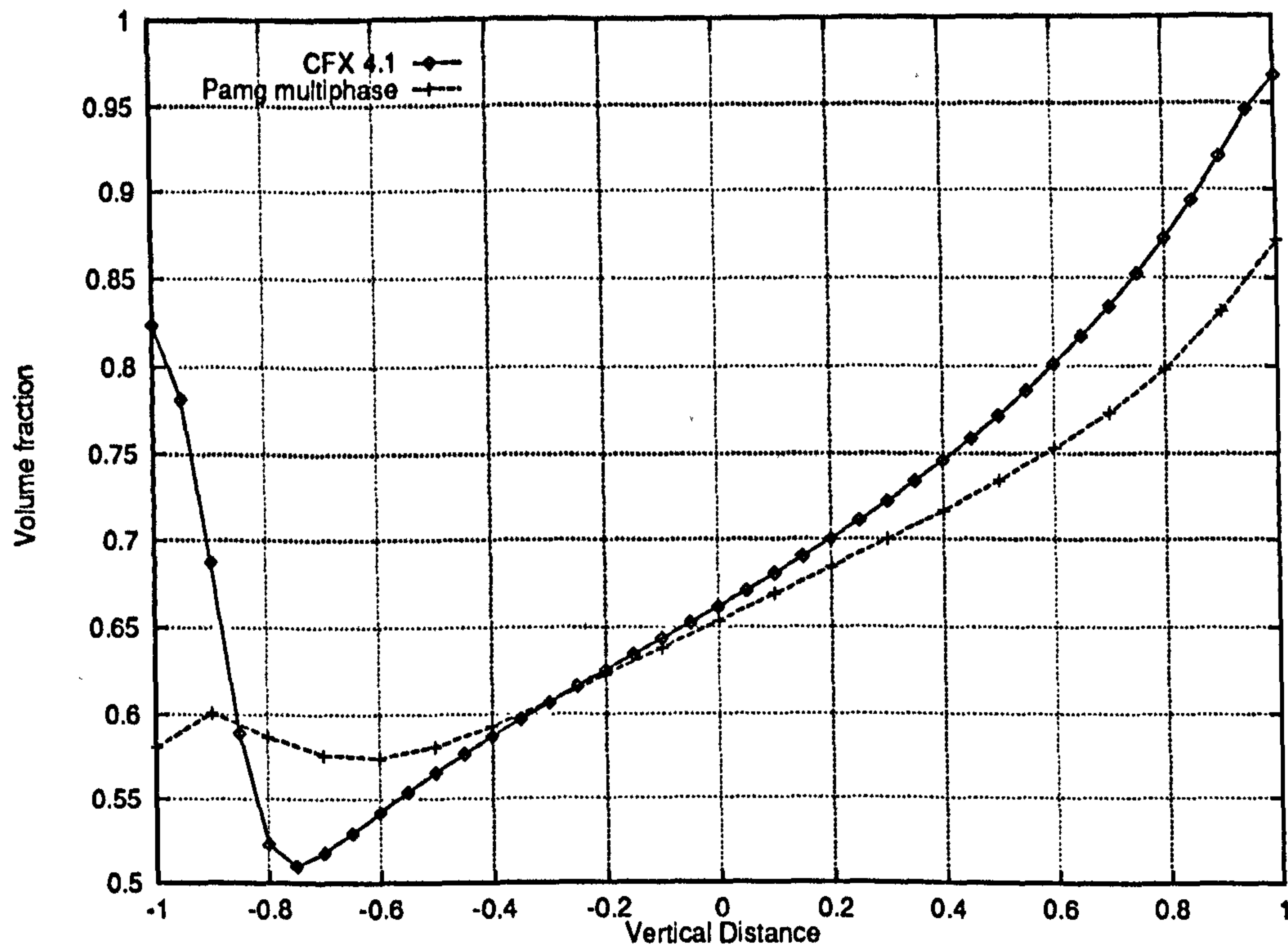


Figure 5.106: Two-phase backward-facing problem – Volume fraction profile along the line $x = 10.0$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – See text for further comments

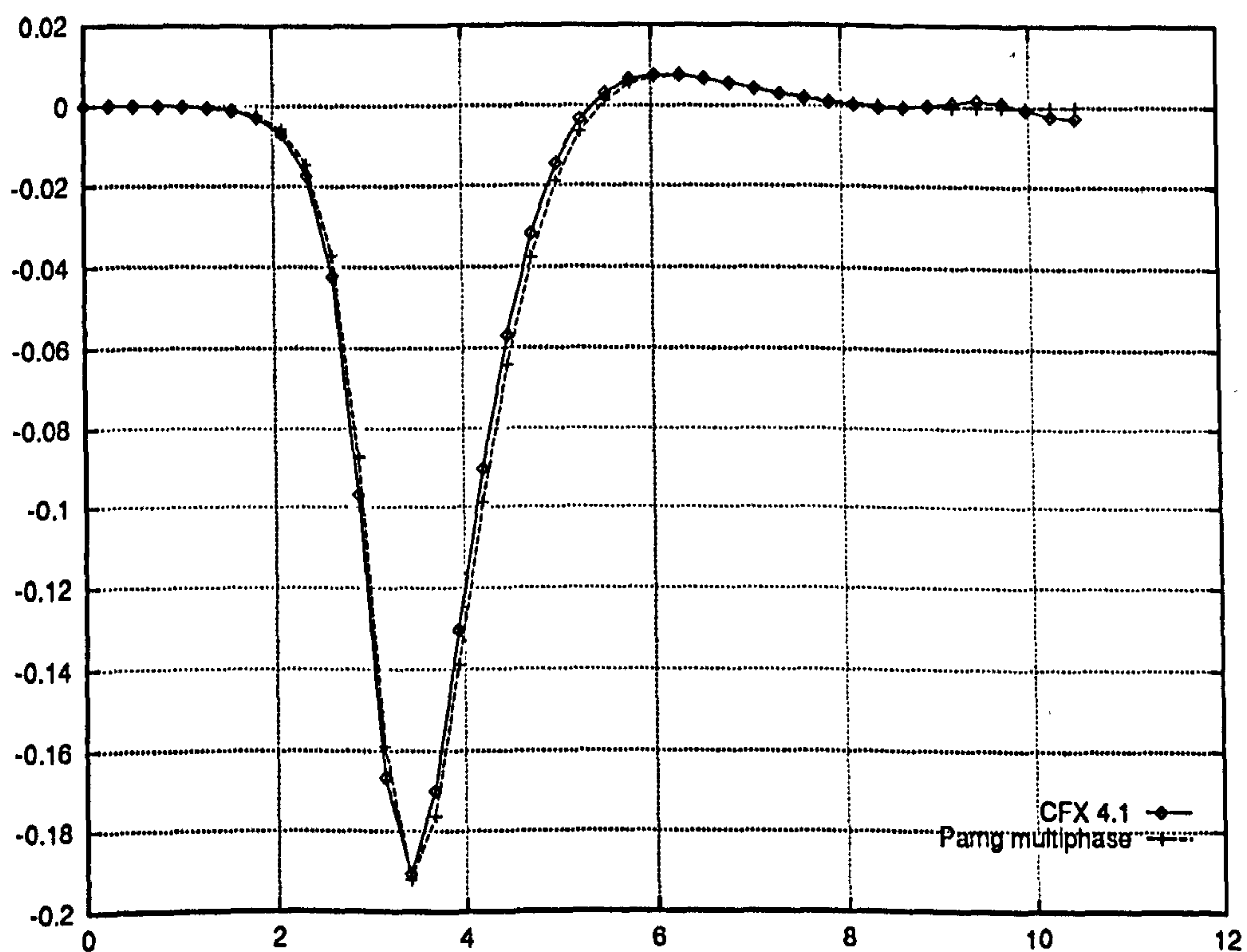


Figure 5.107: Two-phase backward-facing problem – Vertical velocity profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – Note the slight spurious oscillations in the CFX 4.1 solution near the outlet

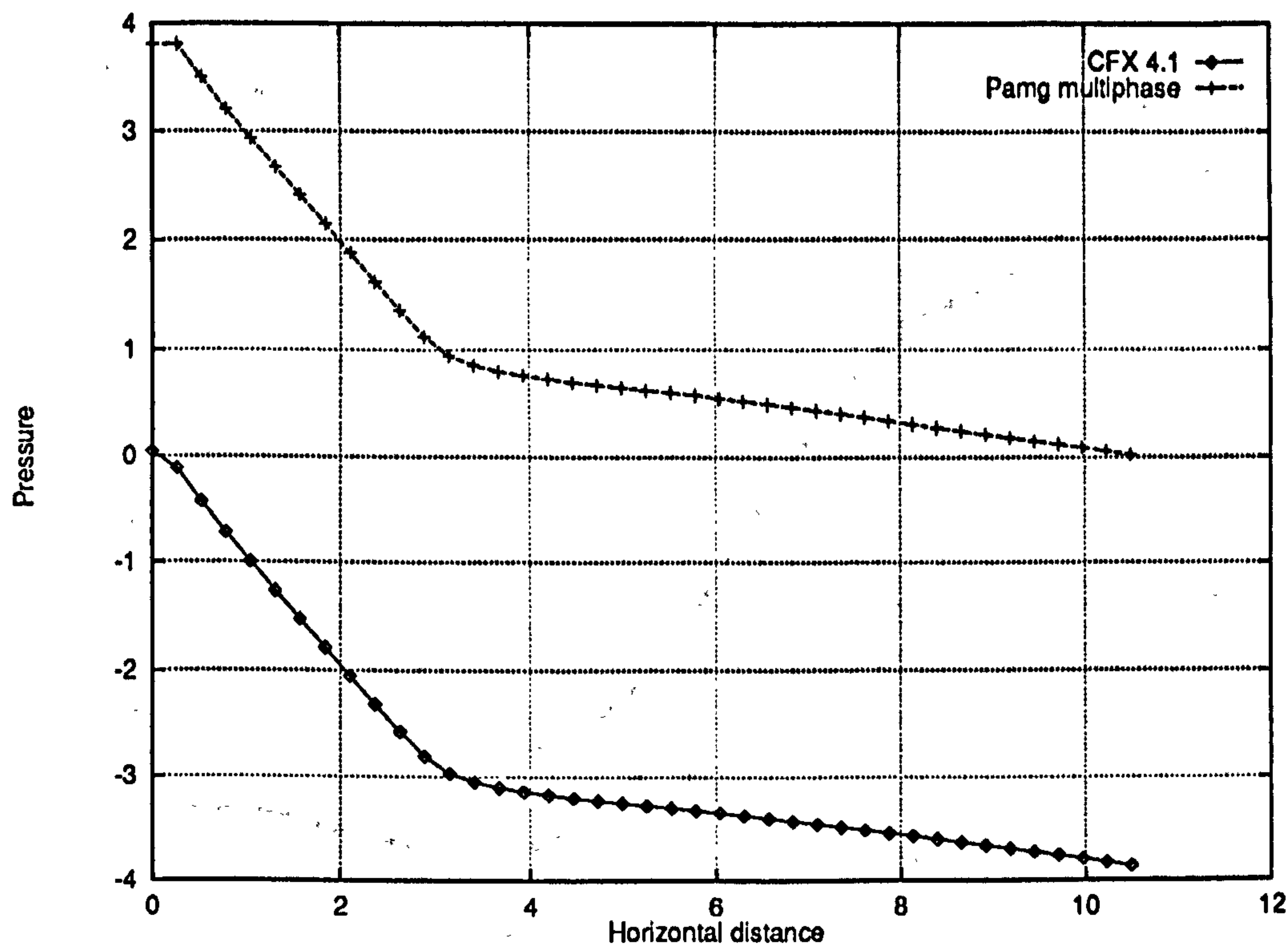


Figure 5.108: Two-phase backward-facing problem – Pressure profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

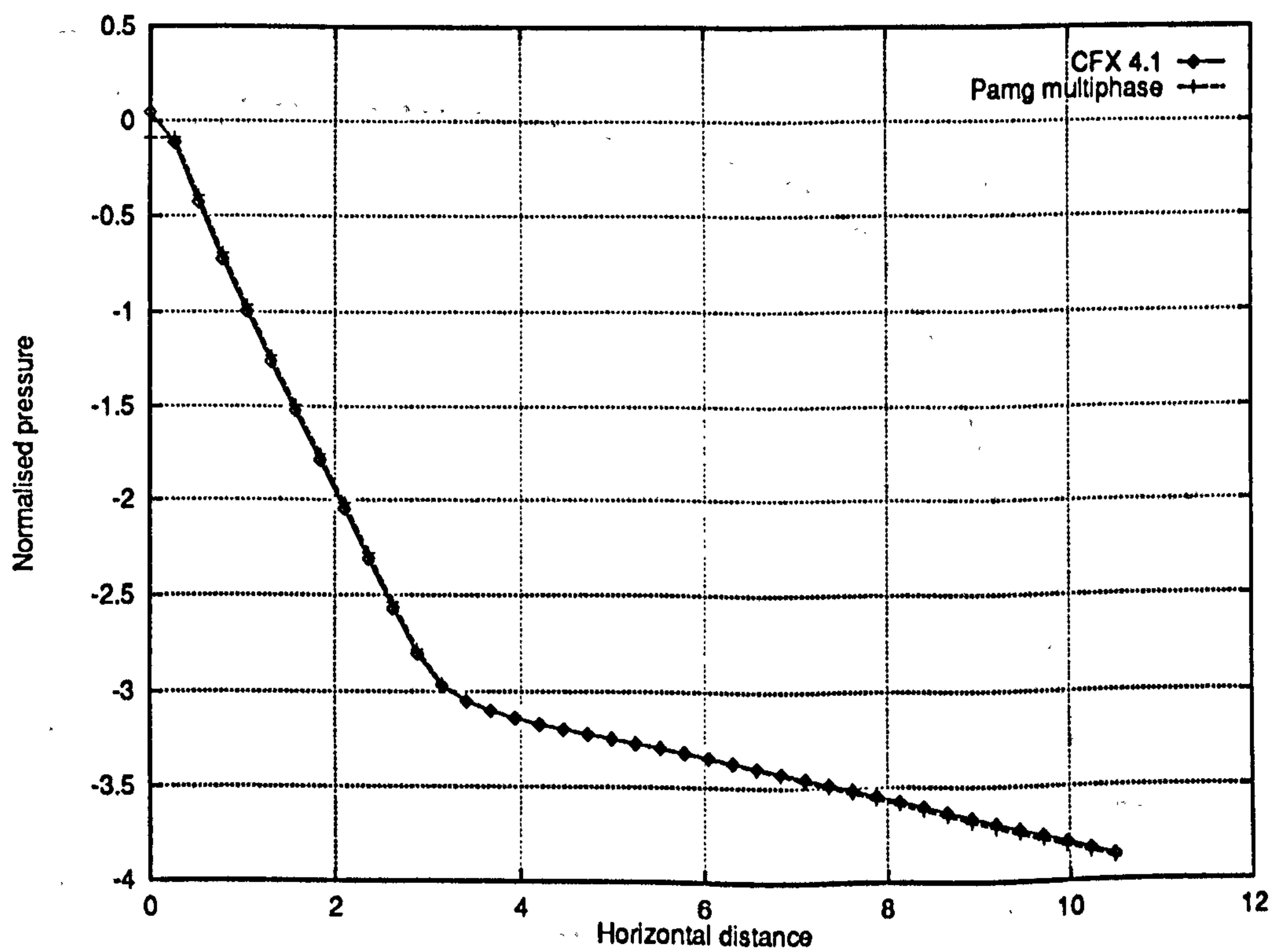


Figure 5.109: Two-phase backward-facing problem – Pressure profile along the line $y = 0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids after shifting

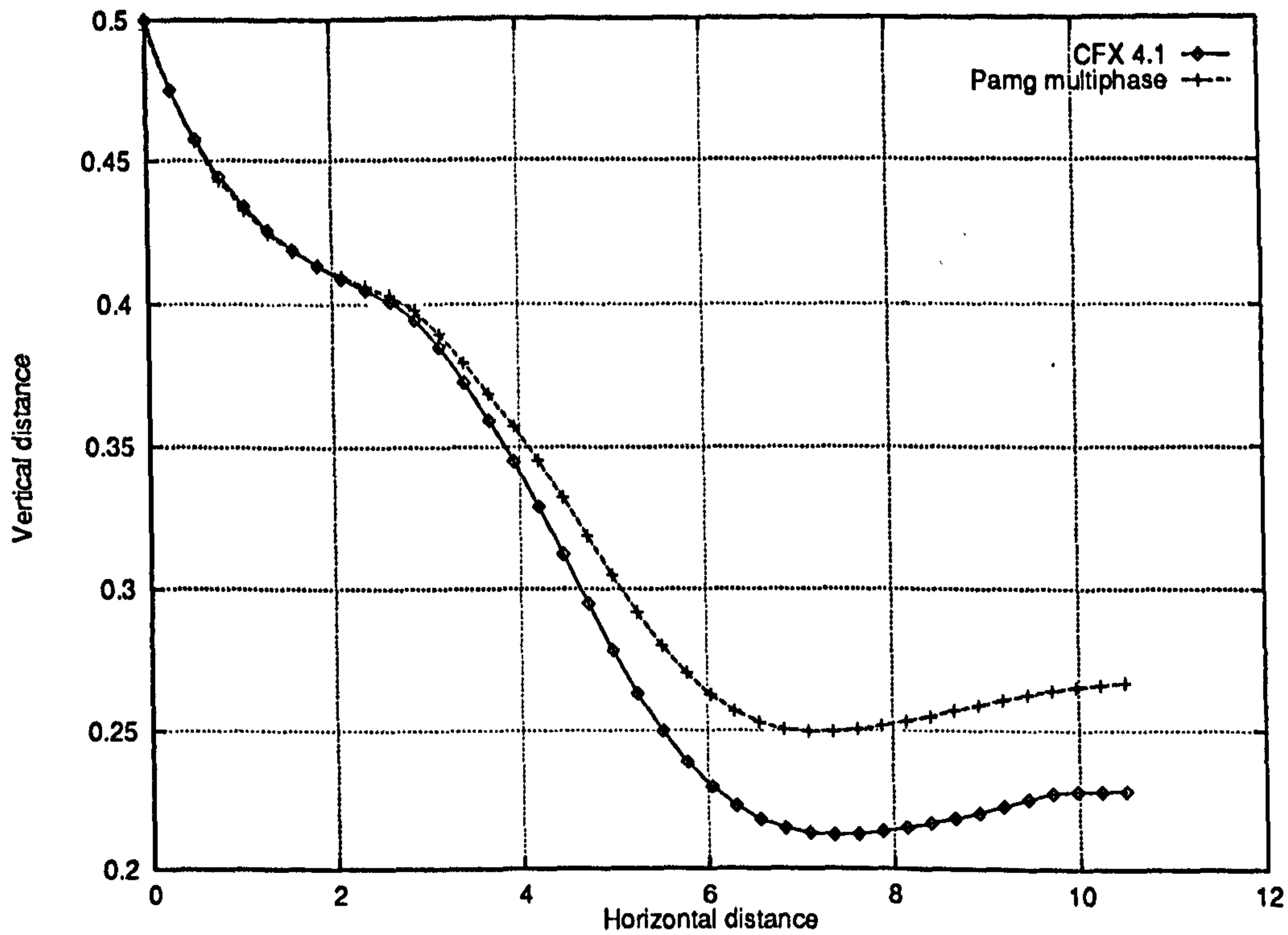


Figure 5.110: Two-phase backward-facing problem – Volume fraction profile along the line $x = 10.0$ (Phase 2) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

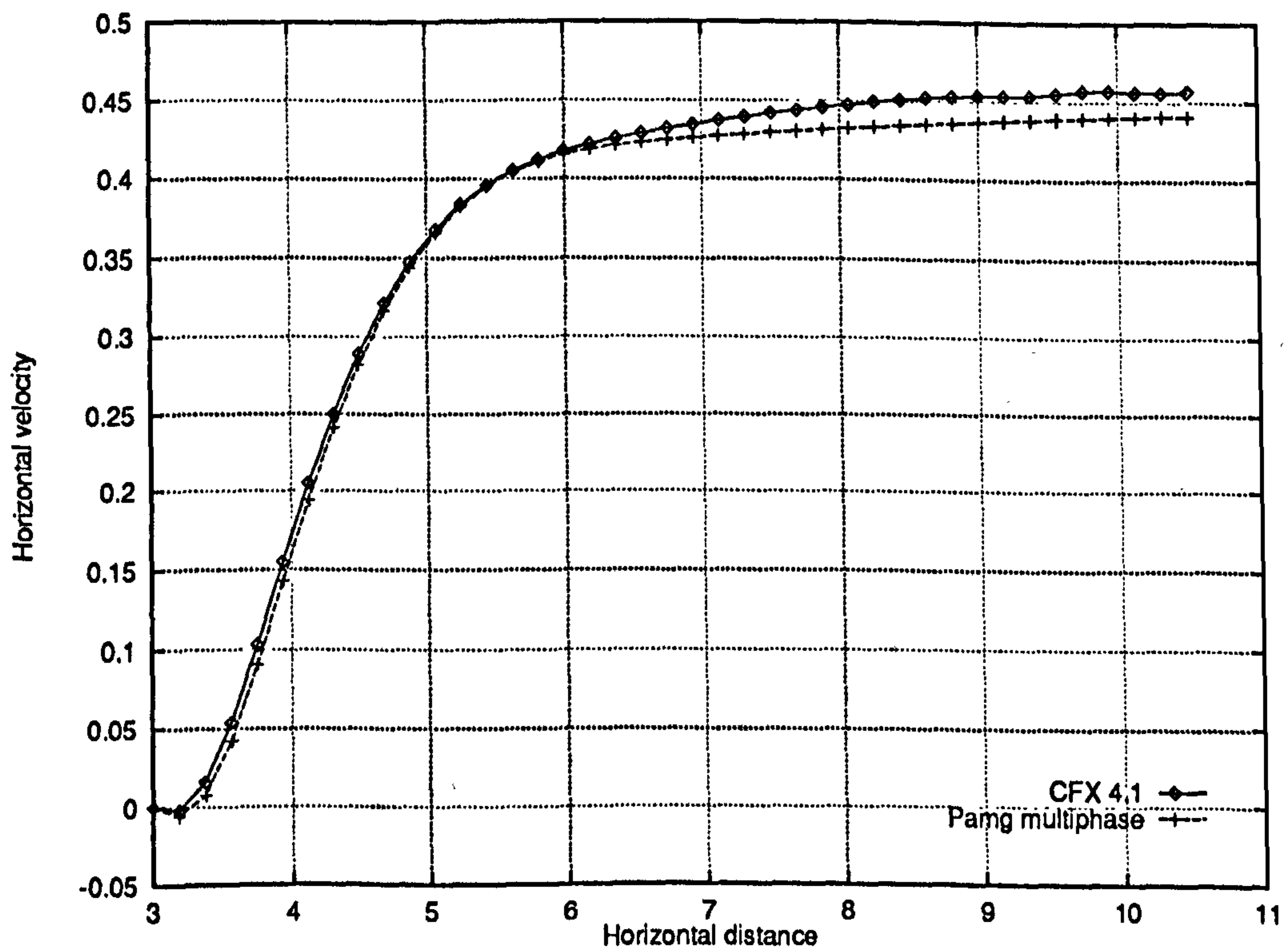


Figure 5.111: Two-phase backward-facing problem – Horizontal velocity profile along the line $y = -0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids

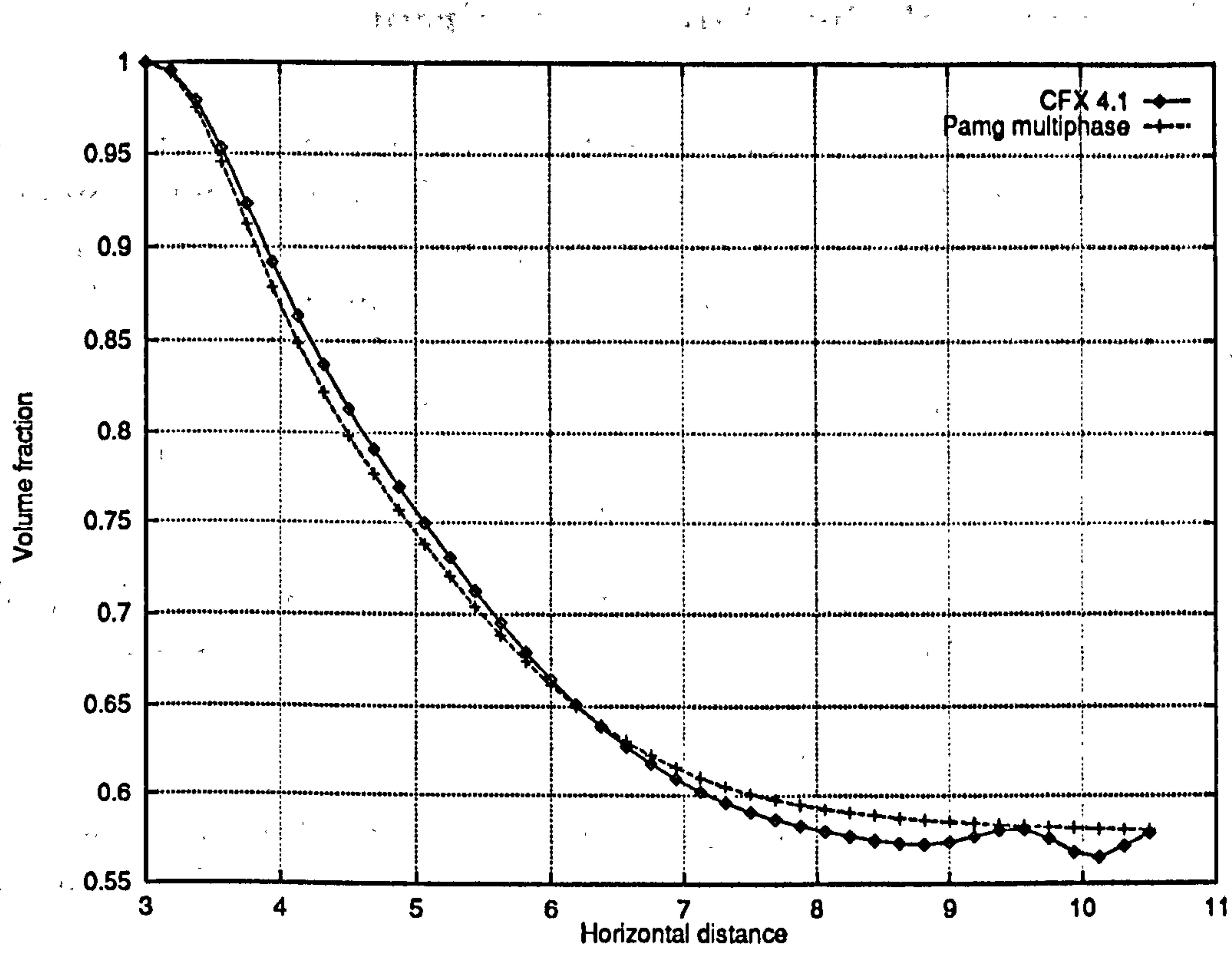


Figure 5.112: Two-phase backward-facing problem – Volume fraction profile along the line $y = -0.5$ (Phase 1) – Comparison of the pamg-multiphase and CFX 4.1 solutions on level 3 grids – Note the slight spurious oscillations in the CFX 4.1 solution near the outlet

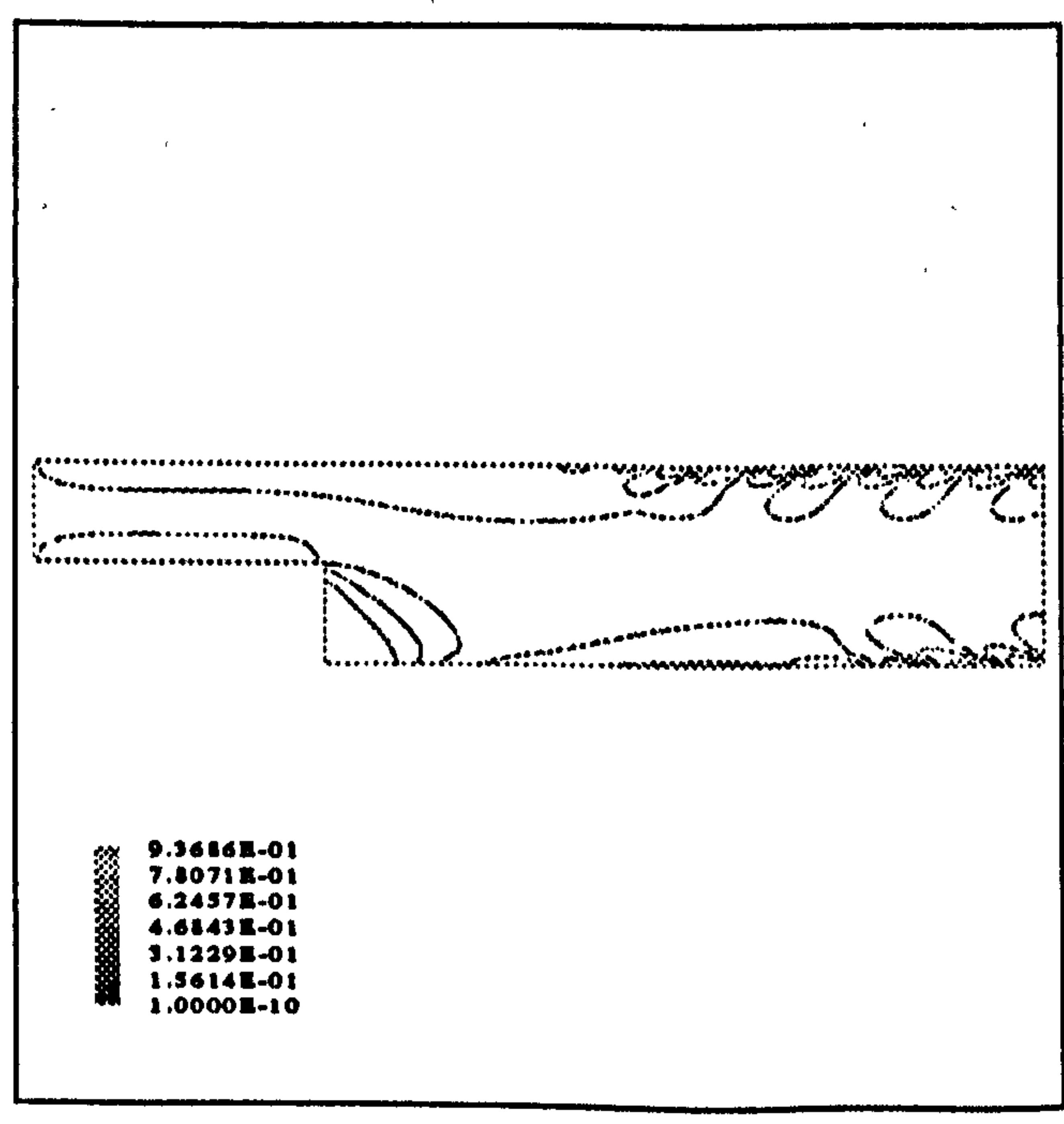


Figure 5.113: Two-phase backward-facing problem – CFX 4.1 solution – contours for the volume fraction of phase 1 – Note the slight spurious oscillations

5.3.4 Estimates of the Truncation Errors

When solving a system of equations by iterative means, an important issue is to determine the accuracy to which the solution should be sought. In the present case where the equations result from a discretisation process, it is usually considered that the residuals, which to some extent measure the numerical error, should be of the same order of magnitude as the truncation error of the discretisation. In this section, we present some estimates of the truncation errors for each of the multiphase problems we have solved. These estimates will be useful when the convergence rates of the multigrid solver are studied in Section 5.4.

Estimates of the truncation errors are computed automatically by the FAS multigrid solver. See Section 2.5.8 where it is shown that the defect $\tau_\kappa^{\kappa-1}$ defined as:

$$\tau_\kappa^{\kappa-1} \equiv \mathcal{L}^{\kappa-1}(\hat{I}_\kappa^{\kappa-1}\tilde{u}^\kappa) - I_\kappa^{\kappa-1}(\mathcal{L}^\kappa\tilde{u}^\kappa).$$

is an approximation of the truncation error on the grid $\Omega^{\kappa-1}$ where Ω^κ is the finest grid. The truncation error varies throughout the computational domain and we choose the median value of defect as a measure.

Since our discretisation is second order accurate for all terms except the volume fractions in the continuity equations, we expect the truncation error to vary by a ratio of 4 between consecutive grid levels, at least when the volume fractions do not change significantly in space.

Examination of Tables 5.1 to 5.3 reveals that, as expected, the median value of the truncation errors is reduced as grids are refined. However, the rate of decrease is usually less than 4. This being particularly noticeable for the channel flow problem. Such an atypical behaviour may be due to the fact that the volume fractions are grid dependent. Correct reduction factors are indeed observed near the median line of the channel (see Table 5.4), where the volume fractions are both relatively constant and grid-independent.

Grid Level	Median truncation error	Reduction factor
1	1.49×10^{-3}	-
2	2.71×10^{-3}	0.55
3	1.74×10^{-3}	1.55
4	1.24×10^{-3}	1.40
5	9.32×10^{-4}	1.33

Table 5.1: Two-phase channel flow – Estimation of the truncation error for different grid levels

Grid Level	Median truncation error	Reduction factor
1	1.35×10^{-2}	-
2	7.59×10^{-3}	1.78
3	4.15×10^{-3}	1.82
4	1.67×10^{-3}	2.48

Table 5.2: Two-phase T-junction problem – Estimation of the truncation error for different grid levels

Grid Level	Median truncation error	Reduction factor
1	1.43×10^{-3}	-
2	1.57×10^{-3}	0.91
3	1.27×10^{-3}	1.23

Table 5.3: Two-phase backward-facing step problem – Estimation of the truncation error for different grid levels

Grid Level	x -coordinate	y -coordinate	$\tau_{\kappa}^{\kappa-1}$	Reduction factor
1	2.25	0.75	1.49×10^{-3}	-
2	2.375	0.625	8.56×10^{-4}	1.74
3	2.4375	0.5625	2.33×10^{-4}	3.67
4	2.46875	0.53125	4.48×10^{-5}	5.20
5	2.484375	0.510625	1.27×10^{-5}	3.52

Table 5.4: Two-phase channel flow – Estimation of the (point-wise) truncation error for different grid levels near the median line of the channel – The use of a staggered grid prevents the estimation to be done at the same point on different grids

5.3.5 Conclusions

We have considered the accuracy of the **pamg-multiphase** solutions for both single phase flows and multiphase flows.

In the single phase case, three validation cases have been chosen: a simple channel flow, a flow through a T-junction and a flow through a backward-facing step. These provide a good selection of difficult flow patterns, mainly mixing and recirculation. The **pamg-multiphase** answers agree with the solutions provided by **pamg** and **CFX 4.1**, two widely validated CFD solvers.

Multiphase flows have been simulated for the same three geometries. In each case, significant phase separation occurs, implying that the solution is not trivial even for simple geometries. In the case of the backward-facing step where recirculation zones are an important feature of the flow, it is possible that for certain flow configurations, the steady problem is not well-posed.

Despite the level of complexity of the equations, correct solutions are obtained. The degree of agreement with the answers provided by **CFX 4.1** is also very good. However, it is not perfect. This is not surprising since the two solvers differ widely in many respects. Furthermore, it has been established that the **pamg-multiphase** solutions are grid independent to a very large extent. The only exception is the volume fraction which may be highly dependent on the grid size, particularly for "easier" problems such as the two-phase channel flow. Finally, both cases where inter-phase momentum transfers are allowed or neglected have been validated against the **CFX 4.1** solutions.

The ability of **pamg-multiphase** to handle difficult multiphase flow patterns such as complete phase separation or recirculation regions has been established, provided that the steady problem is well-posed.

Having established the accuracy of the **pamg-multiphase** solutions, we now turn our attention to the question of the *performance* of the solver and examine the rates of convergence observed for the different test cases which we have studied. We will then consider computations on adaptive grids.

5.3.6 Volume Fraction Maps Obtained from Pamg-multiphase Solutions

We group here, for convenience of reproduction, colour maps of the volume fractions fields for our different multiphase test cases. The figures are referenced in the main text.

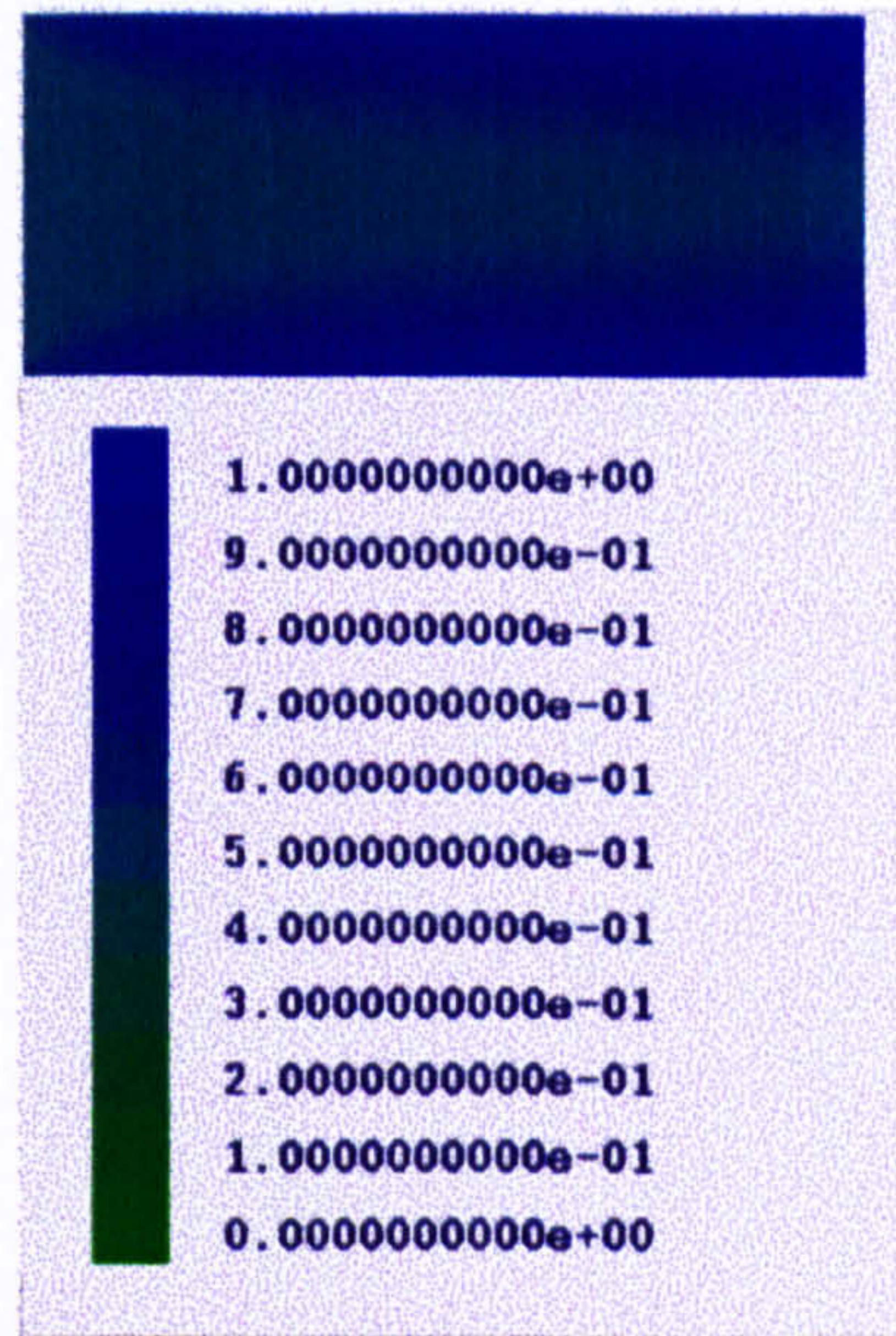


Figure 5.114: Two-phase channel flow – Volume fraction map for phase 1 – See page 134

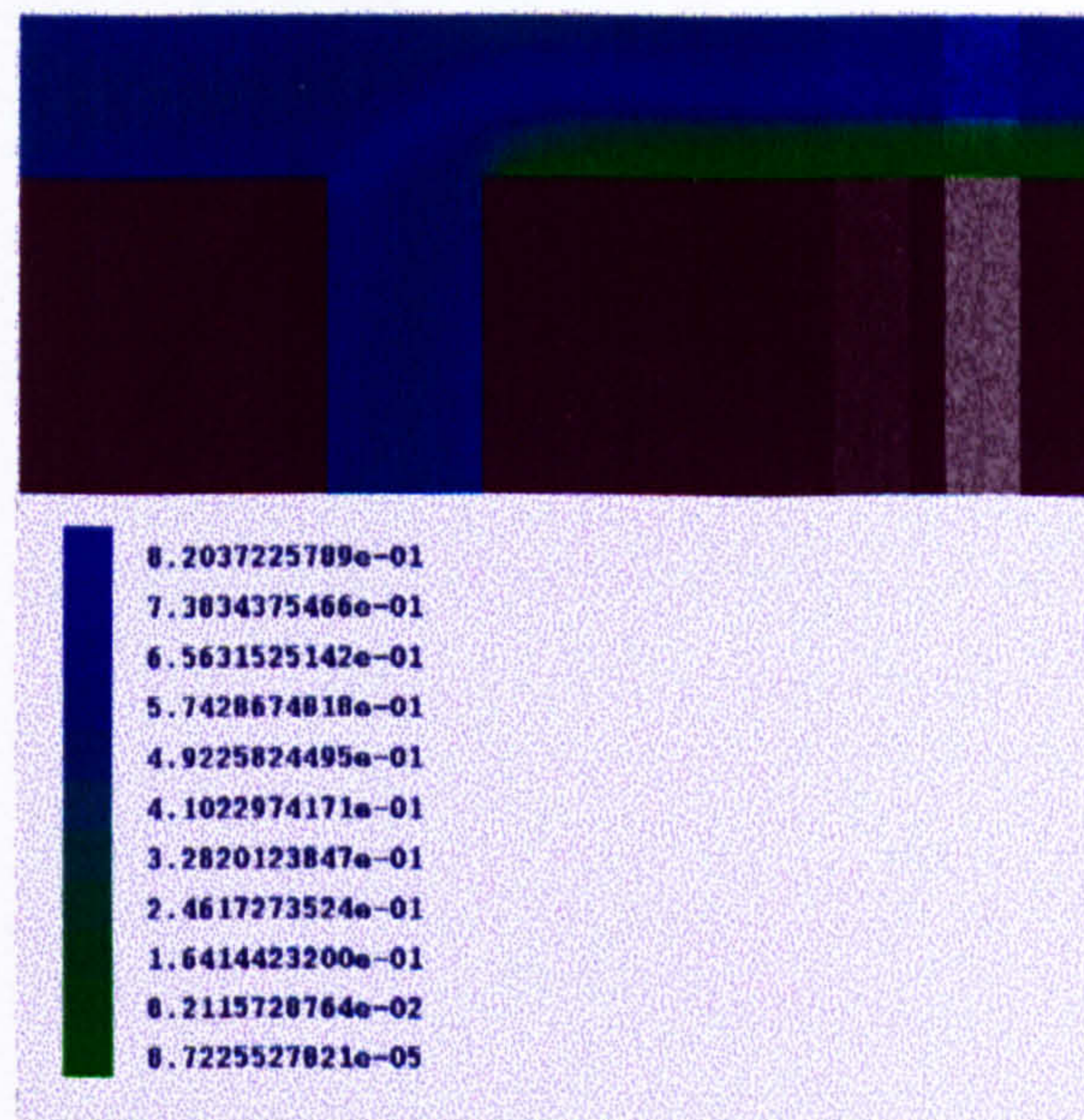


Figure 5.115: Multiphase T-junction problem – Volume fraction map for phase 1 – See page 153

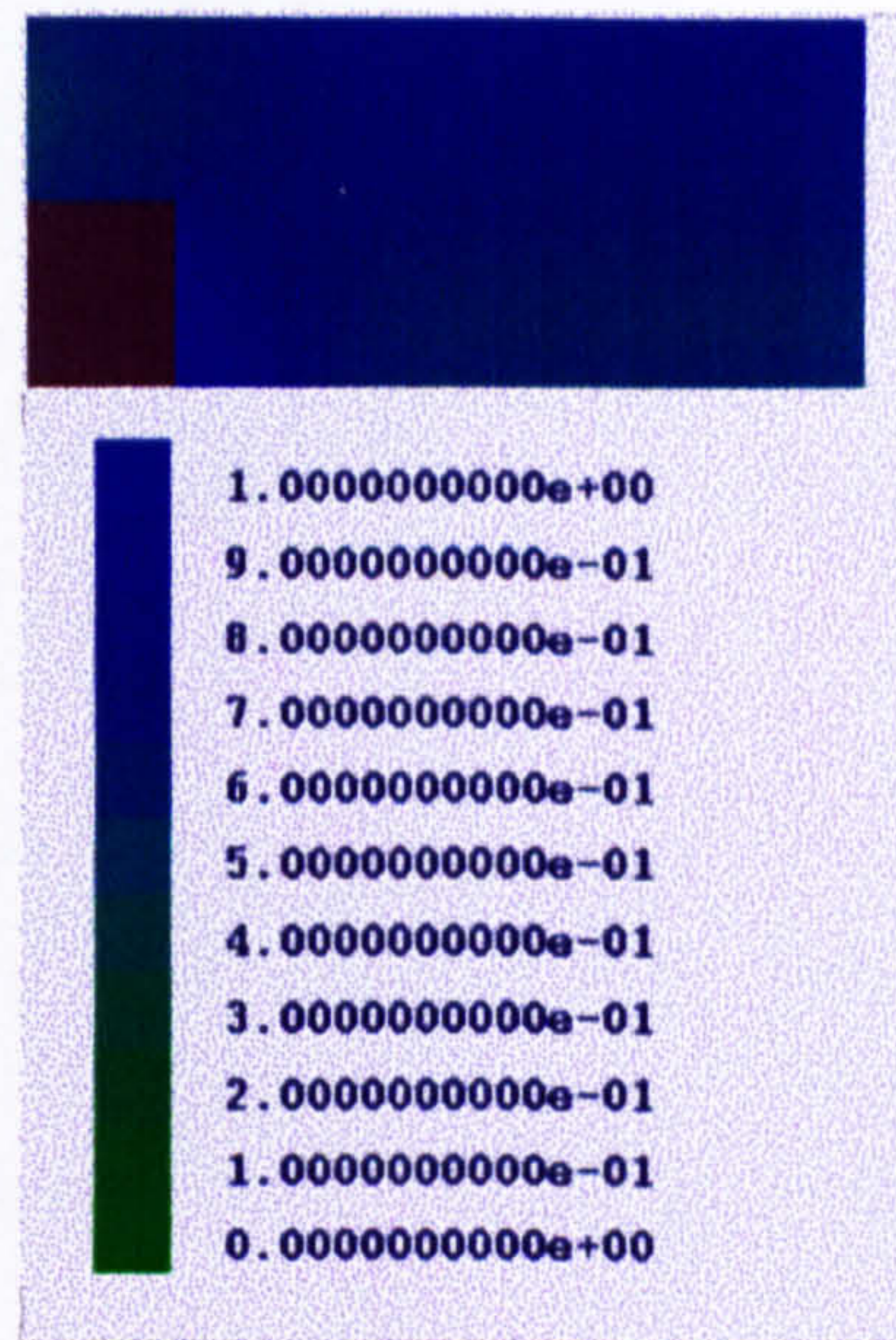


Figure 5.116: Multiphase backward-facing step problem – Volume fraction map for phase 1 (see page 169) showing the exclusion from the region behind the step of the lighter phase (phase 2)



Figure 5.117: Multiphase backward-facing step problem at higher Reynolds numbers – Volume fraction map for phase 1 (see page 233) showing more extended recirculation zones

5.4 The performance of the pamg-multiphase Solver

5.4.1 Introduction

We have established the accuracy of the solutions provided by pamg-multiphase. We will now study the performance of the solver. Given that pamg-multiphase is an iterative solver, performance will be evaluated using the following criteria:

- *Convergence* (or more precisely numerical convergence): this is the property that the computational error, i.e. the difference between the exact solution of the discrete equations and its approximation, is small. Since the error is not available in general, convergence is usually monitored by the reduction of the residual, measured using a suitable vector norm. The quantity which best quantifies efficiency is the convergence factor. It is usually its asymptotic value which is of interest in order to eliminate the influence of the initial guess.
- *Robustness*: this is the property that convergence does not depend on the quality of the initial guess.

The robustness of the pamg-multiphase is immediately established since all the results presented in Sections 5.2 and 5.3 were obtained with simple standard initial guesses. Generally, the velocity fields and pressure fields were set to zero, while the volume fractions were set to 0.5 for each phase. This contrasts sharply with many segregated solvers for which it is very important that the initial guess should be of good quality in order to achieve a converged solution. However, we have also observed novel singularities which are specific to multiphase flows and which may complicate the application of our multigrid solver. See Section 5.4.10 for a further discussion.

In this section, we therefore focus on the convergence factors observed with pamg-multiphase. The first observation is that multigriding greatly improves the convergence rates of the embedded single grid solver — FAS multigrid procedures are very successful acceleration techniques when applied to multiphase flows. The ideal case would be that the convergence factors are grid-independent, ensuring that the method is optimal order-wise. This is not observed. The situation here is quite complex and the effect of grid size on convergence factors is discussed in detail.

It should be noted that asymptotic convergence factors which to a large extent are grid independent, are observed in parts of the convergence histories. This indicates that the multigrid procedure performs as well as can be expected and that the relative degradation of convergence factors is caused by specific features of the equations. It is not possible to blame non-linearities in a general sense. A comparison with the convergence factors of the single phase solver pamg reveals that multigrid convergence factors can be grid independent for *some* non-linear problems. The remainder

of this section is therefore devoted to an examination of the reasons which may explain why multigrid convergence factors are sometimes degraded for the system of the multi-fluid equations. Two lines of enquiries are possible: either (i) the solution algorithm has limitations or, (ii) the equations themselves have characteristics which hinder their resolution by a multigrid procedure, without precluding it.

Both lines of investigations are examined below. Firstly, we assess the effect of under-relaxation, of different multi-grid cycling strategies and of the order of accuracy of the grid transfers. We also consider the possibility that errors in the outlet boundary conditions dominate the convergence factors.

We then focus our attention on specific features of the multi-fluid equations: the influence of source terms, of difficult flow patterns (“dry-out” zones and recirculation regions) and of the multiphase diffusive tensor. In the author’s opinion, the behaviour of this term is central to analysing the multigrid convergence rates. Evidence accumulated during the course of this project indicates that as diffusive effects become dominant, obtaining a solution requires more and more effort. Hypotheses and reasons for this phenomenon are addressed as thoroughly as possible.

To facilitate the discussion of efficiency, we introduce the concept of *work* unit which is the cost of one relaxation sweep on the finest grid on which the solution is sought. The work unit is obviously directly proportional to the number of unknowns on the finest grid. The convergence of *pamg-multiphase* is measured in terms of an average residual which is equal to the L_2 norm of the residual normalised by the number of discrete equations on the finest grid ⁵:

$$resid = \sqrt{\frac{\sum_{ij} \sum_{\alpha=1}^M \sum_k^3 (f_{ijk}^\alpha)^2}{3NM}}$$

where the subscripts i and j refer to a particular computational cell, k sweeps the number of equations defined on a particular cell per phase, α denotes the phase, N is the number of computational cells and M the number of phases. With this choice of vector norm, we have the property that if $f_{ijk}^\alpha = 1 \quad \forall i, j, k, \alpha$, then $resid = 1$.

Whenever an iterative algorithm is used to solve a system of equations, an important issue is to determine the appropriate level of accuracy to which the solution should be obtained. In the present case, where the equations arise from the discretisation of PDEs, it is sufficient to require that the numerical error is of the same order of magnitude as the truncation error. In other words, the discrete solution is acceptable if $resid \sim \|\tau^h\|$. In Section 5.3.4, we provide estimates of the truncation errors for the different multiphase test cases we have considered. In all cases, the residuals were reduced to a level far lower than the truncation error. This is not necessary

⁵The number of equations is taken to be $3MN$ instead of $4MN$ because we have MN closure equations which are always satisfied during the course of the computations

for the purpose of obtaining a meaningful solution but allows a thorough study of the performance of our multigrid solver.

The FAS multigrid algorithm has been described in Chapter Four. For all uniform grid computations, non-conservative transfers are systematically used.

5.4.2 Multigrid Acceleration

We first remark that the multigrid method significantly accelerates the speed of convergence of the single grid quasi-Newton coupled solver.

For the two-phase channel flow problem, a 3 grid computation using a “F” cycle is 5 times faster than the corresponding single grid computation (Figure 5.118) and it was verified that the solutions are identical (Figure 5.119).

For the multiphase T-junction problem, the multigrid method works even better as an acceleration technique since it allows speed-up of 20 compared with single grid computations (Figure 5.120). The comparison here was also done on level 3 uniform grids.

At this juncture, we have established that multigrid procedures work very well for multiphase flows. As in the single phase case, they accelerate the convergence factors of the embedded local quasi-Newton solver. Assessing this acceleration is straightforward because the multigrid and single grid solvers are easily comparable with each other but this only gives a partial answer to the question of the performance of the solver. We will now show that `pamg-multiphase` is also more efficient than a traditional segregated multiphase solver. This is a much more difficult proposition. For the sake of simplicity, we will compare `pamg-multiphase` with CFX 4.1.

The first question to raise is which measure of accuracy should be adopted. In the case of coupled solvers like `pamg-multiphase`, it makes sense to use the L_2 norm of the residual to quantify the residuals since all equations are treated in the same way. By contrast, many segregated solvers emphasise the continuity equations so that good mass residuals are quickly obtained while momentum residuals are reduced much more slowly. The effects of this type of strategy are complex and the resultant error depends on the discrete operator \mathcal{L}^h .

The other issue is the measure of computational costs. The ideal measure is execution time to a certain level of accuracy. However, this ignores the effect of optimisations. `pamg-multiphase`, as mentioned in Section 4.4.3, is not optimised and it is inappropriate to compare it with a commercial code with many man-years of effort in optimisation. The number of iterations performed by a solver is a good indication of the potential of a procedure. This measure is chosen here, although it ignores the fact that the cost of one iteration could be very different between two algorithms. It should be noted, however, that the cost of one IPSA relaxation cycle

is comparable to that of a `pamg-multiphase` work unit.

We have attempted an approximate comparison of the efficiency of `pamg-multiphase` and `CFX 4.1` by:

- Considering the computational cost to be measured in work units i.e. the number of (equivalent) fine grid iterations;
- Assuming that the `pamg-multiphase` and `CFX 4.1` work units identical;
- Ignoring the cost of any preprocessing intended to improve the quality of the initial guess;
- Estimating the accuracy using all equation residuals (for the `CFX 4.1` solutions, we have formed a global 1-norm of the residual by adding the residuals of each equations);
- Normalising the residuals with respect to their initial values.

The results are summarised by Figures 5.121 and 5.122. They clearly demonstrate that `pamg-multiphase` out-performs a good segregated solver. This is the final justification for the approach chosen to the solution of the multi-fluid equations. The `CFX 4.1` computations were performed with the default under-relaxation factors ($\lambda_r = 0.65$ for all unknowns except the pressure for which $\lambda_r = 1.0$). For many problems, a greater amount of under-relaxation is necessary to obtain a converged solution. Figure 5.123 shows that the performance of the solver can be greatly affected by under-relaxation.

It may appear surprising at first glance that multigrid works better on a complex problem such as the T-junction than on the comparatively much simpler channel flow. This indicates that the dominant errors are different in each cases. For the T-junction, the flow patterns are strongly determined by the geometry. Multiphase effects are obviously present and important but they do not dominate the solution. By contrast, the geometry of the channel flow is very simple so that the evolution of the flow is entirely governed by multiphase effects, namely phase separation.

The remainder of this chapter will elaborate on this aspect: which multiphase effects dominate the convergence rates?

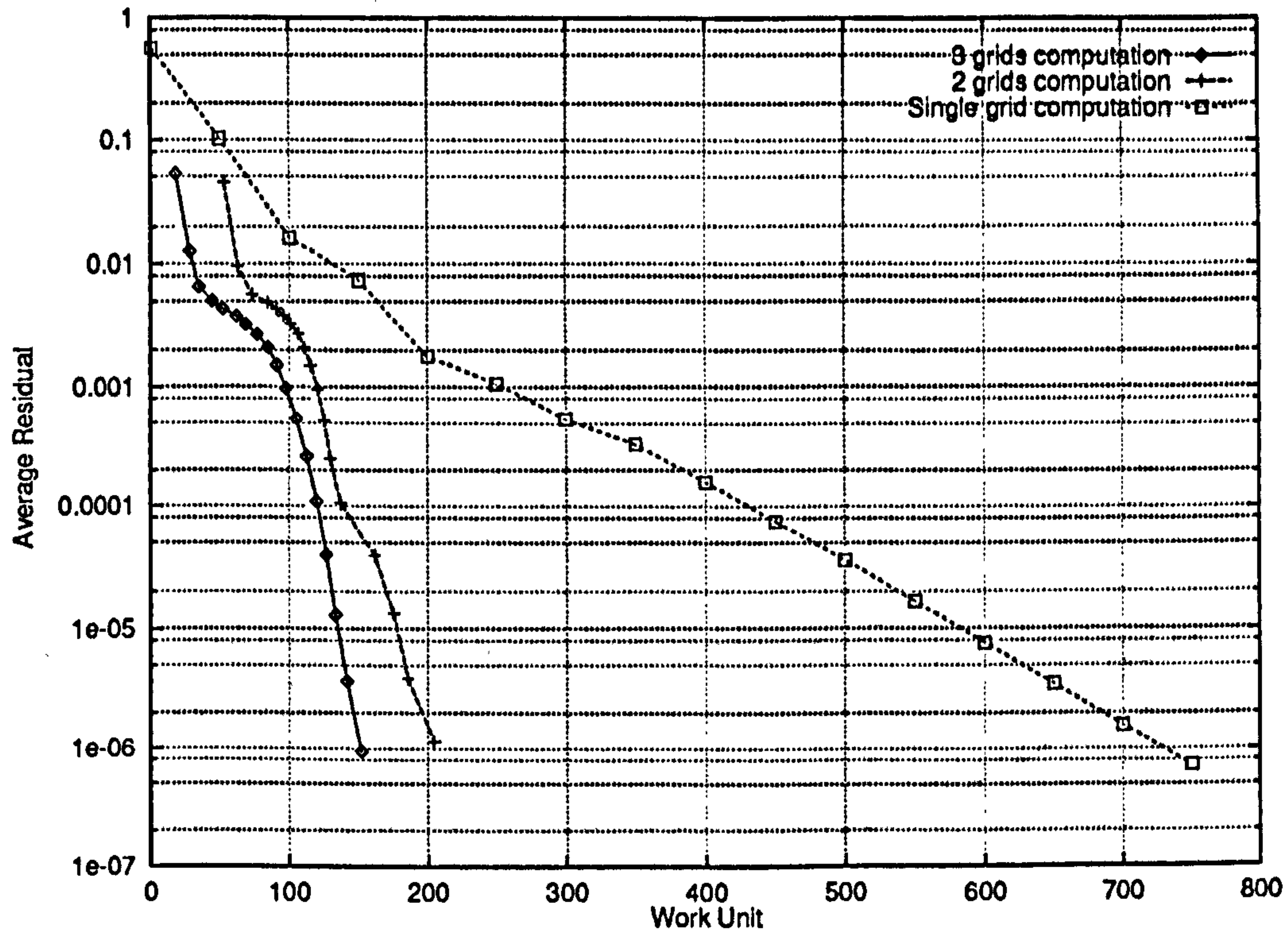


Figure 5.118: Two-phase channel flow – Convergence of solution for multi-grid computations and the equivalent single grid computation

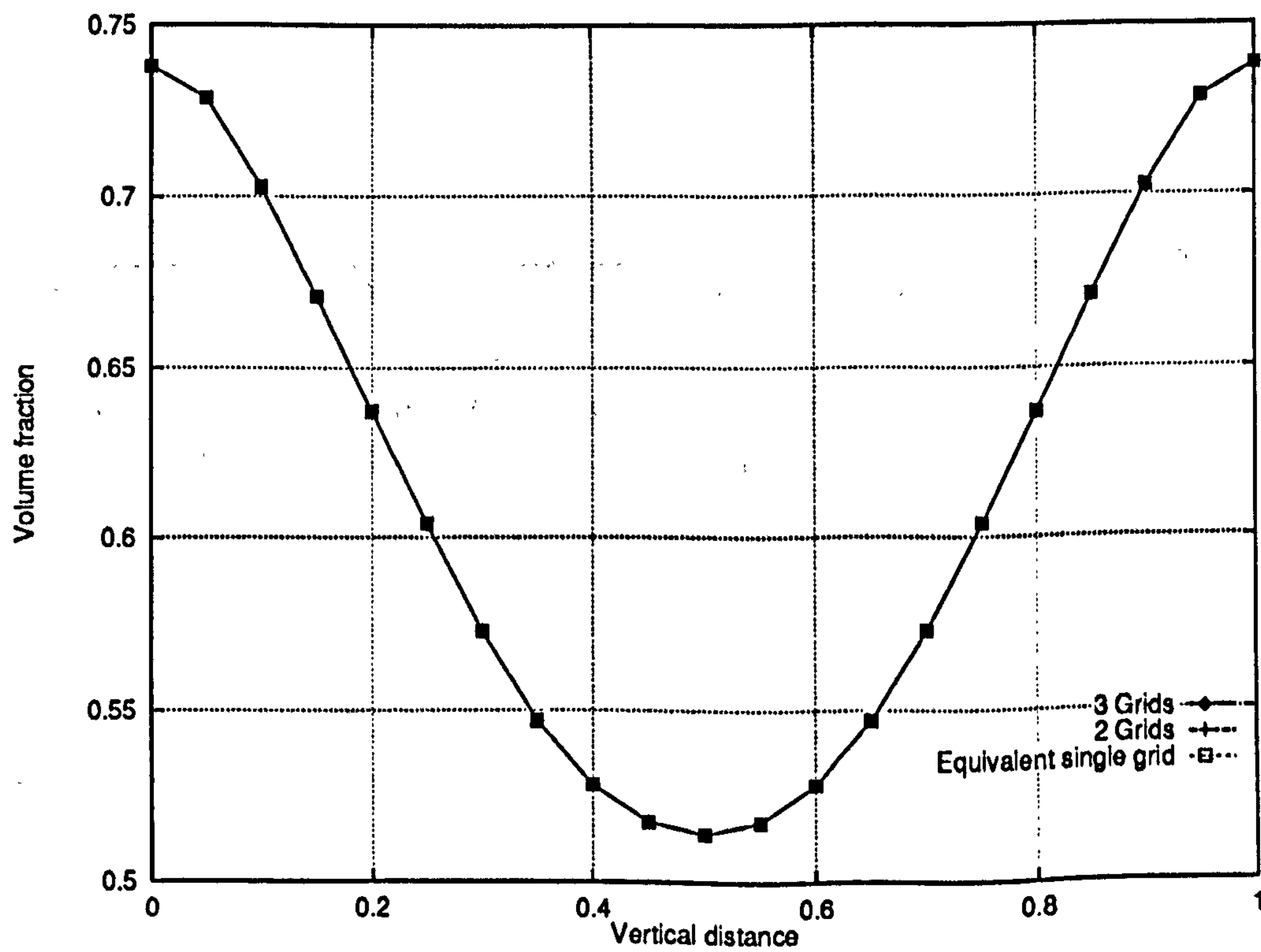


Figure 5.119: Two-phase channel flow – Volume fraction profiles along the line $x = 2.5$ – Comparison of multigrid and single grid solutions

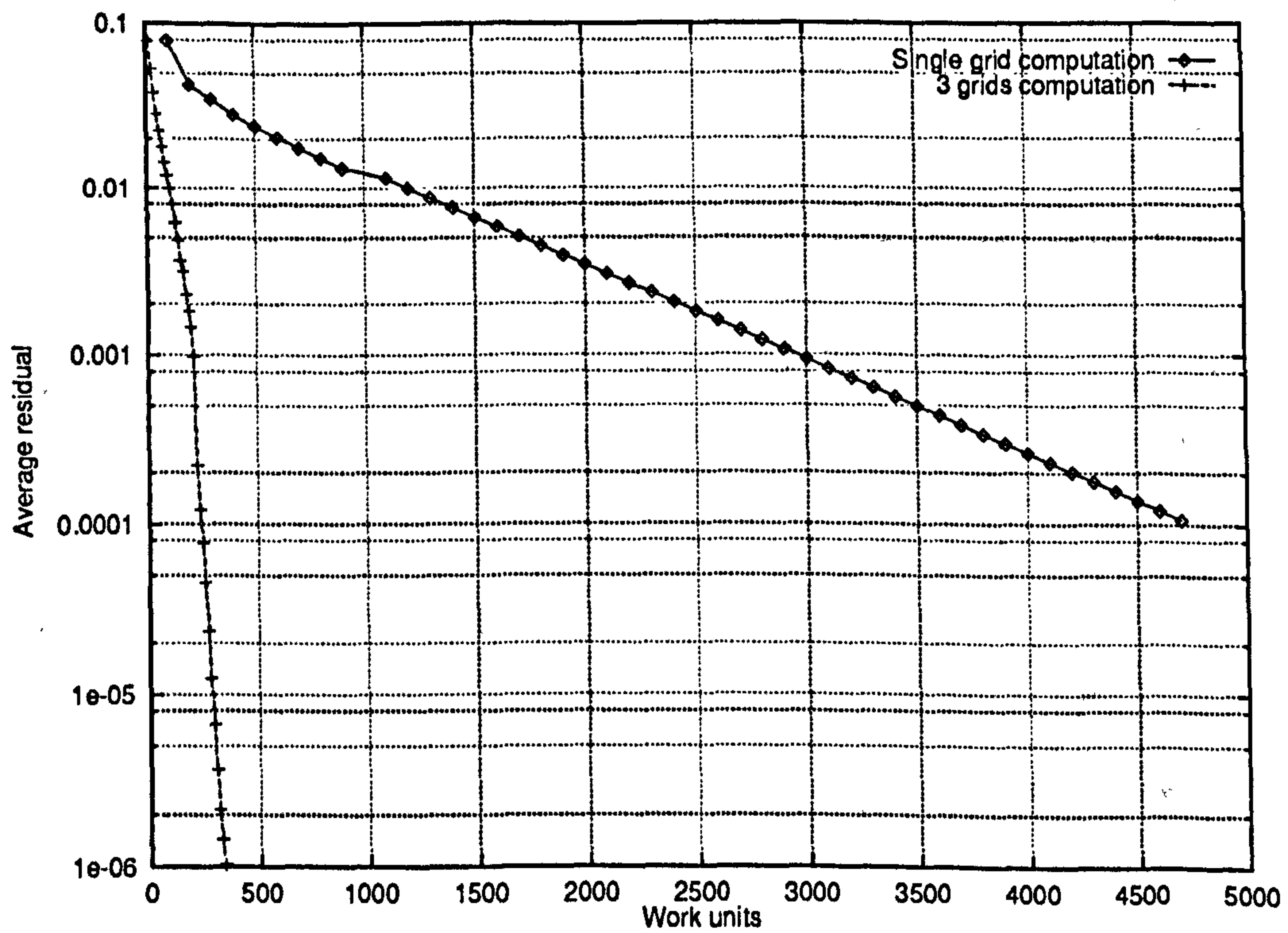


Figure 5.120: Multiphase T-junction problem – Comparison of the convergence histories for single and multigrid computations at level 3

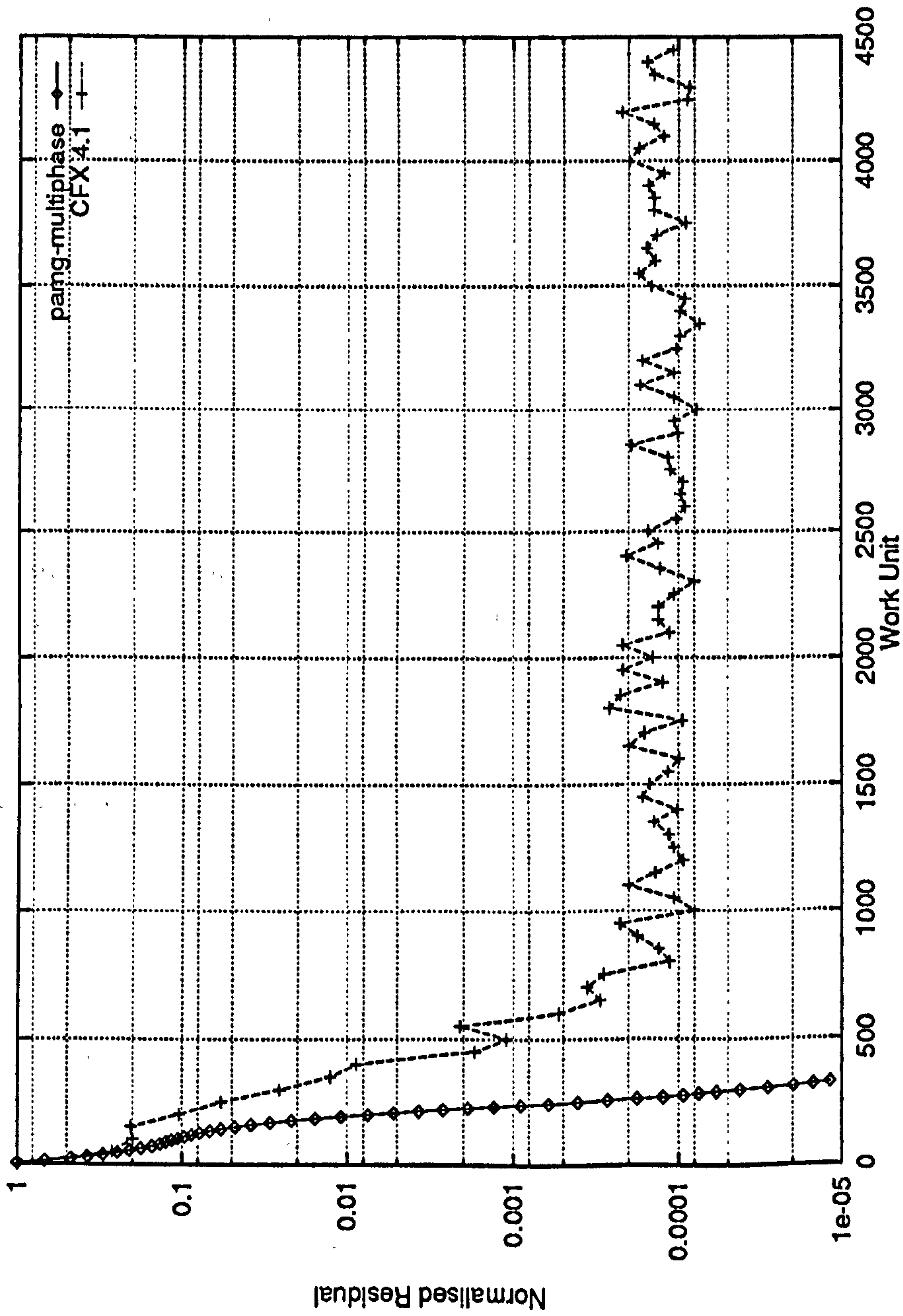


Figure 5.121: Multiphase T-junction problem – Comparison of the convergences histories for pang-multiphase and CFX 4.1 (with default under-relaxation)

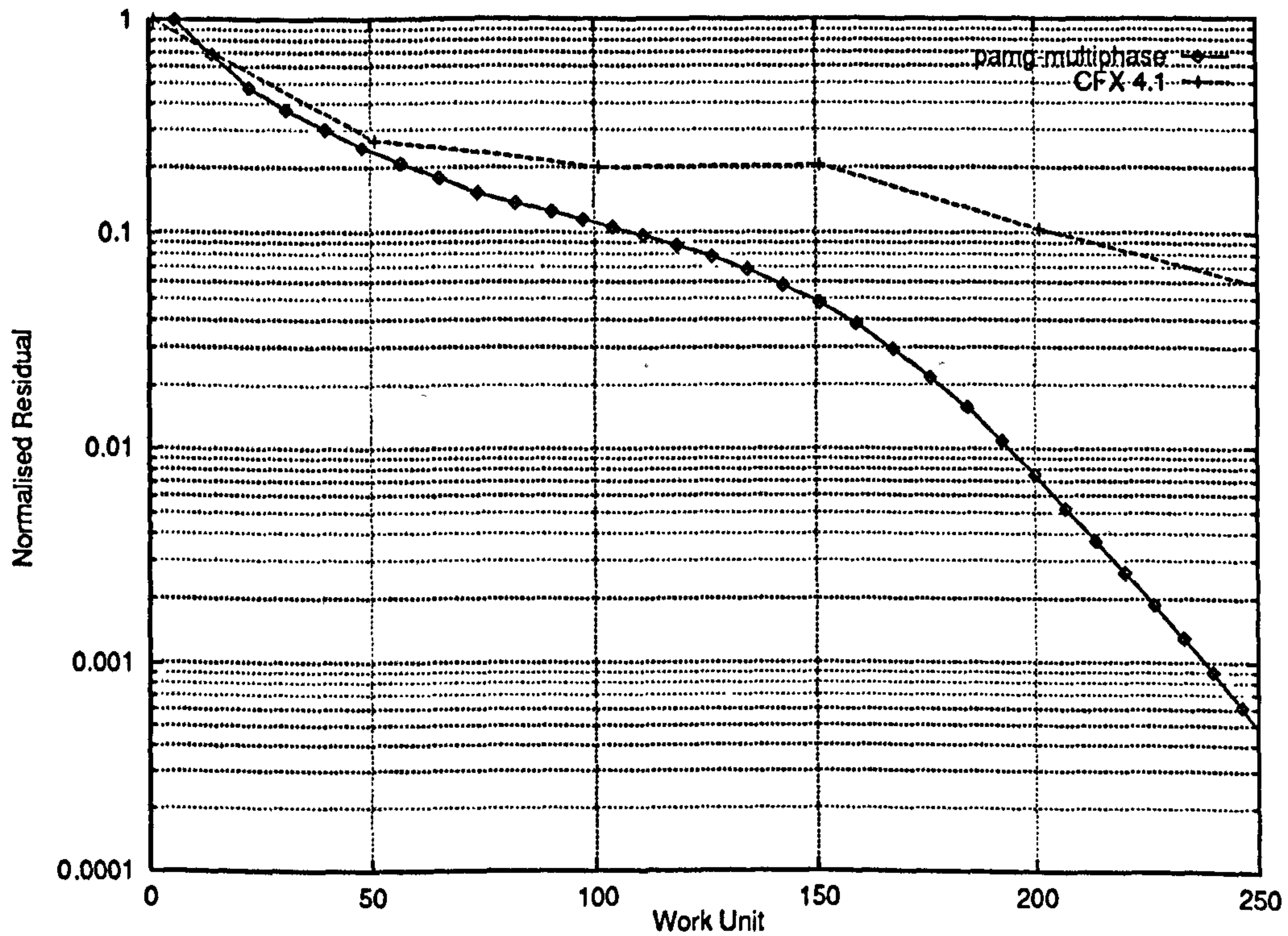


Figure 5.122: Multiphase T-junction problem – Comparison of the convergence histories for pamg-multiphase and CFX 4.1 for the initial iterations, showing that pamg-multiphase is always faster than CFX 4.1

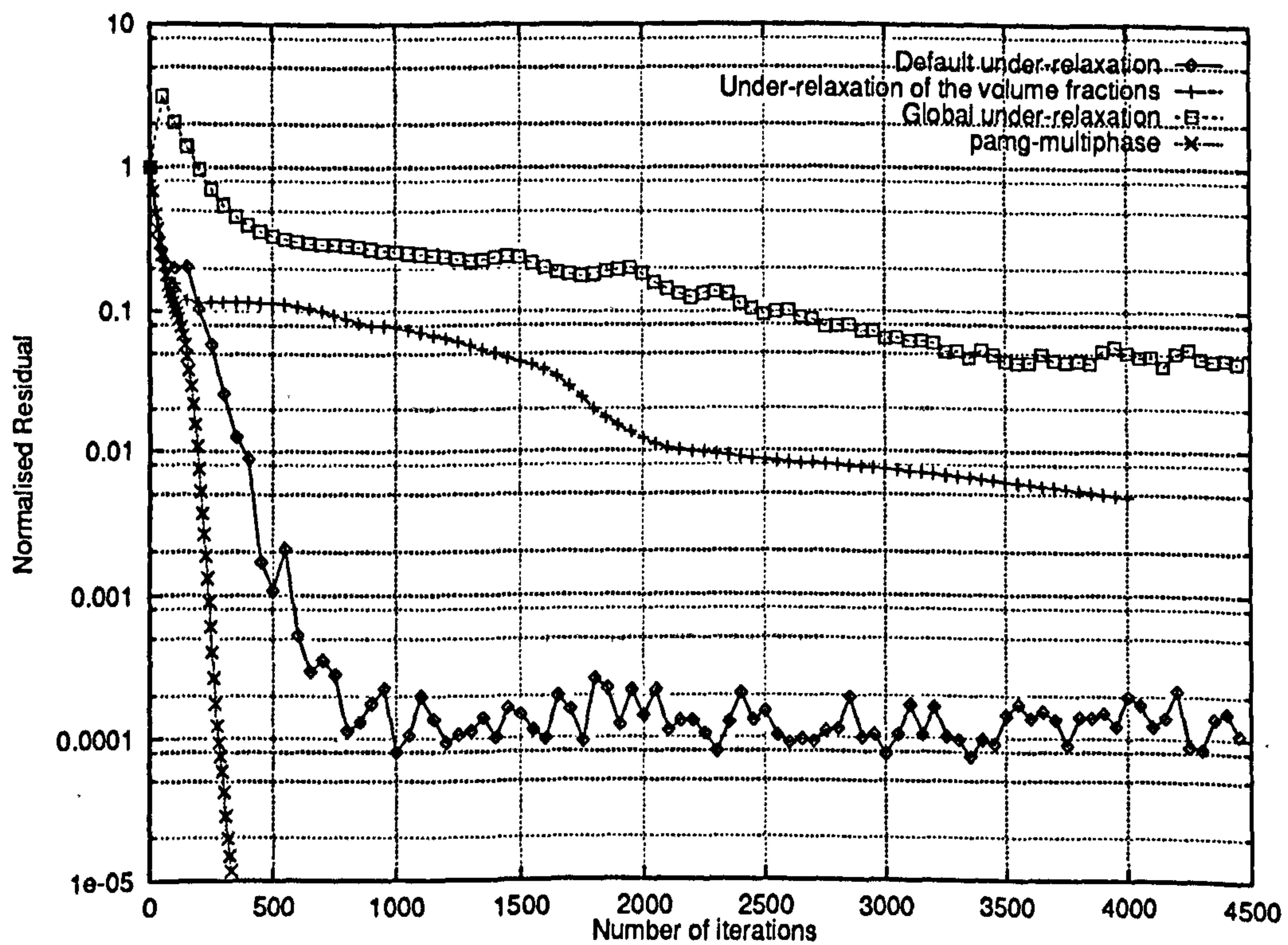


Figure 5.123: Multiphase T-junction problem – Comparison of the convergence histories for CFX 4.1 solutions and for different under-relaxation strategies. In each case, the under-relaxation factors are either the default values or set to $\lambda_r = 0.1$

5.4.3 Multigrid Convergence Factors and Grid Resolution

The multigrid algorithm of `pang-multiphase` can be considered as very successful on two accounts. It provides: (i) computational speed because by accelerating the convergence of the underlying single grid solver and (ii) robustness because it implicitly defines a continuation method.

If a multigrid method works well, one of its further benefits is that convergence rates are very largely grid independent. This is another way of saying that multigrid methods converge in $\mathcal{O}(N)$ operations where N is the number of unknowns. A good multigrid method is therefore an optimal solver (order-wise) for discrete equations. This property was *not* observed for our multiphase computations.

This is particularly striking for the two-phase channel flow problem. Convergence histories for computations at levels 3, 4 and 5, when the finest grid size is $\Delta x = \Delta y$ is 1/16, 1/32 and 1/64 respectively, are shown in Figure 5.124. It is obvious that the convergence factors do not scale well with the grid size (see Table 5.5).

Grid Level	Number of Cells	Average convergence factor (per relaxation work unit)
2	192	0.900
3	768	0.914
4	3072	0.942
5	12288	0.964

Table 5.5: Average convergence factors for the two phase channel flow problem

Multigrid solvers have a complexity which is $\mathcal{O}(N)$ for linear problems such as Laplace equation. Some Navier-Stokes solvers also achieve a similar level of performance despite their non-linearity. The single phase `pang` solver is a prime example (see Section 5.4.4). Although FAS is directly applicable to non-linear problems and actually limits the need for a linearisation, a non-linear problem is still harder to solve since the grid operators have to be approximated as well as the solution. For complex non-linear problems, the operators may be very different on different grids and their adjustment (via the defects) will require time.

In the author's opinion, the dependency of convergence factors on the grid size is primarily a feature of the multi-fluid equations. The situation is quite similar to the multigrid simulation of hyperbolic equations: multigrid is a very successful acceleration technique but it is not optimal order-wise [89, 14].

Certainly grid-independence studies have already indicated that multiphase solutions, particularly the volume fraction fields, can be sensitive to the grid size. Furthermore, careful examination of Figure 5.124 reveals that in some regions of the convergence history, the convergence factors are almost grid-independent (see Table 5.6). Globally, the history can be divided into three main regions, according to the

average residual:

- Up to 10^{-5} , the convergence factors are very grid dependent. It can be argued that during this phase, the non-linearities are being treated and the approximation of the operators are progressively more accurate. We refer to this as the “plateau” phase because convergence factors are strongly degraded.
- Between 10^{-5} and 10^{-8} , convergence factors are grid independent to a large extent.
- Below 10^{-8} , a region which is not usually investigated, convergence factors are again grid dependent. Some evidence (single precision computations) suggests that this is not a consequence of round-off errors. Rather, it is likely that in this region, another part of the operator dominates.

Grid level	Average convergence factor			
	to 10^{-4}	between 10^{-4} and 10^{-6}	between 10^{-6} and 10^{-8}	below 10^{-8}
2	0.902	0.902	0.906	0.890
3	0.941	0.864	0.876	0.918
4	0.965	0.901	0.886	0.945
5	0.978	0.936	0.896	0.968

Table 5.6: Average convergence factors for the two phase channel flow, depending on the value of the average residual

By contrast, for the two-phase flow through a T-junction, a much better measure of grid independence exists, as Figure 5.125 and Table 5.7 show.

Grid Level	Number of Cells	Average Convergence factor (per relaxation unit work)
2	576	0.9879
3	2304	0.9666
4	9216	0.9656

Table 5.7: Average convergence factor for the two phase T-junction problem

In terms of grid independent convergence factors as well as in terms of multigrid acceleration, the conclusion seems to be that the more complicated the flow, the better the performance of the multigrid solution algorithm. However, it should be noted that a comparison of Table 5.7 with Table 5.5 shows that particularly for coarser grids, the convergence factors for the channel flow are still faster than for the T-junction.

An explanation for this phenomena could be as follows: in Sections 5.4.13 and 5.4.14 we present evidence to support the view that the channel flow convergence factors

are dominated by the effect of cross derivative terms which appear in the diffusive flux $\nabla \cdot (r\mathbf{T})$ in the momentum equation. This only happens because many terms are identically zero as a result of the flow being one-dimensional.

This term is proportional to the spatial variation of the volume fractions (see Section 3.3.2) and, in the case of the channel flow, errors are greatest near boundaries where volume fractions vary considerably. As a result, the cross derivative terms will take large values and could dominate the other terms. By contrast, in the T-junction problem, errors are mostly associated with a small region in which the flow changes direction i.e. most errors are concentrated near the T-junction itself. In this region, all terms are significant and the cross-derivative terms are not allowed to dominate the error. It should also be noted that the volume fraction fields are largely grid dependent for the channel flow problem so that the cross derivative terms can be expected to have different effects on different grids. This is not so true for the multiphase T-junction since volume fractions are more grid independent (compare Figures 5.46 and 5.81). The refinement patterns observed for each problem when adaptive gridding is allowed (see Sections 6.2.4 and 6.3 together with Figures 6.54 and 6.57) confirm the location of the dominating errors.

It may be concluded that because the T-junction flow is more complicated, errors specifically associated with the nature of the multi-fluid equations are not dominant and `pang-multiphase` offers multigrid acceleration and convergence factors which are comparable with the single phase solvers, even though the flow is much more complex than most single phase flows.

In this section we have discussed the fact that the multigrid convergence factors observed with `pang-multiphase` are not always grid independent. The importance of this needs to be put in perspective since some grid independent convergence factors are observed for some complex problems and/or for some part of the convergence history. Furthermore, some grid-dependent convergence regimes occur for very low values of the residuals compared with the truncation errors (see Section 5.3.4). In this context it is interesting to note that qualitatively very similar convergence histories have been recorded for the Euler equations using conceptually-related algorithms, namely a Newton-Krylov-Schwarz method [112].

A hypothesis that the grid-dependence of the multigrid convergence factors is a feature of the equations and is connected with the diffusive terms (and hence is a cell Reynolds number effect) has been formulated, and will be discussed in Sections 5.4.11 to 5.4.14. First, we compare the convergence factors for `pang` and `pang-multiphase`.

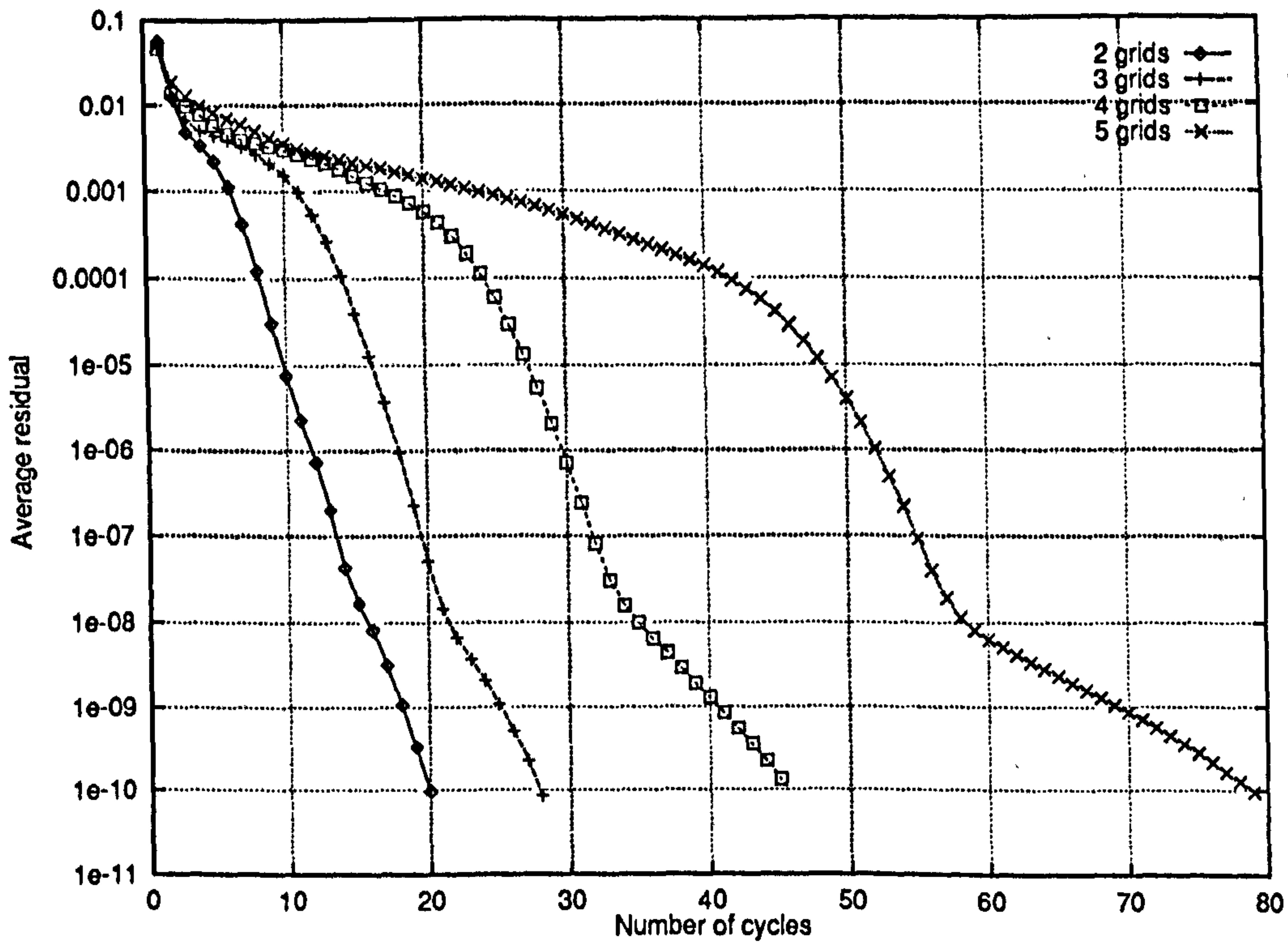


Figure 5.124: Two-phase channel flow – Convergence of solution for different numbers of grids

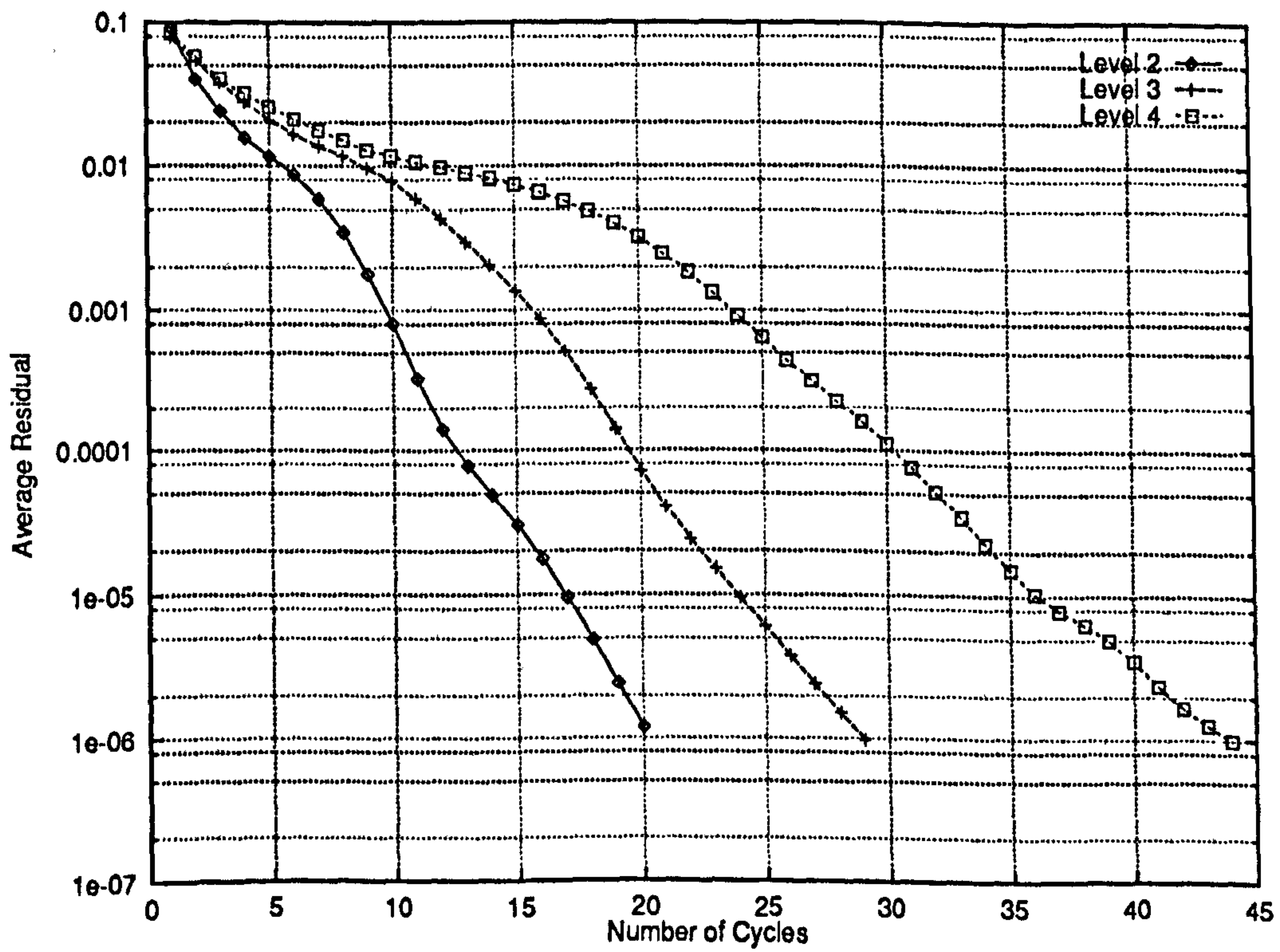


Figure 5.125: Multiphase T-junction problem – Convergence factors for multi-grid computations at different levels

5.4.4 Comparison with pamg Convergence Factors

In the case of the pseudo-two-phase channel flow described in Section 5.2.1, the volume fractions very quickly reach their final values. From that moment, the computation does not differ significantly from a single phase flow computation. It is therefore expected that the multigrid method will perform much better. This has been partly verified but significant differences between pamg and pamg-multiphase have been observed. These differences lead to the conclusion that details of the implementation of the quasi-Newton coupled solver are important in determining the multigrid performance. In other words, the extent to which the LQN solver is a good multigrid smoother is quite sensitive to the formulation of the local Jacobians.

The convergence history for the two phase pseudo-multiphase channel flow (Figure 5.126) reveals that:

- There is no “plateau” phase in the convergence history. The plateau phase is therefore a characteristic which is specific to multiphase flows where there are spatial variations of the volume fractions;
- The convergence factors are not quite grid independent: again fine grid computations are less efficient than those done on coarser grids. However, the difference is not as marked as with truly multiphase flows.

This contrasts with results obtained with the pamg single phase code which shows (Figure 5.127) a very good measure of grid independence. Indeed, as the grid gets finer, the convergence factors recorded with pamg improve slightly due to the increased discrete ellipticity [10, 90]

The main difference between the two codes lies in the Jacobians used for the Newton corrections. In pamg, an approximate Jacobian is obtained by neglecting the non-diagonal velocity components (see Section 4.2.1). This choice has been proven to work very well and forms the core of a highly efficient multigrid method [1]. A useful property of the pamg quasi-Newton solver is that it does not require any measure of under-relaxation to converge.

It became apparent early during the development of pamg-multiphase that a quasi-Newton step based on a similar approximation for the Jacobian did not work for multiphase flows, due to the greater degree of non-linearity. This suggests that the coupling between local variables needs to be treated very accurately. This conclusion motivated the use of automatic differentiation to obtain an *exact*, local Jacobian for the discrete equations.

It appears, however, that the deterioration of convergence factors for finer grids is linked to more accurate expressions for the Jacobian. This hypothesis is corroborated by two observations:

1. The phenomenon is not specific to the two phase computations. Indeed, it can be observed to a lesser degree for a single phase pipe flow if the approximate Jacobian of [13] is replaced by an exact Jacobian obtained by automatic differentiation. (Figures 5.127 and 5.128).

This conclusion follows from the inverse experiment: in `pamg-multiphase`, the automatic differentiation Jacobians are used for single phase computations so deteriorating convergence factors are observed. However, if they are replaced by the approximate Jacobians, we obtain good grid independent convergence rates, as shown in Figure 5.129.

2. Under-relaxation during the smoothing phase of the multigrid cycle (see Sections 4.3.1 and 5.4.4) renders the convergence factors much more grid independent in the single phase case (Figure 5.130) as in the two phase case (Figure 5.131). Hence, the deterioration in performance is due to the fact that the basic quasi-Newton solver is not a good enough smoother for the multigrid method to work optimally. By adding under-relaxation, the smoothing properties of the solver appear to be greatly improved. There is a parallel situation with the Jacobi iterative method: under-relaxation is necessary if the high frequency components of the error are to be eliminated quickly [79].

Figure 5.131 includes results obtained for two values of the under-relaxation parameter λ_r (defined by equation 4.38). If too much under-relaxation is applied, convergence factors are strongly deteriorated because once again the performance of the smoother is reduced due to the corrections not being as large as they need to be.

A further confirmation comes from comparing the actual entries of the approximate and automatic differentiation Jacobians. Tables 5.8 and 5.9 show the Jacobian matrices for a typical interior cell, computed from the converged solution. Some non-diagonal entries of the local Jacobians, which are neglected in the simplified Jacobian, actually take large values ^{6 7}.

$$\begin{pmatrix} 27.2849 & 0.0 & 2.56 & -2.63306 & 16.0 \\ -22.1245 & 27.3620 & -2.56 & 2.48228 & -16.0 \\ 2.56 & -2.56 & 32.3645 & -5.15948 & 16.0 \\ -2.55806 & 2.56 & -5.12 & 32.2184 & -16.0 \\ -16.0 & 16.0 & -16.0 & 16.0 & 0.0 \end{pmatrix}$$

Table 5.8: Single phase pipe flow – Exact (automatic differentiation) Jacobian for a level 3 cell ($x = 2.036125 \pm 0.03125$, $y = 0.5036125 \pm 0.03125$) – See Section 4.2.1 for the arrangement of equations and unknowns in the Jacobian

⁶Entry (1,2) of the automatic differentiation Jacobian is zero due to the upwinding.

⁷Entries (3,4) and (4,3) of the automatic differentiation Jacobian could take large values in other parts of the flow.

$$\begin{pmatrix} 27.2482 & 0.0 & 0.0 & 0.0 & 16.0 \\ 0.0 & 27.3270 & 0.0 & 0.0 & -16.0 \\ 0.0 & 0.0 & 32.4085 & 0.0 & 16.0 \\ 0.0 & 0.0 & 0.0 & 32.2571 & -16.0 \\ -16.0 & 16.0 & -16.0 & 16.0 & 0.0 \end{pmatrix}$$

Table 5.9: Single phase pipe flow – Approximate Jacobian for a level 3 cell ($x = 2.036125 \pm 0.03125$, $y = 0.5036125 \pm 0.03125$) – See Section 4.2.1 for the arrangement of equations and unknowns in the Jacobian

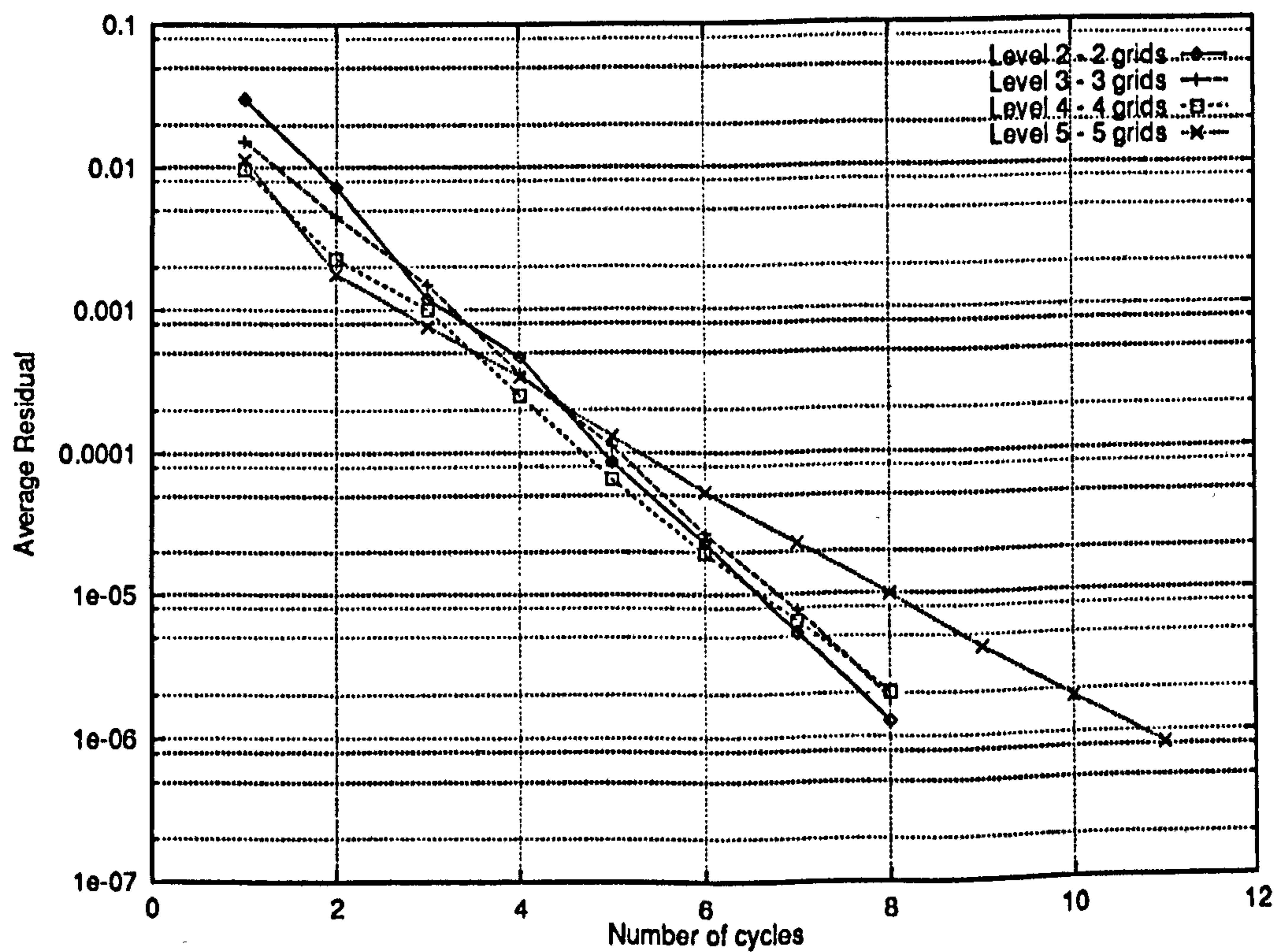


Figure 5.126: Convergence histories of the pang-multiphase solver for the pseudo-multiphase channel flow without relaxation

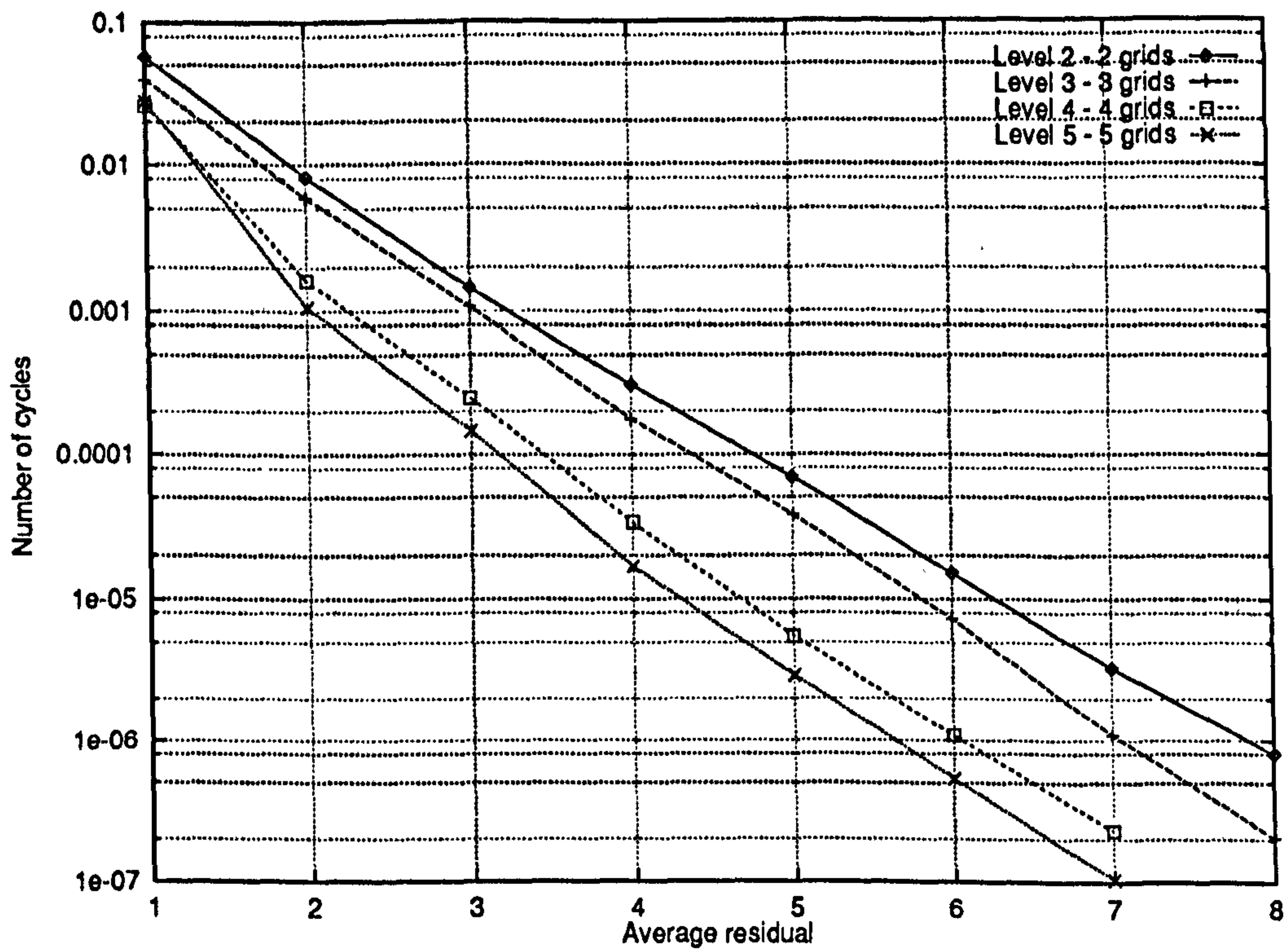


Figure 5.127: Convergence histories for a *single phase* channel problem – single phase pang solver

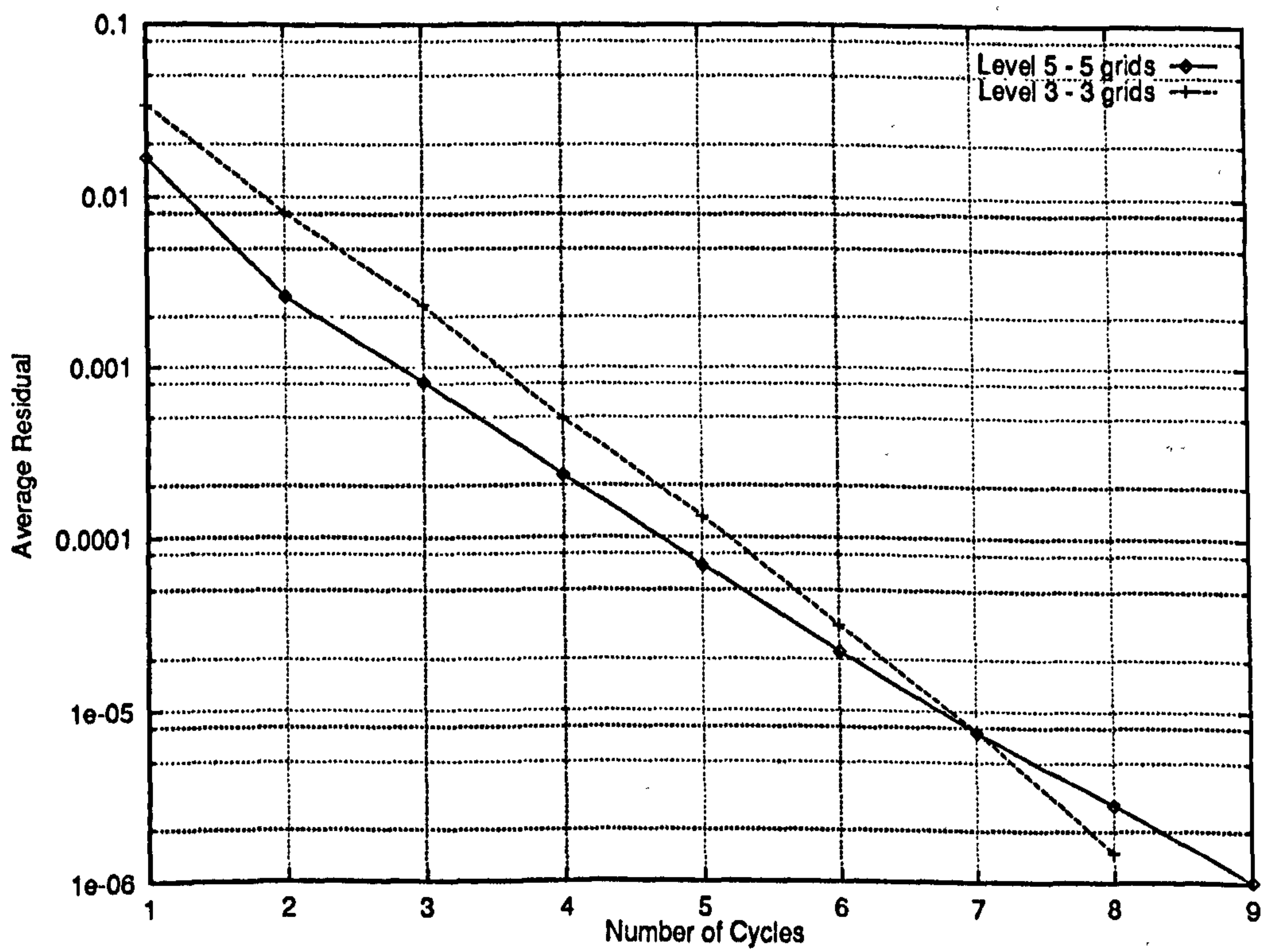


Figure 5.128: Convergence histories for a *single phase* channel problem – pang multiphase with automatic differentiation Jacobians

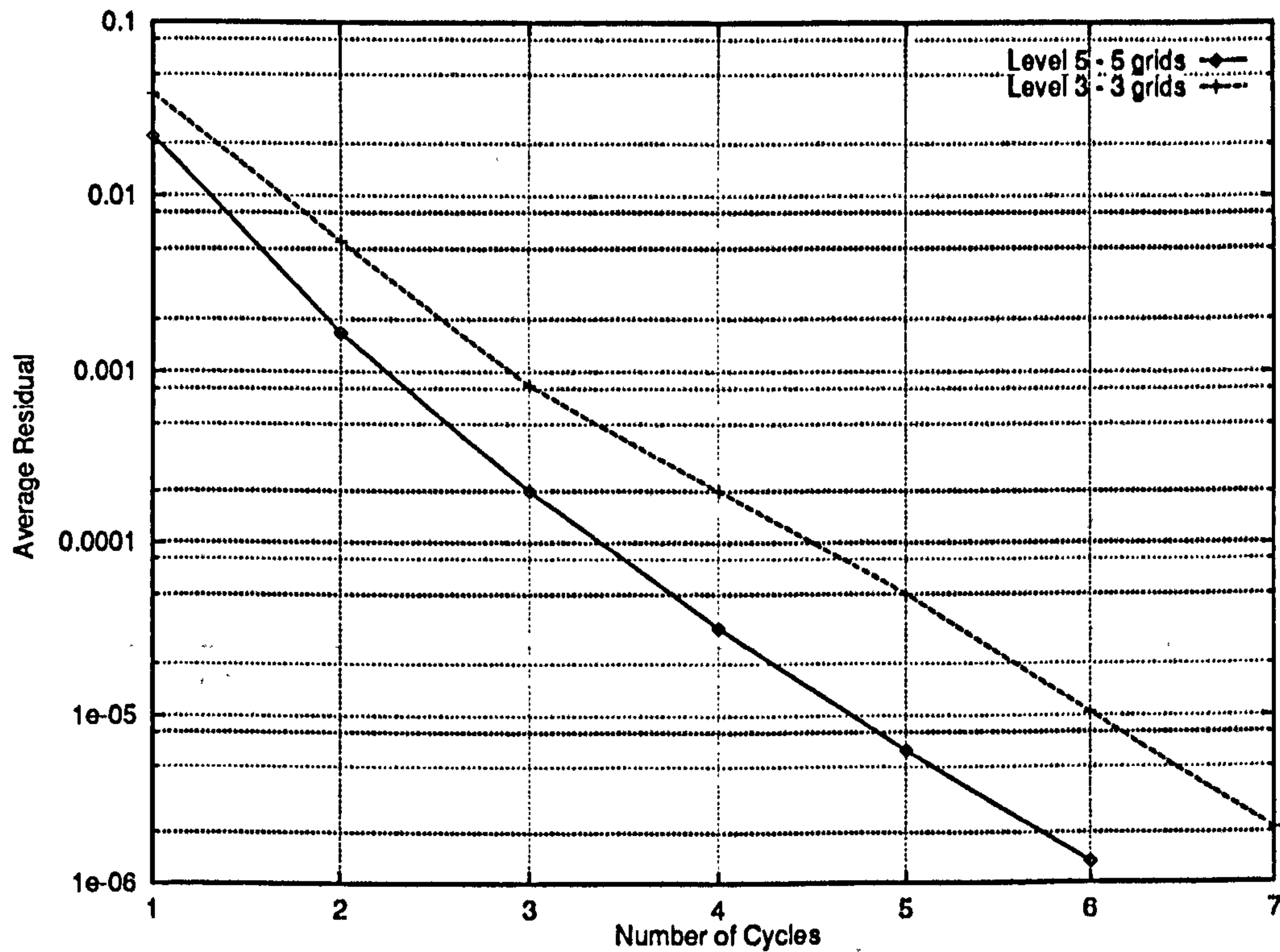


Figure 5.129: Convergence histories for a single phase channel problem – pang multiphase with approximate Jacobians and without under-relaxation

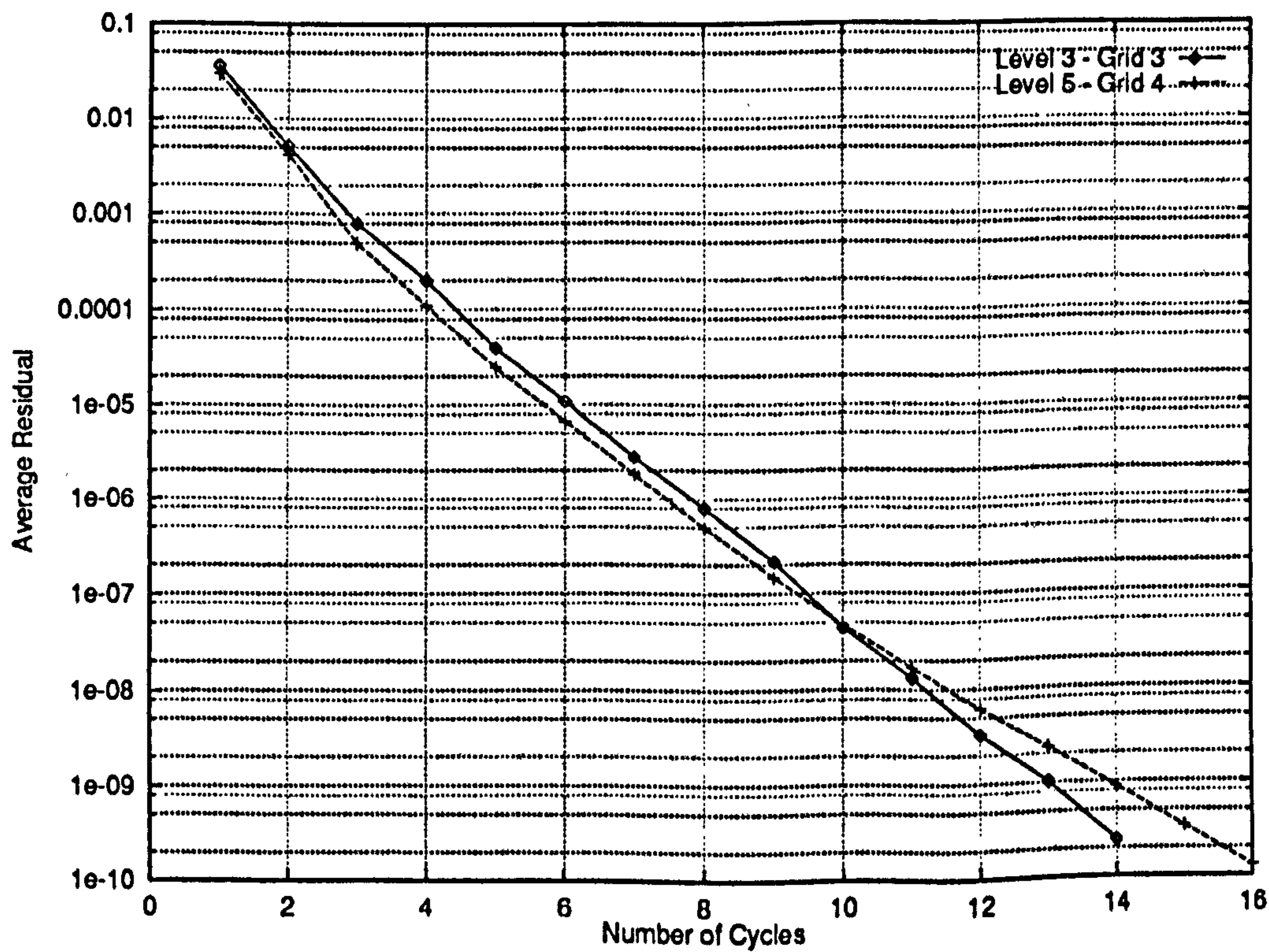


Figure 5.130: Convergence histories for a single phase channel problem – pang multiphase with automatic differentiation Jacobian and under-relaxation factor ($\lambda_r = 0.6$)

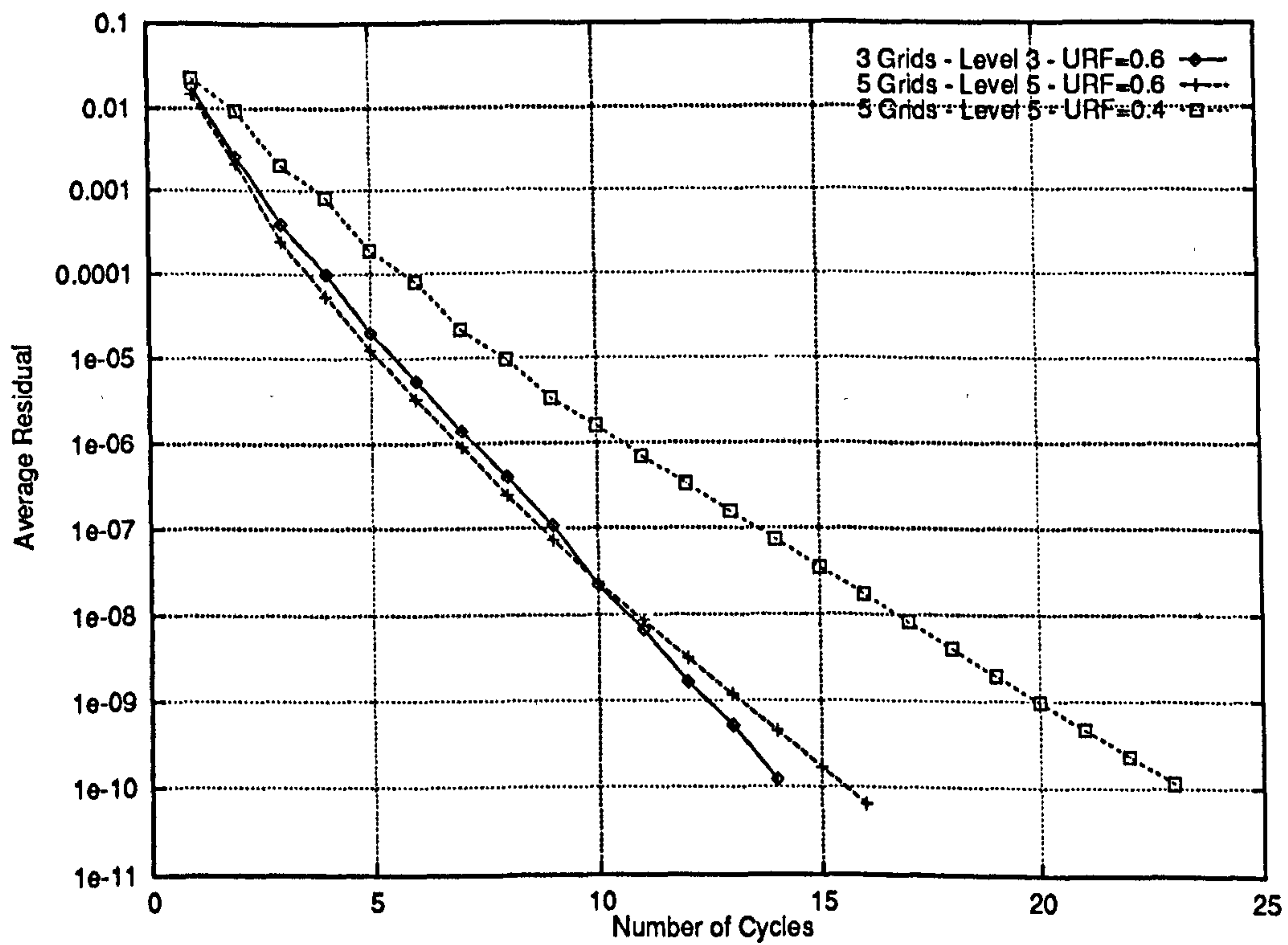


Figure 5.131: Convergence histories for a pseudo two-phase pipe problem – pang multiphase with automatic differentiation Jacobian and under-relaxation showing mesh independent convergence with under-relaxation

In *pamg-multiphase*, the Newton corrections are not only under-relaxed but a line-searching procedure is added (see Section 4.3.1). This has proved necessary whenever multiphase computations are attempted. Otherwise, the algorithm diverges after a few iterations. Line-searching offers two benefits. Firstly, it renders Newton's method globally convergent and removes the need for the initial guess to be sufficiently close from the solution. Secondly, it allows the Newton corrections to be monitored so that the volume fractions are always bounded between 0 and 1. Another advantageous feature of *pamg* is that no line-searching is needed to obtain convergence. This appears to be a characteristic of the particular approximation chosen for the Jacobians since if they are replaced by exact Jacobians, then the single phase computations diverge in a few iterations unless line-searching is added.

At first glance, it may appear that comparing the *pamg* and *pamg-multiphase* solvers is of very little interest for understanding the behaviour of the latter for multiphase flows. It is, however, very instructive. The role that the quasi-Newton solver plays in determining the convergence factors is clarified. On the one hand, the absence of a plateau phase in the pseudo two-phase case indicates that the plateau is connected with spatial variations in the volume fractions — the governing equations therefore play an important role. On the other hand, an investigation of the properties of the quasi-Newton coupled solver based on exact Jacobians has revealed that on fine grids, under-relaxation is necessary to obtain approximately grid-independent convergence factors.

This suggests that the solution algorithm may have some shortcomings which would explain the fact that convergence factors are dependent on the grid size. Three hypotheses along this line may thus be formulated:

- The LQN solver does not perform as well for low Reynolds numbers as it does for higher Reynolds numbers. In Section 5.4.11 we investigate the influence of the Reynolds number on efficiency.
- The multiphase flow solutions are highly non-linear due to variations in the volume fraction while the interpolation operators are linear. The cure may be to use operator-based interpolation. This could be a long term objective. For now, the effect of order of accuracy of the grid transfers on convergence factors is examined in Section 5.4.8.
- Although FAS is directly applicable to non-linear problems, a crucial assumption is that the coarse grid correction is also a good fine grid correction, at least as far as the low frequencies are concerned. The fact that the multigrid method is convergent and fast even for complex multiphase flows indicates that this assumption is largely valid. However, it may well be that due to the relative dependence of the volume fraction on the grid size, the coarse grid and fine grid operators are quite different. The multigrid procedure cannot be optimal until the discrepancy is accurately corrected by the defect. In this case, the plateau phase is nothing more than a symptom reflecting the extent to which the problem is non-linear. Varying the parameters which define

Grid level	Convergence factors
2	0.8760
3	0.8161
4	0.7862
5	0.7477

Table 5.10: Single phase entry flow – Average convergence factor for pang computations on different grid levels

the multigrid procedure should thus have a large impact on the convergence factors. This is the subject matter of the following sections.

Finally, and for completeness, the performance of pang for single phase Navier-Stokes flows is illustrated using two problems: the single phase entry flow of Section 5.4.14 (see also Appendix B and the single-phase T-junction problem of Section 5.2.2). These problems are solved on uniform and adaptive grids respectively. Figures 5.132 and 5.133 give the convergence histories: very few iterations are necessary to bring the residuals down to very low values and, by contrast with the multiphase flow computations described in this chapter, the convergence factors are not significantly degraded on fine grids. For the entry flow, the average convergence factors are shown in Table 5.10 and are significantly better than for the two phase computations. Indeed, as the grids become finer, the convergence factor are significantly lower, due to the effect of h-ellipticity.

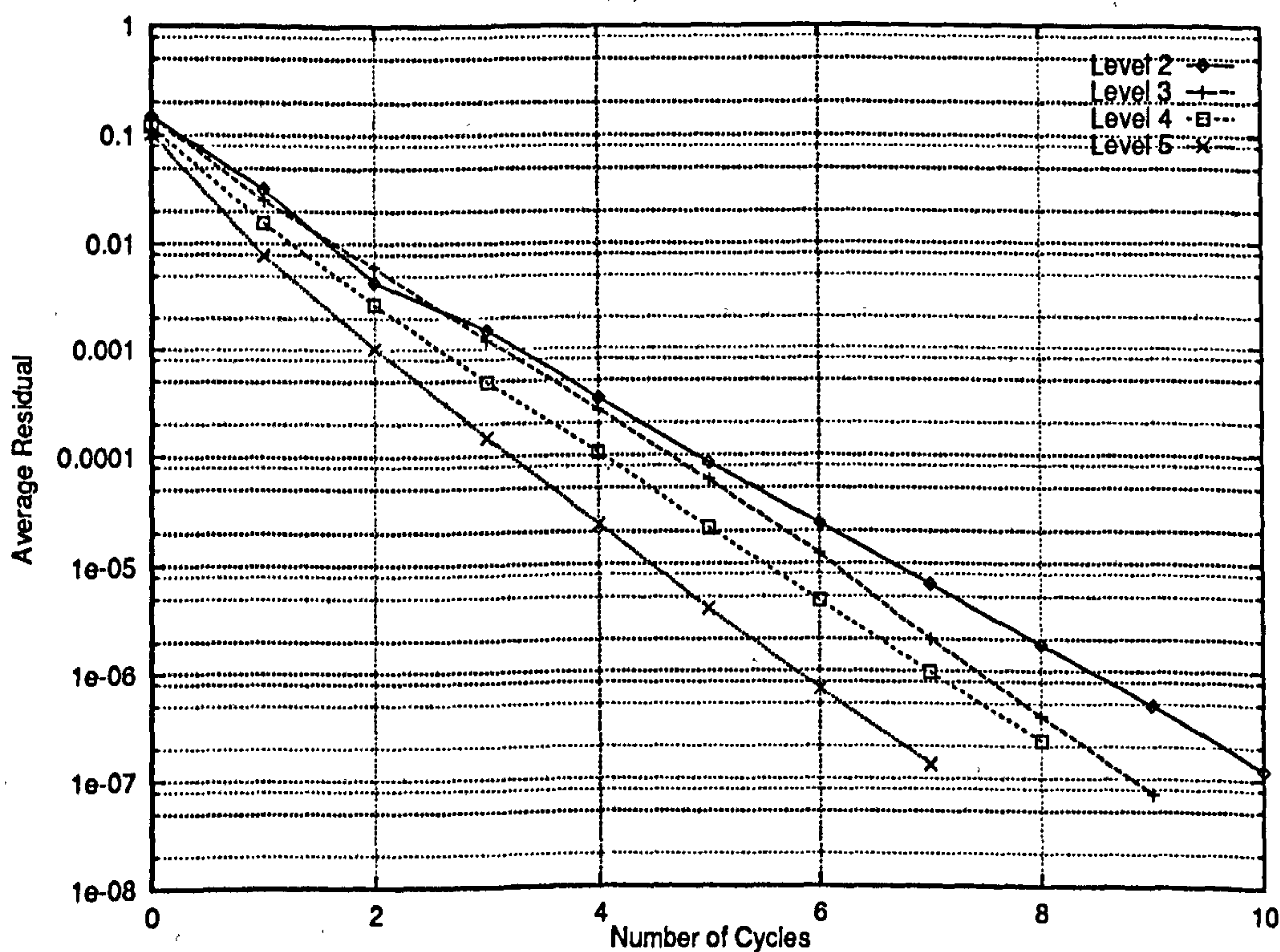


Figure 5.132: Single phase entry flow – Convergence histories with the pang code for different (uniform) grid levels

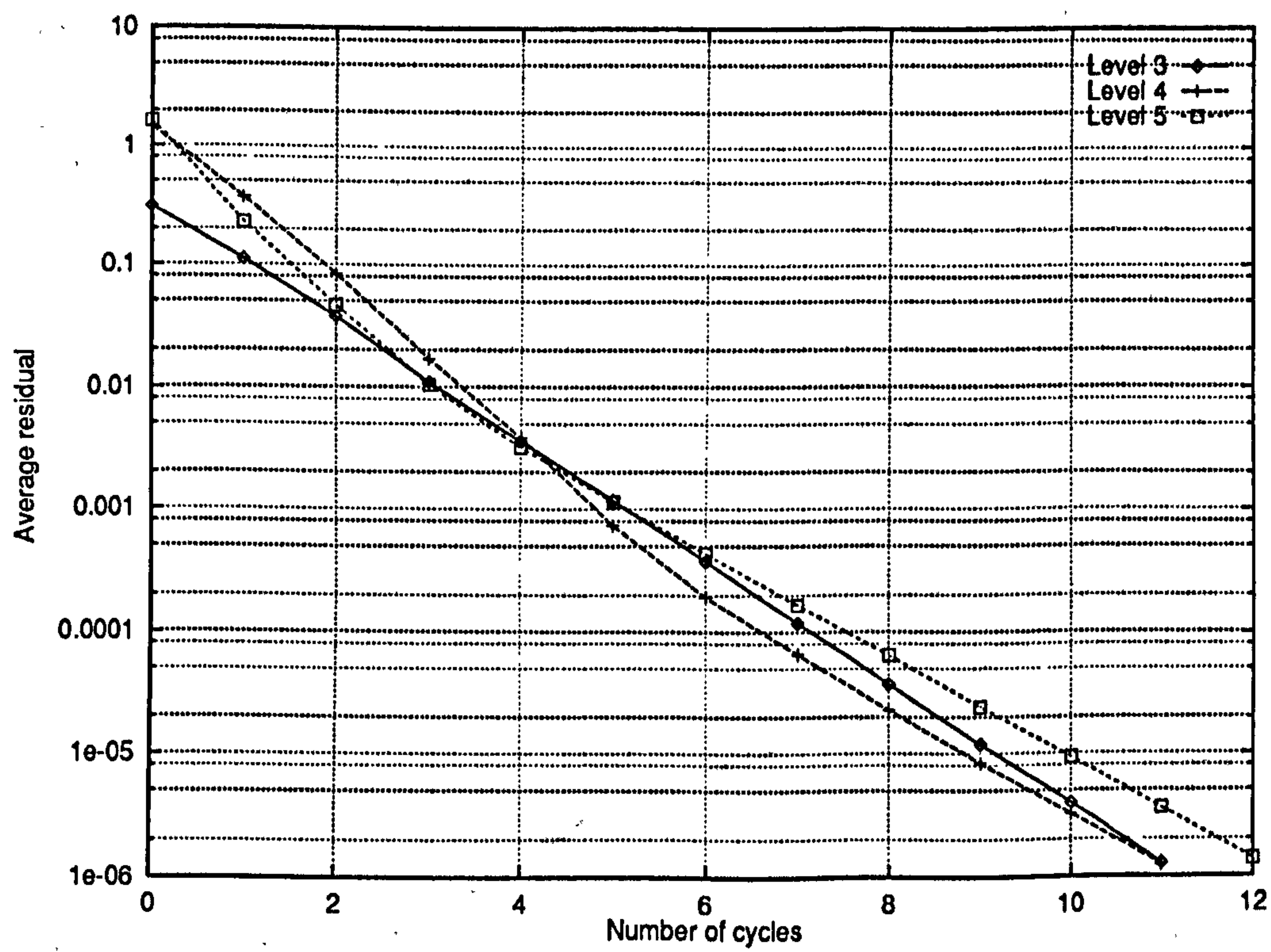


Figure 5.133: Single phase T-junction – Convergence histories with the pang code for different (adaptive) grid levels

5.4.5 Brief Investigation of Multigrid Cycling Strategies

The cycling strategy used in all the previous examples was the “F” cycle, which stands for *full* multigrid. Other cycling strategies have been briefly tested. *On this test case*, the “V” cycle was unsuccessful and the multigrid computations diverged. By contrast, computations using the “W” cycle converged and it was relatively surprising to note that the convergence history is almost the same as for the “F” cycle (Figure 5.134)⁸. Since the “W” cycle is the more costly of the two, the “F” cycle was systematically used.

It is well known that the “V” cycle, which is the least costly of the three cycling strategies, is also the least stable choice for non-linear problems. Its failure indicates that the problem whose solution is attempted is sufficiently non-linear to render the correction scheme inadequate. In short, more computational work needs to be done in order to obtain a good coarse grid correction. The fact that the “W” and “F” cycles lead to similar convergence factors on the fine grid suggests that a large part of the error on the fine grids is related to the transfer of the correction to the fine grid. As explained above, errors may arise from two sources: (i) the linear transfer operators and (ii) discrepancies between the fine and coarse grid operators.

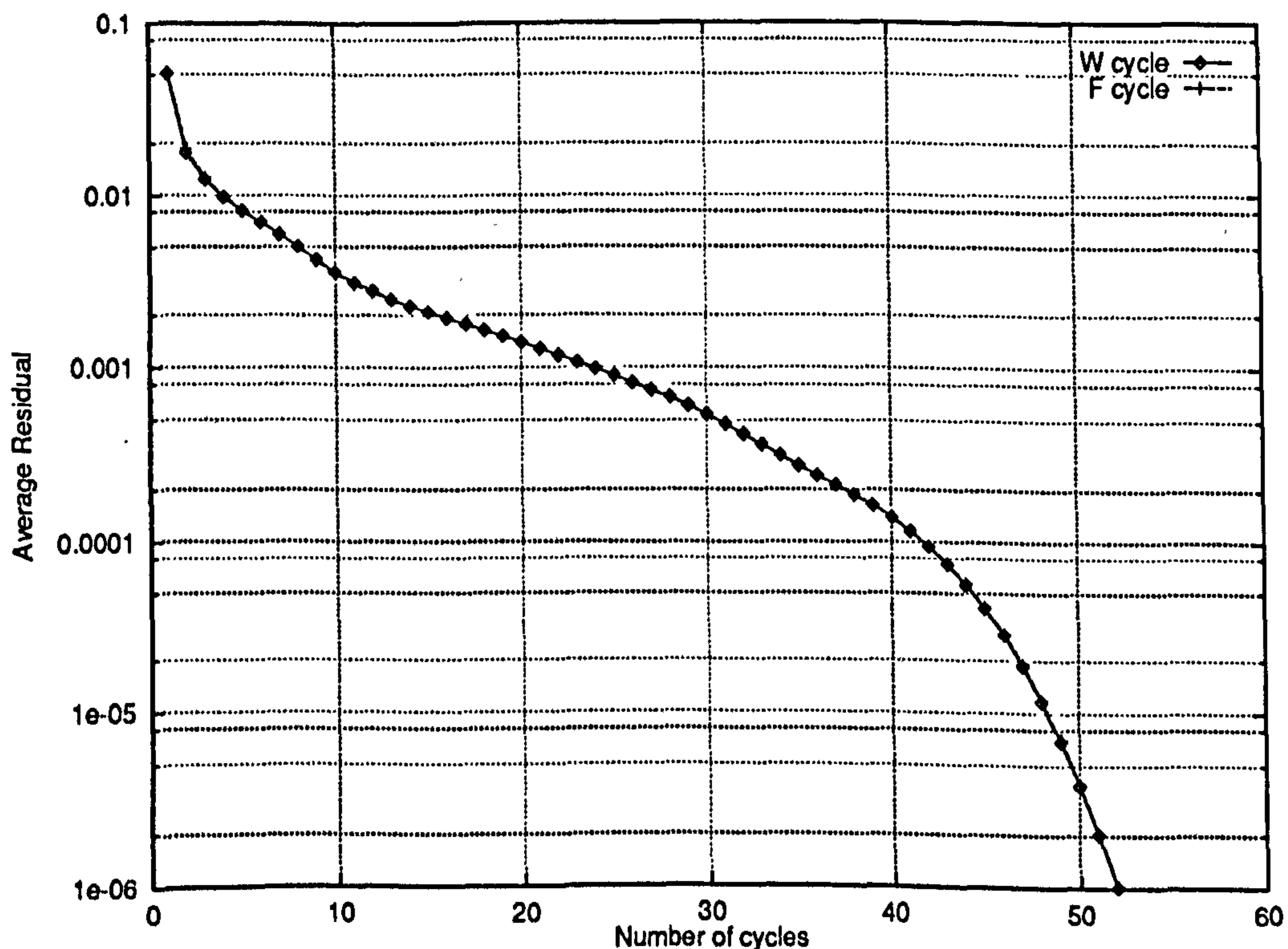


Figure 5.134: Two-phase channel flow – Convergence of solution on a level 5 grid for two cycling strategies: the “W” and “F” cycles

⁸Although a similar behaviour was observed with the single phase pang solver.

5.4.6 Effect of Under Relaxation

Under-relaxation is widely used as a means of dealing with non-linear systems. The aim of this section is to quantify its effect on the convergence factors of multiphase computations by building on the results of Section 5.4.4, which is mainly devoted to the single phase case.

For this purpose, the two-phase channel flow problem is solved for varying values for the under-relaxation parameter λ_r (defined by equation 4.38, see Section 4.3.1). The resulting convergence histories are shown in Figures 5.135 to 5.137.

On coarse grids ($k \leq 2$), Figure 5.135, it would appear that under-relaxation is not necessary to achieve convergence. Indeed, as the amount by which the computations are under-relaxed increases, the plateau phase during which the convergence factors are severely degraded is much longer (Figures 5.135 and 5.136). In view of the results presented in Section 5.4.7, a possible explanation is that during the plateau phase, the fine grid error contains mostly high frequency components which are quickly removed by the solver since it is more efficient on coarse grids. Following the addition of under-relaxation, the high frequencies are removed less quickly and consequently convergence is slower, particularly when residuals are relatively high (so that large corrections are needed).

Even though under-relaxation is undesirable on coarse grids, most of the results presented in this thesis were obtained with a certain (and fixed) amount of under-relaxation on all grids. As finest grids are refined, increasing amounts of under-relaxation are necessary to maintain smooth convergence factors. This parallels the situation for single-phase flows and appears to be characteristic of the automatic differentiation Jacobians on the one hand and of highly non-linear flows on the other.

The pattern for multiphase computations is as follows: on a level 3 grid, as Figure 5.136 shows, convergence histories are less smooth when the Newton corrections are not under-relaxed than those with a small amount of under-relaxation ($\lambda_r = 0.8$). However, they are still more efficient than more strongly under-relaxed computations ($\lambda_r = 0.6$). On grid 4, see Figure 5.137, under-relaxation becomes necessary and the results become very poor as the amount of under-relaxation is decreased. Obviously excessive under-relaxation also degrades the convergence factors (e.g. the case $\lambda_r = 0.4$).

These results indicate that as the grids get finer, the quasi-Newton solver, based on the automatic differentiation Jacobian, becomes less efficient at removing the high frequencies of the error, and under-relaxation facilitates this process.

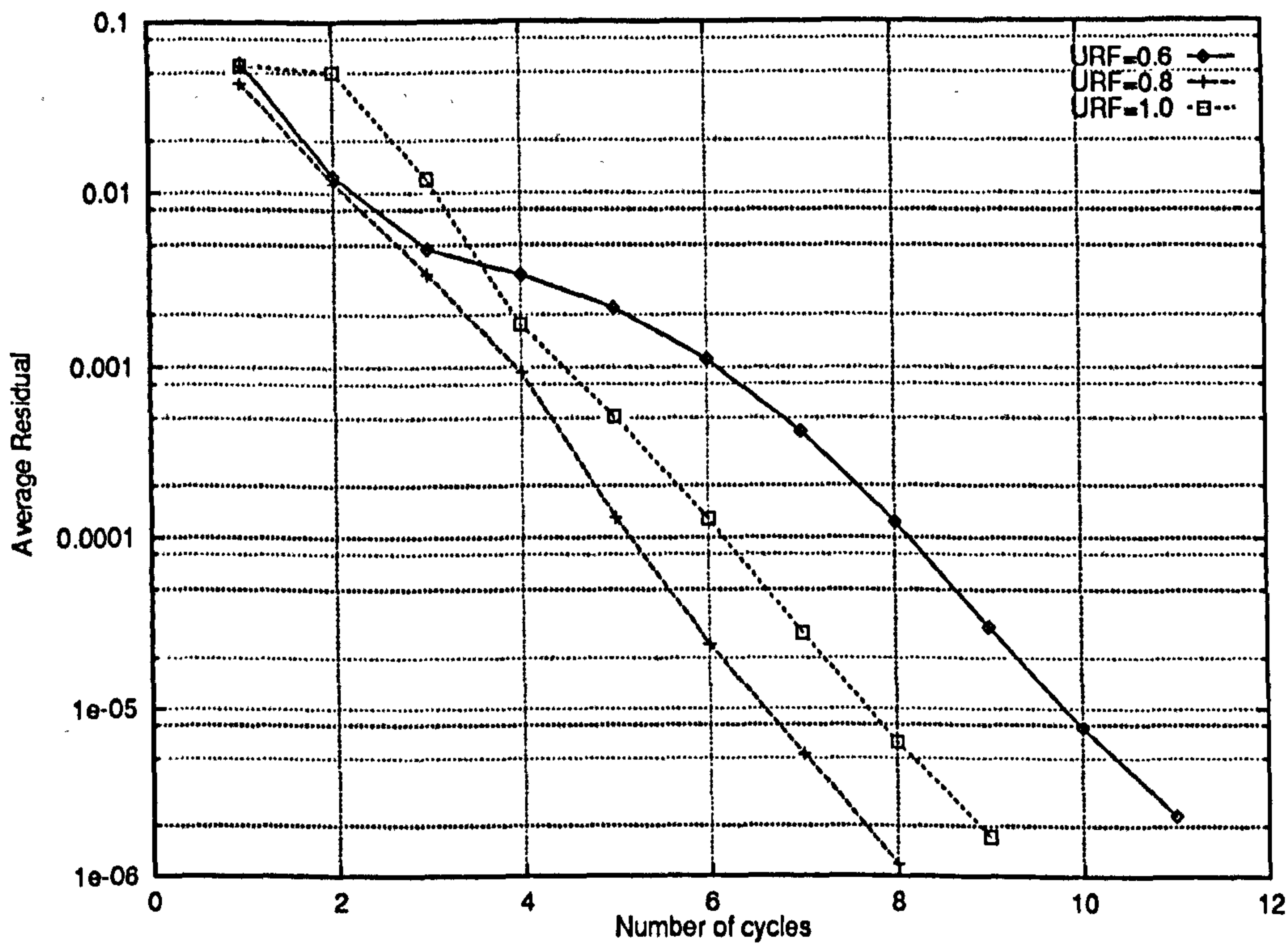


Figure 5.135: Multiphase channel flow – Convergence history for 2 grid computations depending on the amount of under-relaxation

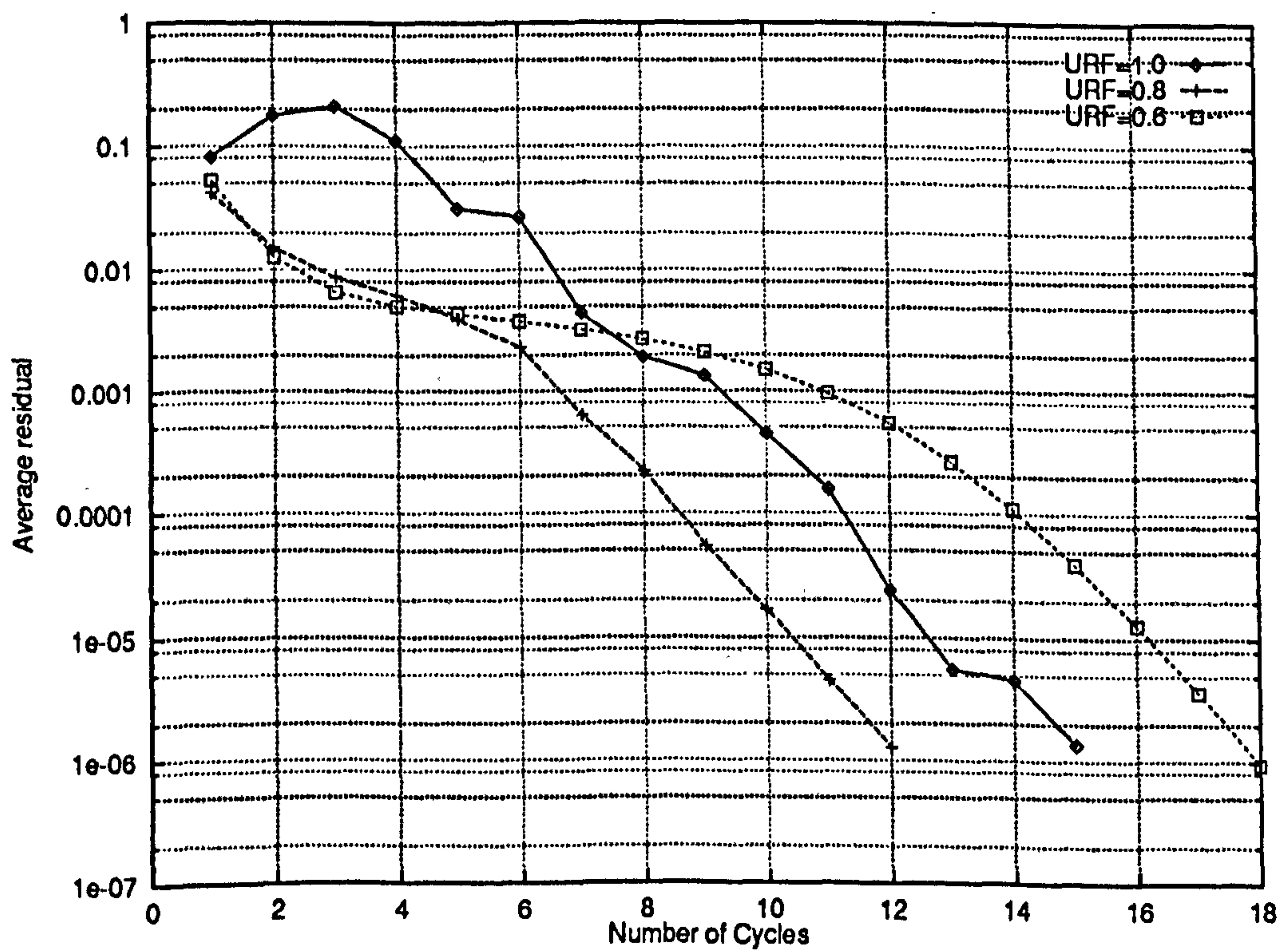


Figure 5.136: Multiphase channel flow – Convergence history for 3 grid computations depending on the amount of under-relaxation

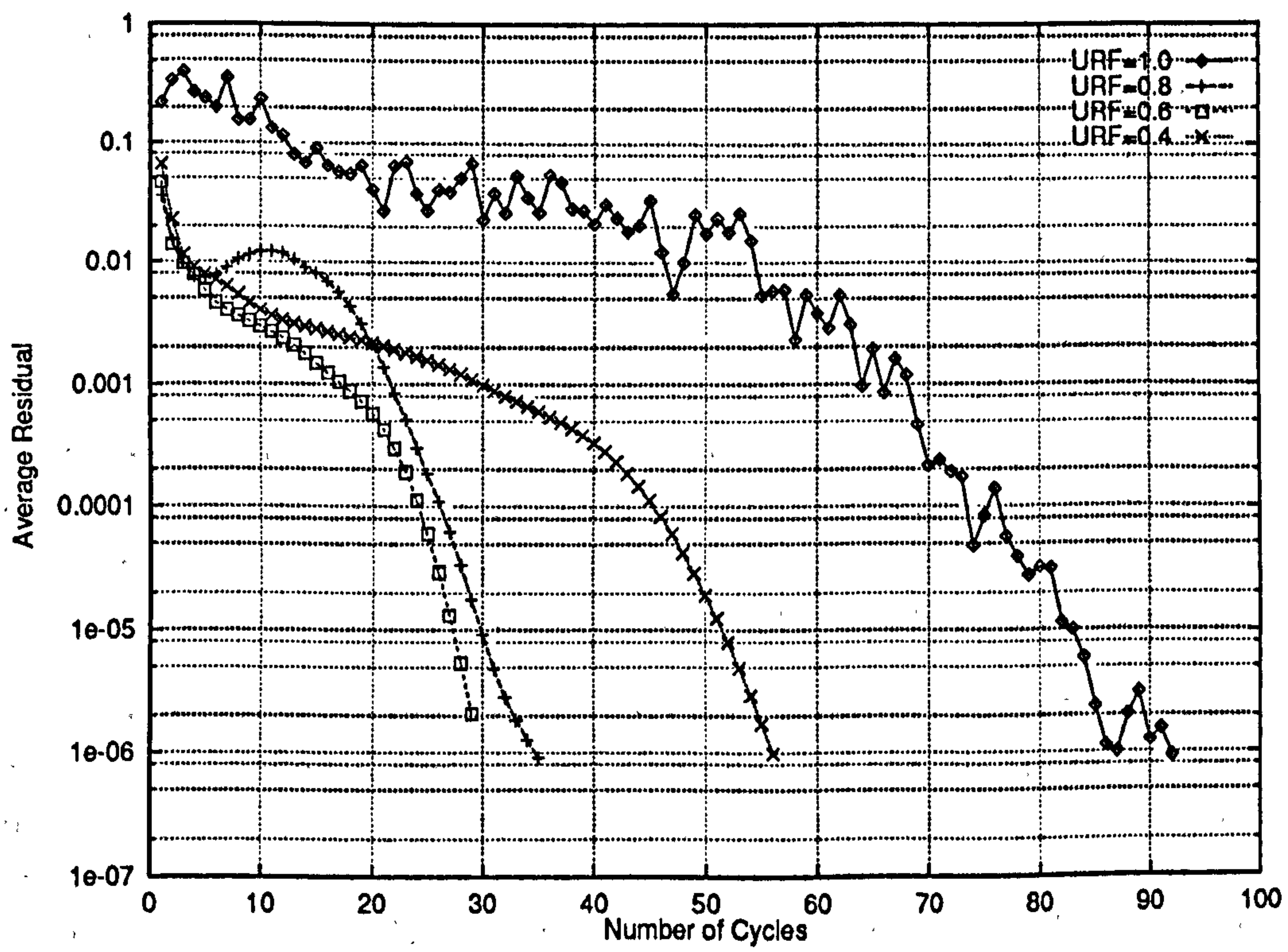


Figure 5.137: Multiphase channel flow – Convergence history for 4 grid computations depending on the amount of under-relaxation

5.4.7 Effect of Multigrid Cycling Parameters

The number of relaxation sweeps performed after prolongation, ν_p , and before a restriction, ν_r , have a large impact on the computational cost of any multigrid method. For an efficient procedure, the values of ν_r and ν_p have to be chosen as low as possible whilst respecting the constraint that sufficient relaxation sweeps should always be performed to reduce the high frequency errors on a given grid.

We adopt the notation $F(\nu_p, \nu_r)$ to completely define the behaviour of the multigrid cycle on finer grids, where the letter F refers to the “F”-cycle cycling strategy.

Unless otherwise stated, the results presented in this thesis were obtained using an $F(2, 2)$ algorithm. These values proved sufficient to obtain convergence but given the high degree of non-linearity of the system, it is worth investigating the effect of devoting more effort to the smoothing of high frequencies error on fine grids ($k \geq 2$). It may be that for this complicated set of equations, the smoother requires more sweeps to significantly reduce the high frequency components of the error.

First, we present a $F(1, 1)$ computation for the two-phase channel flow. The convergence history in terms of number of multigrid cycles is given in Figure 5.138. Compared with a $F(2, 2)$ computation, twice as many cycles are required to obtain the same level of accuracy. However, each cycle is half as costly meaning that the total costs of the computations are almost identical. $F(2, 2)$ cycles are usually preferred to the $F(1, 1)$ alternative because they are slightly more stable.

Similar results for $F(4, 4)$ computations ($\nu_p = \nu_r = 4$) are now discussed: on a level 3 grid, more numerous sweeps on fine grids have a very beneficial effect on convergence factors – see Figure 5.139 which compares the $F(2, 2)$ and $F(4, 4)$ convergence histories. In particular, the plateau phase is visibly much reduced in duration. This is consistent with the observations made in Section 5.4.6: here the larger number of sweeps balances the effects of under-relaxation and suggests that the errors are mostly due to the interpolation operators as these primarily involve high frequency modes. The heavy penalty paid for doubling ν_r and ν_p appears clearly on Figure 5.140 where it can be seen that although far less cycles are required to reduce the residuals, the computational cost is only marginally improved.

Hence, $F(1, 1)$, $F(2, 2)$ and $F(4, 4)$ computations are almost equivalent. This strongly suggests that, for 3 grid computations, the error on the finest grid mostly contains high frequency modes which are efficiently dealt with by the relaxation sweeps.

It is also interesting to note that if under-relaxation is then added, the asymptotic convergence factor is not altered but residuals are prevented from increasing at the very beginning of the computations. This is not surprising: if the number of relaxation sweeps on fine grids is large, then the correction applied on the finest grid will tend to be larger (as the correction equation is more accurately satisfied). However, if the finest grid solution is still very approximate, this correction is likely

to have gone too far. Under-relaxation will tend to decrease the amplitude of the correction.

Finally, $F(2,2)$ and $F(4,4)$ computations were also compared for a level 5 grid (see Figure 5.141) The conclusions are sharply distinct: increasing ν_r and ν_p had no effect on the number of cycles necessary to obtain convergence and each cycle had a much heavier computational cost! We conclude therefore that on fine grids, the error is dominated by low frequency components, which have become relatively more important since the magnitude of the interpolation errors is largely reduced by the use of fine grids. The fact that the error on fine grids is dominated by the low frequency modes suggests that the coarse grid correction is not as effective for finer grids, and again reflects the non-linearity of the operator.

The last cycling parameter which remains to be investigated is the maximum number of coarsest grid relaxation sweeps ν_c . When the average residual on grid 1 becomes smaller than the average residual on grid 2 by a certain factor, relaxation is stopped and the corrections are prolonged immediately to grid 2. For the two-phase channel flow problem, it is usual that only a few relaxation iterations are necessary to reach that criterion and therefore results are not very sensitive to the value of ν_c . Experiments have also been carried out with different values for the threshold for prolongation to grid 2 (the default value is $\epsilon_c = 1/10$). Smaller values for ϵ_c have not significantly altered the convergence factors. This is entirely consistent with our main conclusion that the multigrid convergence factors are limited by grid transfer effects rather than the quality of the coarse grid correction (assessed on the coarse grid by the degree to which it satisfies the coarse grid correction system of equations).

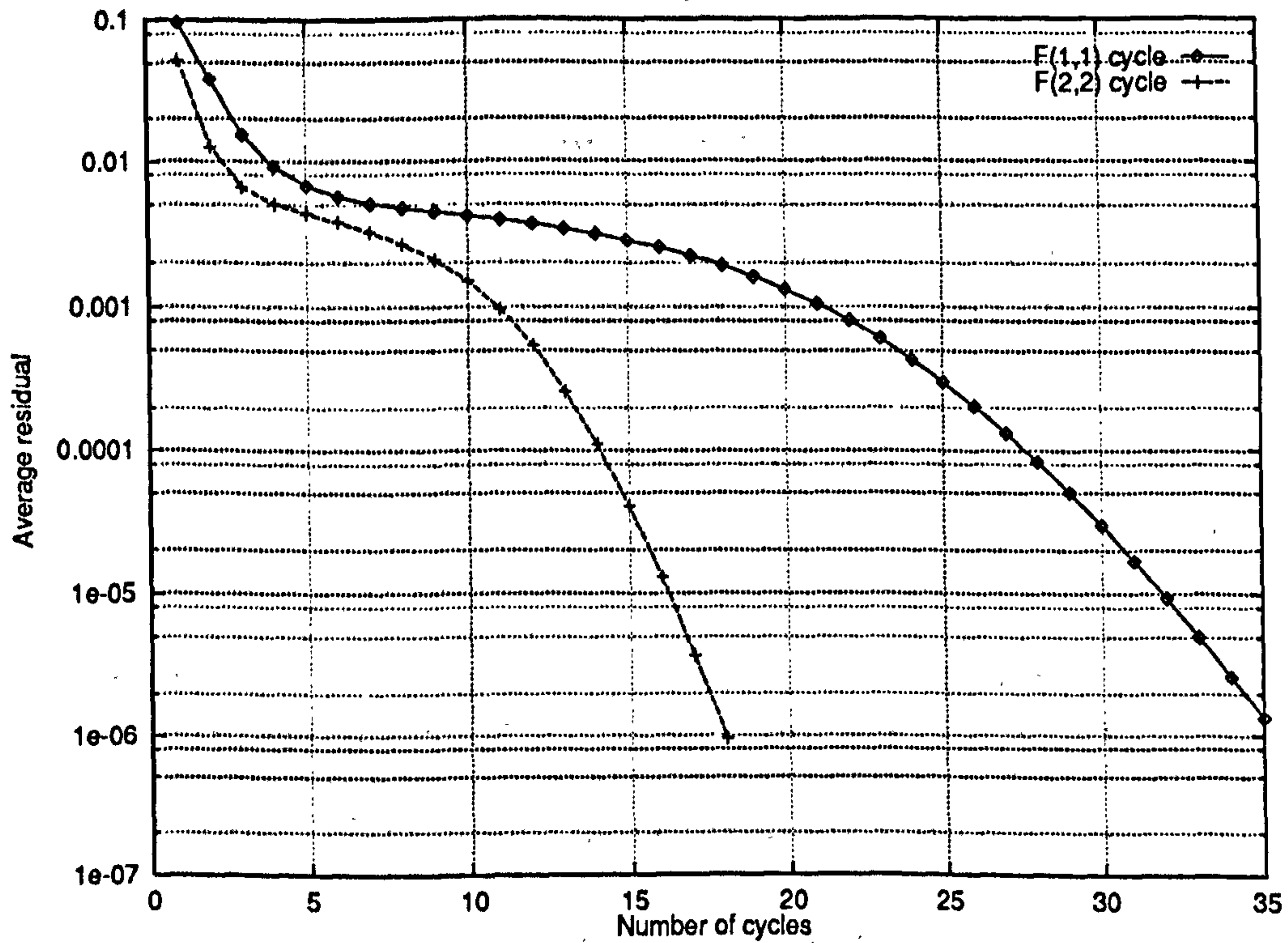


Figure 5.138: Two-phase channel flow – Comparison of the convergence histories of a level 3 computation for different number of relaxation sweeps after prolongation and restriction

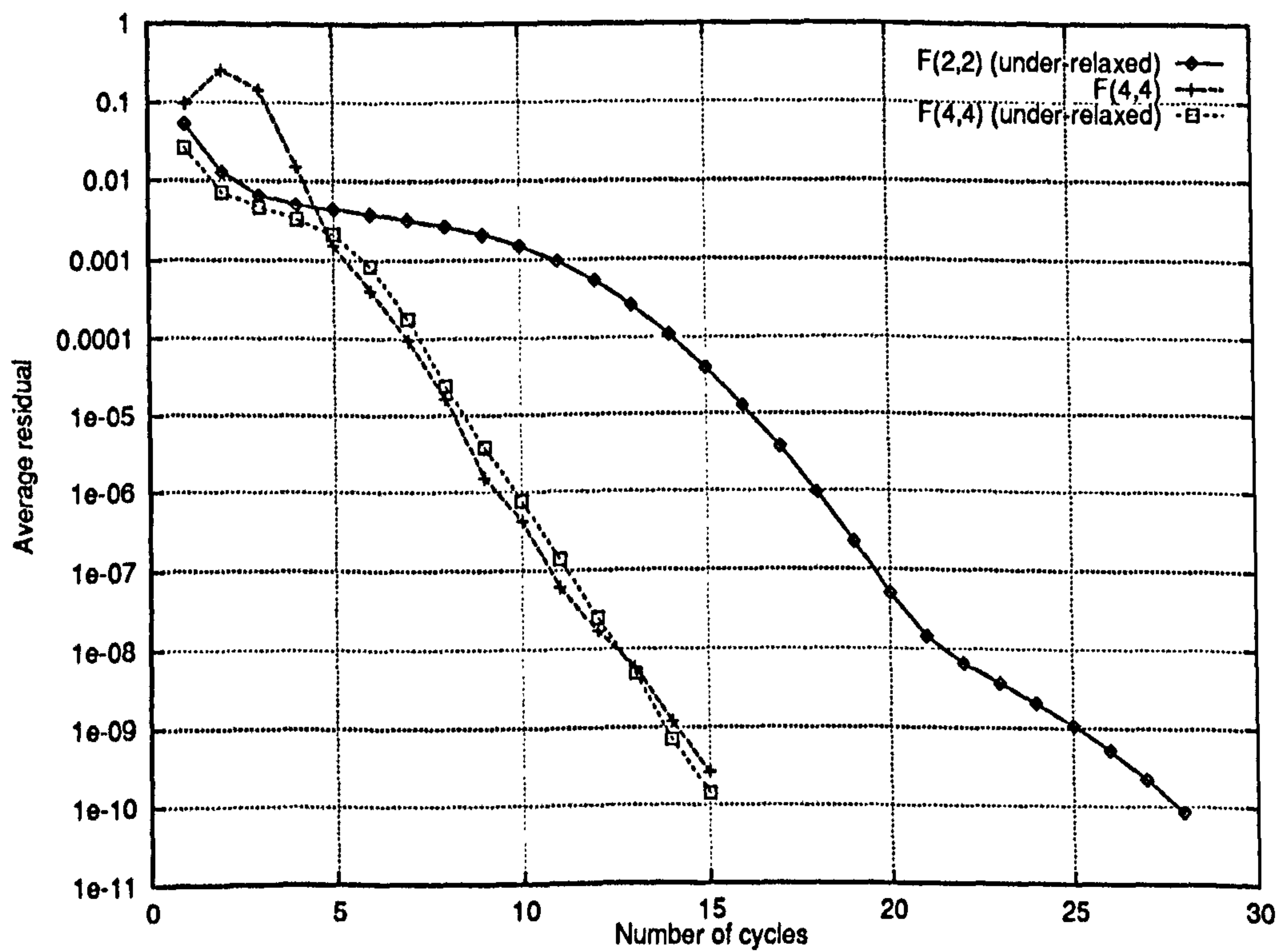


Figure 5.139: Two-phase channel flow – Effect on the convergence history of a level 3 computation for different numbers of relaxation sweeps after prolongation and restriction

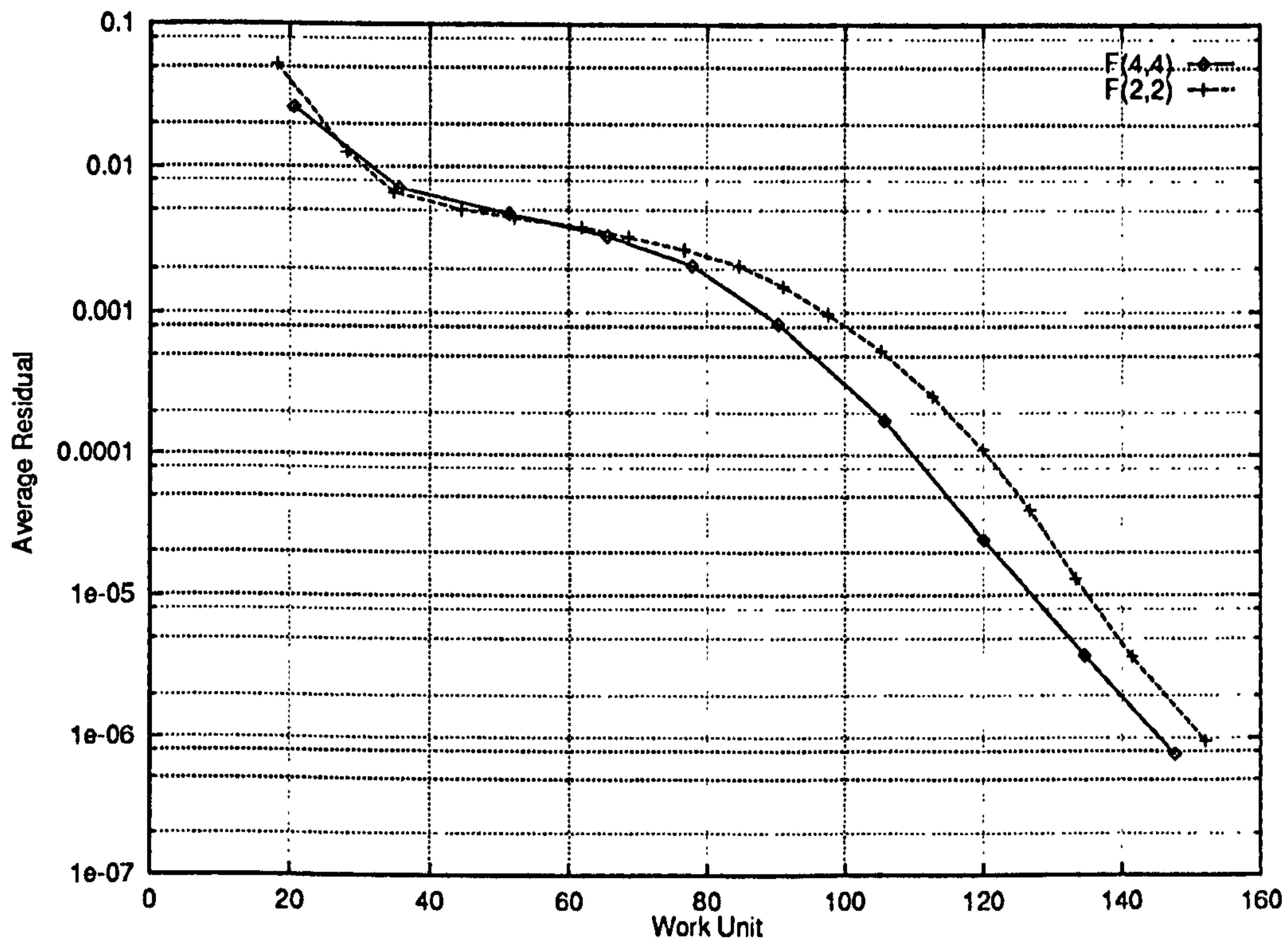


Figure 5.140: Two-phase channel flow – Convergence histories as a function of computational work for a level 3 computation and for different numbers of relaxation sweeps after prolongation and restriction

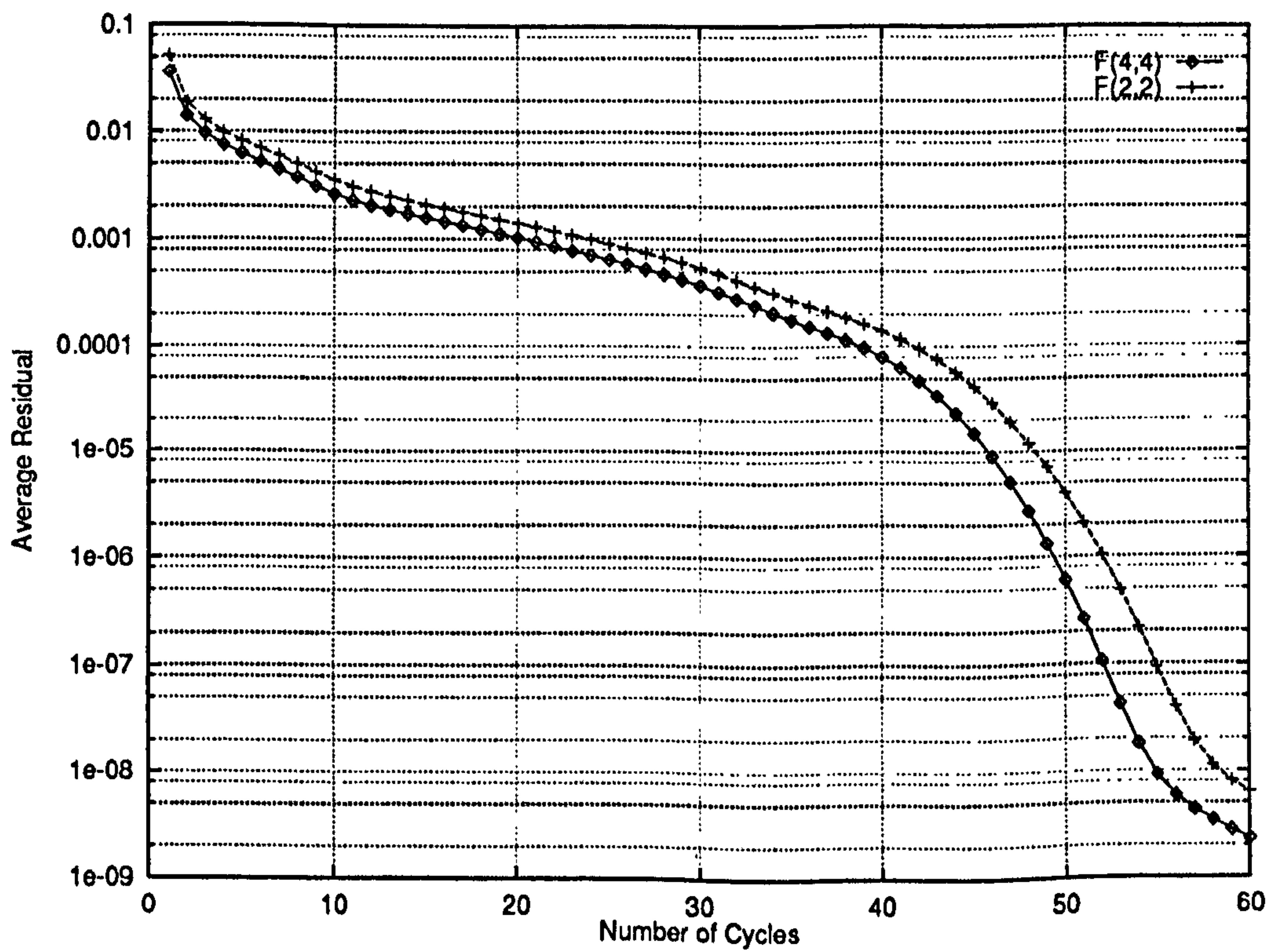


Figure 5.141: Two-phase channel flow – Comparison of the convergence histories of a level 5 computation for different numbers of relaxation sweeps after prolongation and restriction

5.4.8 Some Experiments on Interpolation Operators for the Volume Fractions

In this section, we conclude our analysis of the different parts of the multigrid algorithm implemented in `pang-multiphase`, by investigating the efficiency of the inter-grid transfers. The results presented so far are based on first order interpolation of the pressure and volume fraction corrections during the prolongation stage of the FAS computations (see equations (4.21) to (4.24)).

Although geometric interpolation operators in a multigrid method do not influence the solution on the fine grid (if it is uniform), they may have a very significant bearing on the convergence factors of the method since interpolation errors will need to be eliminated during relaxation. It is therefore legitimate to ask whether the absence of grid independence is due to inaccurate prolongation of the cell centred quantities which is performed to a low order of accuracy. The prolongation of the velocities, being located on edges, is accurate up to second order as explained in Section 4.2.2, and hence should not cause any problems on finer grids.

The pressures will still be corrected using the first order procedure described by equations (4.21) to (4.21). By contrast, the volume fractions are corrected using the second order formulae (4.28) to (4.28). These relationships are valid for “interior” fine grid cells (see Figure 5.142) but modifications are necessary for the boundary cells. It should be noted that the prolongation of the volume fraction corrections on the boundary itself is unnecessary to since boundary conditions are applied on the fine grid just after the prolongation stage.

At cells located just near a solid wall, the volume fraction corrections are still determined by bilinear interpolation but the weights are different (see Figure 5.143). A simple derivation leads, in the case of an upper boundary, to:

$$\delta r_{if,jf}^{(f)} = \frac{1}{8}(3\delta r_{ic,jc}^{(c)} + \delta r_{ic+1,jc}^{(c)} + 3\delta r_{ic,jc+1}^{(c)} + \delta r_{ic+1,jc+1}^{(c)})$$

$$\delta r_{if-1,jf}^{(f)} = \frac{1}{8}(3\delta r_{ic,jc}^{(c)} + \delta r_{ic-1,jc}^{(c)} + 3\delta r_{ic,jc+1}^{(c)} + \delta r_{ic-1,jc+1}^{(c)})$$

At corners, the fine grid correction $\delta r^{(f)}$ are taken to be equal to the coarse grid corrections and the interpolation therefore reverts to first order.

With such a procedure for the prolongation of the volume fractions, experimental evidence suggests that the algorithm is more likely to generate non-physical fine grid values than with the first order procedure. It is therefore critical to monitor the corrected values by implementing one of the tests given by equations (4.44) and (4.45).

It has consistently been observed that with second order interpolation, the algorithm converges less well for fine grids (Figure 5.144). In particular, a larger number of

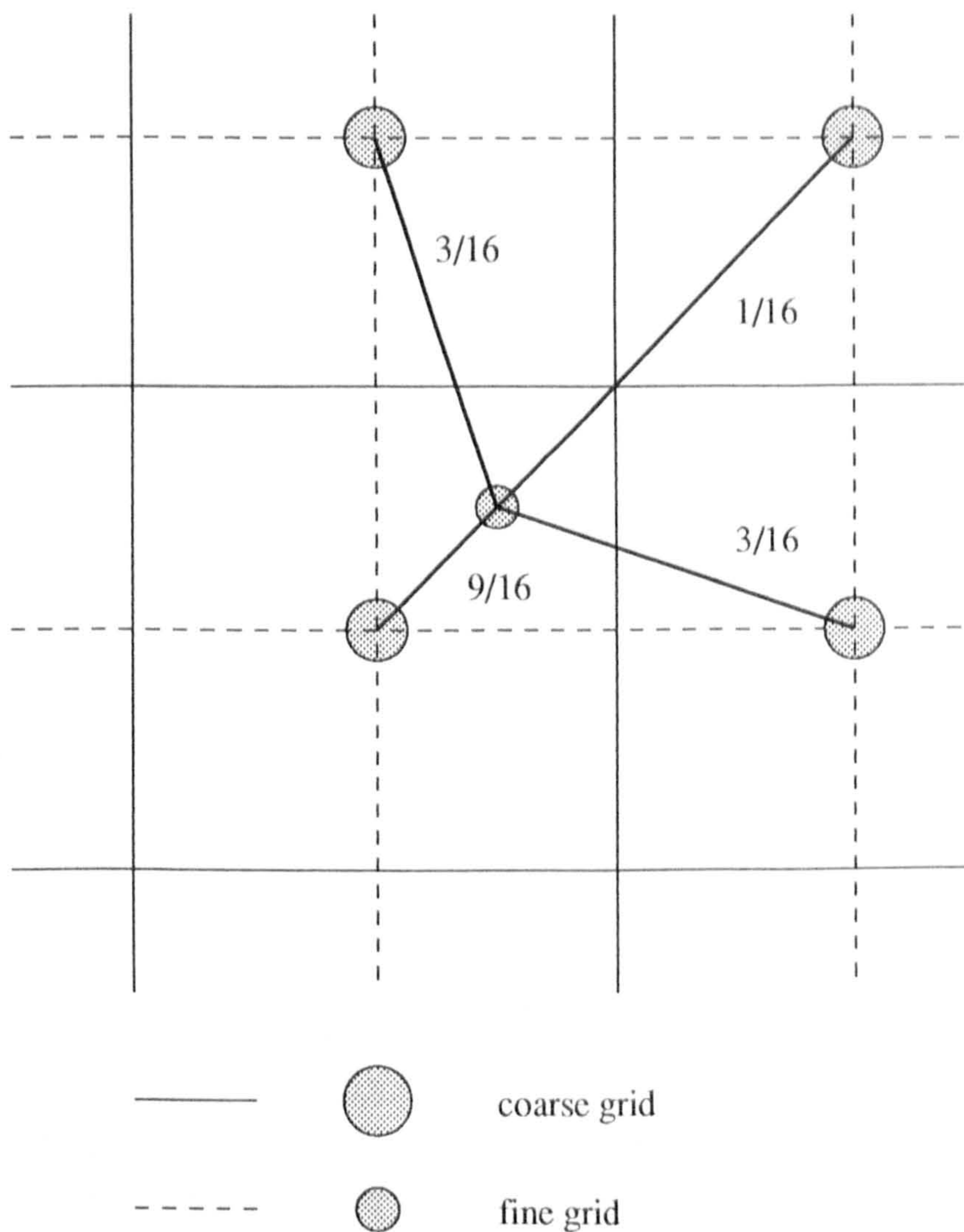


Figure 5.142: Bilinear interpolation for the volume fractions for interior cells

relaxation steps are necessary to get smoother convergence factors (see Figure 5.145) but the number of fine grid relaxations does not appear to affect the underlying convergence factors. It is very interesting to note that between levels 4 and 5, “asymptotic” convergence factors are largely grid independent (see Figure 5.146 and Table 5.11).

On coarse grids, by contrast, second order interpolation does not greatly affect the convergence histories, which are very comparable to the first order case (Figure 5.147) although they appear smoother. In the case of coarse grids, as was noted before in Section 5.4.7, the number of fine grid relaxation significantly affects the convergence factor at large residuals (Figure 5.148) but this is achieved at a heavy price in terms of computational work. Here again, grid independent convergence factors can be observed between levels 2 and 3 (see Figures 5.149 and 5.150 together with Table 5.11)

On coarse grids, the error is dominated by interpolation effects but since grid sizes are still large, improving the accuracy of the interpolation operators only has a marginal effect. For such grids, we argue that the error contains high frequency

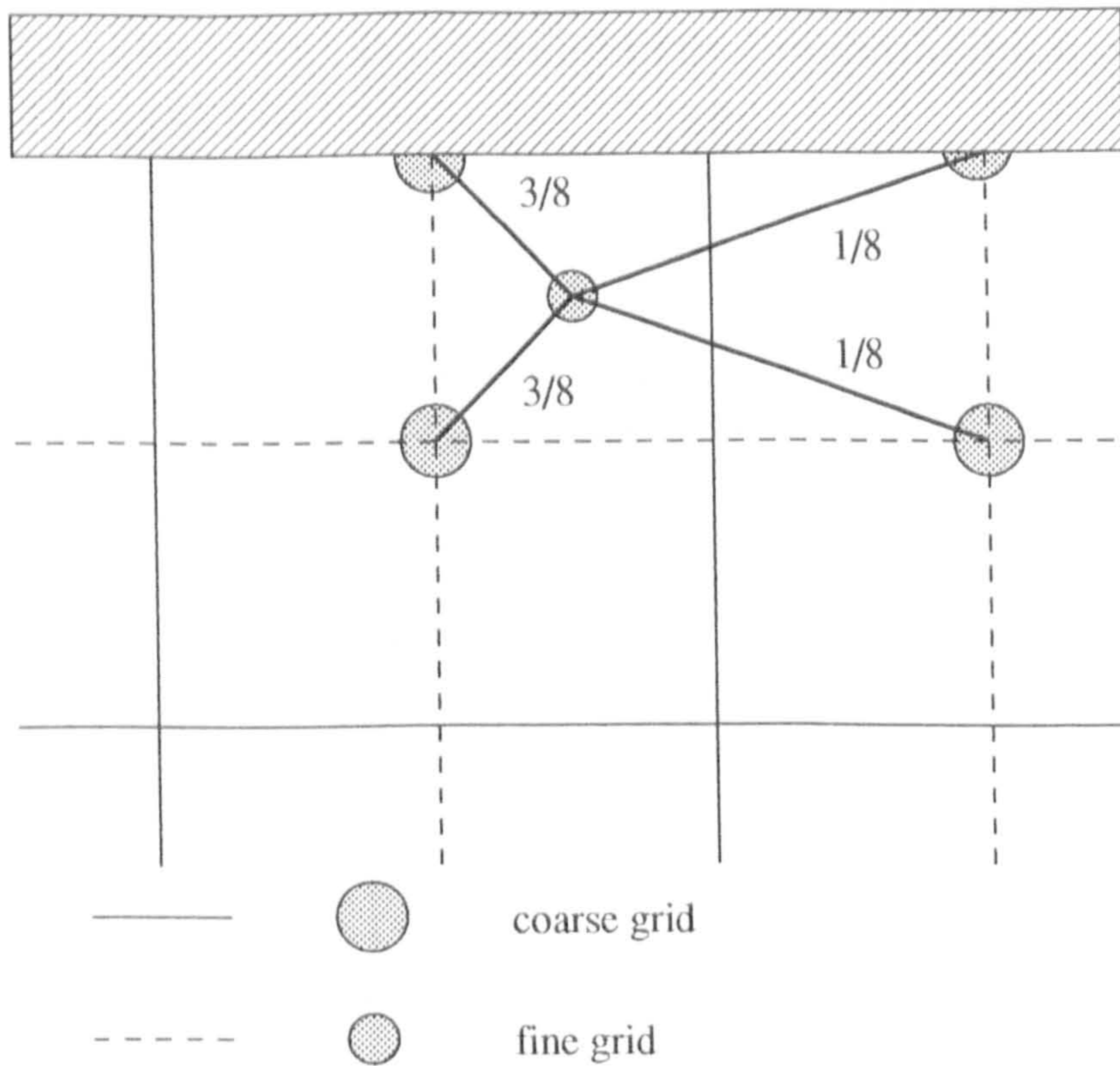


Figure 5.143: Bilinear interpolation for the volume fractions for border cells

modes. More relaxation sweeps on the fine grid will greatly reduce these errors and consequently, multigrid convergence factors should appear more grid independent. A comparison of Figures 5.149 and 5.150 supports this conclusion.

On fine grids, by contrast, interpolation errors are not dominant. As a result, the benefit of high accuracy transfer operators is not visible. Indeed, higher order grid transfers appear to hinder the solution algorithm. However, after the solution algorithm has passed the plateau phase, the influence of interpolation operator is more apparent: the more accurate second order transfer leads to asymptotic convergence factors (during phase 2 – see page 202) which are more grid independent than their counterparts for first order transfers (compare Tables 5.6 and 5.11).

It can be concluded that on coarse grids, as on fine grids, first order prolongation of the volume fraction corrections is sufficient. Implementing second order accurate prolongation operators for the volume fractions slightly increases the grid-independence of the asymptotic convergence factors but also appears to hinder the solver since it introduces higher frequency errors which are not damped effectively. This is counter-intuitive. Moreover, conventional multigrid wisdom states that even higher order interpolation should be beneficial. We believe that the complexity and non-linearity of the system (especially the realisability constraints) prevent the normal mechanisms from working.

Finally, we note that Figure 5.151 clearly shows that convergence factors have different characteristics according to the coarseness of the final grid. These experiments

(and others) have revealed strong cell-Reynolds number effects which could be investigated in more detail.

Grid level	Asymptotic convergence factor
2	0.8997
3	0.8748
4	0.9583
5	0.9560

Table 5.11: Two-phase channel flow problem – Asymptotic convergence factors for different finest levels and for second order prolongation of the volume fraction corrections

This section concludes our investigation into the effects of the parameters which define the multigrid algorithm implemented in `pamg-multiphase` on the convergence factors. We have consistently observed that the error on fine grids is not dominated by high frequency modes. This suggests that the coarse grid correction is less effective as grids get finer. This could well be a direct consequence of the non-linearity of the discrete equations. Operators are different on different grids and need to be approximated and updated as the solution progresses. This will slow the convergence. In the nomenclature defined on page 202, this corresponds to phase 1 of the convergence histories. When the solutions and operators on different grids have adjusted to each other, improved convergence factors are observed (phase 2) up to a certain point when another part of the operators clearly becomes dominant (phase 3).

We will now identify the multiphase effects which are responsible for the degradation of the convergence factors in phases 1 and 3. This is the goal of the following sections.

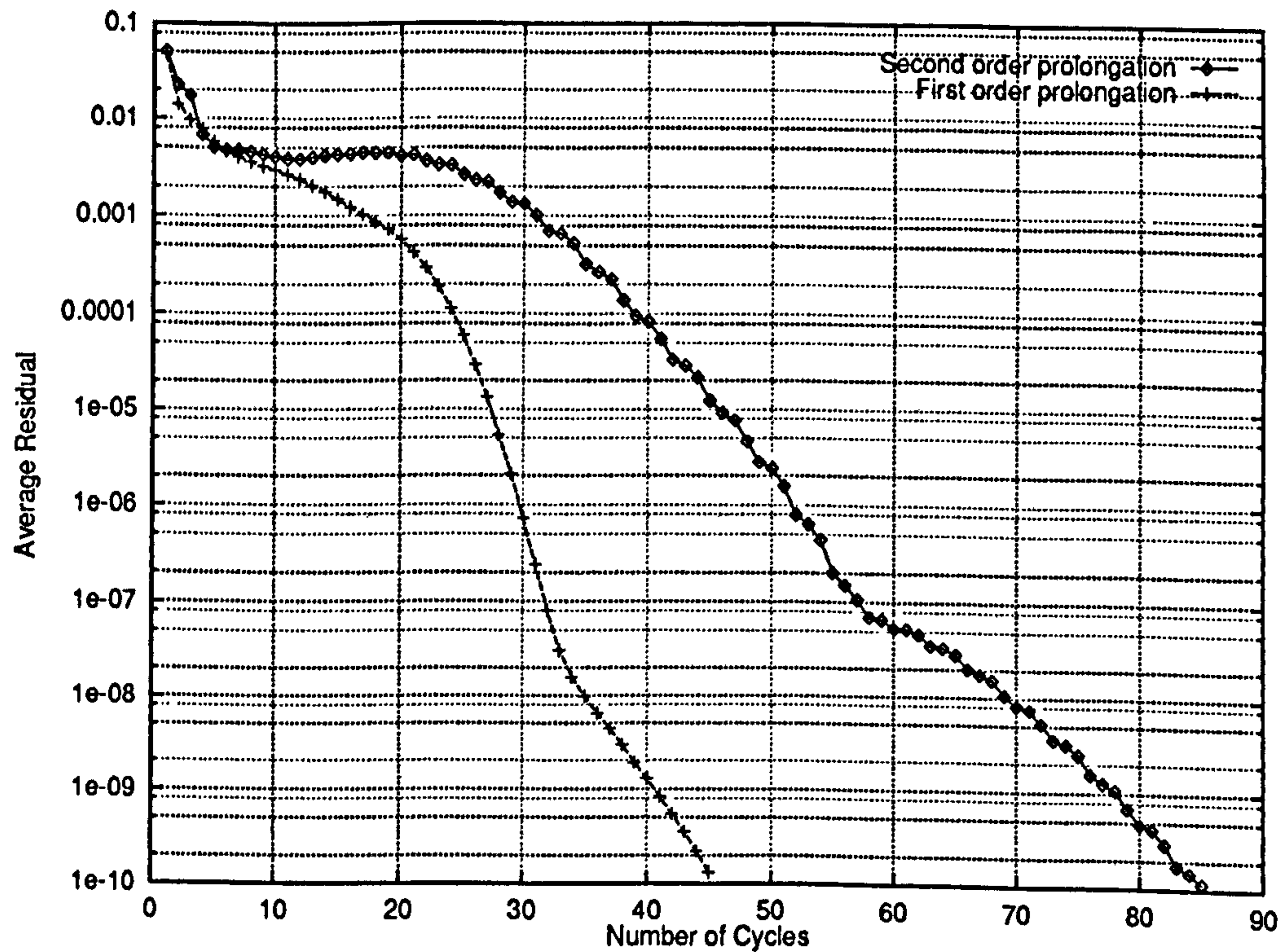


Figure 5.144: Two-phase channel flow problem – Convergence history on a level 4 grid according to the order of interpolation of the volume fraction correction to fine grids

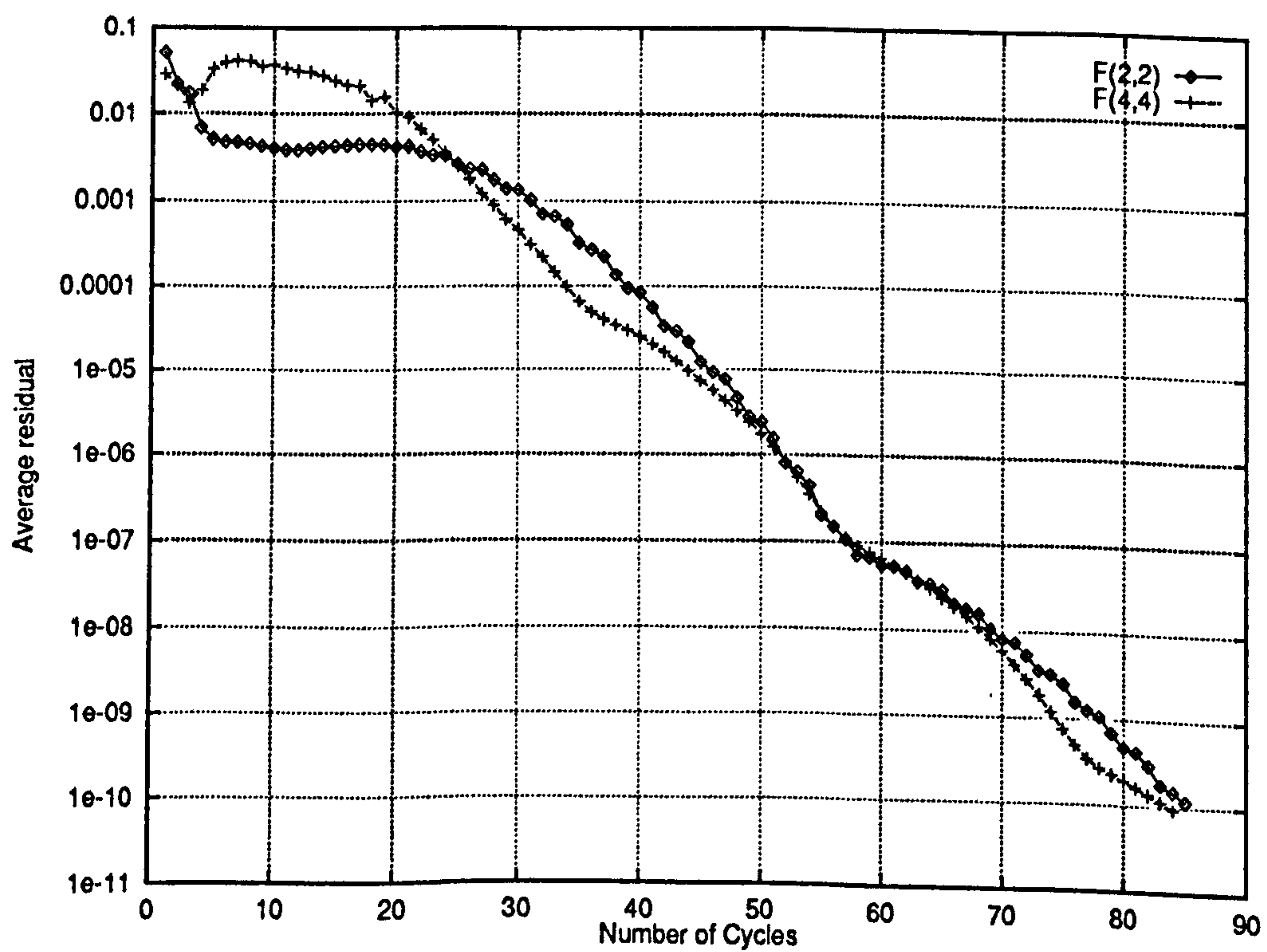


Figure 5.145: Two-phase channel flow problem – Convergence history on a level 4 grid according to the number of fine grid relaxations sweeps $\nu_r = \nu_p$, for second order prolongation of the volume fraction corrections

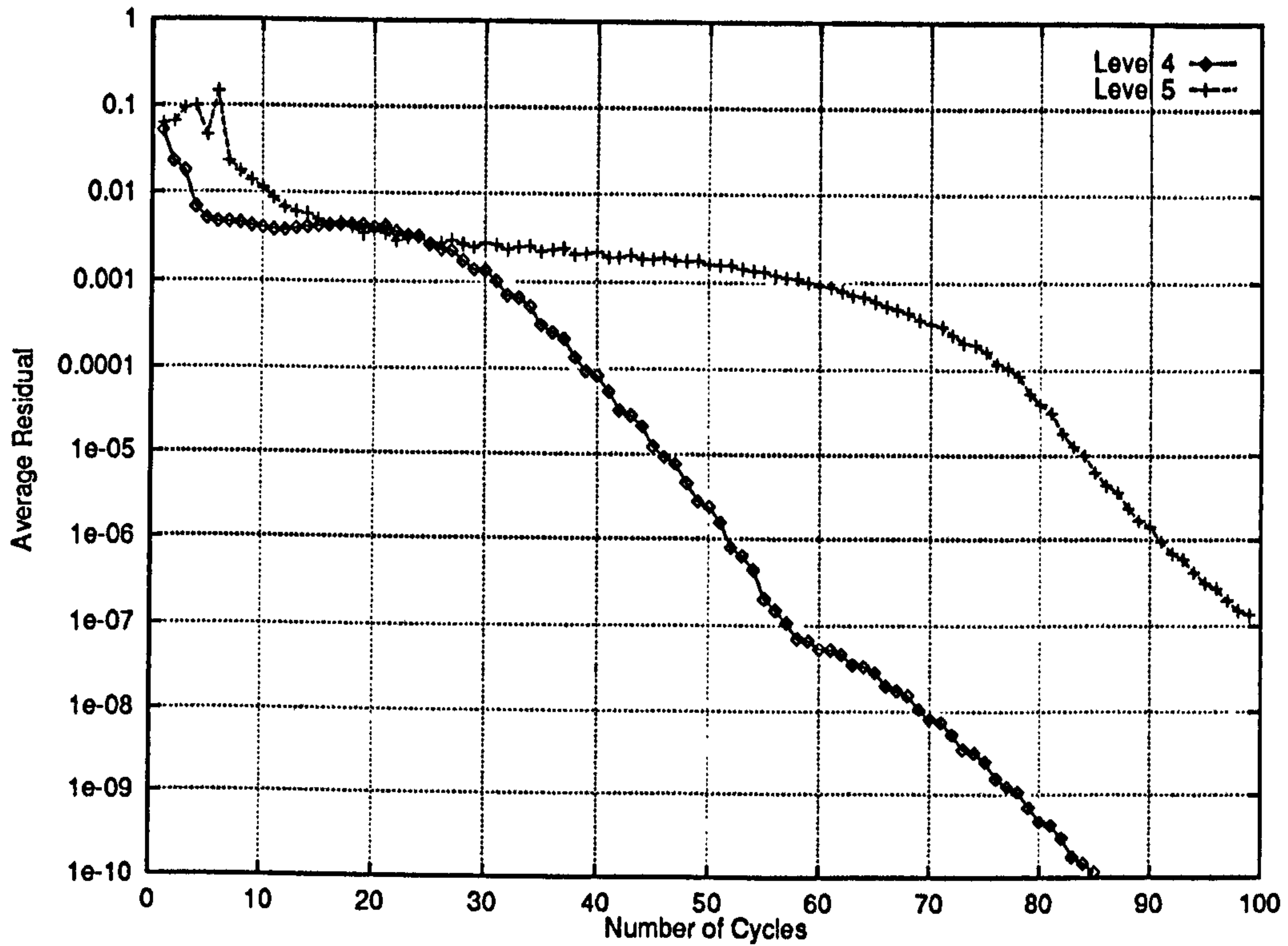


Figure 5.146: Two-phase channel flow problem – Convergence history on finer grids for second order prolongation of the volume fraction corrections

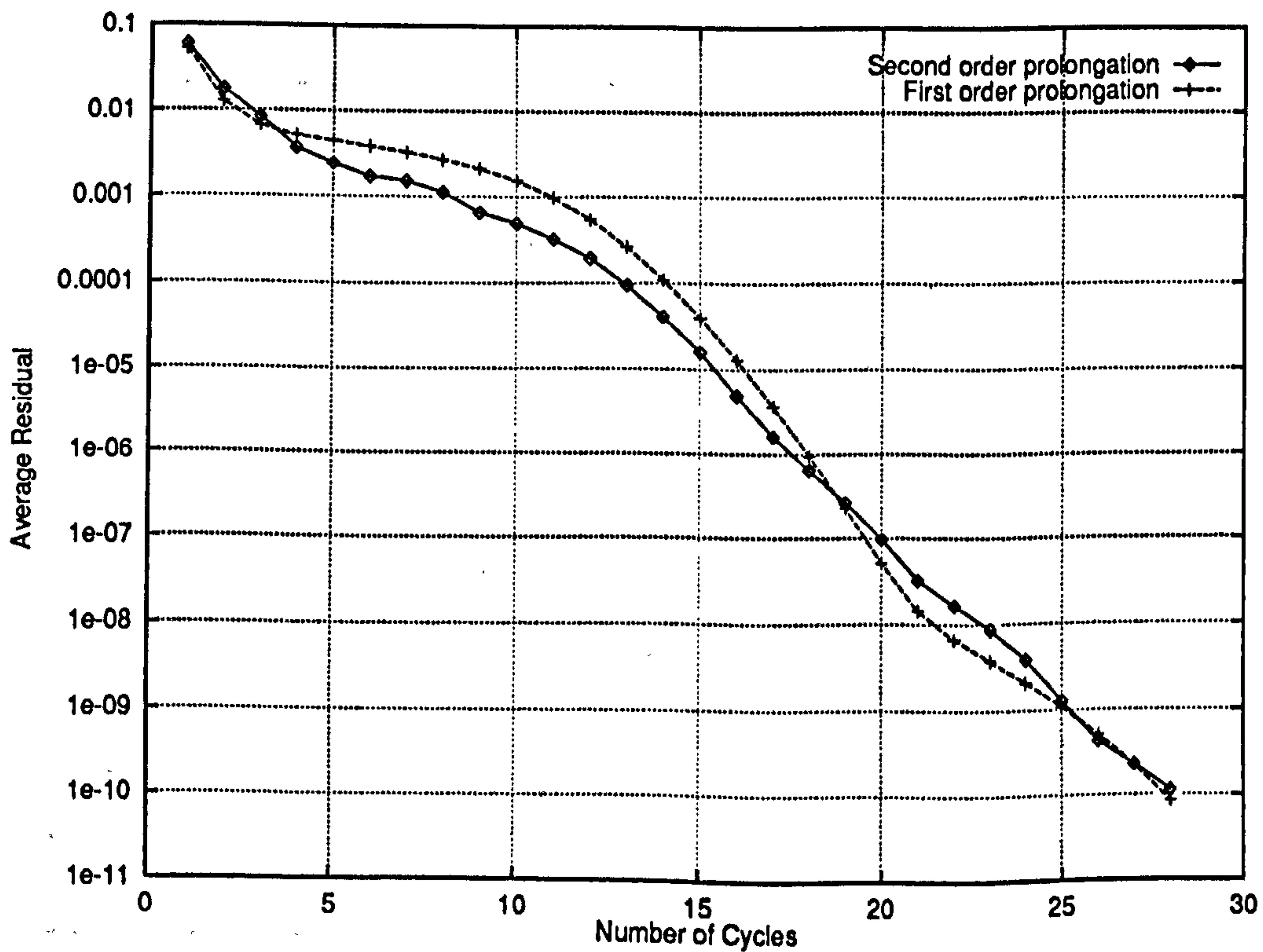


Figure 5.147: Two-phase channel flow problem – Convergence history on a level 3 grid according to the order of interpolation of the volume fraction correction to fine grids

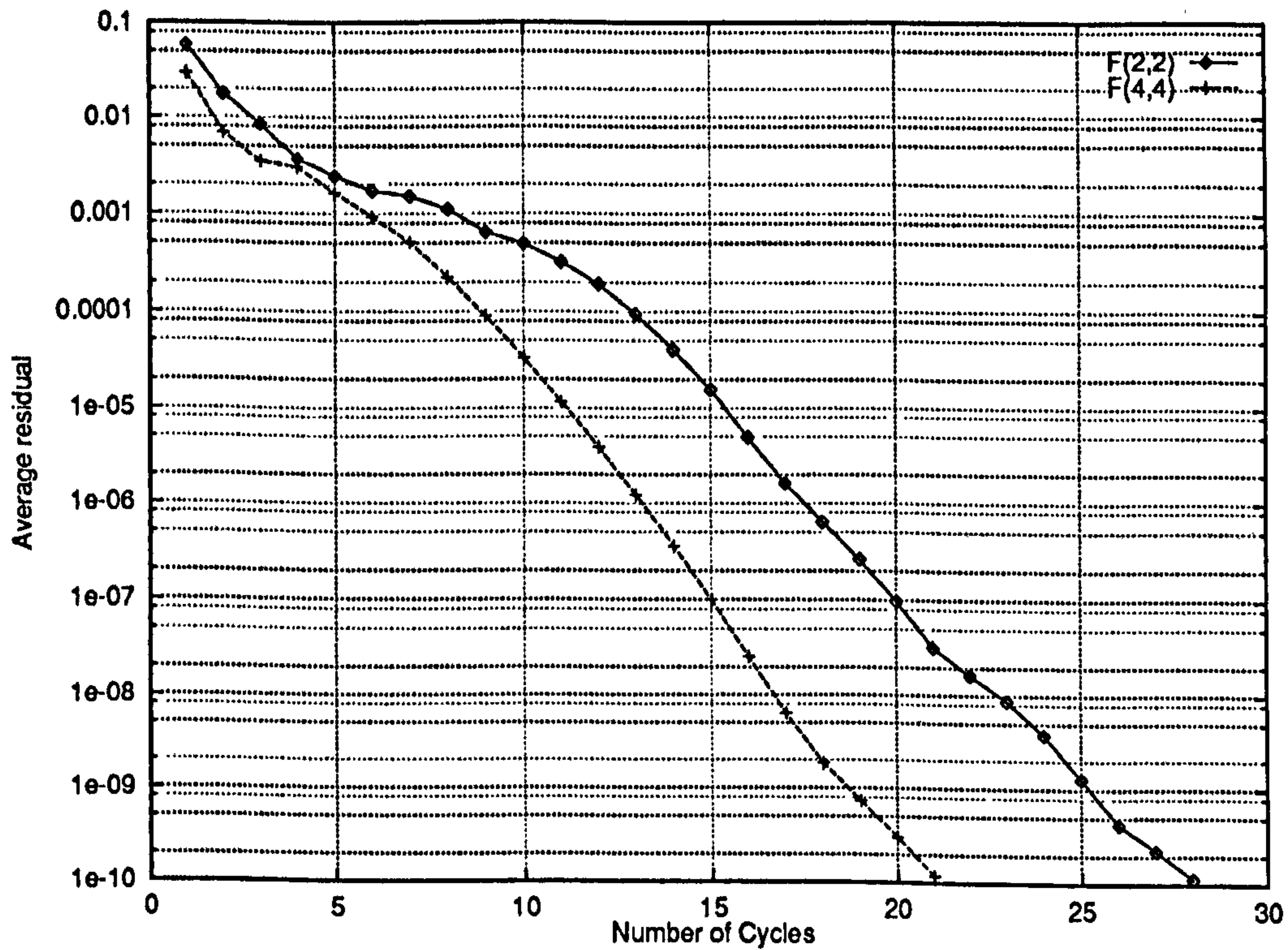


Figure 5.148: Two-phase channel flow problem – Convergence history on a level 3 grid according to the number of fine grid relaxations sweeps $\nu_r = \nu_p$, for second order prolongation of the volume fraction corrections

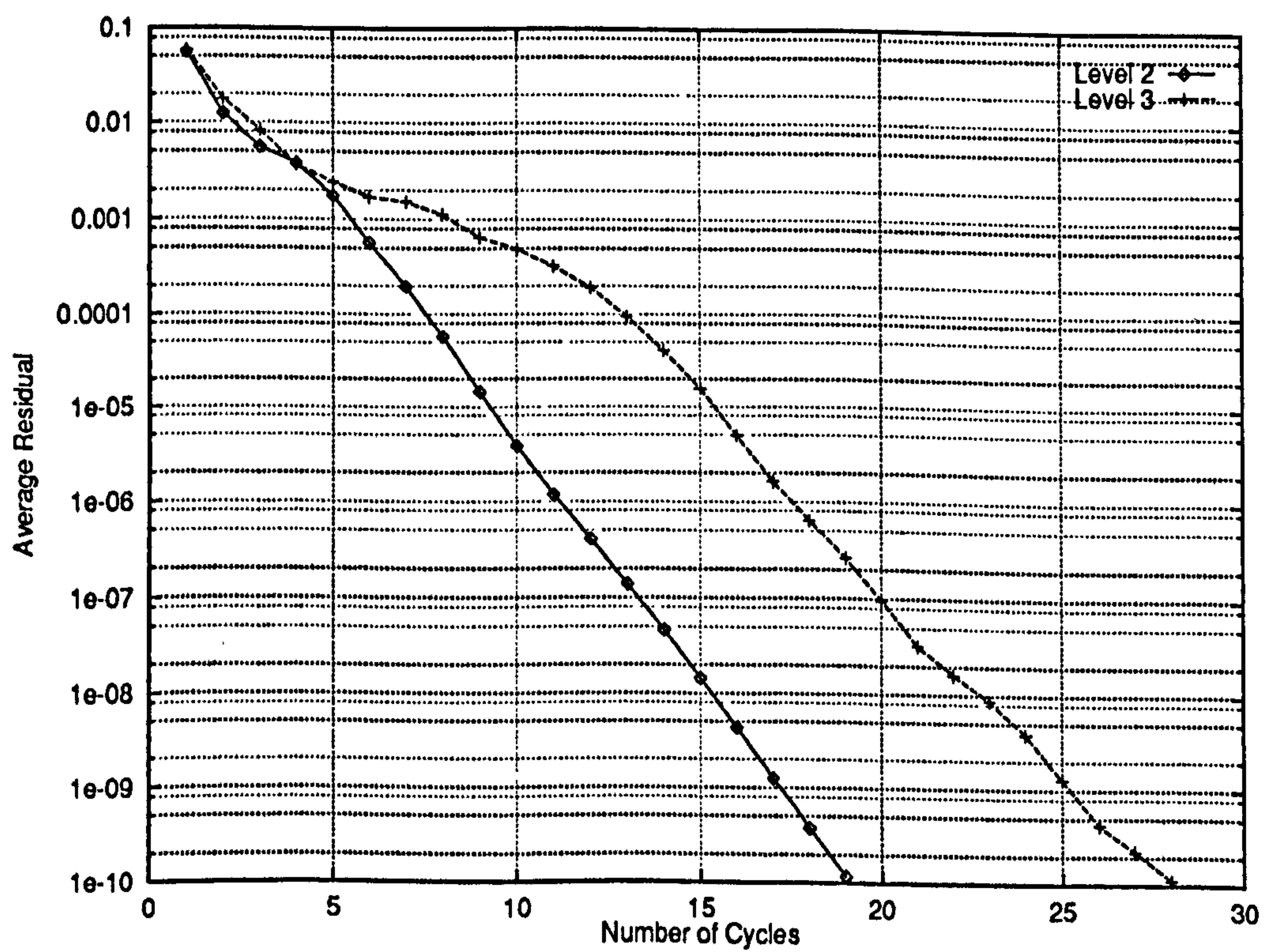


Figure 5.149: Two-phase channel flow problem – Convergence history on coarser grids for second order prolongation of the volume fraction corrections – $\nu_r = \nu_p = 2$

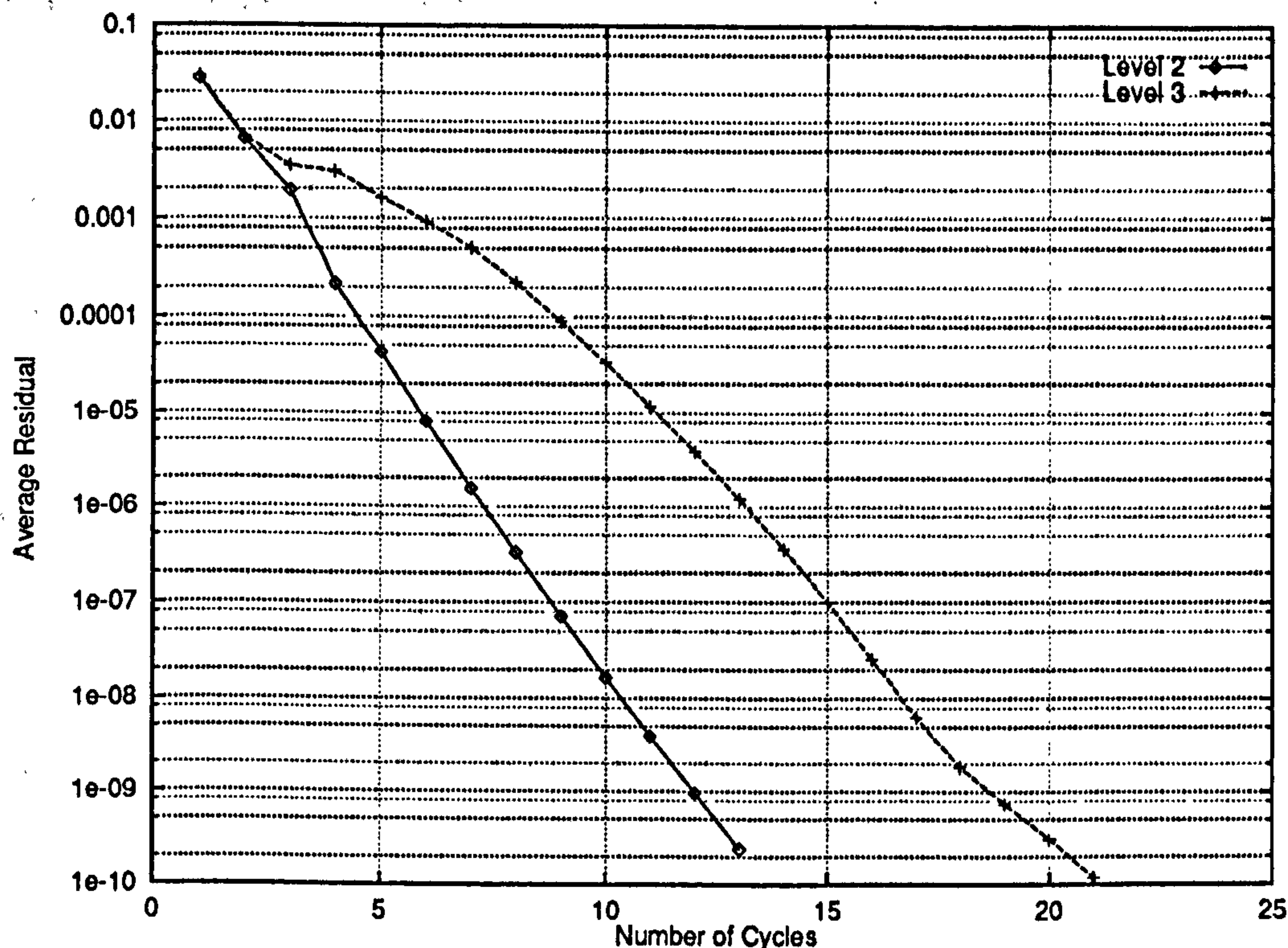


Figure 5.150: Two-phase channel flow problem – Convergence history on coarser grids for second order prolongation of the volume fraction corrections – $\nu_r = \nu_p = 4$

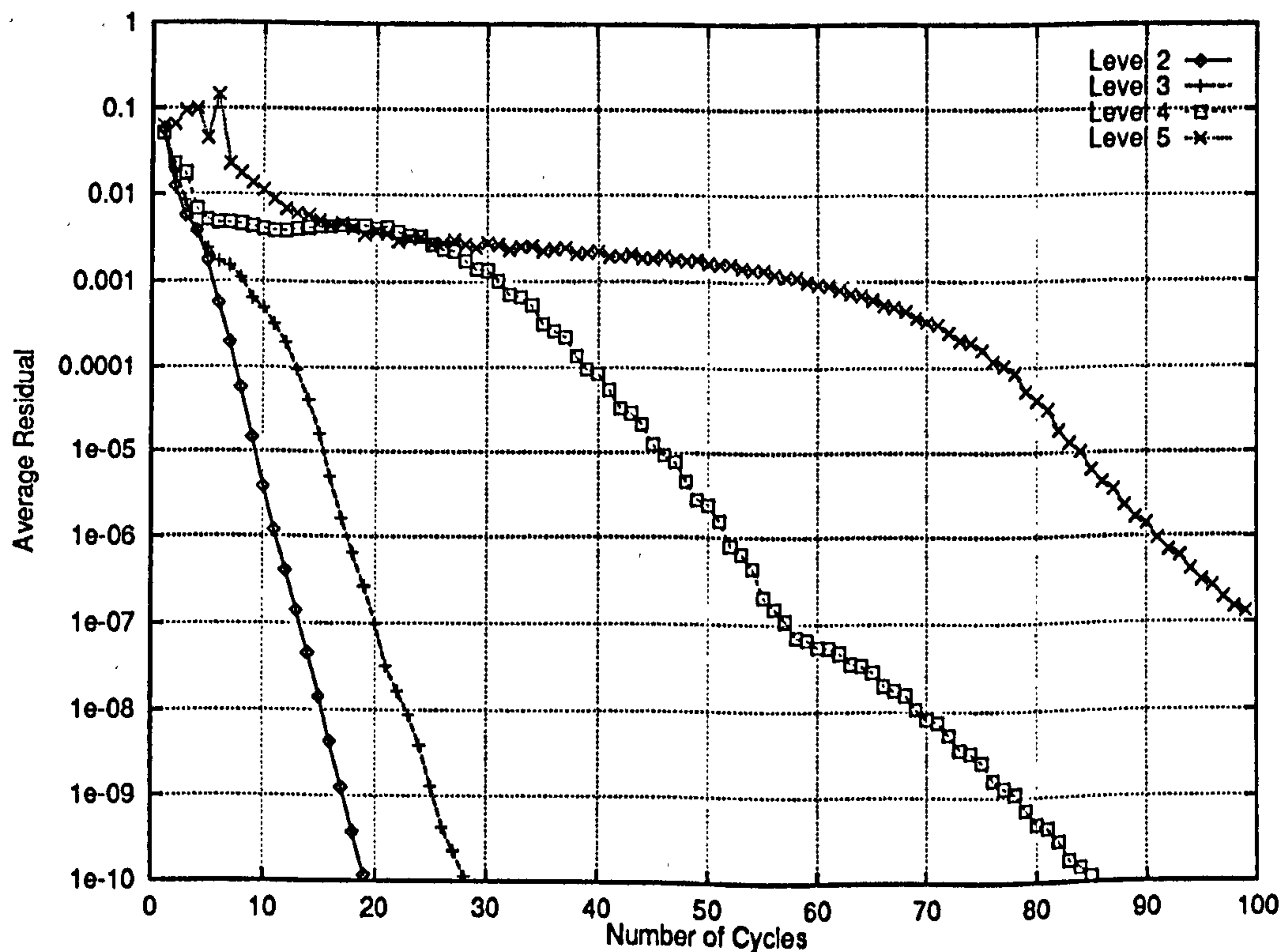


Figure 5.151: Two-phase channel flow problem – Comparison of the convergence history for finer and coarser grids with second order interpolation of the volume fractions and $\nu_r = \nu_p = 2$

5.4.9 Effect of the Source Terms for Inter-Phase Momentum Transfers

In Section 5.3, we alluded to the fact that the source terms which model inter-phase momentum transfers stabilise the multiphase computations. It is reasonable to ask whether, for already stable computations, the convergence factors are improved when momentum transfers are not neglected.

Figure 5.152 shows that this does not appear to be the case. Although the fluids are less separated, the volume fractions are still largely grid-dependent (see Figure 5.153). This is further evidence to suggest that the convergence factors are dominated by the need to approximate different non-linear operators on different grids.

Conversely, and more positively, it further shows that the presence of source terms does not result in deteriorating convergence factors. Since convergence patterns are very close, it can be said that the differential operators effects largely dominate over the source terms.

This does not imply that the algorithmic treatment of the source terms is of no effect. `pamg-multiphase` treats the momentum transfer terms implicitly: their contribution to the Jacobians are taken into account. This feature is greatly facilitated by the use of automatic differentiation to obtain the Jacobians (Section 4.3.1). Some experimental evidence suggests that the implicit treatment of the source term accelerates convergence compared with an explicit treatment.

Experiments have also been conducted for higher values of the transfer coefficients $c_{\alpha\beta}$ (see Section 3.3.1 for a definition). Even in this case, where the relative influence of the source terms is increased, convergence was successfully obtained. In the author's view, it can therefore be concluded that the treatment of the inter-phase transfer source terms in `pamg-multiphase` is good in the sense that it does not adversely affect the rate of convergence of the solver. This is a benefit compared with many segregated solvers. Moreover, in many multiphase flows, source terms are dominant so that a good treatment is essential to convergence.

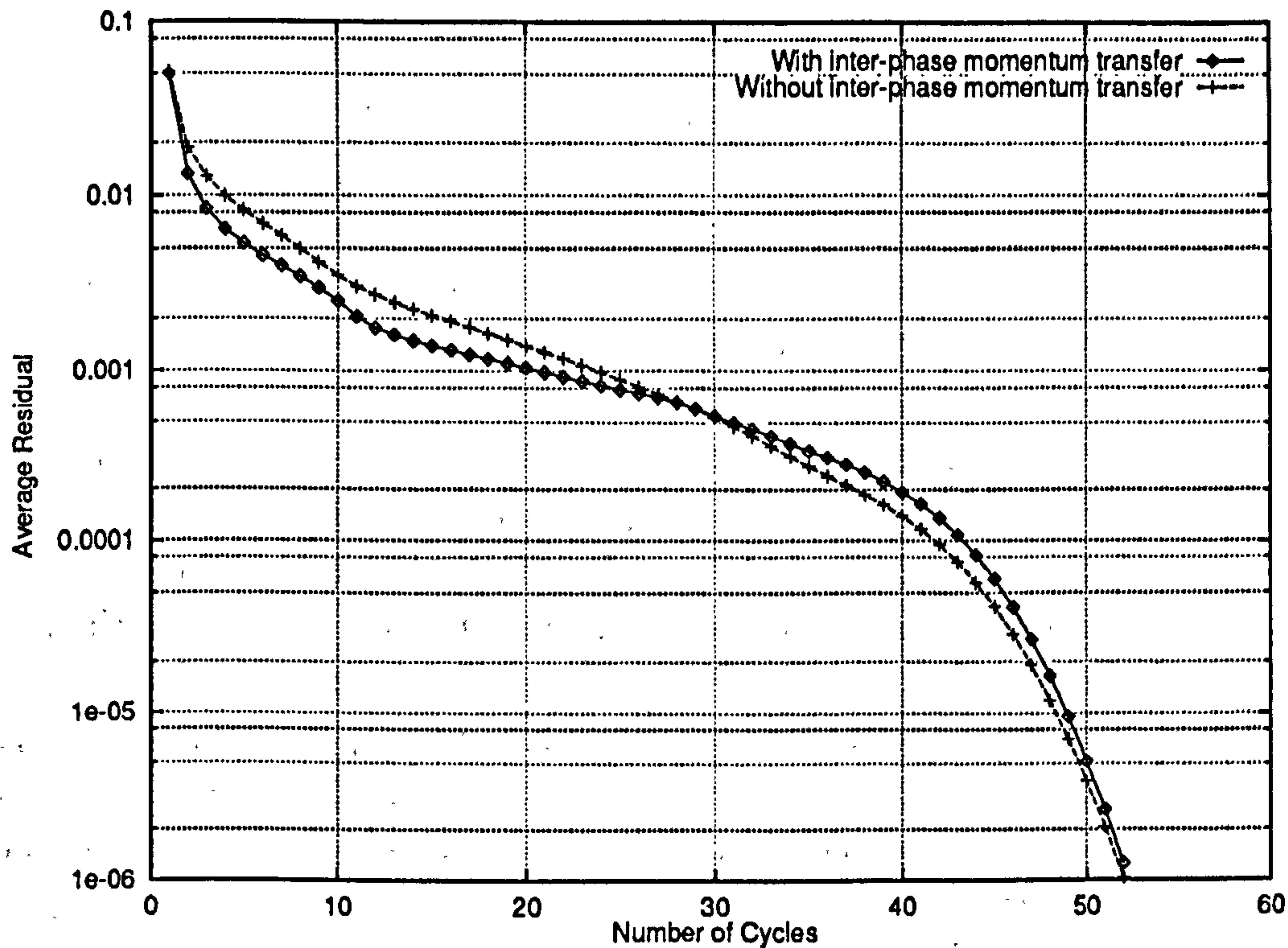


Figure 5.152: Convergence history when inter-phase momentum transfer terms are included

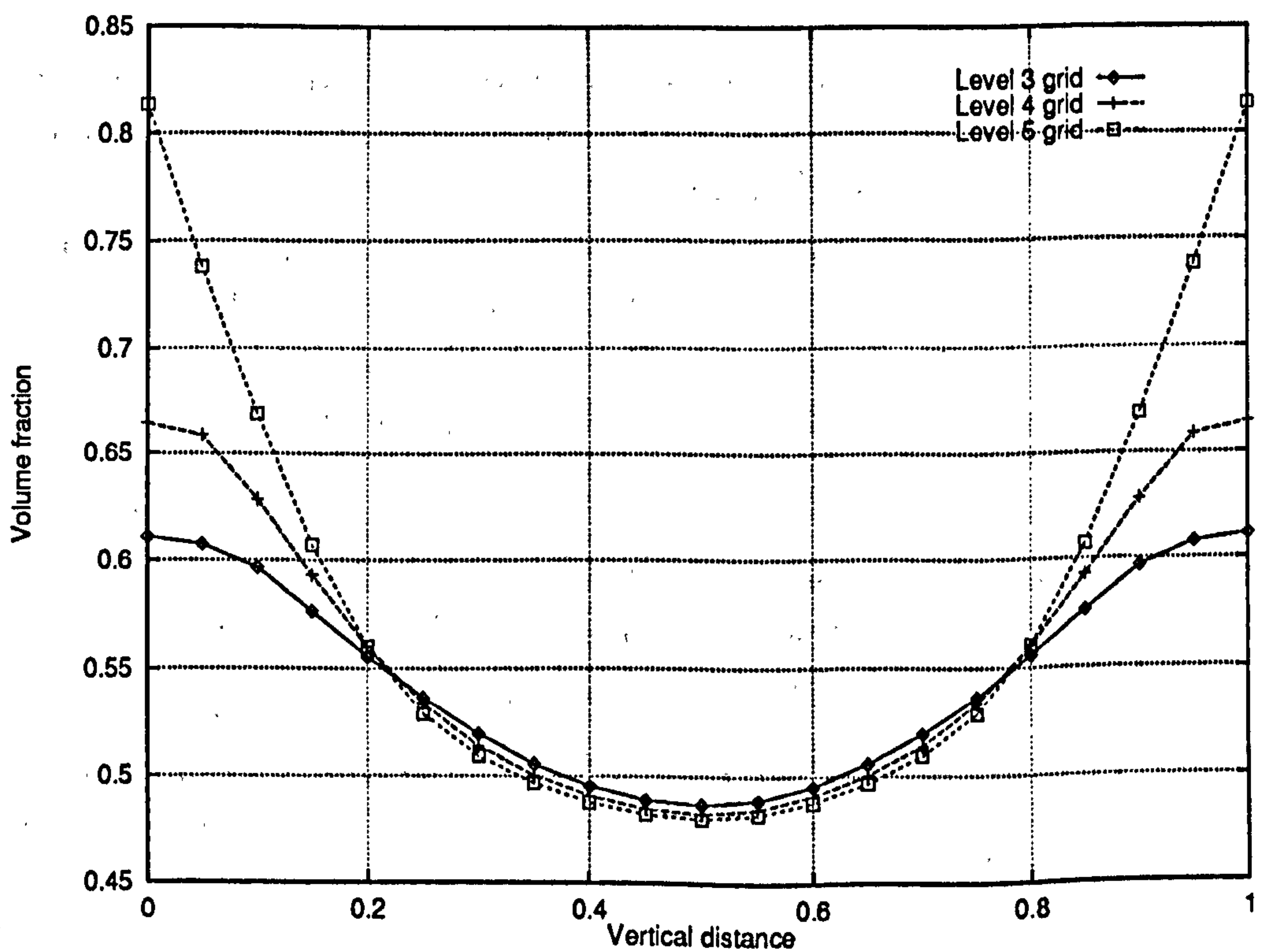


Figure 5.153: Two-phase channel flow problem – Volume fraction profiles along the line $x = 2.5$ for different grid levels when inter-phase momentum transfer terms are included

5.4.10 Convergence Factors in “Difficult” Flow Regions

The solutions of the Navier-Stokes equations are certainly not trivial, even in the relatively simpler case of two-dimensional, laminar, steady flows which is the application domain of `pamg`. Multiphase flows, as demonstrated in the course of this chapter, display a far higher level of complexity. We focus here on two multiphase flow patterns which appear to be particularly challenging:

- Firstly, *there are regions where one phase is driven out*. This renders the discrete equations badly conditioned, due to the fact that volume fractions appear as factors in most of the entries of the local Jacobians. As volume fractions for one phase get close to zero, the condition number of the Jacobian will greatly increase because it will usually admit small eigenvalues in modulus. Overall, `pamg-multiphase` has proven its ability to handle such “dry-out” regions as the example of the T-junction demonstrates. At the T-junction outlet, the flow evolves toward a stratified pattern which is well resolved by `pamg-multiphase` (see Section 5.3.2 and Figure 5.154), without degradation of the convergence factors on sufficiently coarse grids. This may be a direct consequence of using line-searching.

However, as the grids get finer, volume fraction fields in “dry-out” regions approach the limiting values of 0 or 1. In such cases, the potential for generating non-physical volume fractions on the fine grids after correction, increases (see Section 4.3.3). The corrected volume fractions need to be monitored to ensure that only physical values are allowed. In other words, one of the tests (4.44) or (4.45) needs to be implemented. As discussed in Section 6.3, the test chosen can affect the convergence factors very significantly.

In the author’s opinion, difficulties arising in the correction of the volume fractions on fine grids explain why on a level 5 uniform grid, the convergence history for the T-junction is not as regular as for coarser grids (see Figure 5.155). This last convergence history was obtained with the “no-change” strategy (see equation 4.45). Some evidence established for adaptive computations indicates that the “no-change” strategy (equation 4.45) is more robust than the “cut-off” strategy (equation 4.44). It was also verified that the solution after 45 cycles, i.e. just before the convergence singularity, is correct and very close to the solution obtained after 80 cycles.

- Secondly, *there are regions where the fluids may be recirculating*. In some cases, the equations for steady flow will be ill-posed. This arises from the fact that it may be impossible to fix the proportion of the volume occupied by each phase in a recirculation zone since by definition, there is no mass transfer between the main flow and the recirculation region ⁹.

Evidence accumulated during this work suggests that whenever a multiphase recirculation occurs, it is often combined with a “dry-out” zone: the recircu-

⁹If the numerical diffusion of mass introduced by cell-donor differencing is neglected.

lation zones tends to contain only one phase. Physically, it may be possible to correctly model the recirculation zone by reverting to a single phase flow model and prevent the absent phase from entering the region. In the present version of `pang-multiphase`, the governing equations are not reformulated in recirculation zones.

There is some evidence that a recirculation zone can in some, but not all, cases greatly degrade the performance of the multigrid method. A good model problem is provided by a two-phase flow past a backward-facing step. Two-phase flows for small Reynolds numbers ($Re \approx 10$) have been successfully computed (see Section 5.3.3) and good convergence factors are obtained, as Table 5.12 and Figure 5.156 show.

Grid level	Number of Cells	Convergence Factors (per work unit)		
		to 10^{-5}	between 10^{-5} and 10^{-6}	Global
2	768	0.9574	0.9413	0.9532
3	3072	0.9601	0.9753	0.9660

Table 5.12: Multiphase backward-facing step problem – Comparison of convergence factors on level 3 and 4 grids for low Reynolds numbers

Computations were not successful for larger values of the Reynolds numbers (see Figure 5.157 for an example at $Re \approx 100$). It was first verified that the failure was due to the multigrid algorithm rather than the coupled solver. The equivalent single grid computation was successful although the convergence history was not very regular (see Figure 5.158). Plots of the streamlines (Figures 5.159 and 5.160) as well as velocity profiles (Figures 5.161 to 5.165) show that the recirculation zone is much larger than for Reynolds numbers of lower values. More significantly, *both* phases have recirculation zones (Figure 5.166). In the author's opinion, only this second type of recirculation in which both phases recirculate is ill-posed, because there is no means of fixing the volume fraction in the smaller recirculation zone. For completeness, the volume fraction distribution associated with the backward-facing problem at higher Reynolds numbers is shown in Figure 5.117.

Examination of these two difficult flow patterns reveals that apart from issues of mathematical well-posedness, `pang-multiphase` is able to simulate these regions successfully, at least for sufficiently coarse grids. Some minor aspects of the solution algorithm may require fine-tuning, but the difficult flow regions are not the primary cause of the degradation of the multigrid convergence rates.

The fact that a multiphase steady state problem is not-well posed when more than one phase is recirculating at any given point is, to the best of the author's knowledge, a novel type of singularity. When an operator is singular, it cannot be inverted. The Fredholm alternative states a solution only exists if certain conditions are met and

are then is non-unique. In the context of fluid flow, singularities can fall into three categories.

1. *Pressure singularities*: in incompressible flows, the pressure is defined with an arbitrary additive constant. This is unimportant since only pressure differences occur in the discrete equations.
2. *Bifurcations*: away from bifurcation points, the solutions are well separated and multigrid solvers are usually successful. A good example is provided by a single phase viscous flow through a sudden expansion. See for instance reference [1].
3. *Non-separated solutions*: in this last case, the solutions are no longer locally unique in solution space. There are real, different but close solutions. This situation, which appears to prevail for multiphase flows when both phases are recirculating at a given point, is much harder to deal with. The fundamental reason why the multigrid solver fails while the single grid solver is successful is that on different grids, it approximates different, non-equivalent, solutions.

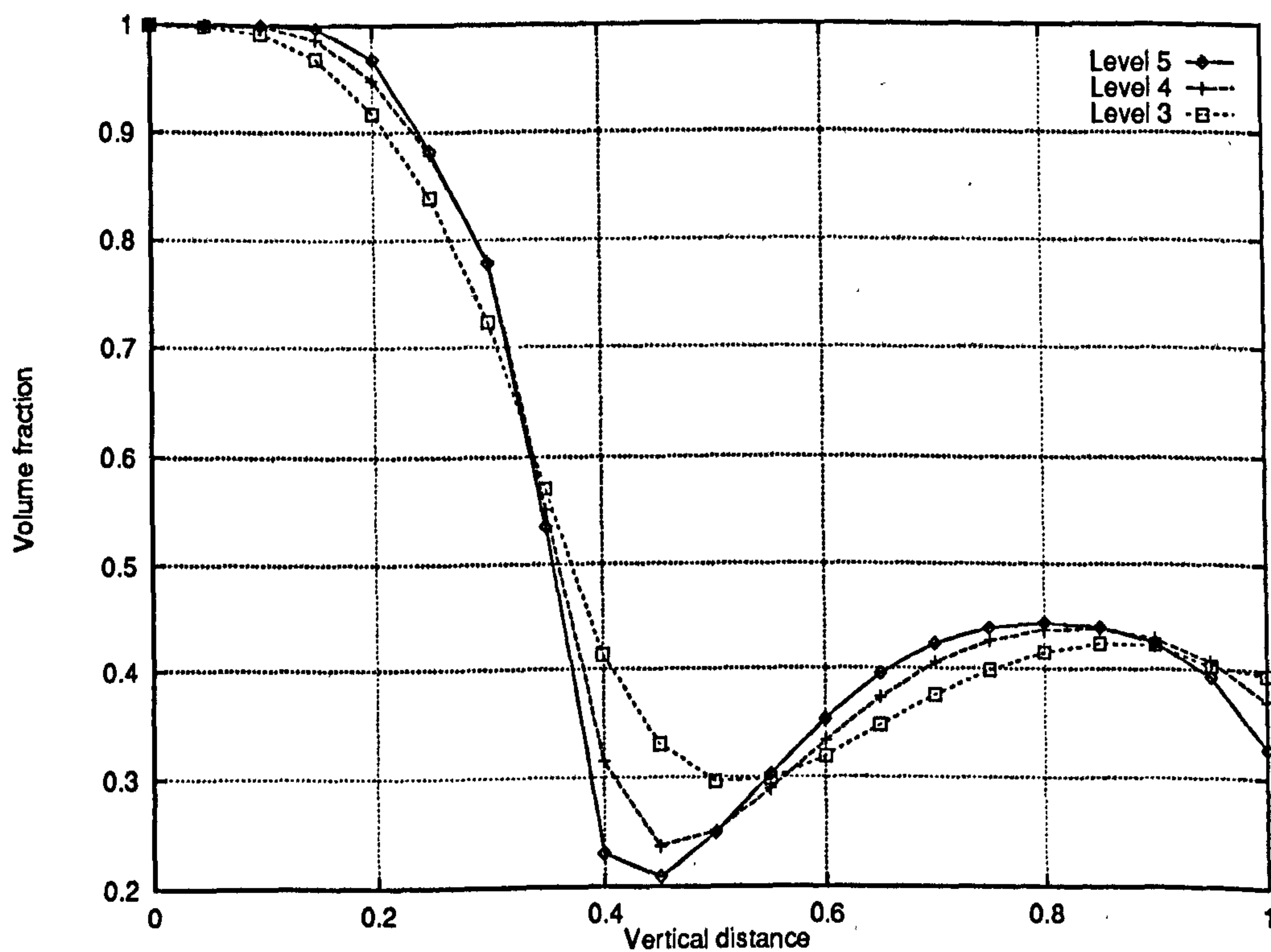


Figure 5.154: Two-phase flow through a T-junction – Volume fraction profile along the line $x = 6.5$ at different grid levels showing a “dry-out” zone, i.e. a complete separation of the flow which is more established at level 5 – See page 123 for the geometry of the problem

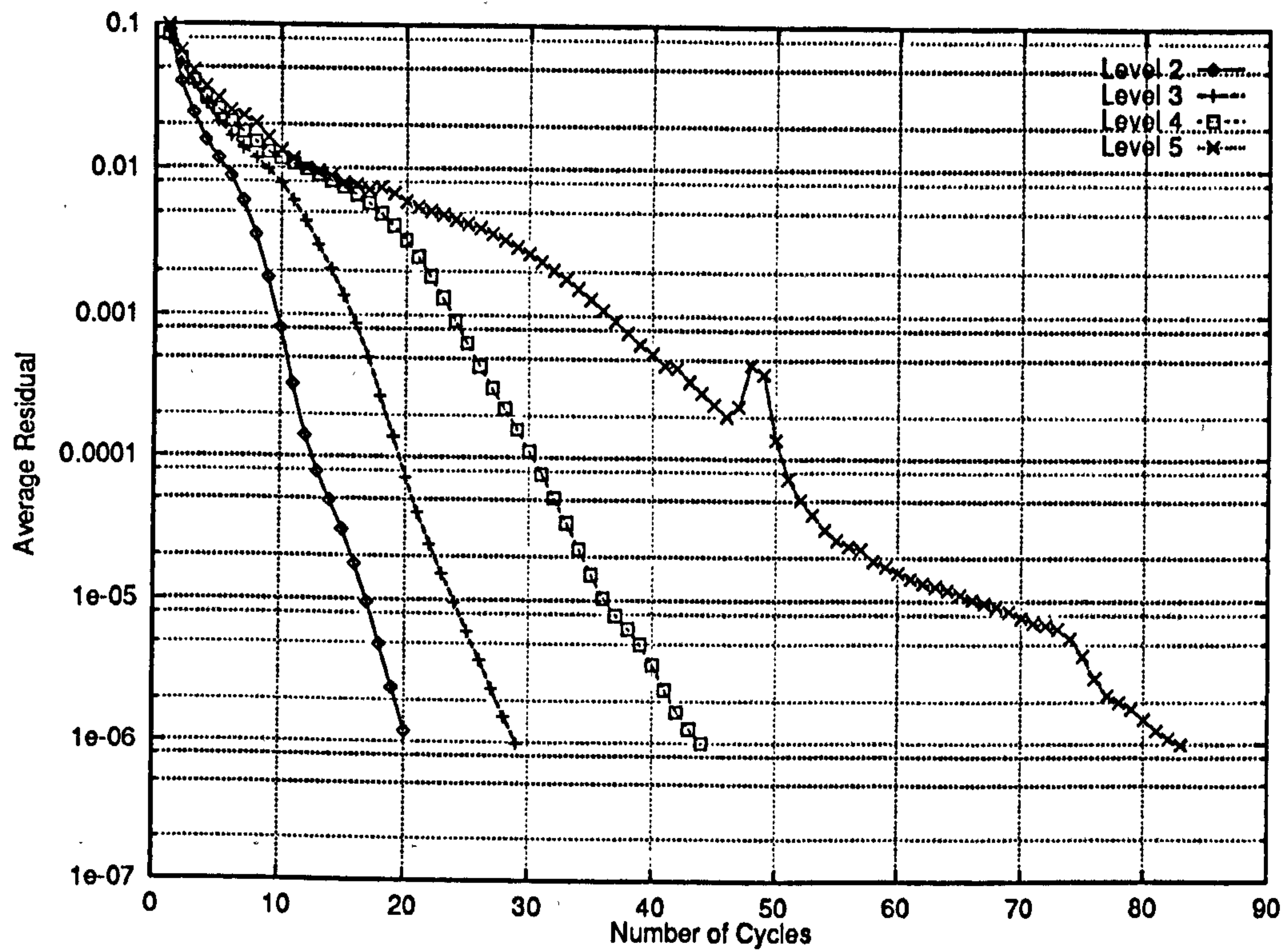


Figure 5.155: Two-phase T-junction problem – Convergence factors for multi-grid computations at different levels. See text for details concerning the irregularities in the convergence history at level 5

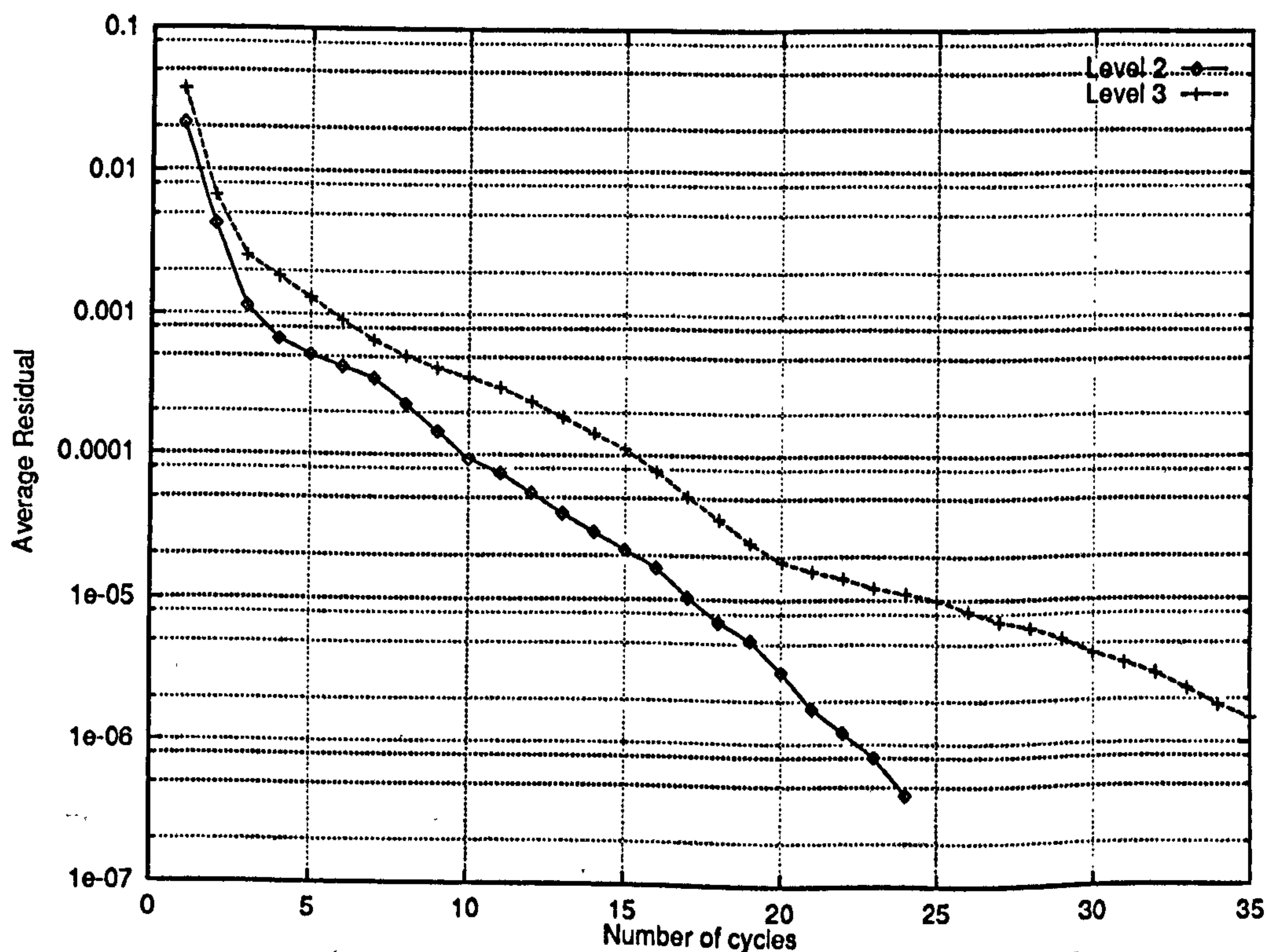


Figure 5.156: Multiphase backward-facing step problem ($Re_1 = 10$ and $Re_2 = 7.5$) – Convergence history on level 2 and 3 grids showing reasonable grid independence

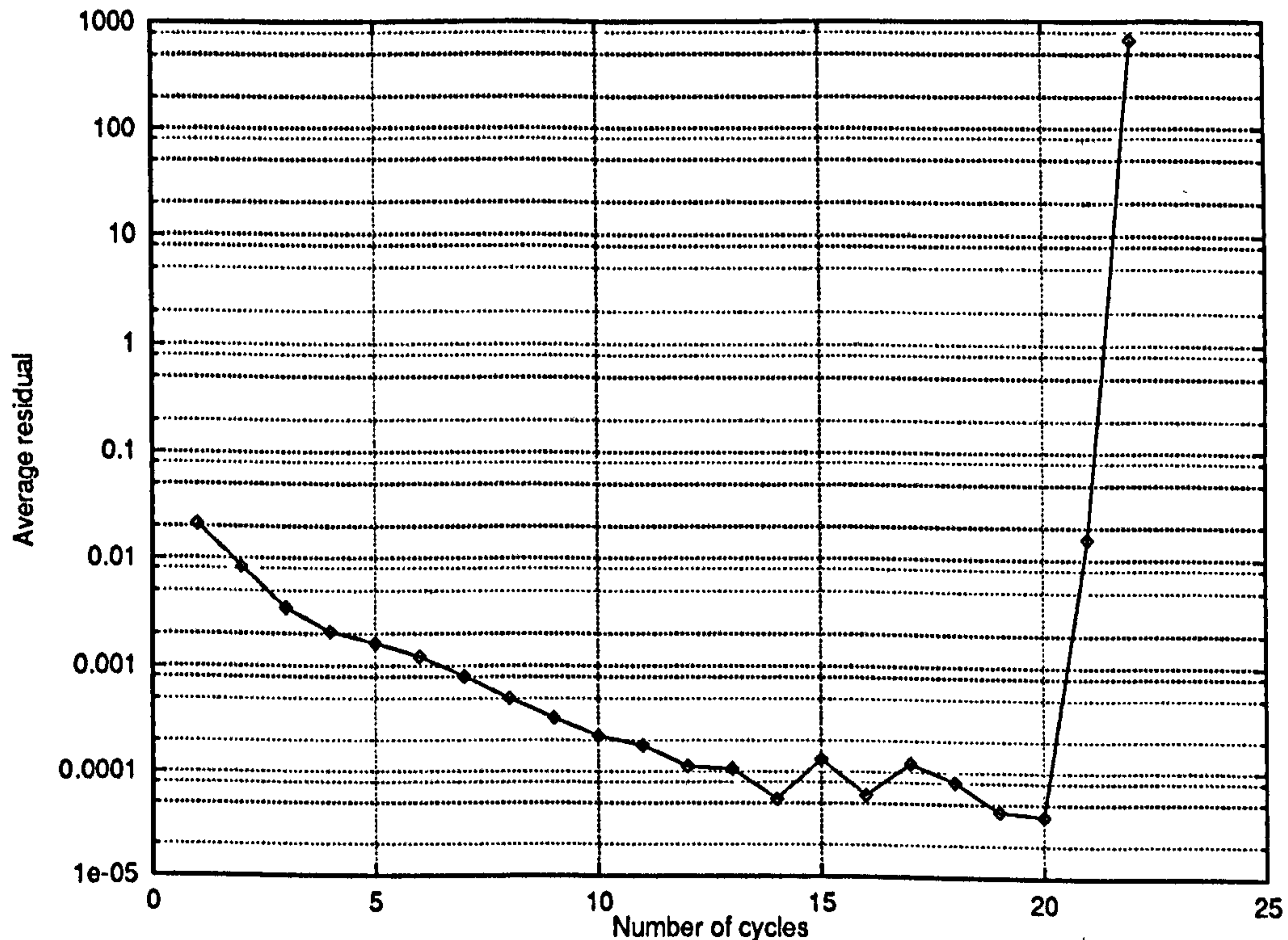


Figure 5.157: Multiphase backward-facing step problem – Convergence history on a level 3 grid for $Re_1 = 100$ and $Re_2 = 75$ showing the divergence of the pang-multiphase solver because both phases are recirculating and the problem is therefore ill-posed mathematically

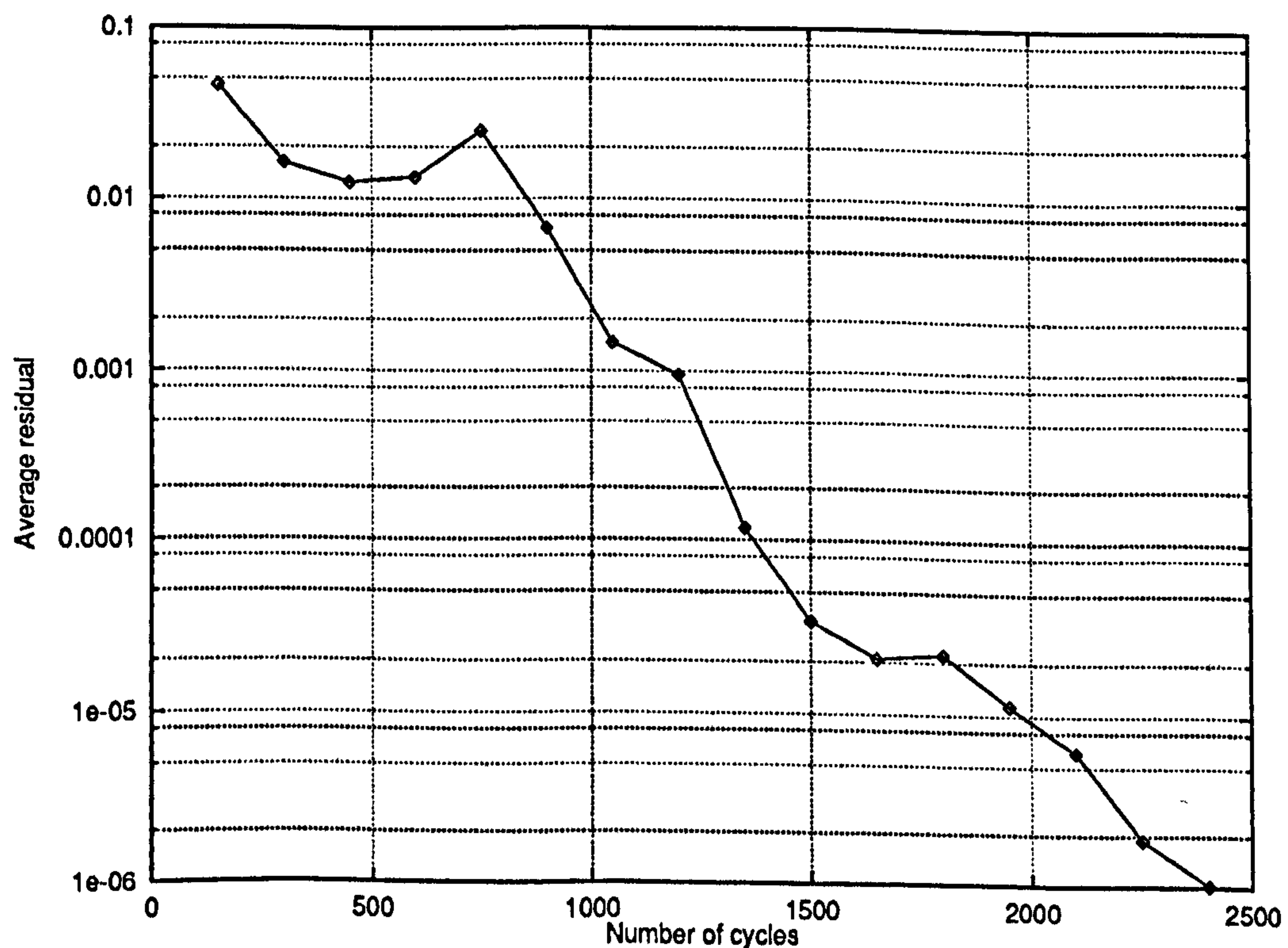


Figure 5.158: Multiphase backward-facing step problem – Convergence history for single grid computation with $\Delta x = \Delta y = 0.625$, equivalent to a level 3 multigrid computation ($Re_1 = 100$ and $Re_2 = 75$) – Compare with Figure 5.157: the multigrid solver is approximately 50 times faster than the single grid solver

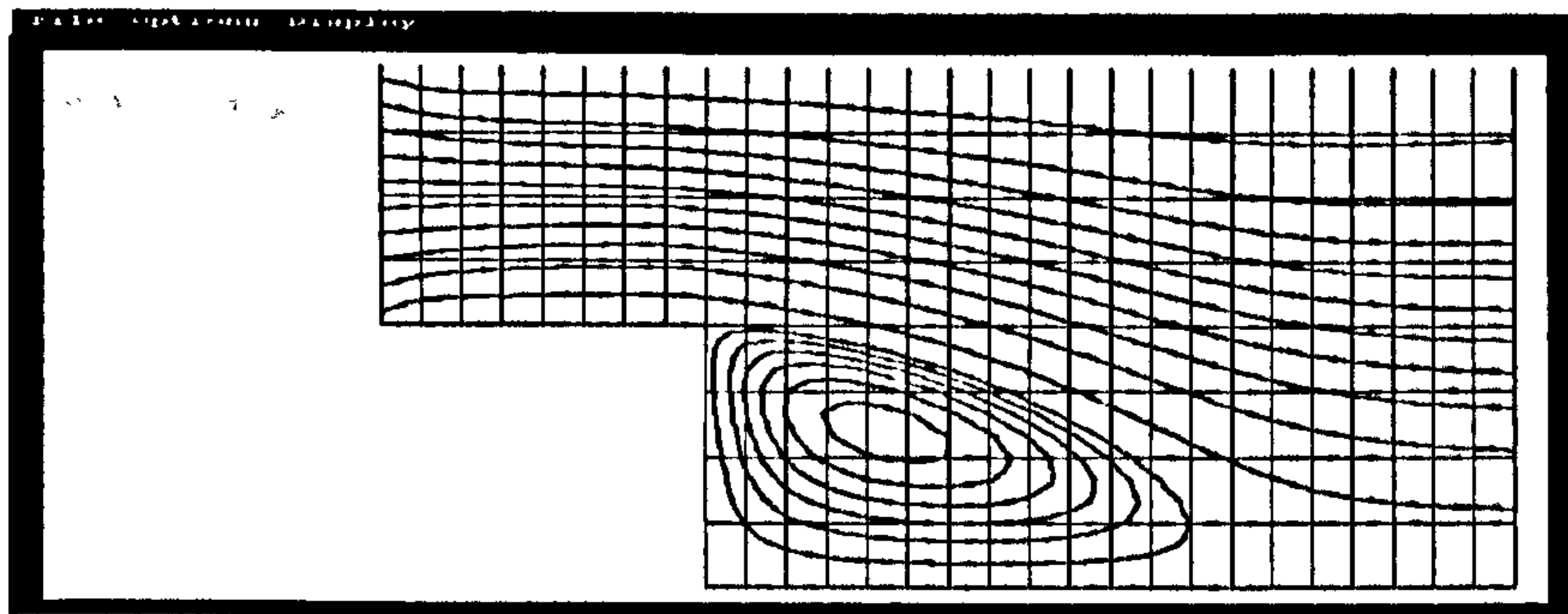


Figure 5.159: Multiphase backward-facing step problem at $Re_1 = 100$ and $Re_2 = 75$ – Streamlines for phase 1 – Compare with Figure 5.85

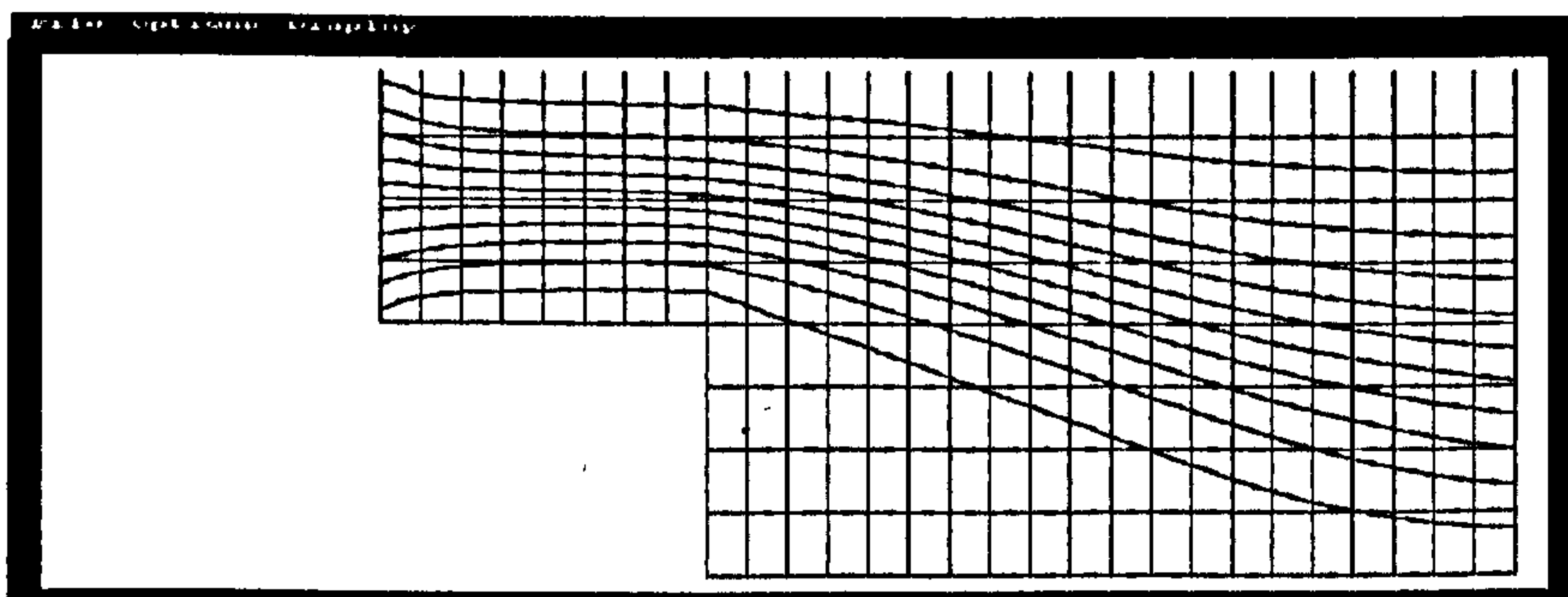


Figure 5.160: Multiphase backward-facing step problem at $Re_1 = 100$ and $Re_2 = 75$ – Streamlines for phase 2 – Compare with Figure 5.86

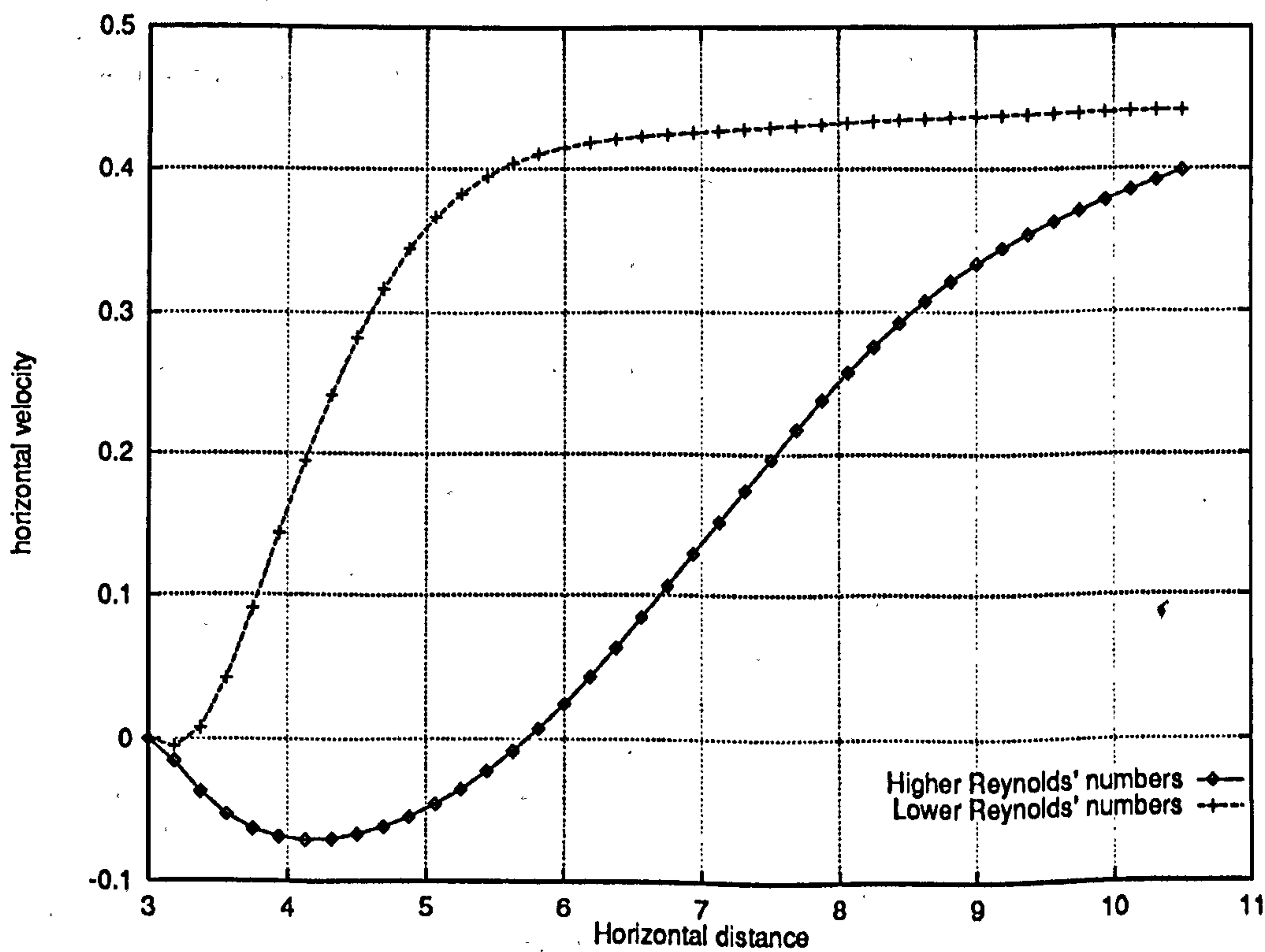


Figure 5.161: Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 1) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established recirculation region at high Reynolds numbers

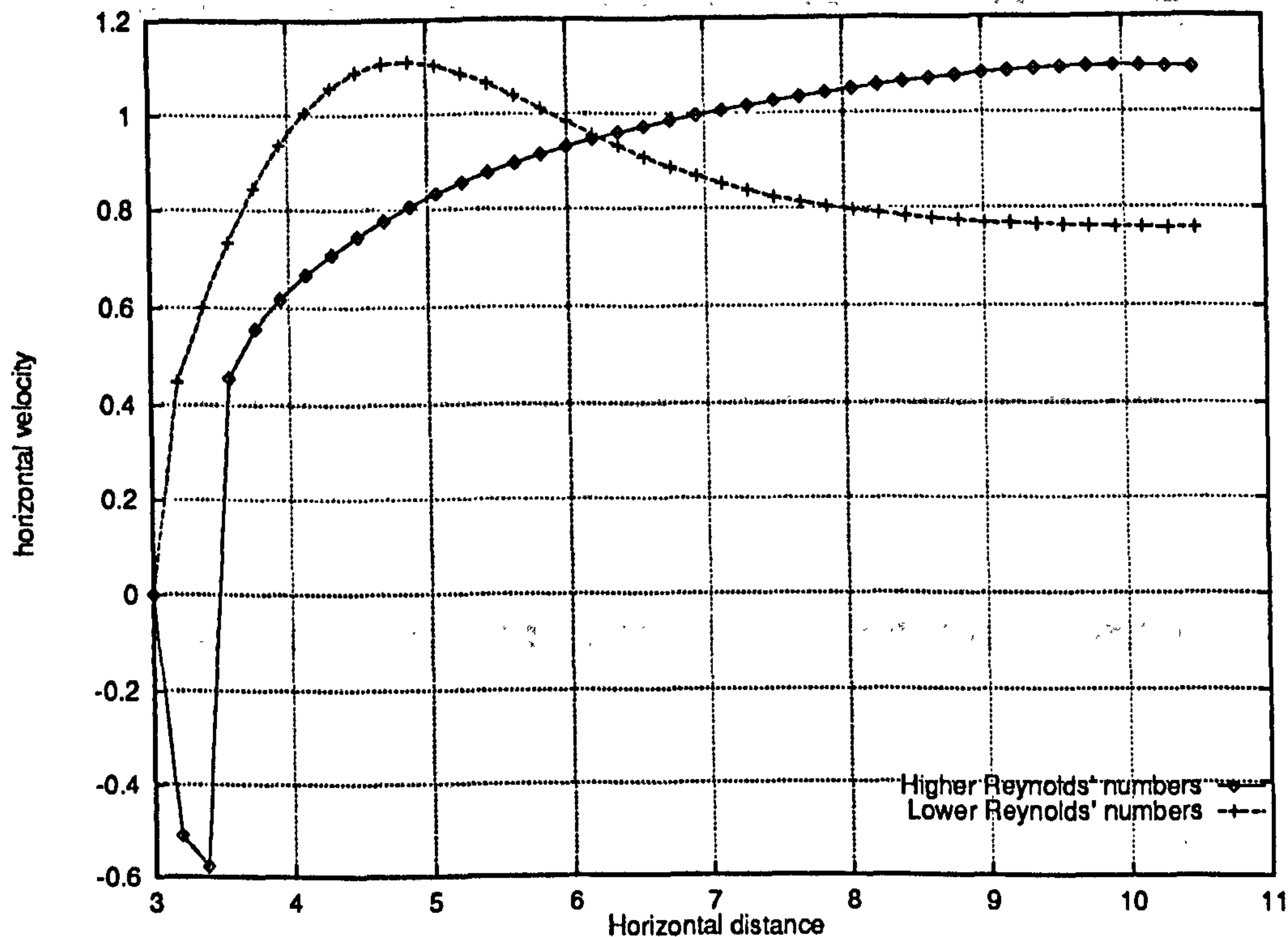


Figure 5.162: Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 2) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing that a recirculation zone only occurs at higher Reynolds numbers

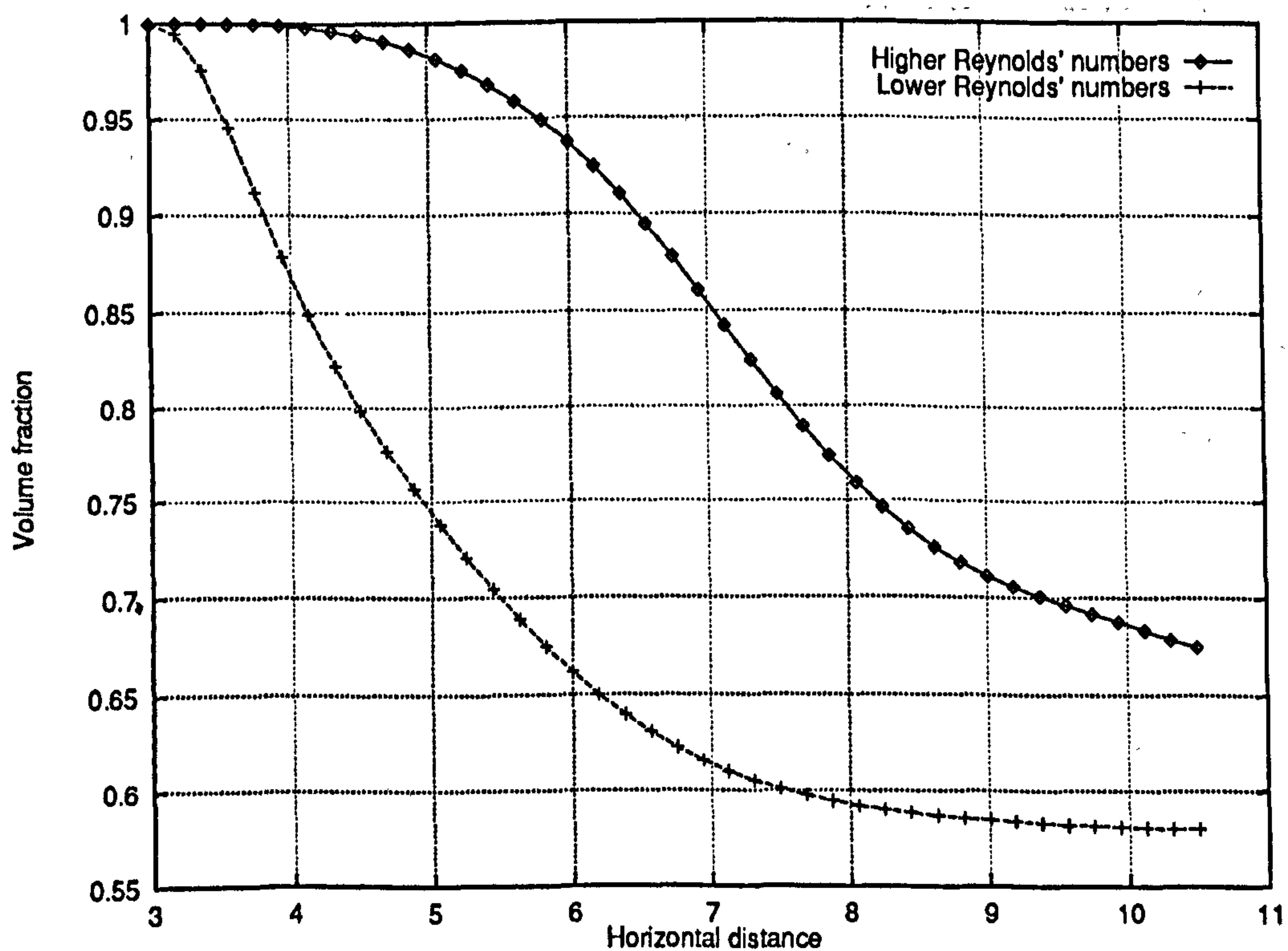


Figure 5.163: Multiphase backward-facing step problem – Volume fraction profiles (for phase 1) along the line $y = -0.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established single phase regions behind the step at high Reynolds number

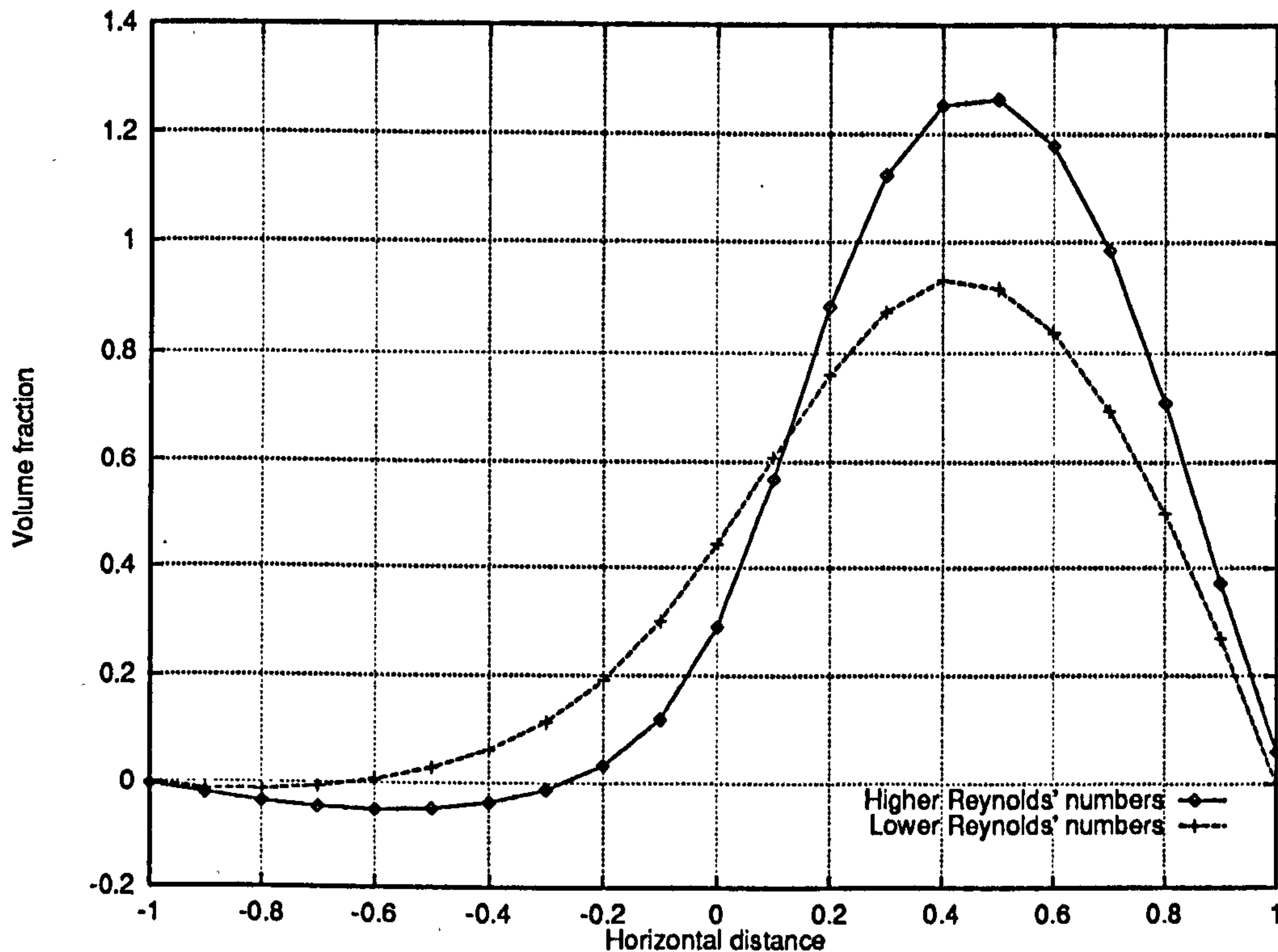


Figure 5.164: Multiphase backward-facing step problem – Horizontal velocity profiles (for phase 1) along the line $x = 3.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing an established recirculation region at high Reynolds numbers

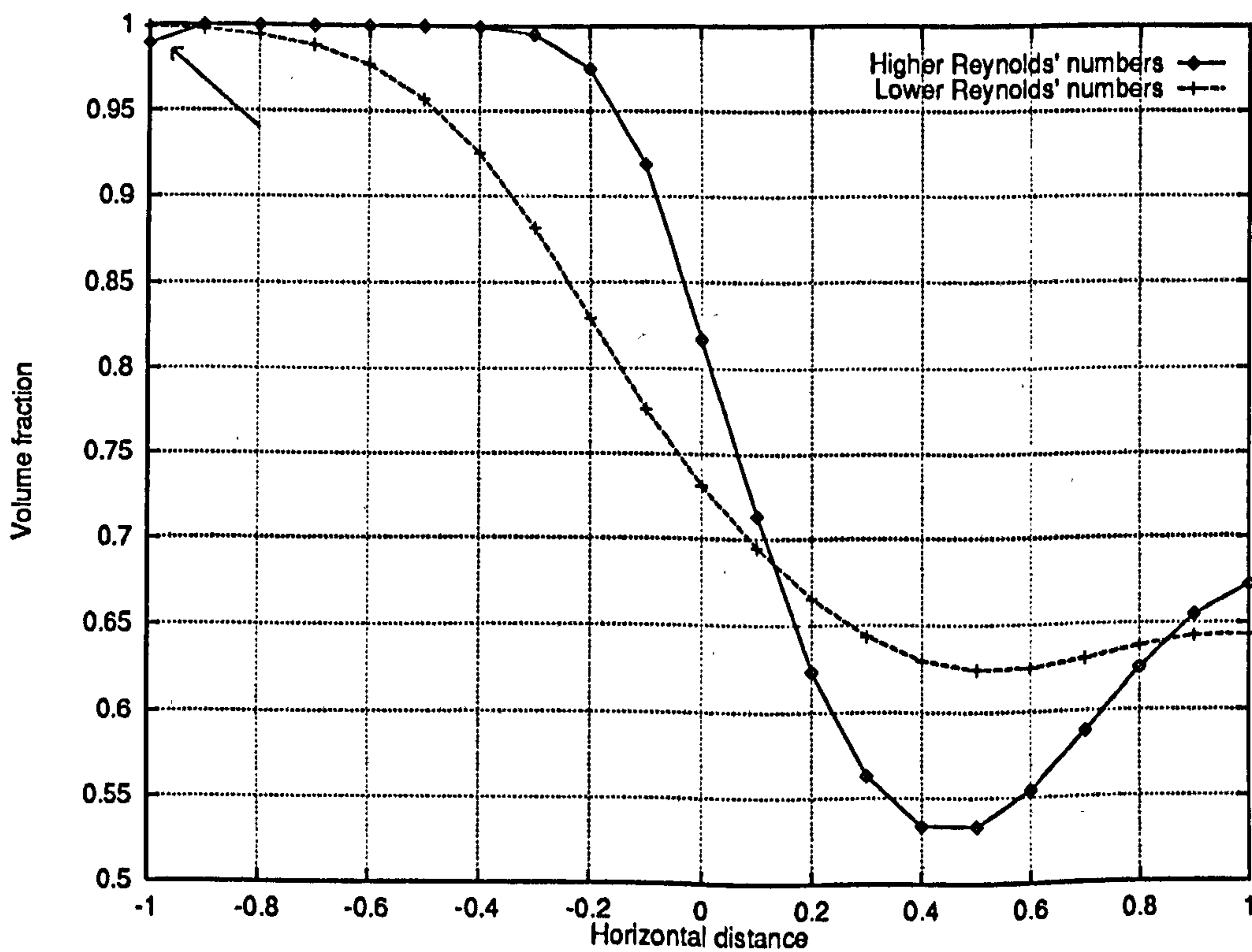


Figure 5.165: Multiphase backward-facing step problem – Volume fraction profiles (for phase 1) along the line $x = 3.5$ for high ($Re_1 = 100$ and $Re_2 = 75$) and low ($Re_1 = 10$ and $Re_2 = 7.5$) Reynolds numbers, showing in the top-left corner, a recirculating region where both phases are present

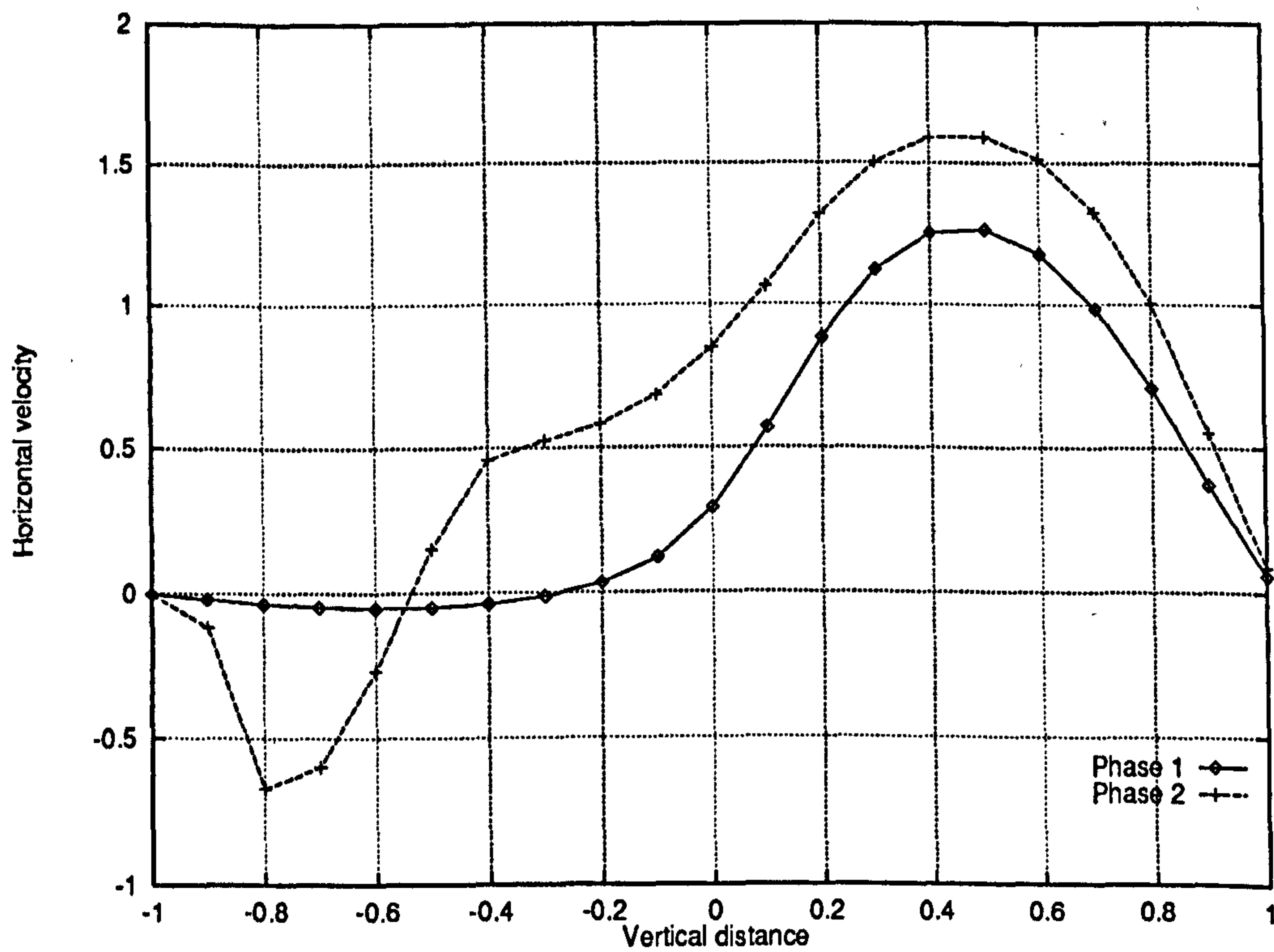


Figure 5.166: Multiphase backward-facing step problem – Horizontal velocity profiles (for both phases) along the line $x = 3.5$ for high Reynolds numbers ($Re_1 = 100$ and $Re_2 = 75$), showing the presence of two recirculation zones up to $y = -0.5$

5.4.11 Some Experiments on Reynolds Numbers Effects

For single phase flows, as the Reynolds number gets smaller, the Navier-Stokes equations become increasingly near linear as the diffusive terms gradually dominate over the convective terms. Consequently, multigrid algorithms converge faster. This indicates that multigrid methods are very good solvers for linear elliptic problems.

For multiphase flows, by contrast, an increasing body of evidence suggests that as the Reynolds numbers become smaller, the pang-multiphase multigrid solver converges more slowly. In this section, we first present evidence to that effect and then discuss the cause. During the course of this project, a large number of experiments have been conducted, relating to the simulation of multiphase flow at low Reynolds numbers ($Re \sim 1$). These are not reported in full as it has proved very difficult to organise the observations and draw solid conclusions.

Consider first the two-phase channel flow: numerical simulations have been carried out for a range of Reynolds numbers on the same computational domain as before (i.e. $l = 3d$). The values chosen were $Re = 1$, $Re = 10$, $Re = 100$ and $Re = 500$ and the results are shown in Figures 5.167 to 5.172. Cases $Re = 1$ and $Re = 10$ suggest that there is a state where the two phases are in equilibrium and the flow can be considered to be fully developed. However, it immediately appears that as the pressure gradient tends toward zero, the length of pipe necessary to reach this equilibrium increases: for $Re = 100$ and $Re = 500$, the flow is clearly not fully developed at the outlet. For these last two cases, large errors may therefore be introduced at the outlet by the application of Neumann boundary conditions. This has implications for the convergence factors and the question is examined in Section 5.4.12.

Figure 5.173 shows the convergence history for multigrid computations on a level 5 grid. It clearly appears that as the diffusive effects dominate the solution, convergence factors are greatly degraded. We make two main observations:

- Firstly, a comparison of the solutions for $Re = 10$ and $Re = 100$ indicates that the plateau phase is connected with diffusive effects — the more dominant the diffusive forces, the longer the plateau.
- Secondly, obtaining convergence for the case $Re = 1$ has proved very difficult. In particular, second order interpolation (Section 5.4.8) for the fine grid volume fraction corrections, which was used for the results reported in Figure 5.173, has proved unstable. The solution profiles shown here for $Re = 1$ were obtained with an approximate prolongation procedure at the boundaries¹⁰ (see Figure 5.174 for the associated convergence history). The inaccuracy in the transfer operators does not have an effect on the solution for the finest grid since only uniform grids are used.

¹⁰This procedure consists of prolongating the volume fraction corrections using equations 4.25 to 4.28 everywhere, even near walls where they should be modified.

The cause of the low Reynolds number instability has not yet been fully established. However, evidence accumulated during the course of this study consistently indicates that the solver performs less well when diffusion effects are important. Four hypotheses can be formulated to explain this effect:

1. The physics of the system is more complicated than in the single phase case. At small Reynolds numbers, the multiphase equations do not become more linear. Actually, as the pressure gradient increases, so does this aspect of inter-phase coupling. Also we have:

$$\nabla \cdot (r\mathbf{v} \otimes \mathbf{v} - r\mathbf{T}) = -\frac{r}{\rho} \nabla p.$$

At $Re \sim 1$, convective effects are of the same order as diffusive effects. Furthermore, the problem is non-linear in the volume fractions and coupled.

2. As the flow becomes more fully separated, the local Jacobian may become more ill-conditioned when the volume fractions approach their limiting values of 0 and 1. This would particularly appear on fine grids (see Section 5.4.13).
3. The diffusive terms in the quasi-Newton solvers are causing difficulties. This is a cell-Reynolds number effect (as opposed to a Reynolds number effect). Figure 5.175, which shows the convergence histories for the case $Re = 1$ on uniform grids for different grid spacing, provides good evidence that the lack of stability is a fine grid effect. Furthermore, it was verified that the single grid solver suffers from the same limitations (see Figure 5.176). In this last case, the appearance of oscillations for both $Re = 1$ and $Re = 100$ contrasts sharply with the very smooth convergence patterns observed on coarser grids (see Figure 5.119) and confirms our hypothesis that the relevant parameter is the cell-Reynolds number.
4. The lack of convergence at low Reynolds numbers and the plateau phase at higher Reynolds numbers may be due to the same cause. In Section 5.4.13, we present evidence that the plateau phase is caused by cross derivative terms appearing in the diffusion flux. However, it was observed that at low Reynolds numbers the computations were not made more stable if the partial stress tensor flux was used.

Next, we consider a two-phase flow through the same T-junction as in Section 5.3.2 but at lower Reynolds numbers. It has been established above that for higher Reynolds numbers, the multigrid algorithm performs very well for the T-junction problem. If our basic hypothesis that the degradation of performance for the channel flow is governed by diffusive (multiphase) effects is correct, then similar symptoms should appear for the T-junctions at lower Reynolds numbers since diffusive effects will be more important.

The physical properties of the fluids are as follows ¹¹:

¹¹As before, the Reynolds number is based on the width of the channel.

Reynolds Number	viscosity	density
$Re_1 = 10.0$	$\mu_1 = 0.1$	$\rho_1 = 1.0$
$Re_2 = 7.5$	$\mu_2 = 0.066$	$\rho_2 = 0.5$

As expected, the diffusion is increased and in particular, the stratification process of the flow is less marked at the outlet, as Figure 5.177 shows. Figure 5.178 shows the convergence history for grids at levels 3 and 4. At lower Reynolds numbers and on sufficiently fine grids, it can clearly be seen that a grid dependent plateau phase occurs (see also Table 5.13). The convergence history, particularly on the level 4 grid, is now very similar to those observed for the multiphase channel flow.

Grid level	Average convergence factors		
	to 10^{-3}	between 10^{-3} and 10^{-6}	globally
3	0.9714	0.9518	0.9625
4	0.9715	0.9825	0.9788

Table 5.13: Multigrid convergence factors for the two phase flow through a T-junction at lower Reynolds numbers

In this section, we have established the fact that multiphase diffusive effects play an important role in determining the convergence pattern of the multigrid solver. Some of the issues raised here, namely the connection between the plateau phase and the diffusive tensor, will be examined and resolved in Sections 5.4.13 and 5.4.14. Other issues, in particular the lack of convergence for very low Reynolds number channel flow, are still not completely understood. However, we have proposed several hypotheses which could be investigated in future.

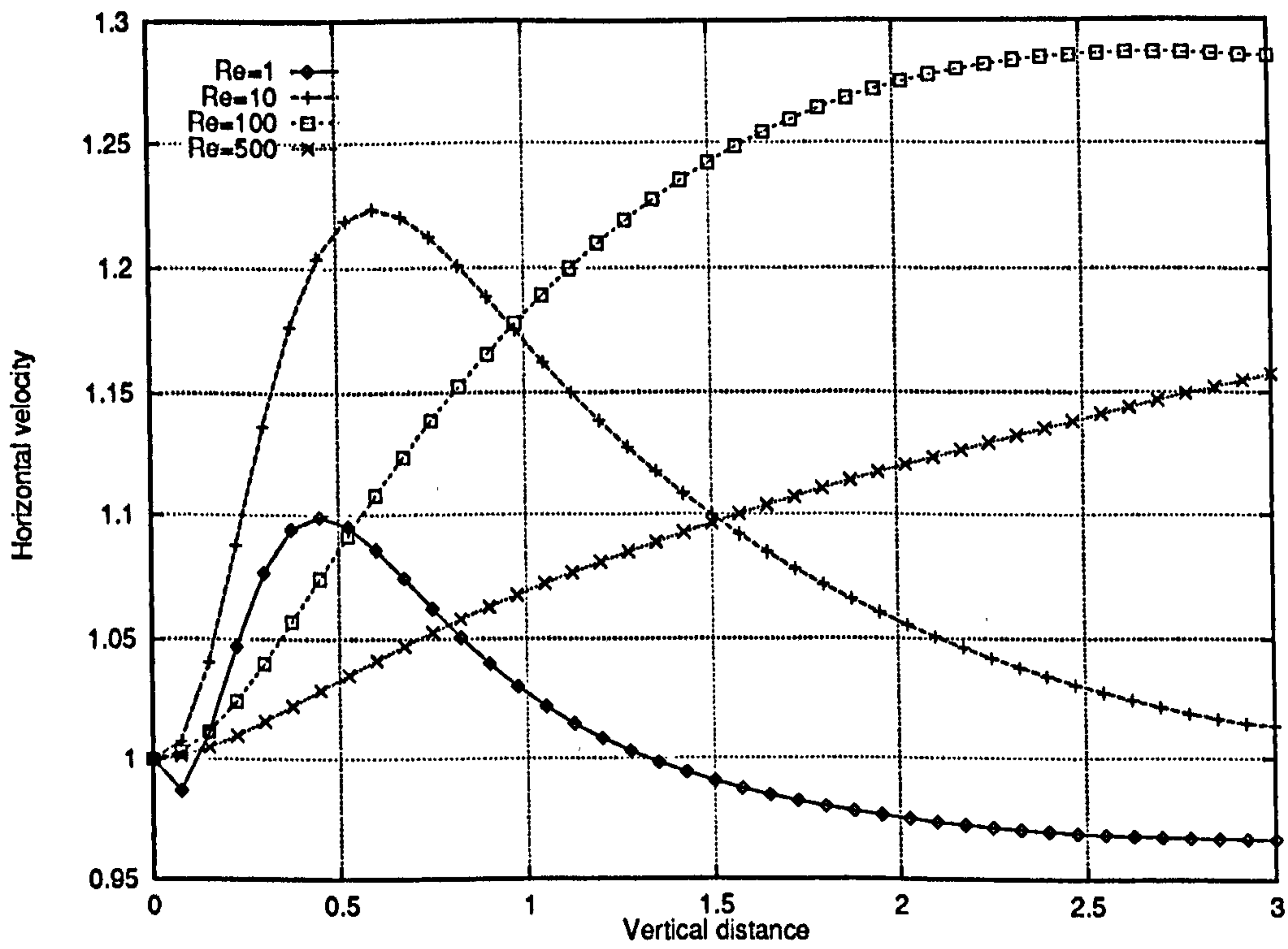


Figure 5.167: Two-phase channel flow problem – Horizontal velocity profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

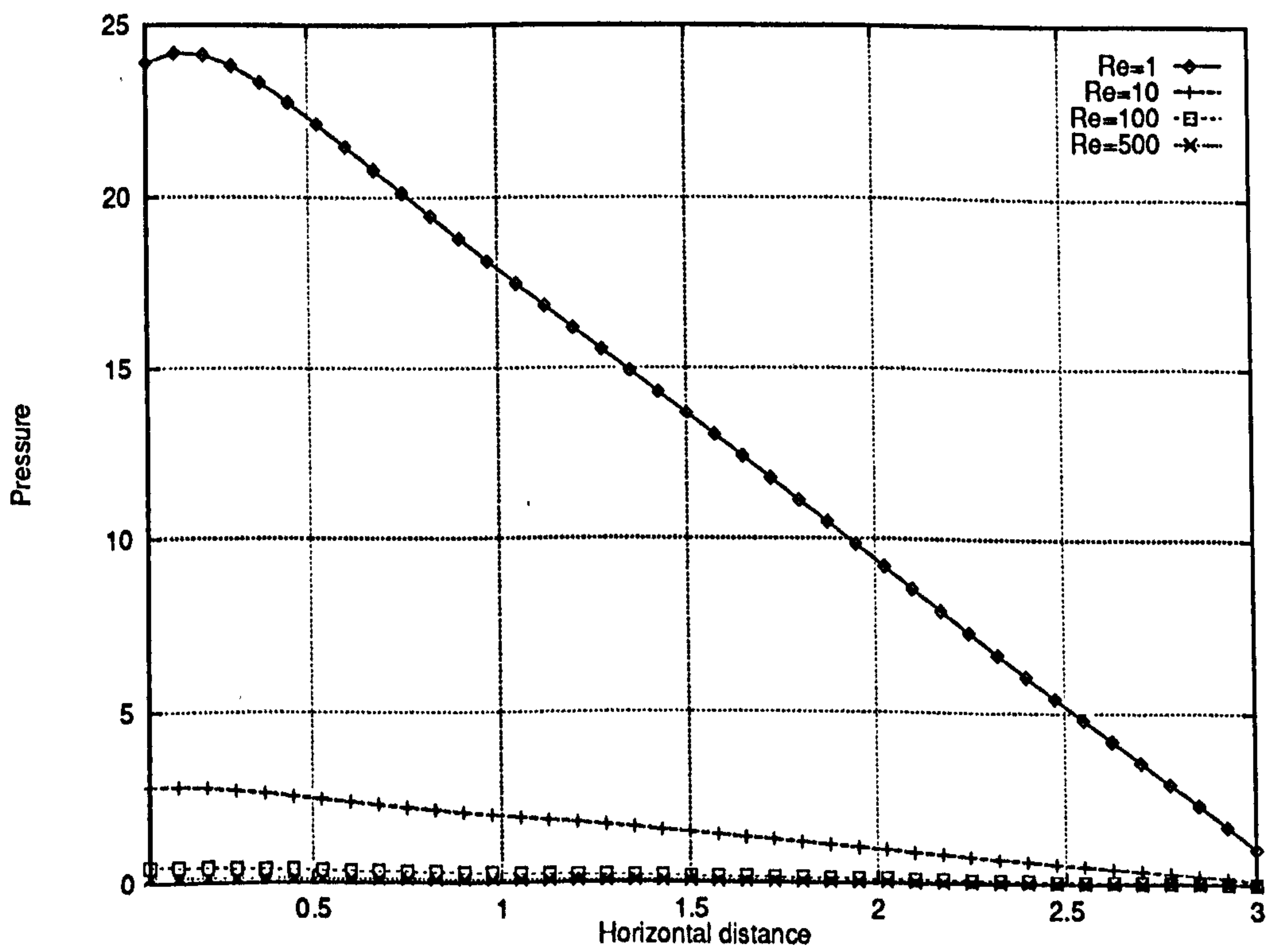


Figure 5.168: Two-phase channel flow problem – Pressure profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

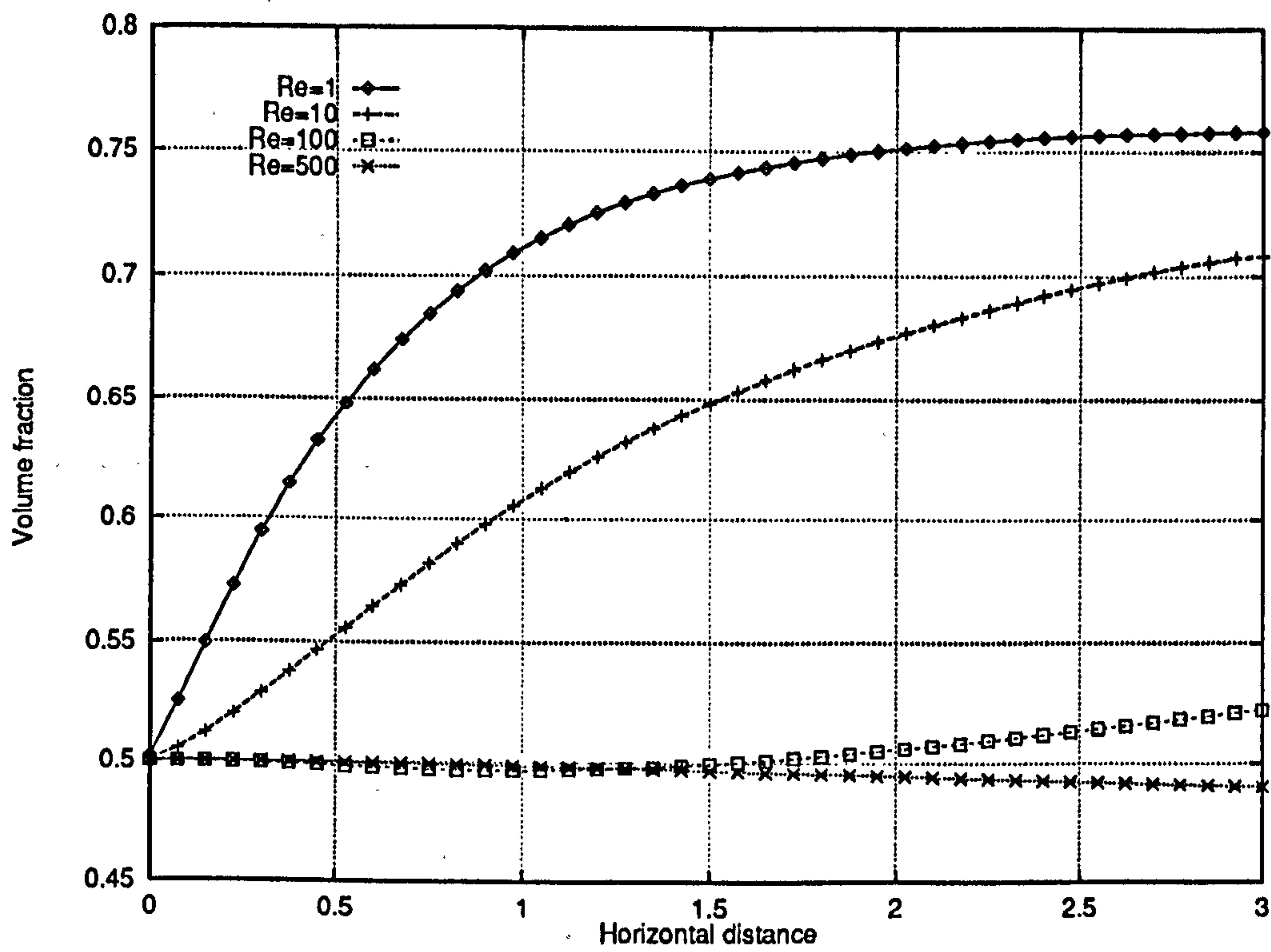


Figure 5.169: Two-phase channel flow problem – Volume fraction profiles along the line $y = 0.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

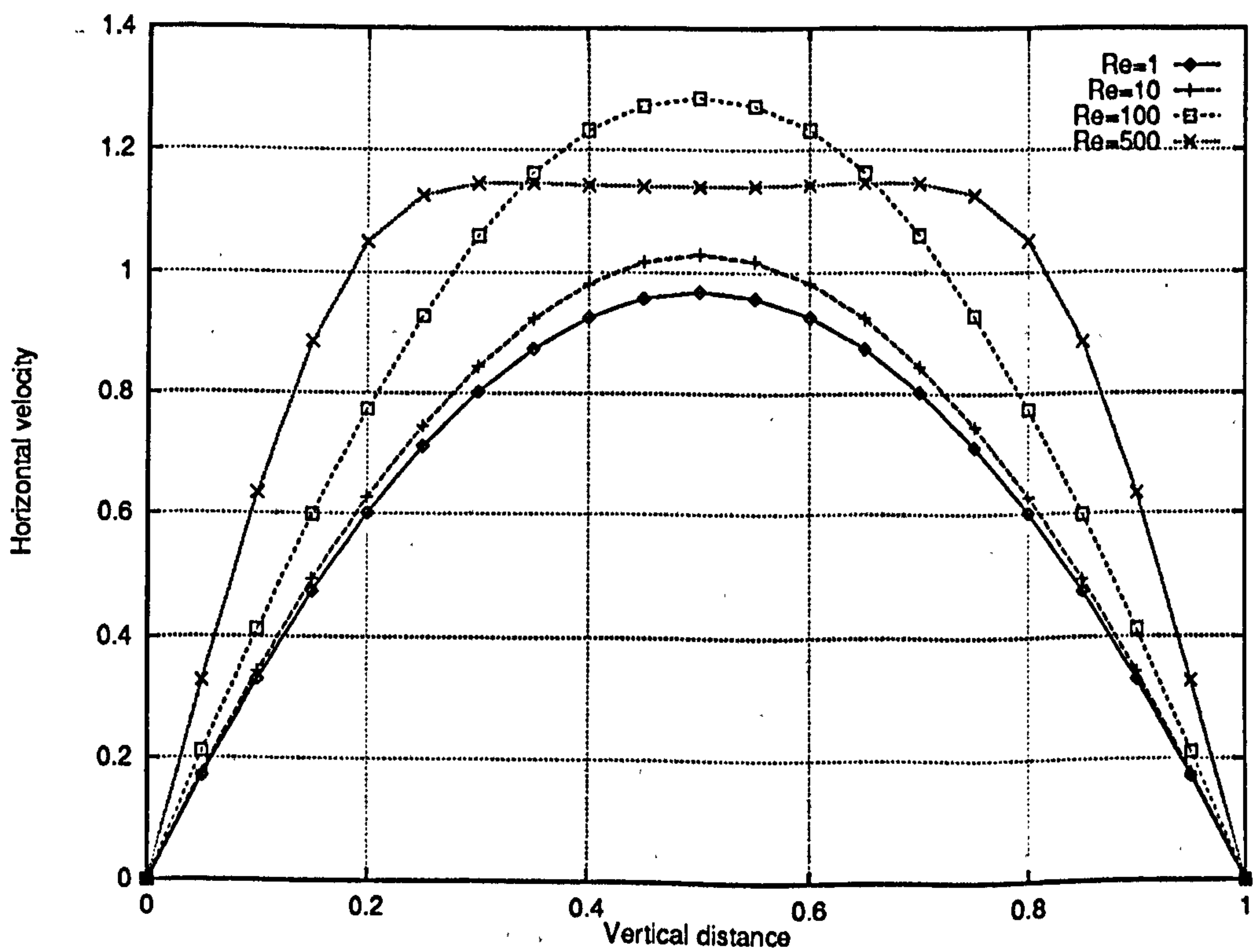


Figure 5.170: Two-phase channel flow problem – Horizontal velocity profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

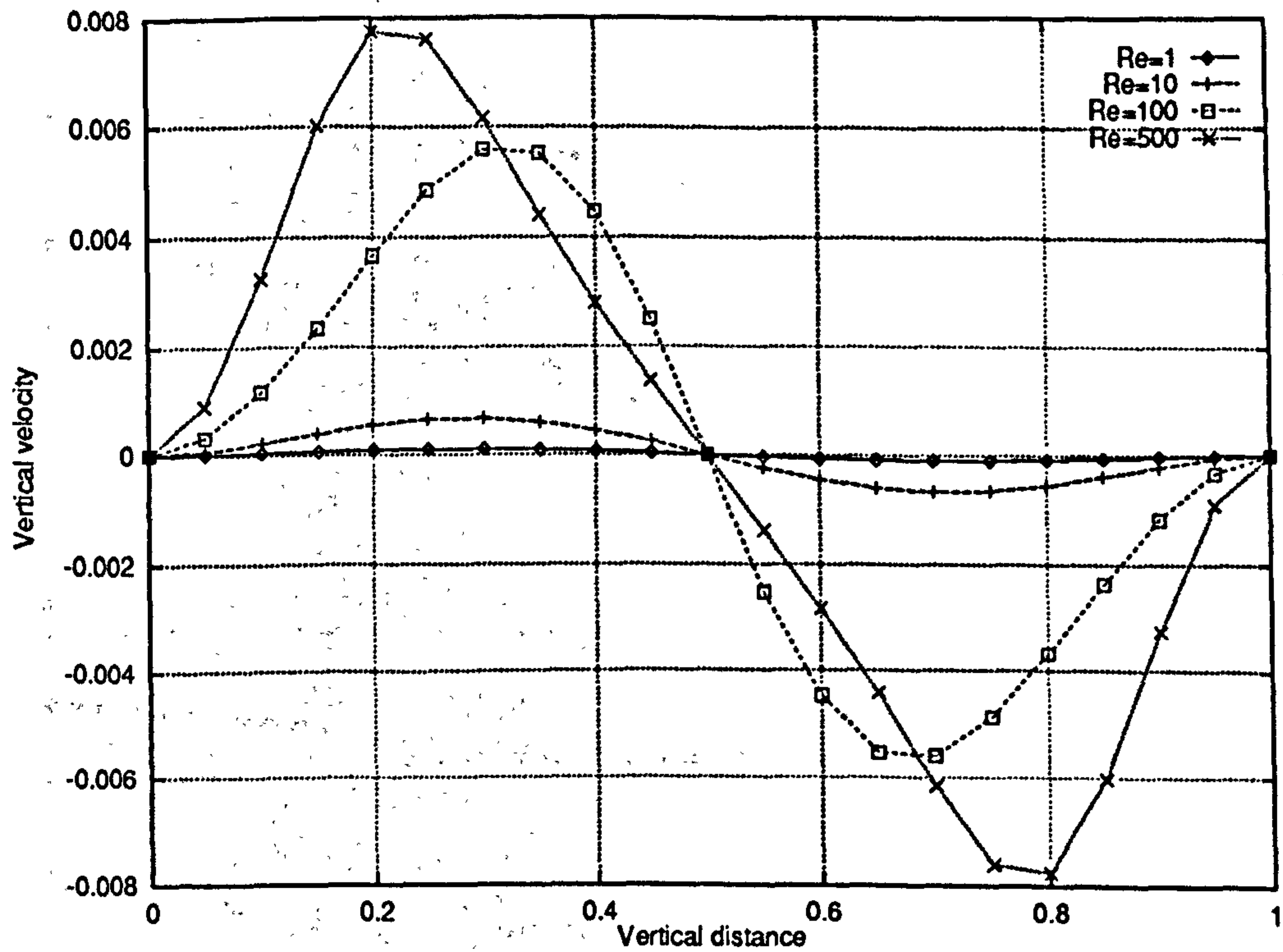


Figure 5.171: Two-phase channel flow problem - Vertical velocity profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

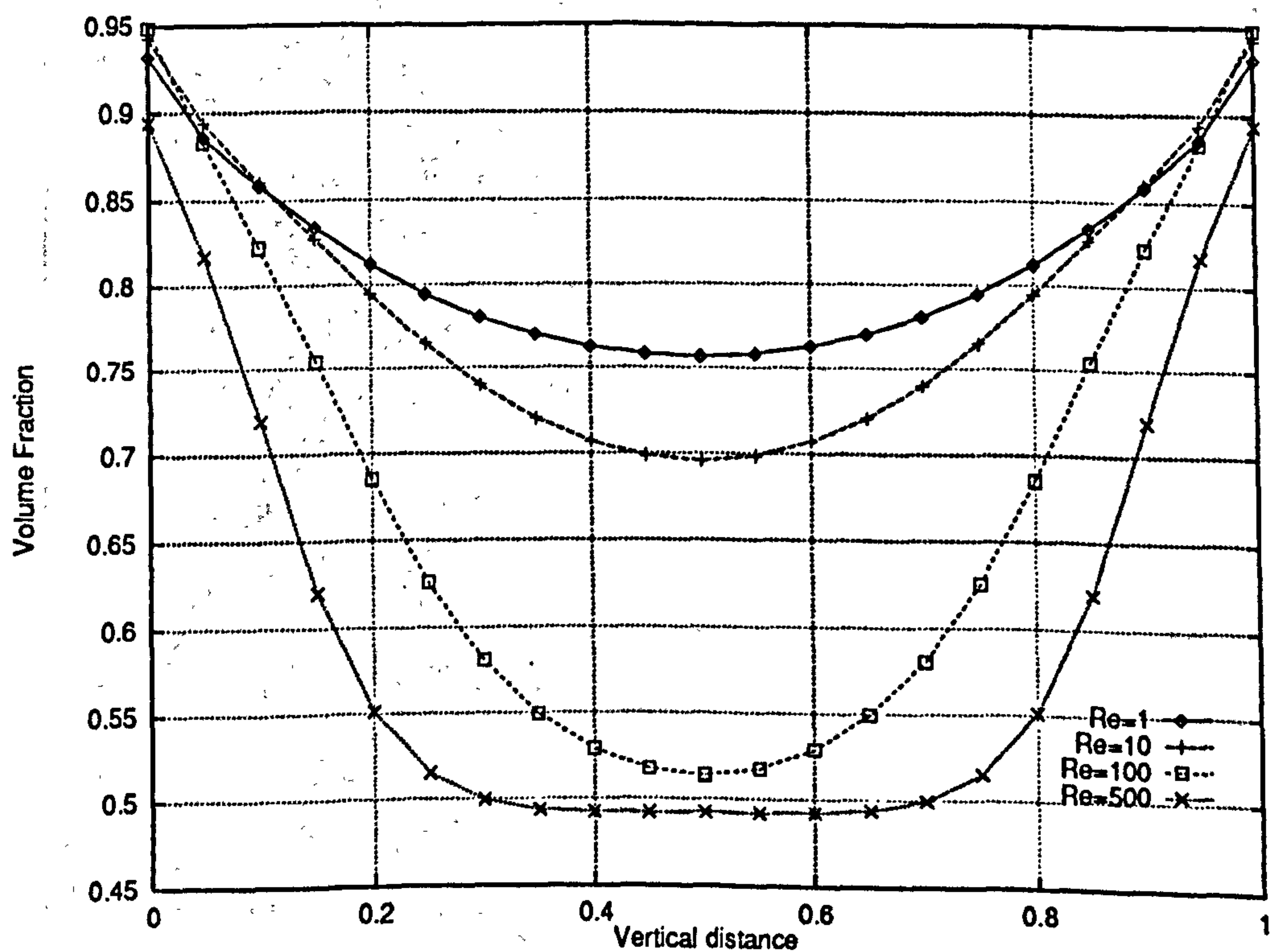


Figure 5.172: Two-phase channel flow problem - Volume fraction profiles along the line $x = 2.5$ on level 5 grids for different Reynolds Numbers (Phase 1)

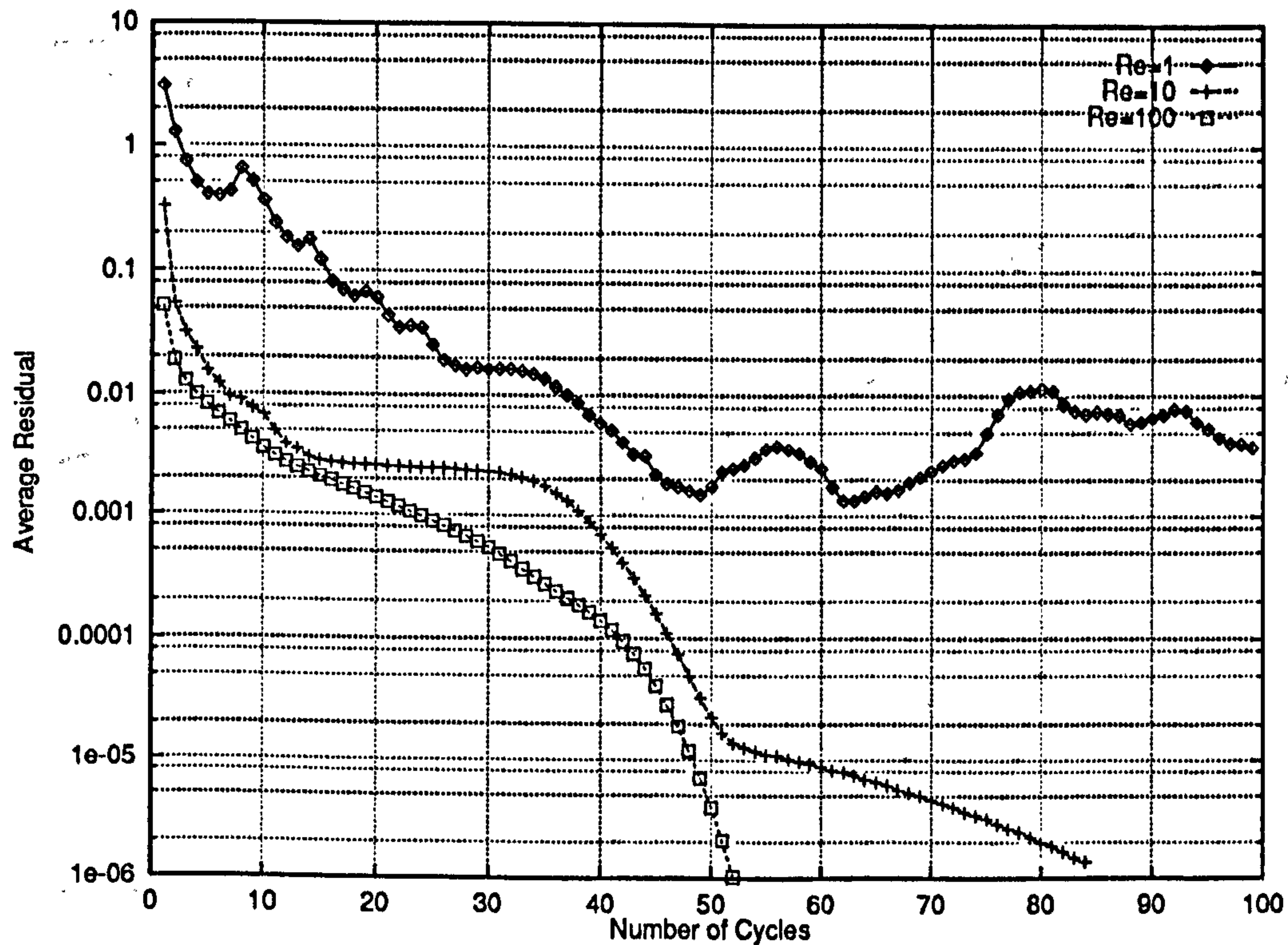


Figure 5.173: Two-phase channel flow problem – Convergence History on level 5 grids for different Reynolds Numbers and a first order interpolation for the fine grid volume fraction corrections

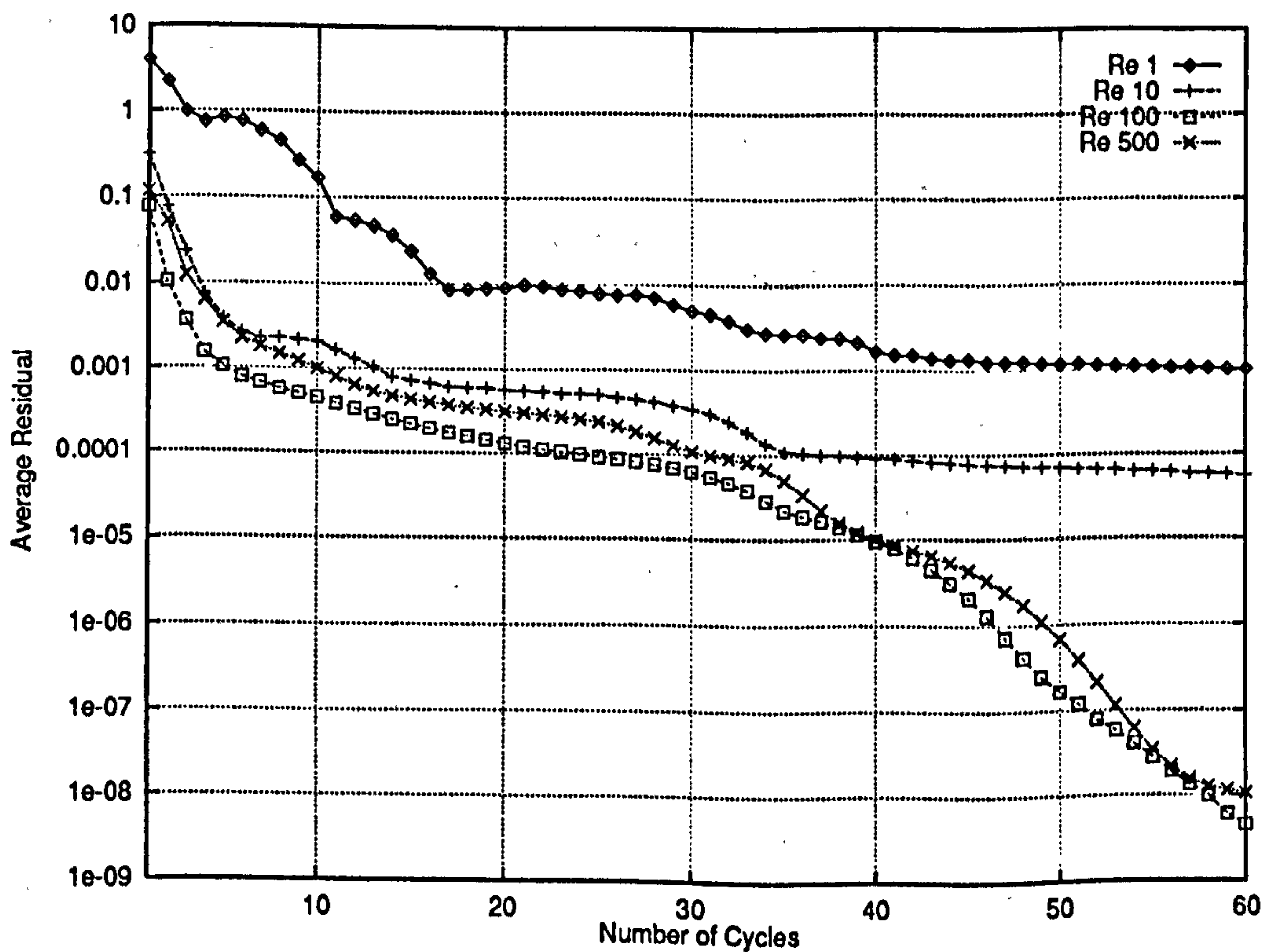


Figure 5.174: Two-phase channel flow problem – Convergence History on level 5 grids for different Reynolds Numbers and a second-order like interpolation for the fine grid volume fraction corrections

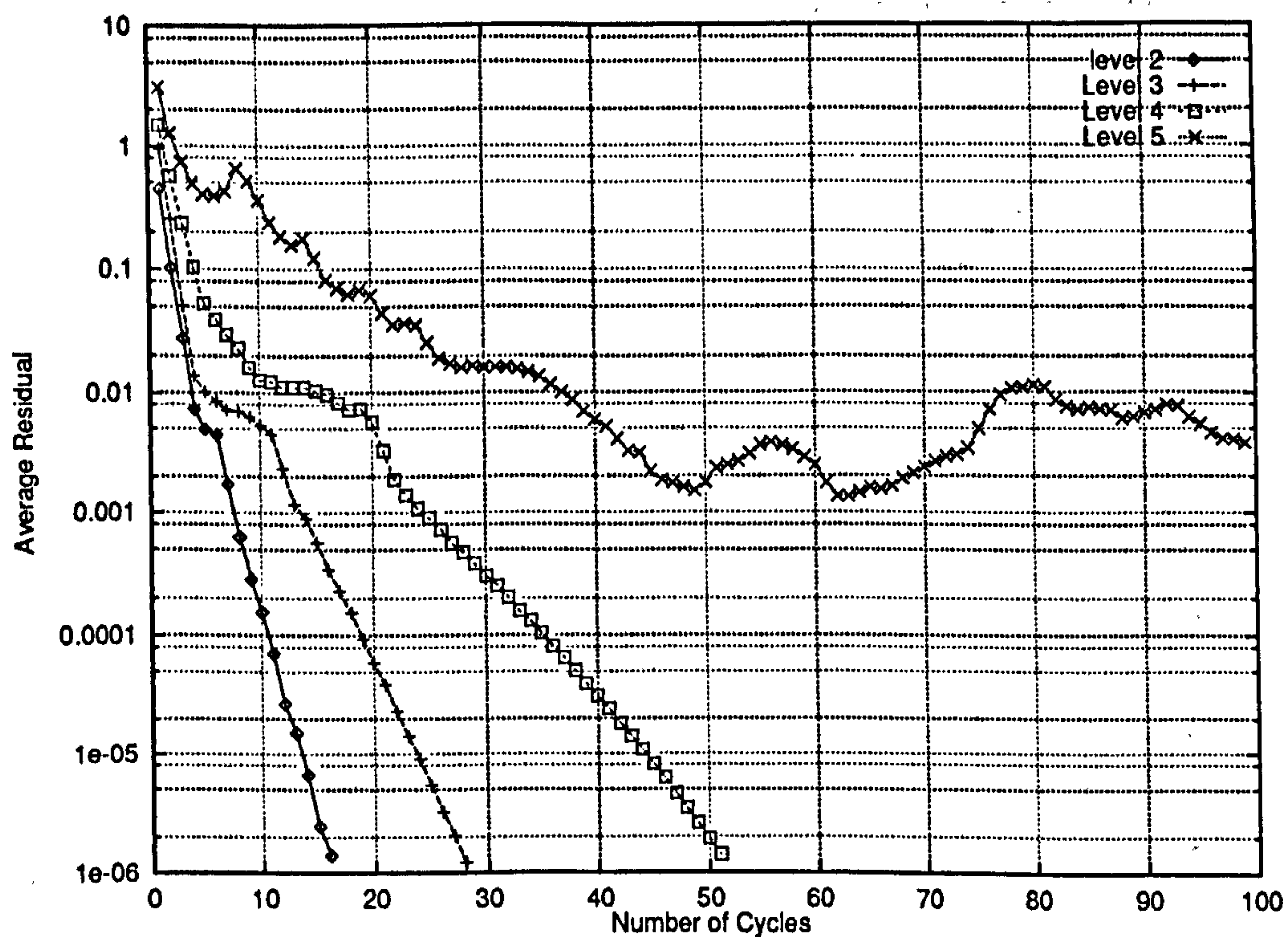


Figure 5.175: Two-phase channel flow problem – Convergence History for a flow at $Re = 1$, on different grid levels

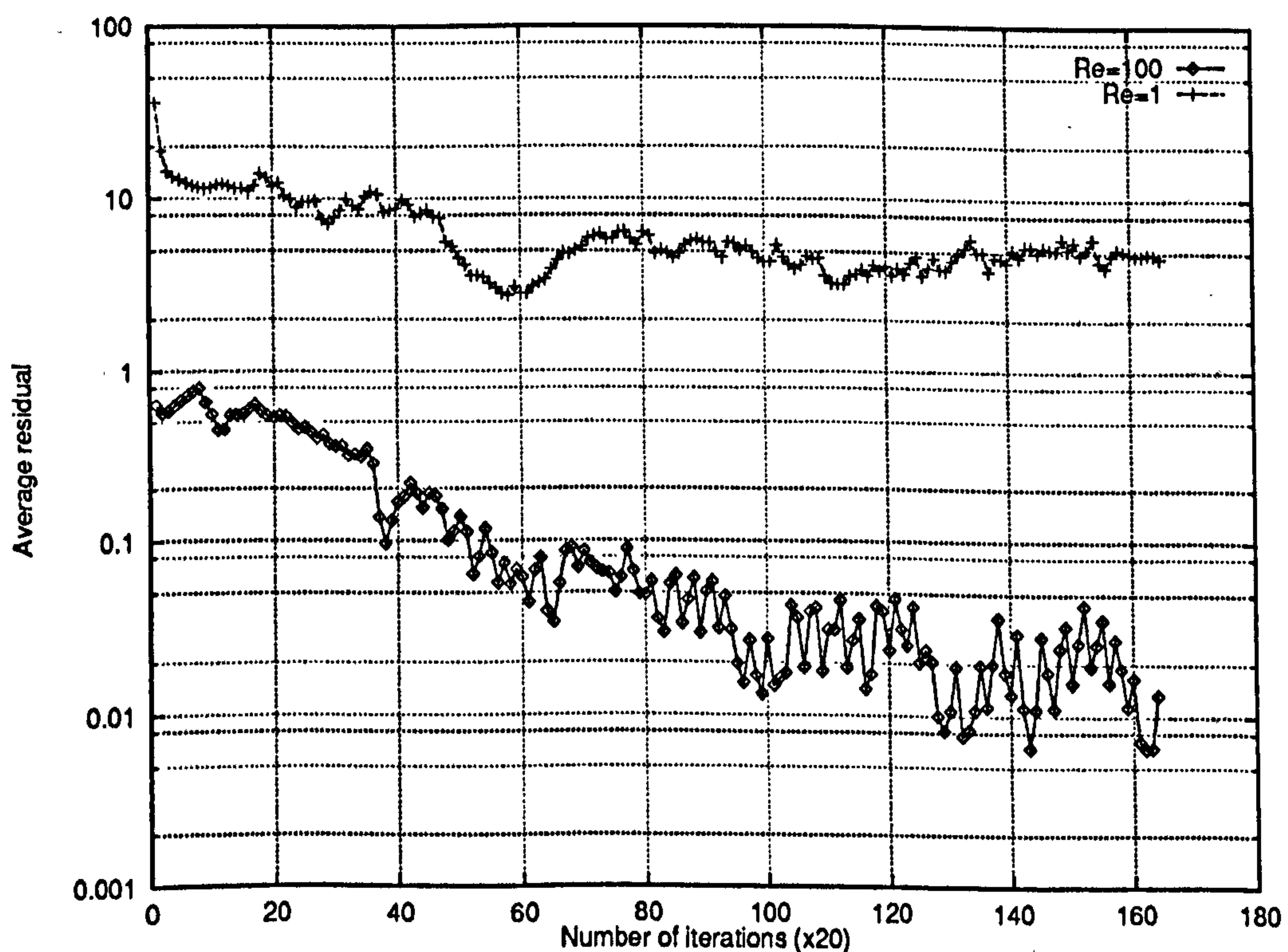


Figure 5.176: Two-phase channel flow problem – Convergence history for a single grid computation for flows at $Re = 1$ and $Re = 100$ – The grid size is $\Delta x = \Delta y = 0.015625$ and the computation is equivalent to a level 5 multigrid computation.

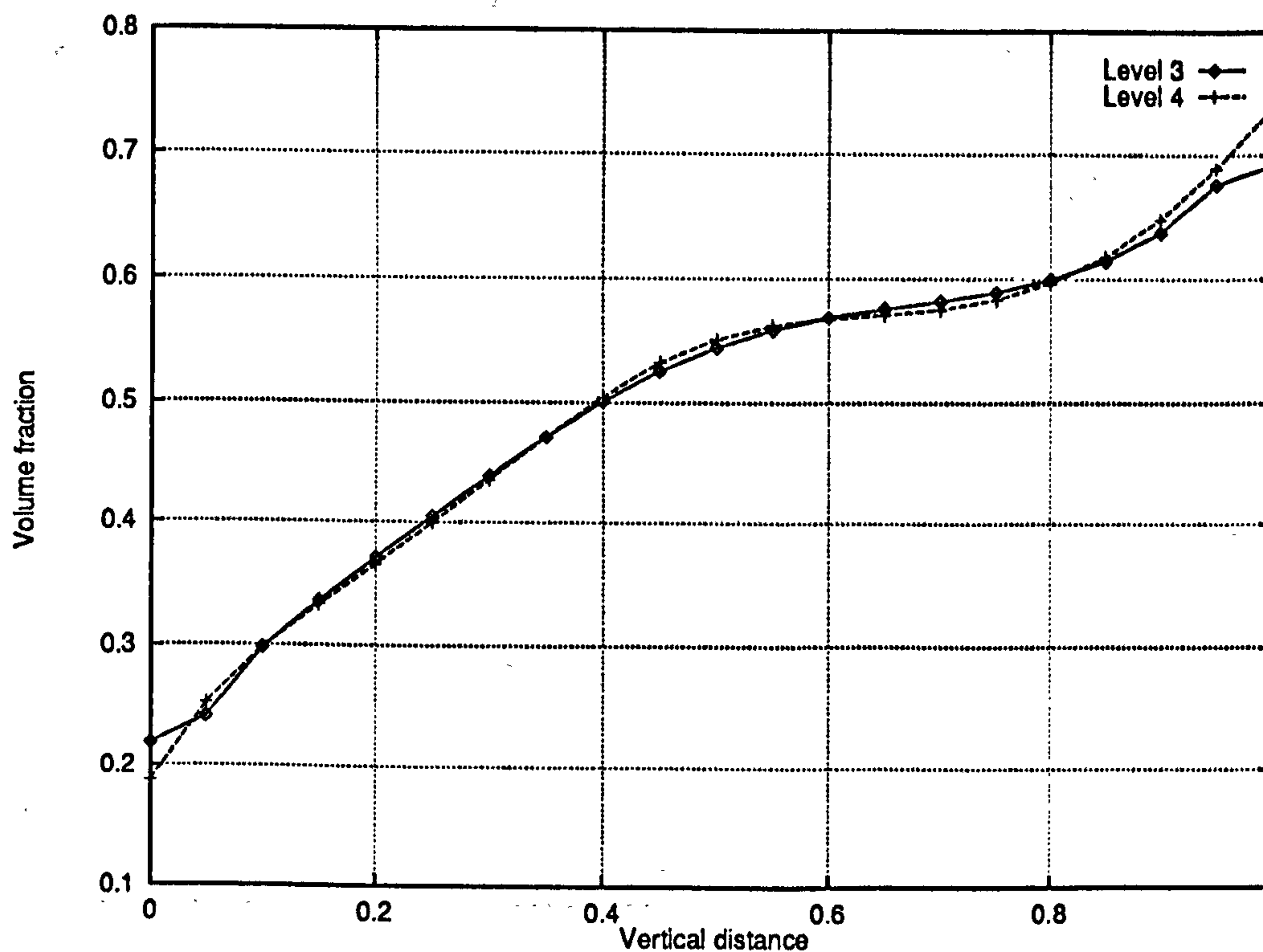


Figure 5.177: Low Reynolds number flow through a T-junction – Volume fraction profile along the line $x = 6.5$ for different grid level showing an increased amount of diffusion (compare with Figure 5.81)

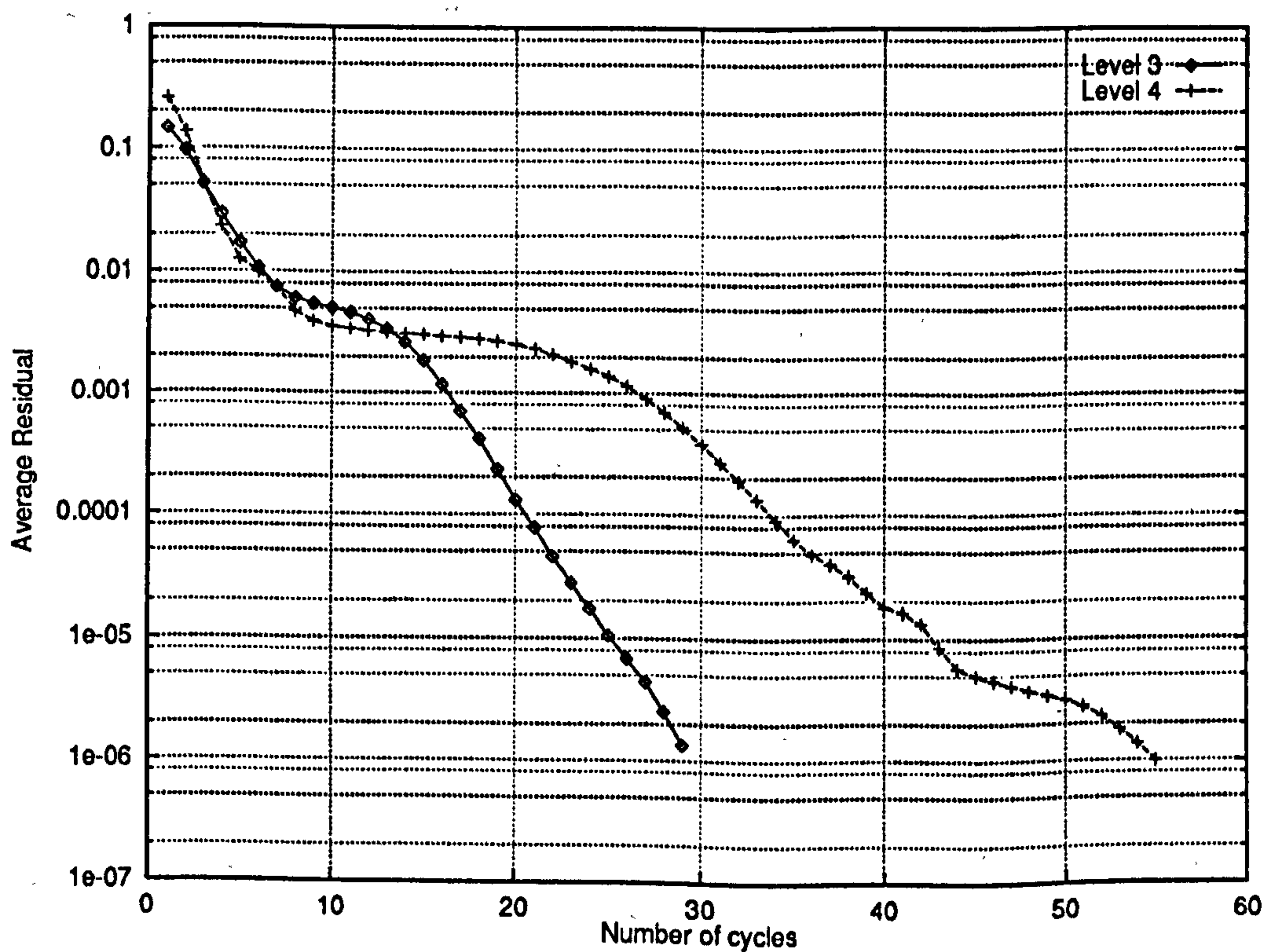


Figure 5.178: Low Reynolds number flow through a T-junction – convergence history for different grid levels

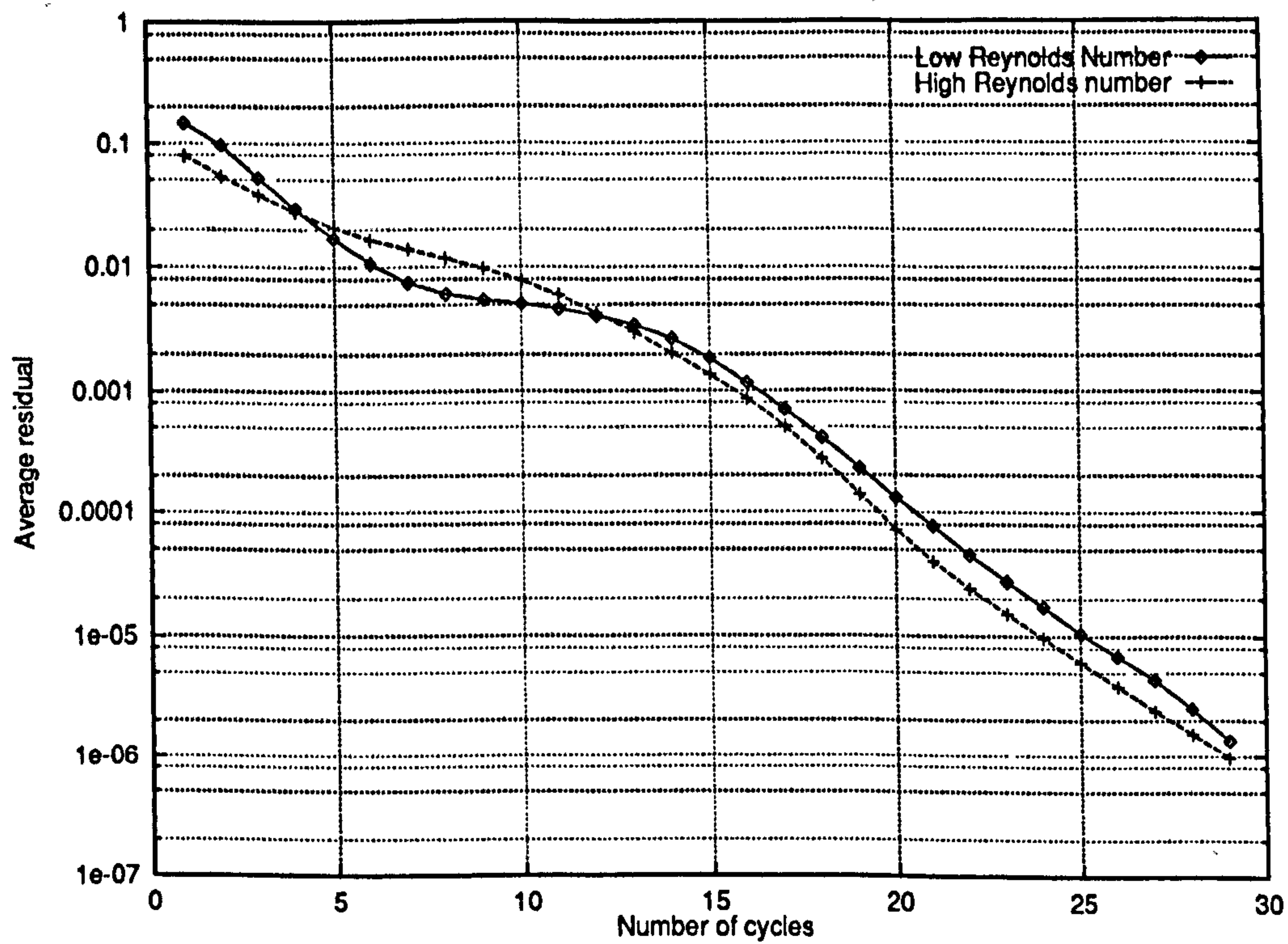


Figure 5.179: Multiphase flow through a T-junction – convergence history on a level 3 grid for different Reynolds numbers

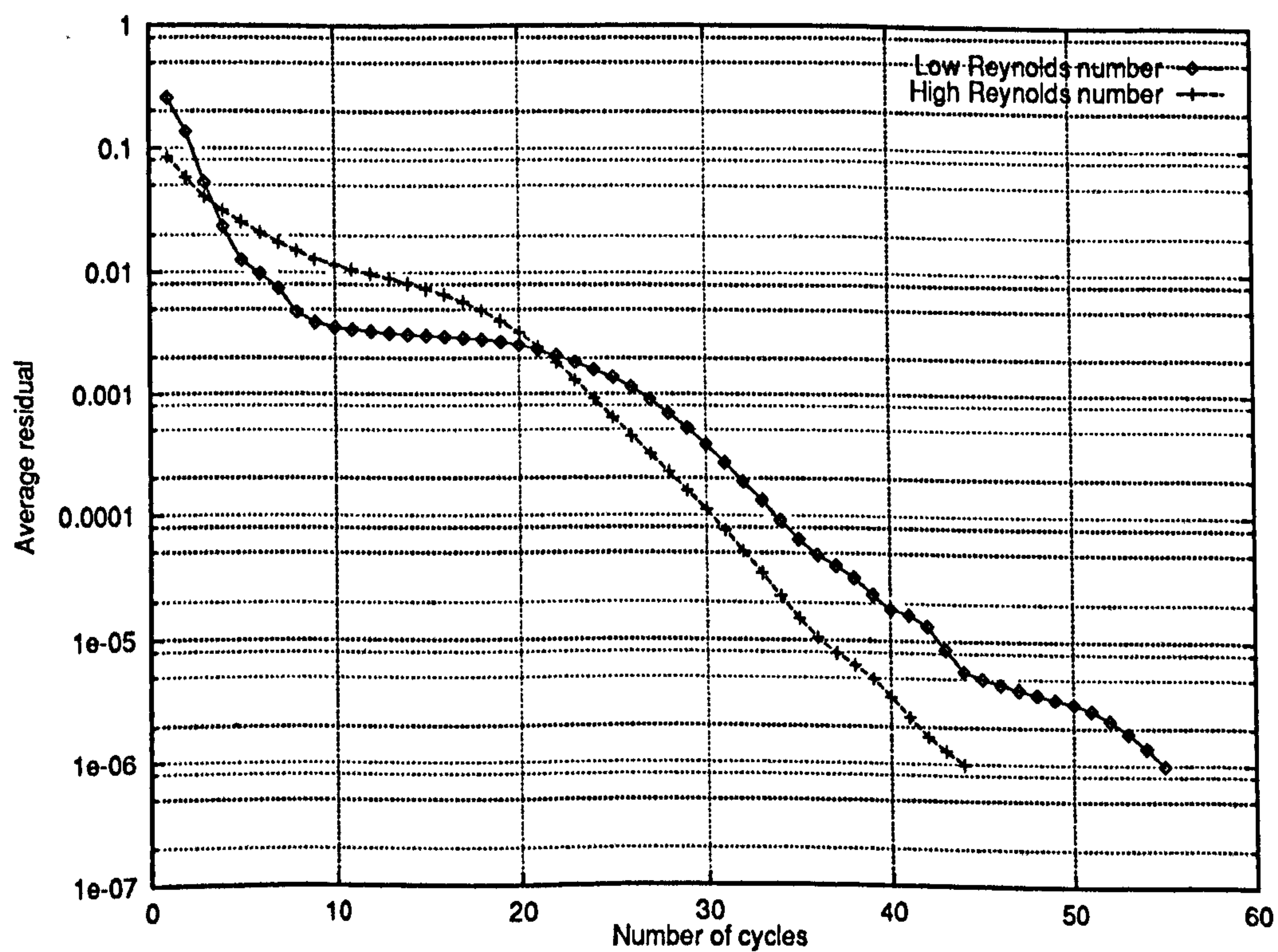


Figure 5.180: Multiphase flow through a T-junction – convergence history on a level 4 grid for different Reynolds numbers

5.4.12 Two-Phase Flow through a Longer Channel – Effect of Outlet Boundary Conditions

The solution profiles shown in the previous section indicate that at $Re = 100$, the two-phase flow in a channel requires more than three channel widths to separate. It is relevant to ask whether the convergence discrepancies reported in Section 5.4.2 are related to the error introduced by applying Neumann-type boundary conditions at the outlet on a still-separating flow.

The two phase channel flow problem was solved over the region $0 \leq x \leq 12$. The solution is illustrated by Figures 5.181 and 5.182. It is clear that at $x = 12$, the flow has not reached a developed pattern yet as one phase is still being accelerated relative to the other. However, the solution for $0 < x < 3$ is very close to the solution obtained for the shorter computational domain (Figures 5.183 to 5.185). Errors associated with the boundary conditions applied at the outlet do not seem to play an important part in determining the solution.

It has also been observed that the convergence factors for finer grids are certainly not improved (see Figure 5.186). Compared with Figure 5.124 which was obtained from identical calculations, except for the geometry, the plateau phase is greatly extended. These results suggest that the dominant errors occur in the neighbourhood of solid walls. The plateau phase appears to be due to these errors not being eliminated efficiently by the multigrid procedure.

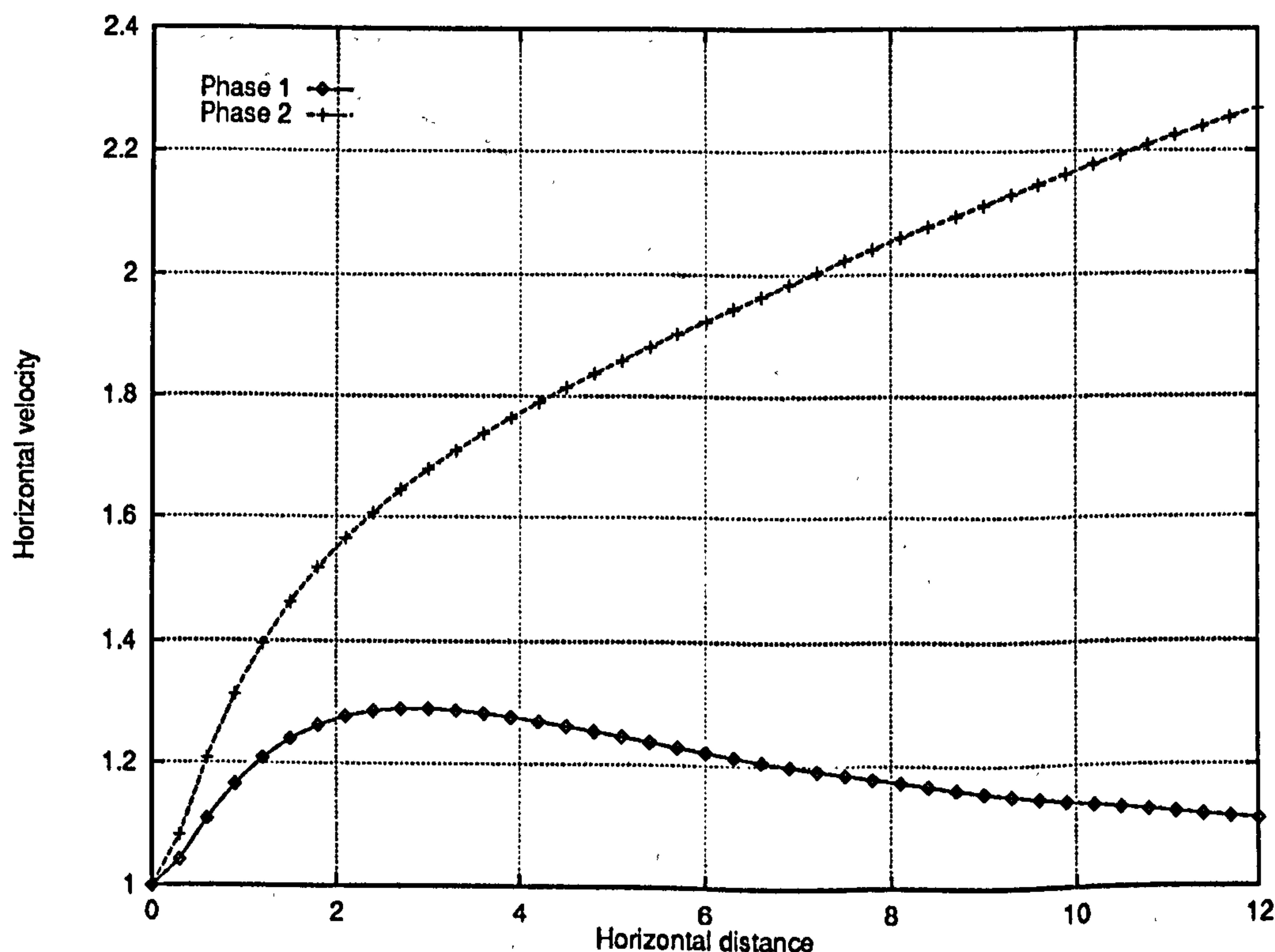


Figure 5.181: Two-phase channel flow – Extended computational domain – Horizontal velocity profile along the line $y = 0.5$

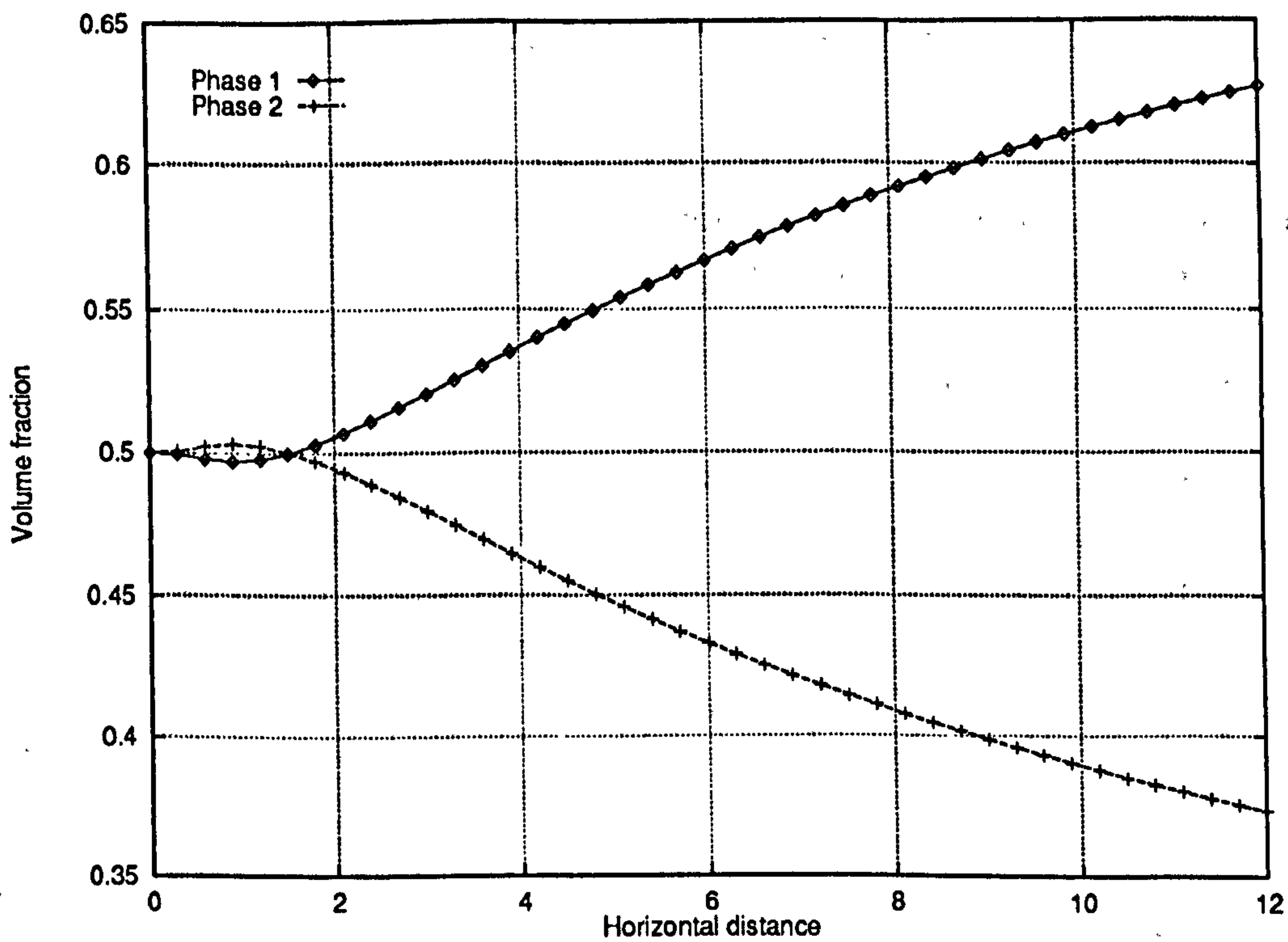


Figure 5.182: Two-phase channel flow – Extended computational domain – Volume fraction profiles along the line $y = 0.5$

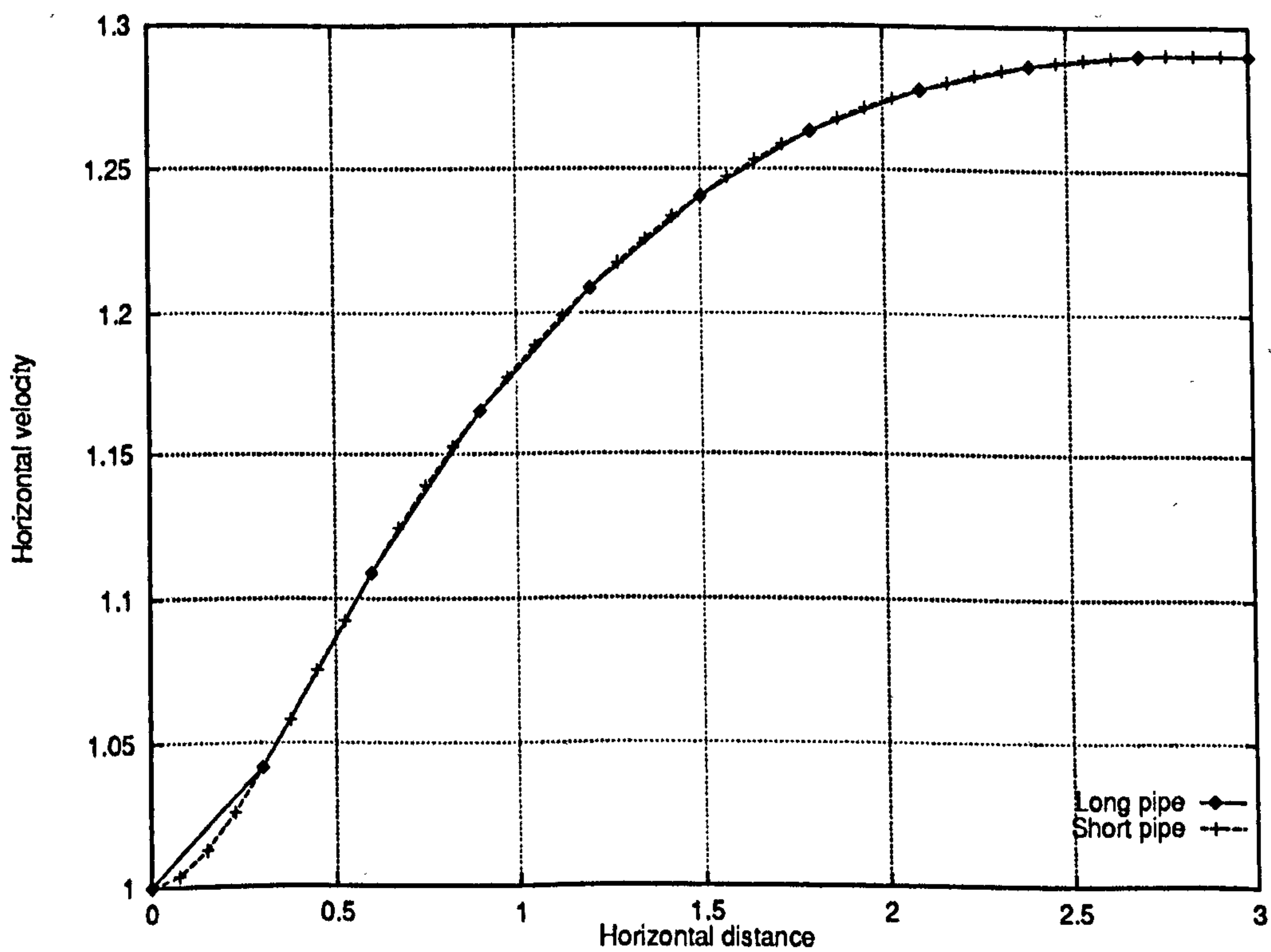


Figure 5.183: Two-phase channel flow – Horizontal velocity profiles for phase 1 along the line $y = 0.5$ for $0 \leq x \leq 3$ for different length of the computational domain

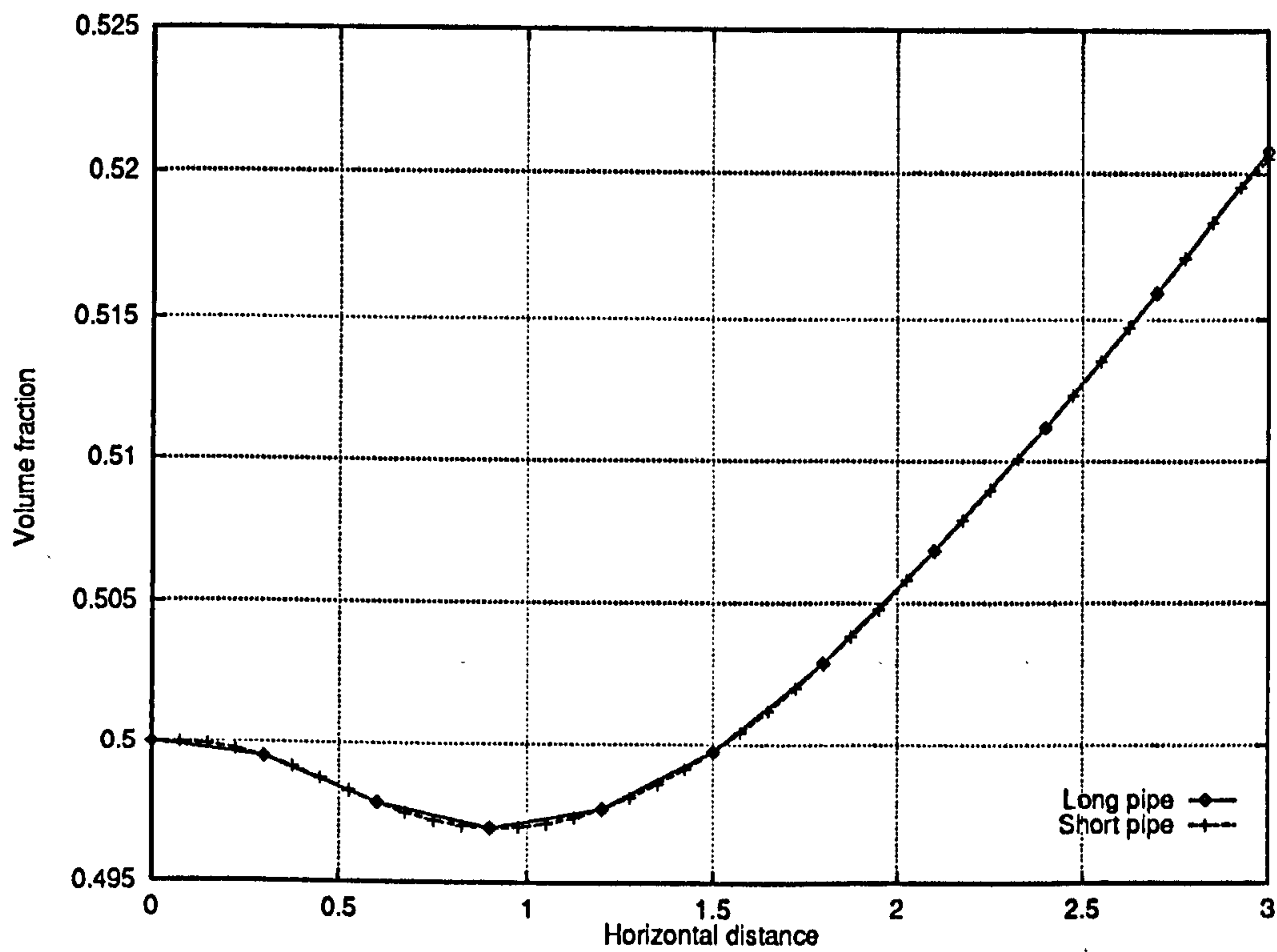


Figure 5.184: Two-phase channel flow – Volume fraction profiles for phase 1 along the line $y = 0.5$ for $0 \leq x \leq 3$ for different lengths of the computational domain

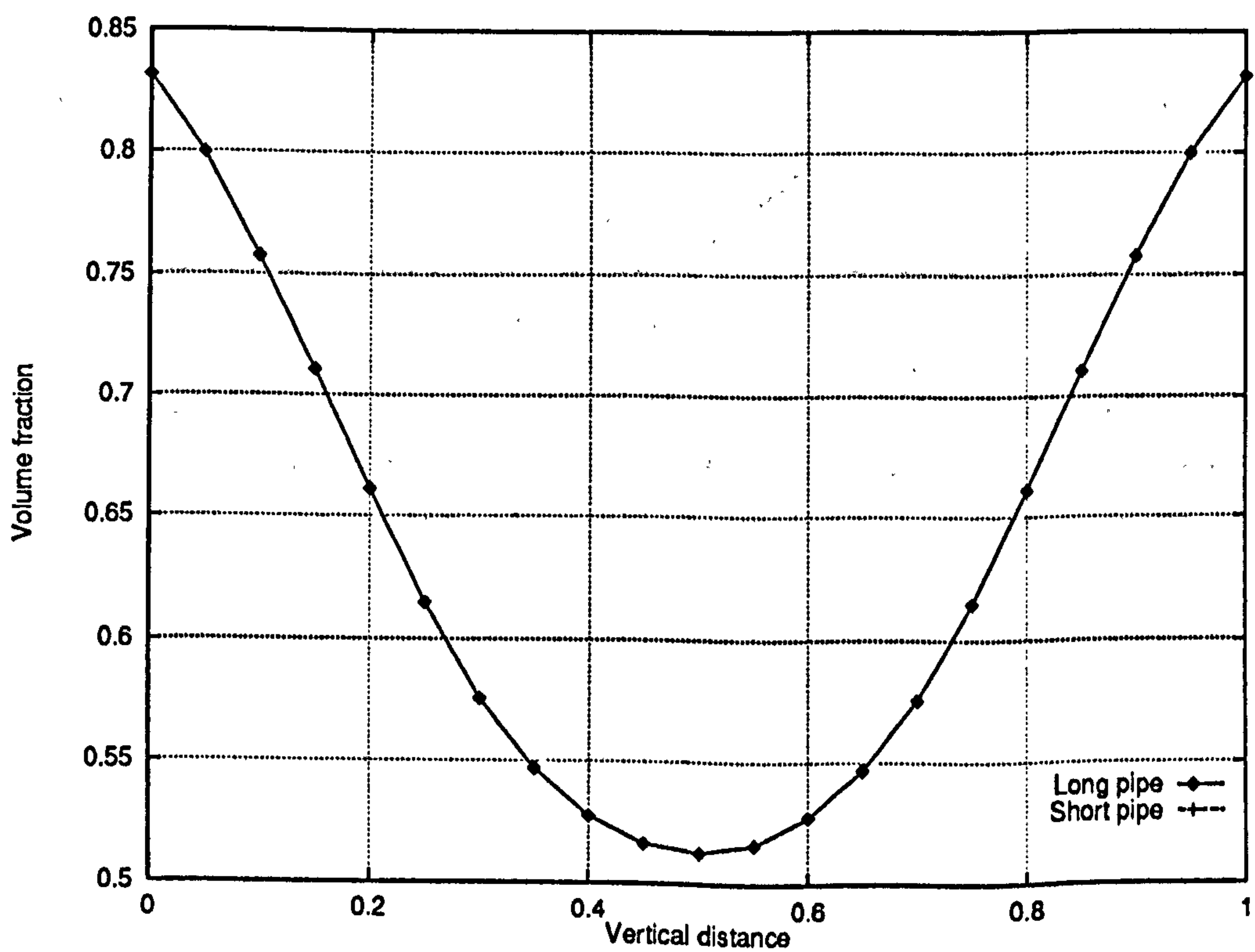


Figure 5.185: Two-phase pipe channel – Volume fraction profiles for phase 1 along the line $x = 2.5$ for computational domain of different lengths

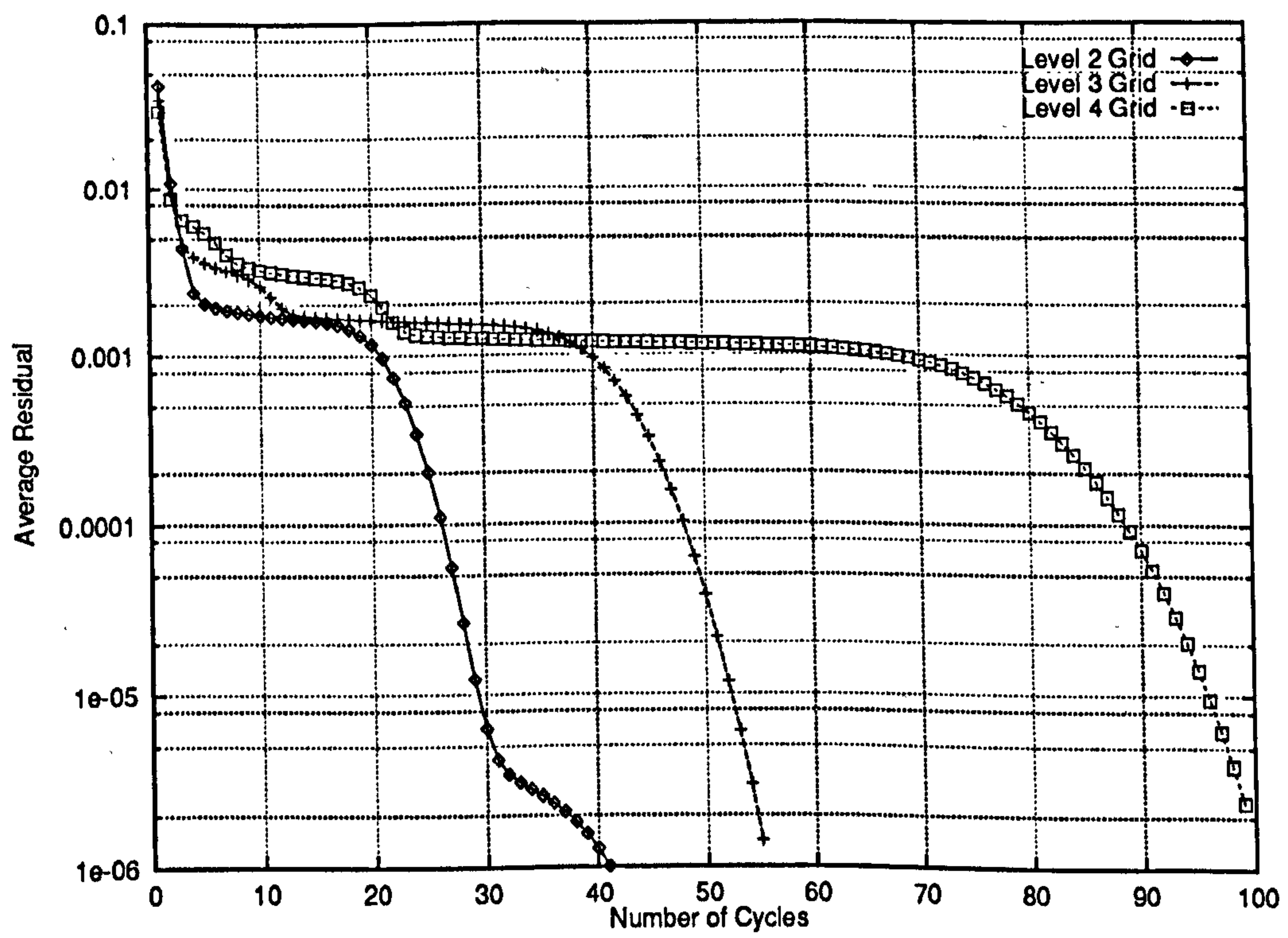


Figure 5.186: Two-phase channel flow – Extended computational domain – Convergence history for various grid levels – Compare with Figure 5.124

5.4.13 Effect of Different Formulations of the Stress Tensor Flux

For multiphase flows as well as single phase flows, a constitutive relationship between the diffusive stress tensor in the momentum equation and the flow variables is required to close the governing conservation laws. For the present thesis, a standard expression has been chosen ¹² :

$$r\mathbf{T} = \mu r(\nabla\mathbf{v} + (\nabla\mathbf{v})^T).$$

In Section 3.3.2, this constitutive relationship is discussed at length. In particular, it is shown that diffusive flux $\nabla \cdot (r\mathbf{T})$ involves the cross derivative terms:

$$\mu \begin{pmatrix} \frac{\partial}{\partial y} \left(r \frac{\partial v}{\partial x} \right) \\ \frac{\partial}{\partial x} \left(r \frac{\partial u}{\partial y} \right) \end{pmatrix}. \quad (5.1)$$

These “extra” diffusive terms further couple the momentum equations together and are analogous to hyperbolic operators. In Section 3.3.2, it is shown that they are associated with spatial variations of the volume fraction. When r is constant throughout the domain, they drop from the momentum equations and the usual single phase diffusive operator for Newtonian fluids:

$$\mu \begin{pmatrix} u_{xx} + u_{yy} \\ v_{xx} + v_{yy} \end{pmatrix}$$

is recovered.

It is well known [89] that multi-grid methods are less efficient when applied to hyperbolic equations compared with elliptic problems. The root cause is referred to as “strong alignment” by Brandt (see for instance [10]). That is, the problem becomes locally one-dimensional *and* aligned with the grid so that equations in the transverse direction(s) are decoupled, precluding the removal of errors by relaxation. Convergence is often achieved but is dependent on the propagation of boundary data which is highly grid dependent. As a result, grid independent convergence factors are seldom observed.

It is therefore tempting to formulate the hypothesis that the loss of efficiency as the grid becomes finer is due to the multiphase cross derivative “hyperbolic-like” terms in the momentum equations.

Four arguments can be given in support of such a conjecture:

1. In Section 5.4.11, we have shown that the convergence factors of the multigrid method are degraded as the Reynolds number diminishes (see Figure 5.173).

¹²The subscripts denoting the phases are dropped for clarity in the next two sections.

This is atypical of multigrid methods in particular and fluid flow simulation in general. For single phase flows, as the Reynolds number tends to zero, the convective terms affect the solution far less, the problem becomes increasingly linear and the measure of discrete ellipticity rises. As a result, multigrid methods perform very well.

Furthermore, at a given Reynolds number, when grids are refined slower convergence factors are observed (Figure 5.175). In such cases, the discretisation of $\nabla \cdot (r\mathbf{T})$ increasingly dominates over the convective terms.

The combination of these two effects indicates that the discretisation of $\nabla \cdot (r\mathbf{T})$ is at least partly to blame for the slow convergence observed in some phases of the multigrid computations.

2. The solution of the channel flow problem contains regions where the volume fractions vary strongly near solid walls (see Figures 5.33 and 5.114). These are regions where the hyperbolic-like terms will take large values.
3. When the length of the computational domain is increased (Section 5.4.12), the plateau phase becomes significantly larger (Figure 5.186). This suggests that the dominating errors during this phase are associated with solid walls, i.e. locations where extra-diffusive terms may take large values.
4. In more complex problems, such as a two-phase flow through a T-junction (see Section 5.3.2), results are more grid independent (see Table 5.7). In such problems, the errors are concentrated in regions where the flow changes significantly, as opposed to the channel flow problem where the solution is determined to a large extent by boundary and/or multiphase effects. The pattern of grid refinement (see Figures 6.54 and 6.57) confirms that the error distributions are very different in the two problems. In regions where the flow changes rapidly in space, all differential terms are important so that the relative effect of the hyperbolic-like terms diminishes. Furthermore, we have seen in Section 5.4.11 that for lower Reynolds numbers, the convergence history of the two-phase T-junction problem, on sufficiently fine grids, becomes qualitatively similar to those of the channel flow problem.

Our hypothesis was tested by simply turning off the extra terms and replacing:

$$\nabla \cdot (r\mathbf{T}) = \mu \left(\begin{array}{l} 2 \frac{\partial}{\partial x} \left(r \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(r \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(r \frac{\partial u}{\partial y} \right) \\ \frac{\partial}{\partial x} \left(r \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial x} \left(r \frac{\partial u}{\partial y} \right) + 2 \frac{\partial}{\partial y} \left(r \frac{\partial v}{\partial y} \right) \end{array} \right) \quad (5.2)$$

by:

$$\nabla \cdot (r\mathbf{T}) = \mu \left(\begin{array}{l} \frac{\partial}{\partial x} \left(r \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(r \frac{\partial u}{\partial y} \right) \\ \frac{\partial}{\partial x} \left(r \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left(r \frac{\partial v}{\partial y} \right) \end{array} \right) \quad (5.3)$$

in the momentum equations. The fluxes defined by equations 5.2 and 5.3 are referred to as the full and partial tensor fluxes respectively. In places, the word flux may be dropped where no ambiguity exists.

Grid level	Partial diffusion		Full diffusion	
	Global	Asymptotic	Global	Asymptotic
2	0.9107	0.9018	0.9004	0.8987
3	0.9230	0.9145	0.9141	0.9031
4	0.9358	0.9115	0.9417	0.9232
5	0.9517	0.9322	0.9636	0.9520

Table 5.14: Multiphase channel flow problem – Comparison of global and asymptotic convergence factors for computations on different finest levels for equations with full and partial stress tensor flux $\nabla \cdot (r\mathbf{T})$ and for $Re = 100$

Flow solutions for the channel problem (with $Re_1 = 100$ and $Re_2 = 100$) are shown in Figures 5.187 to 5.195. It clearly appears that the extra terms have a diffusive effect on the volume fraction (compare Figures 5.195 and 5.196). This last figure, which shows the variation of the volume fraction across the channel for different grid levels, shows that the evolution towards the annular flow pattern is faster if the extra terms are not included since the phases will separate faster.

We will now consider the convergence history. It can be concluded that the slow multi-grid convergence factors during the “plateau” phase can indeed be attributed to the presence of extra terms in the stress tensor flux (compare Figure 5.197 with Figure 5.198 and Figure 5.199 with 5.124). We also observe that on fine grids, the degradation of the convergence factors for very low residuals (see page 202) is removed when the extra diffusive terms are turned off.

Asymptotic convergence factors are also much more grid independent (see Table 5.14) except for grid 5 where the conditioning is very significantly worse than on level 4 (see Table 5.15)¹³. These condition numbers should be compared with those observed when the full diffusion tensor is used. On a level 5 grid, these are of the order of 10^4 for both the average and maximum values.

It should be noted that on coarse grids the effect of the extra diffusive terms is far less noticeable: compare Figures 5.200 and 5.201 with 5.202 and 5.203. Again, this is not surprising since on coarser grids, numerical diffusion (either from hybrid differencing in the momentum equations or cell-donor differencing in the continuity equations) becomes dominant over the physical diffusion.

A similar set of experiments was performed on the T-junction problem. Figures 5.204 to 5.208 show the solution obtained. It can clearly be seen that the only quantities significantly affected are the volume fractions. If the partial tensor is chosen, volume fraction fronts are significantly sharper (Figure 5.208 in particular) than if the full tensor is chosen. Again this is consistent with the interpretation of the extra tensor term being large where there is spatial variation of the volume

¹³The maximum condition numbers reported in Table 5.15 occur near the walls where the flow is most separated.

Finest grid Grid level	Level 5		level 4	
	Average	Maximum	Average	maximum
1	7.0×10^1	7.2×10^1	7.1×10^1	7.3×10^1
2	1.7×10^2	2.5×10^2	1.5×10^2	1.6×10^2
3	6.3×10^2	3.5×10^3	3.0×10^2	3.7×10^2
4	5.8×10^3	1.3×10^5	6.1×10^2	1.7×10^3
5	8.4×10^4	5.1×10^6	-	-

Table 5.15: Comparison of the *average* and *maximum* condition numbers of the Newton correction systems on different grid levels, and for computations on different finest levels, with the partial stress tensor flux $\nabla \cdot (r\mathbf{T})$ at $Re = 100$.

fraction.

Figure 5.209 shows the convergence history for both formulations of the stress tensor flux. It is very interesting to note that far from improving the convergence factors, the choice of the partial tensor significantly slows convergence for low residuals. When the residuals are large, the convergence rates for both formulations are of a very similar level. This indicates that in keeping with the basic hypothesis of this section, the cross derivative terms do not play the same role in this problem as for the two-phase channel flow problem. They do not slow convergence down because they are dominated by other terms and actually have a beneficial effect on performance by diffusing the volume fraction more.

In this section we have established that for simple problems where multiphase diffusive effects dominate, the formulation of the diffusive flux $\nabla \cdot (r\mathbf{T})$ has a critical effect on convergence. The stress tensor chosen for the present work,

$$r\mathbf{T} = \mu r(\nabla \mathbf{v} + (\nabla \mathbf{v})^T),$$

is a standard expression. In the momentum equations, it introduces hyperbolic-like cross derivative terms which are proportional to the spatial variation of the volume fractions. These terms are responsible for the degradation of the multigrid convergence factors both for high and low residuals. They are diffusive but we remark that for the channel flow, the variation of the volume fractions are strongly one-dimensional and aligned with the computational grid. This may therefore be a case of strong-alignment. With the partial formulation of the diffusive flux, the *asymptotic* convergence factors can be grid independent but *global* convergence factors are not. This is due to the behaviour of the solution algorithm for higher residuals and can be attributed to the grid adjustments made necessary by the non-linearity of the governing equations.

However, we also note that another adverse diffusive appears to be at play in this complex problem (see Section 5.4.11). In particular, verification computations were attempted for channel flows at lower Reynolds numbers with the partial formulation

of the stress tensor. These computations diverged. The extra-diffusive terms are therefore an important stabilising factor, a fact confirmed by computation on the T-junction problem.

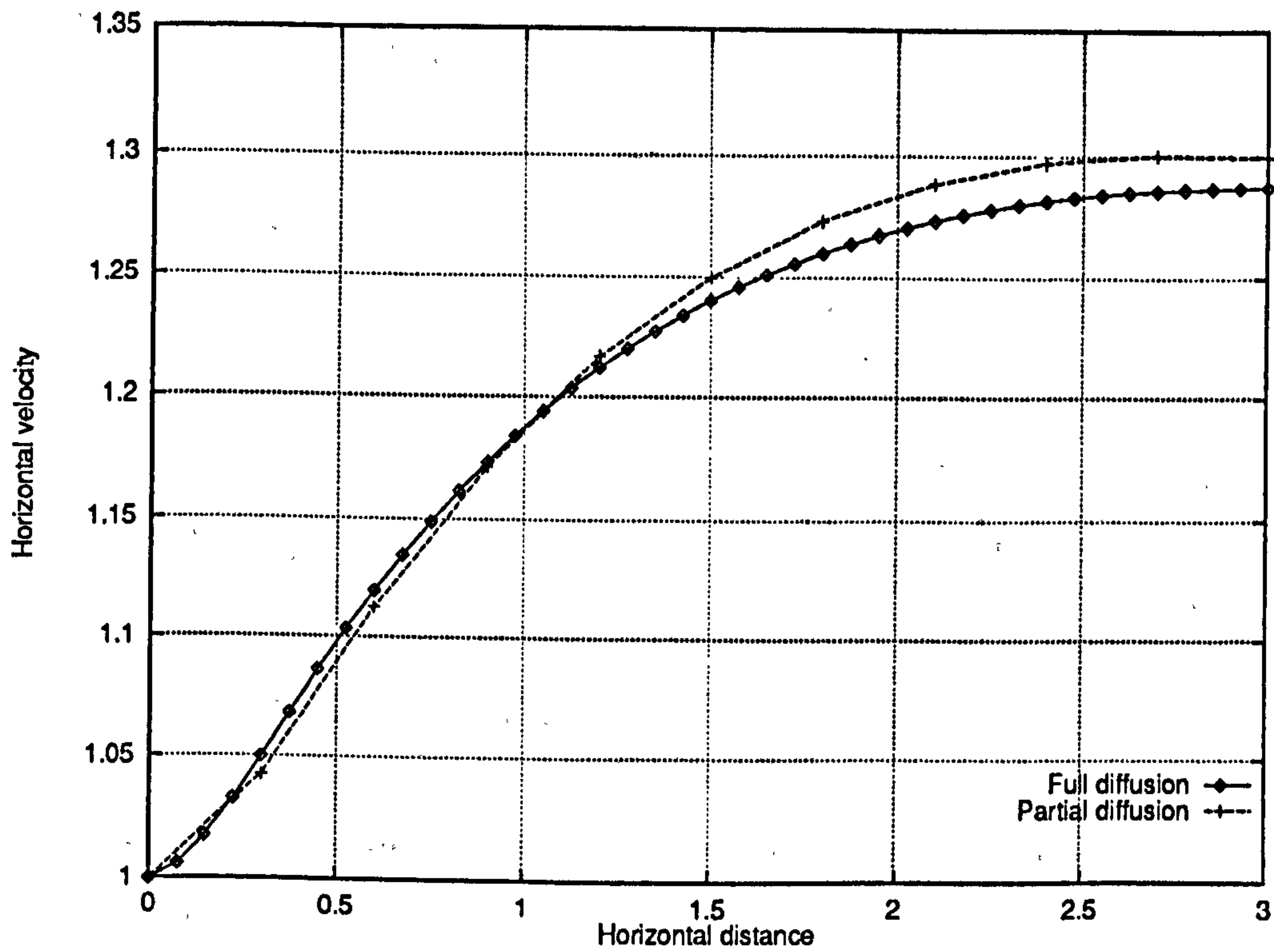


Figure 5.187: Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (rT)$ – Phase 1

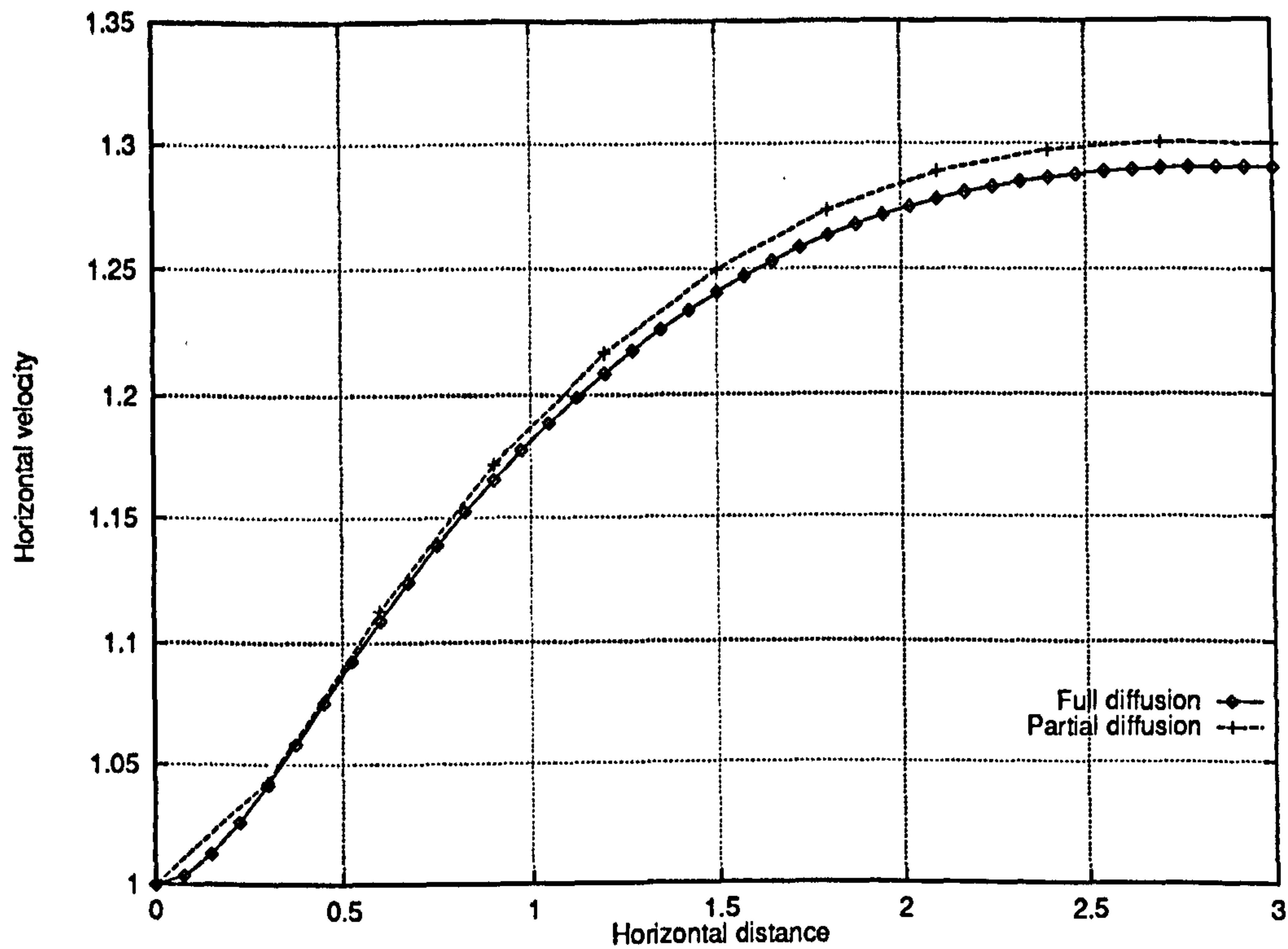


Figure 5.188: Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\tau \mathbf{T})$ – Phase 2

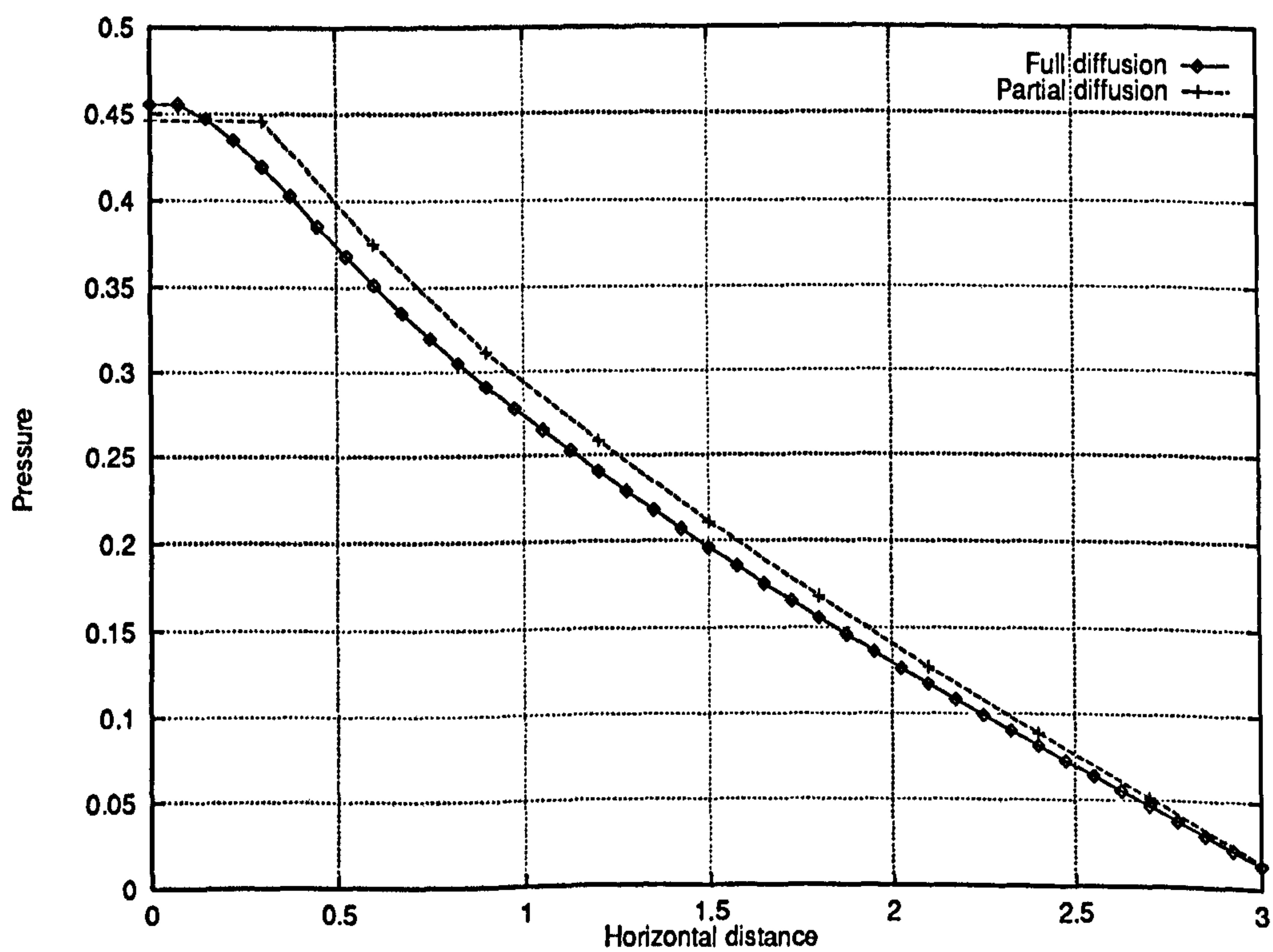


Figure 5.189: Two-phase channel flow – Comparison of pressure profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\tau \mathbf{T})$

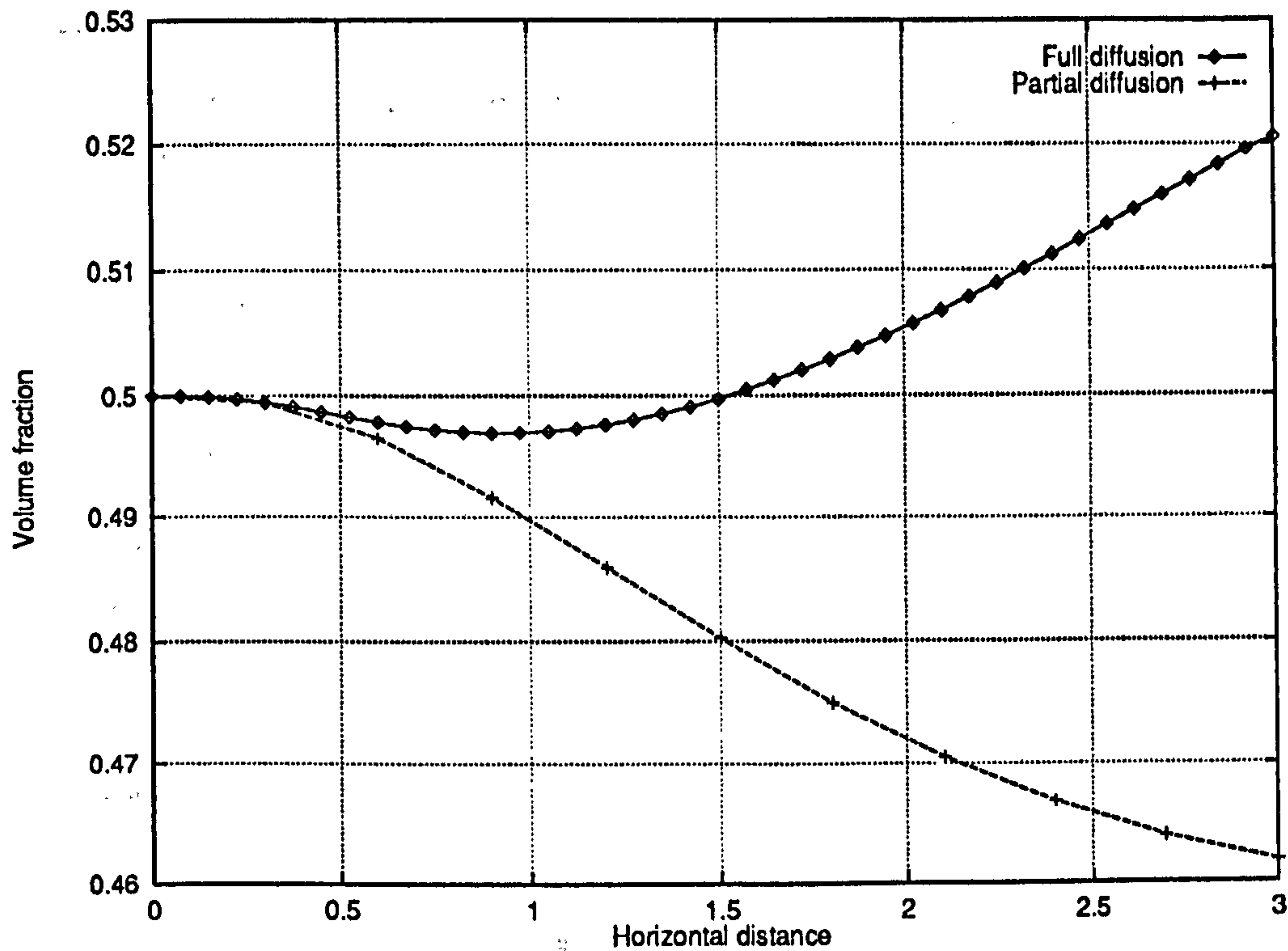


Figure 5.190: Two-phase channel flow – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\mathbf{r}\mathbf{T})$ – Phase 1

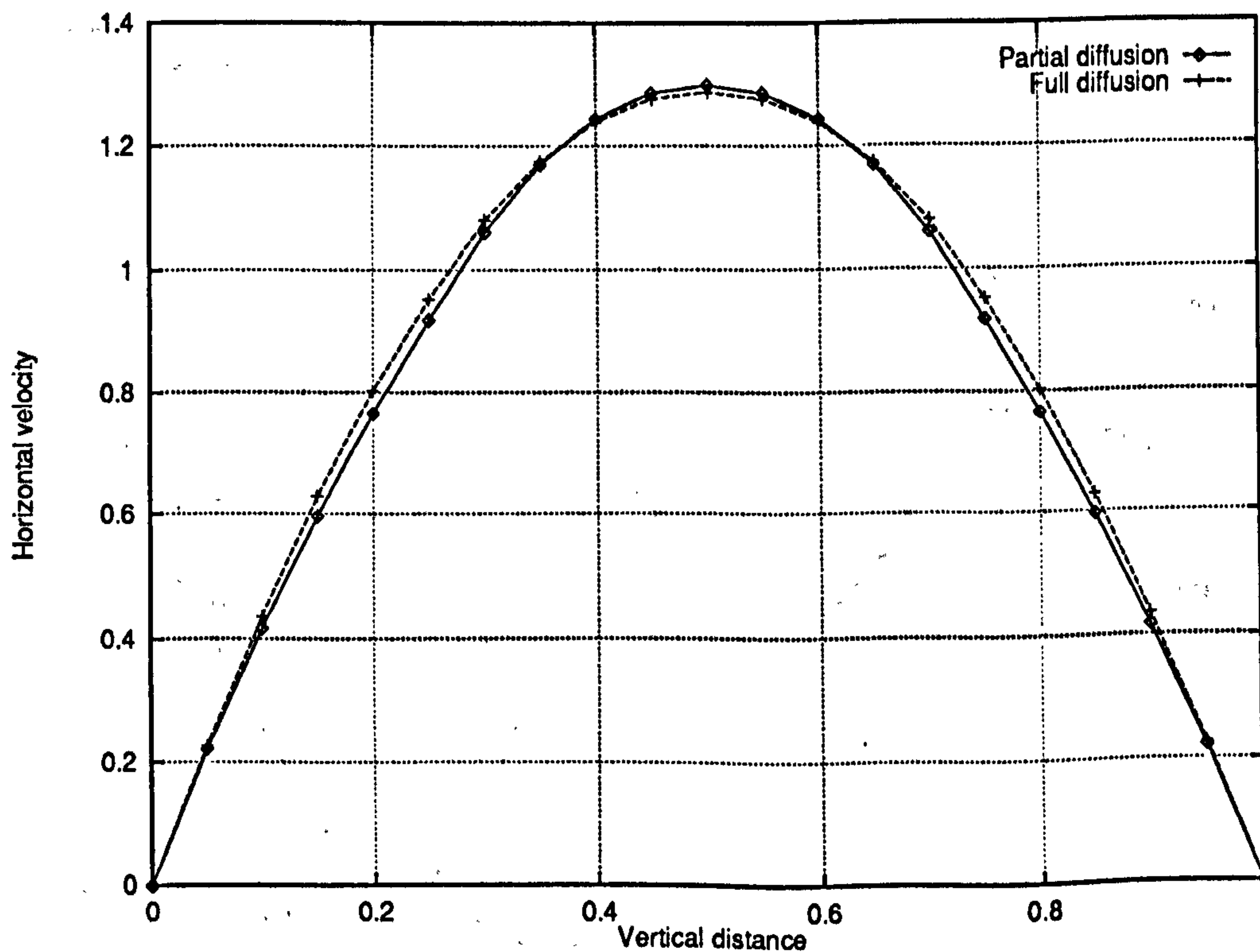


Figure 5.191: Two-phase channel flow – Comparison of horizontal velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\mathbf{r}\mathbf{T})$ – Phase 1

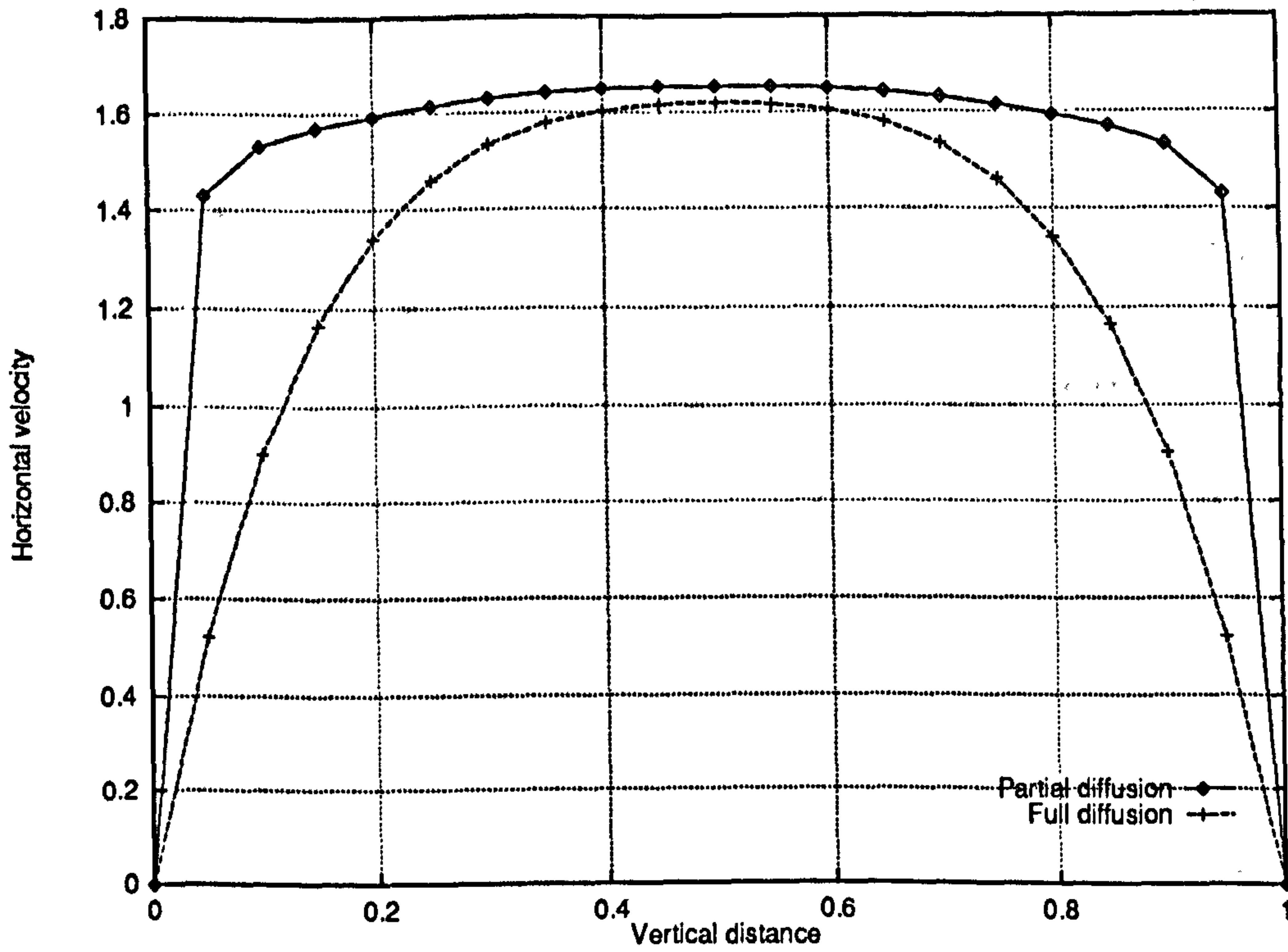


Figure 5.192: Two-phase channel flow – Comparison of horizontal velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (rT)$ – Phase 2

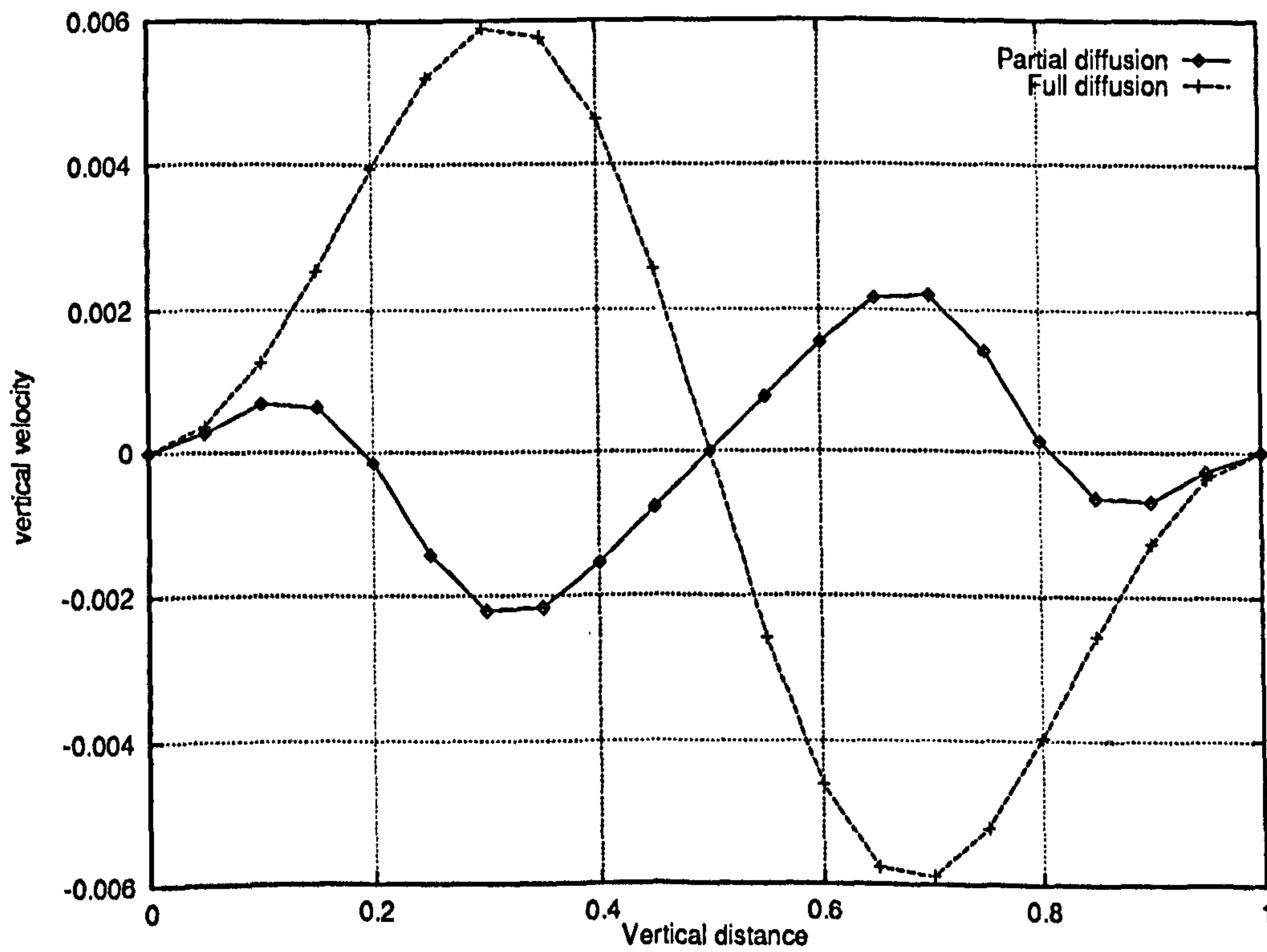


Figure 5.193: Two-phase channel flow – Comparison of vertical velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (rT)$ – Phase 1

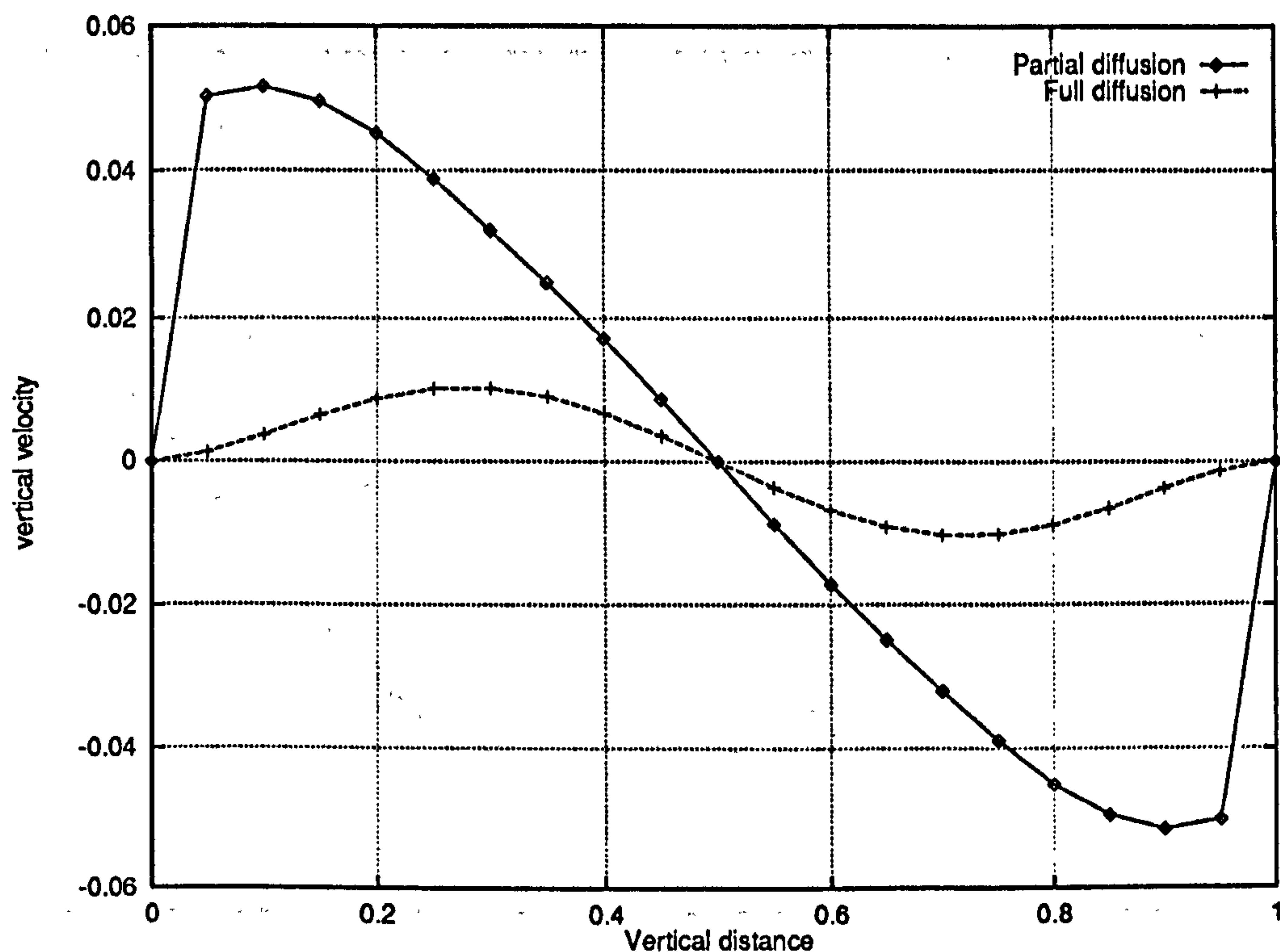


Figure 5.194: Two-phase channel flow – Comparison of vertical velocity profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 2

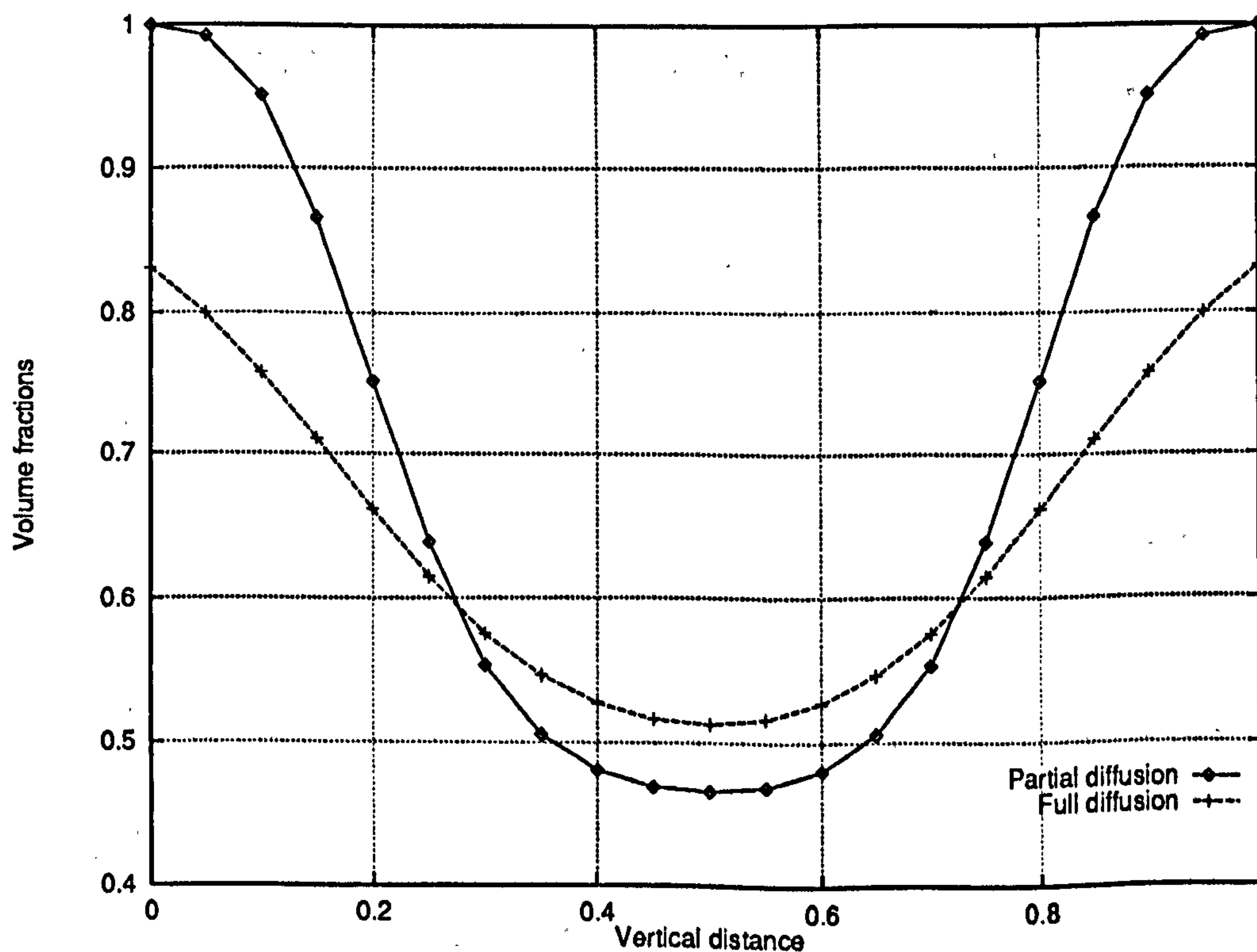


Figure 5.195: Two-phase channel flow – Comparison of volume fraction profiles along the line $x = 2.5$ for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Phase 1

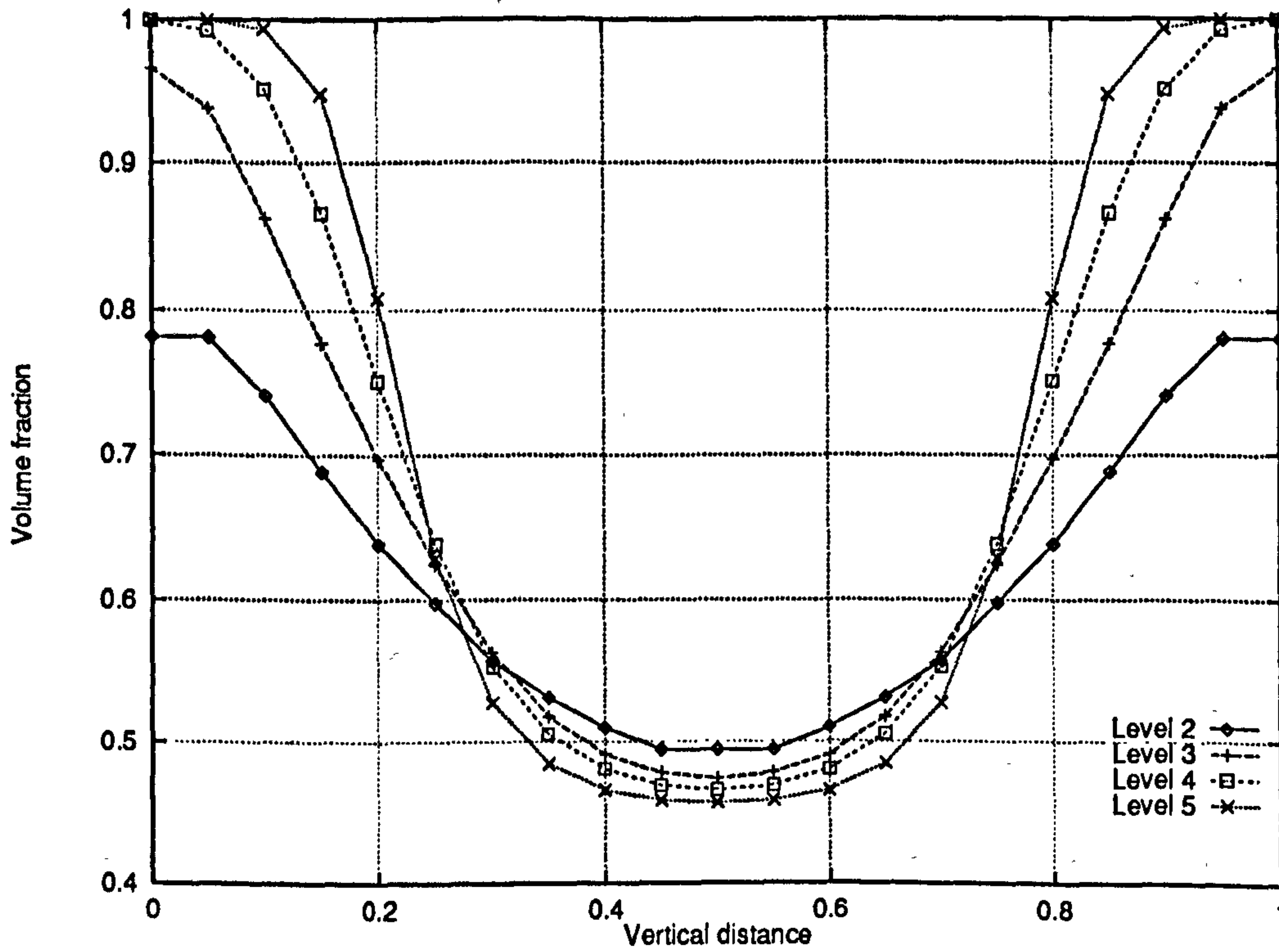


Figure 5.196: Two-phase channel flow - Comparison of volume fraction profiles along the line $x = 2.5$ for the partial diffusion model at different grid levels - Phase 1

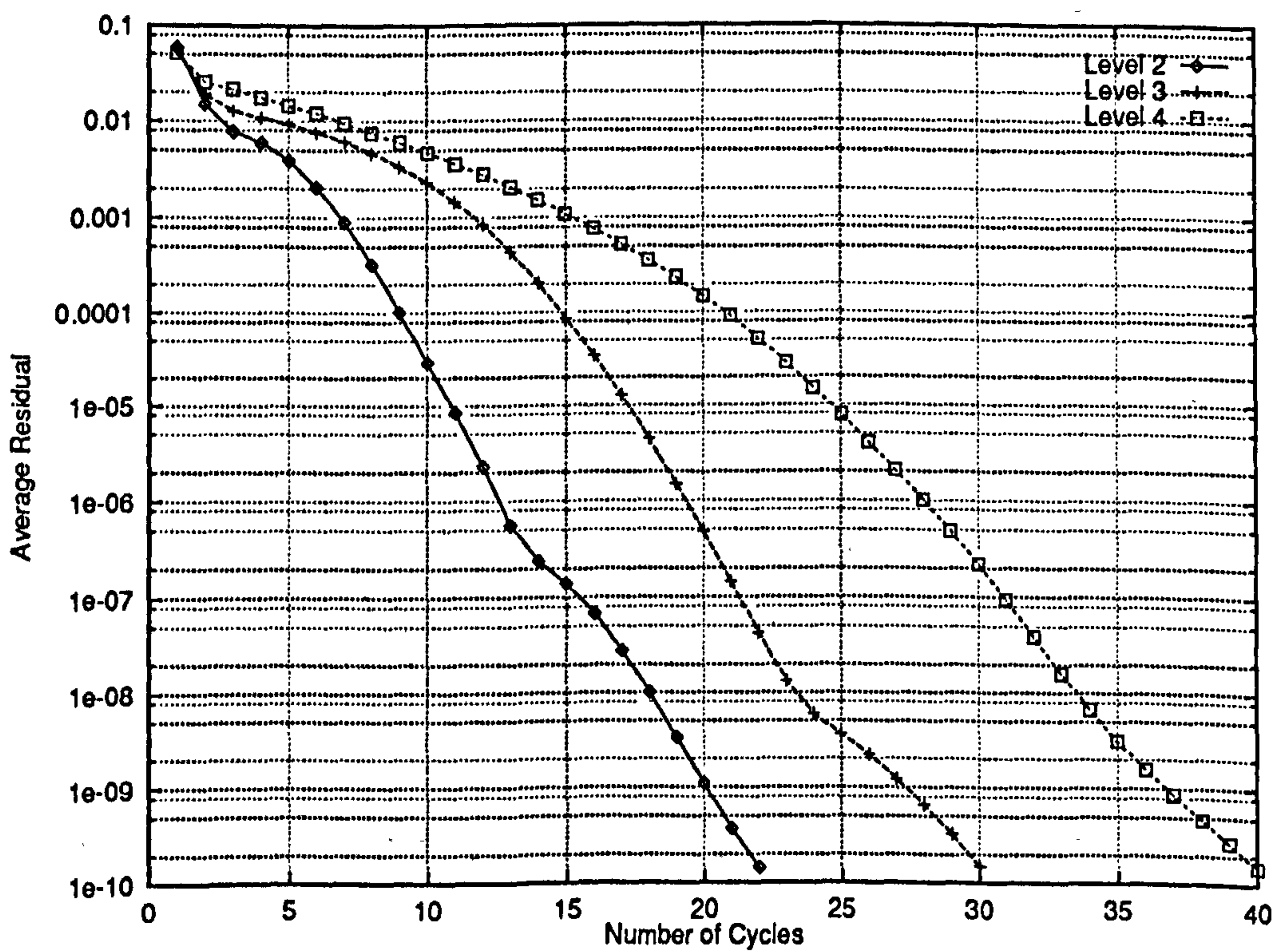


Figure 5.197: Two-phase channel flow - Comparison of the convergence histories for the partial diffusion model at different grid levels

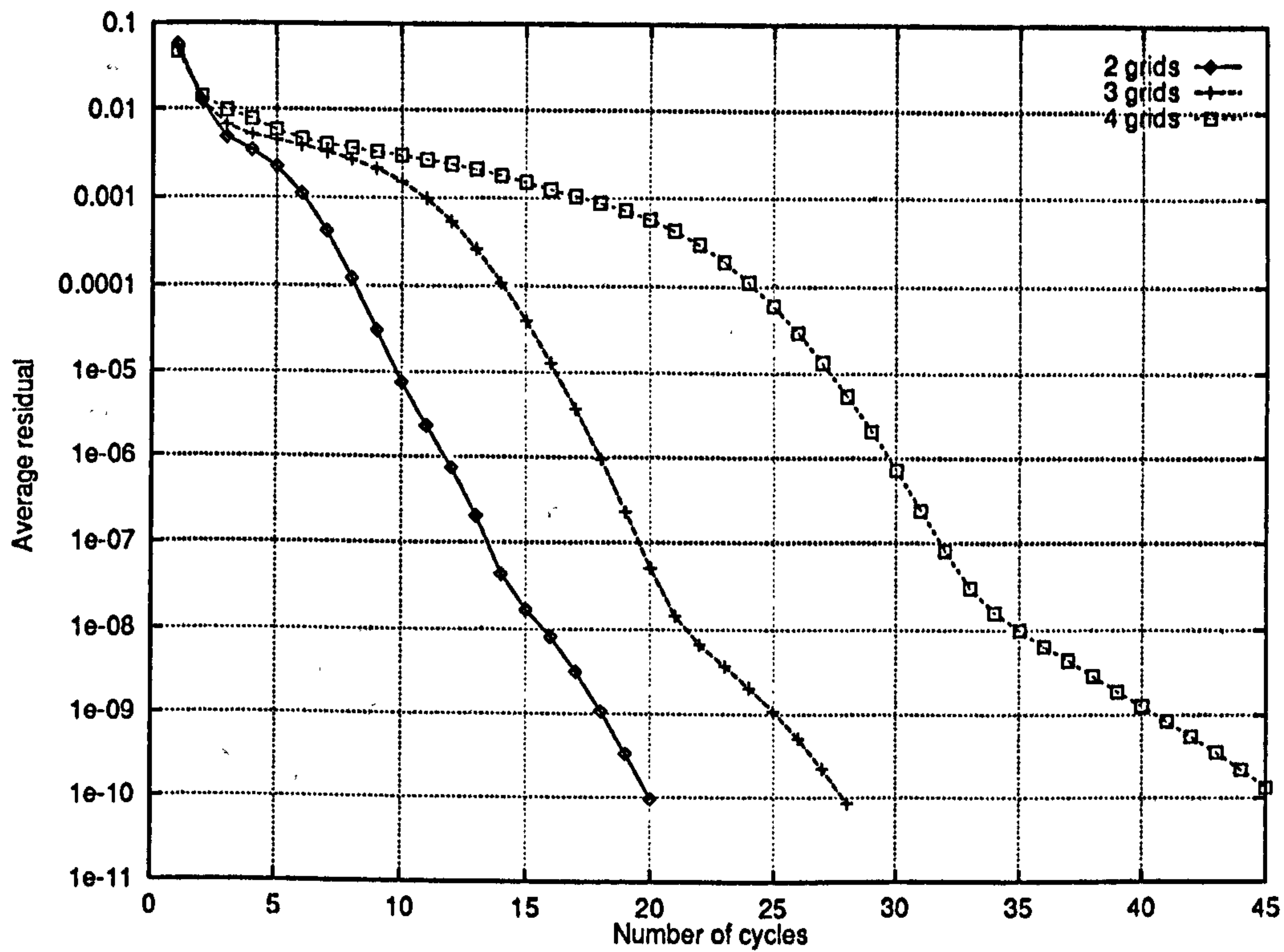


Figure 5.198: Two-phase channel flow – Comparison of the convergence histories for the full diffusion model at different grid levels

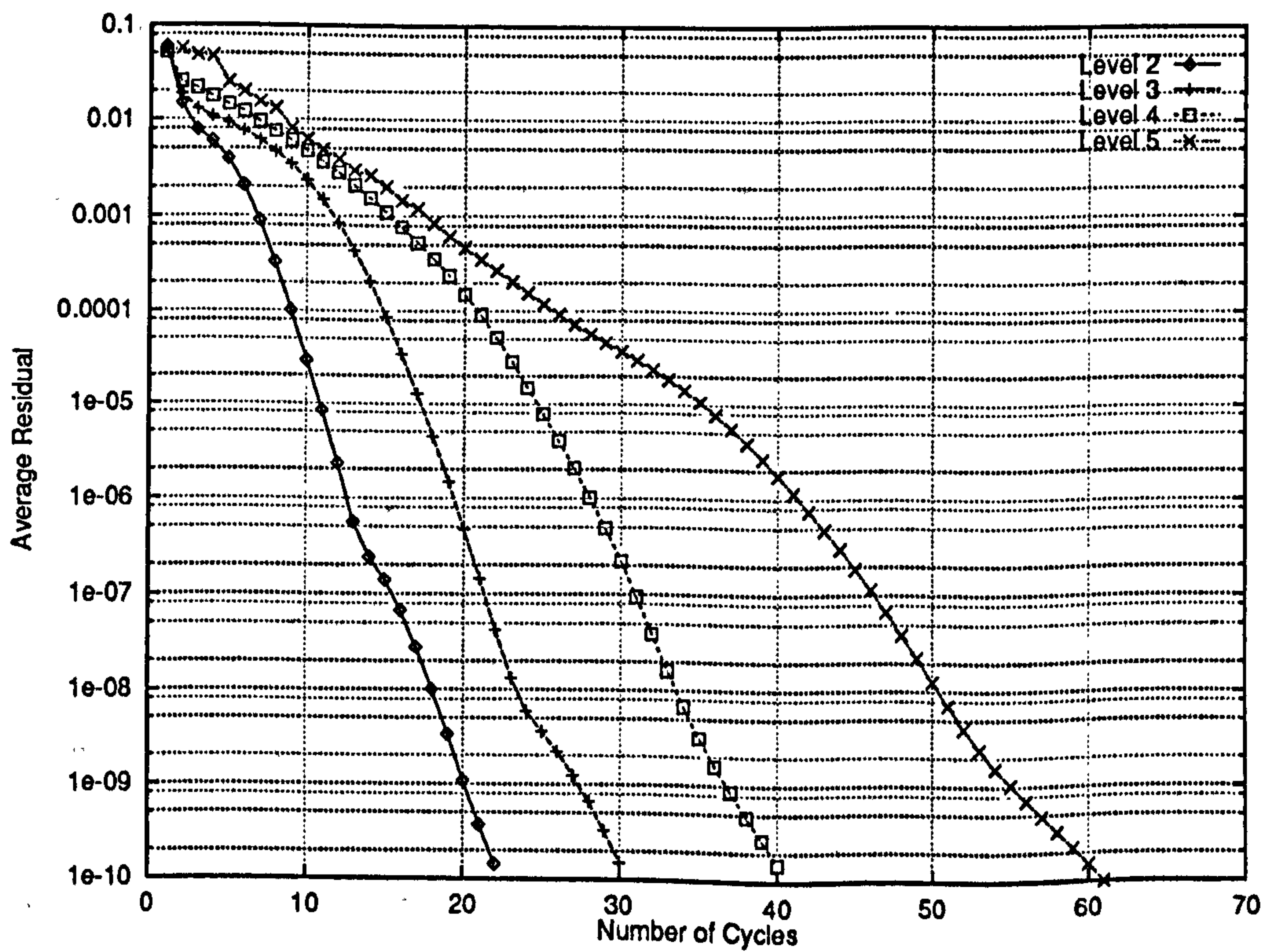


Figure 5.199: Two-phase channel flow – Comparison of the convergence histories for the partial diffusion model at different grid levels

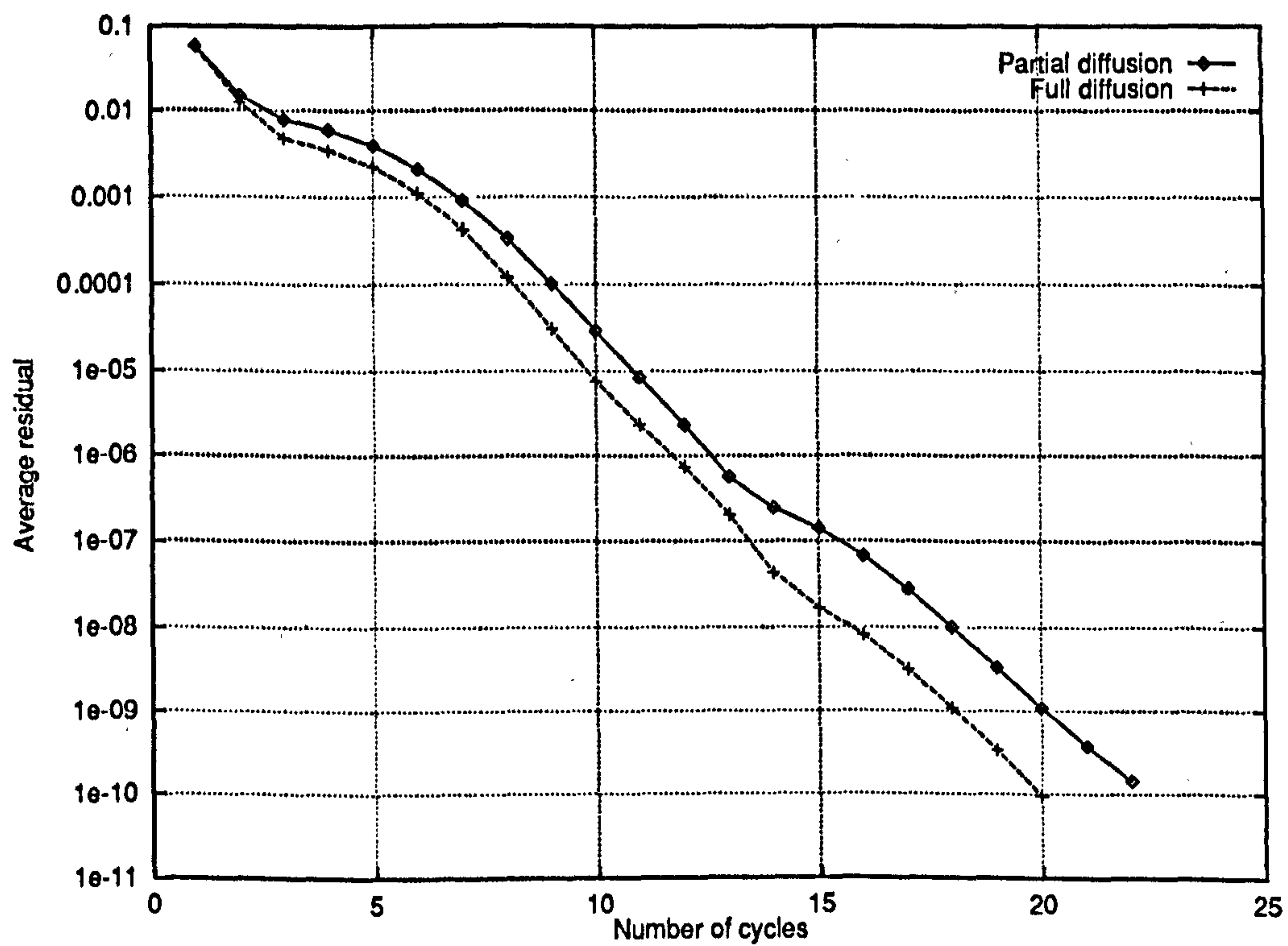


Figure 5.200: Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 2 grid

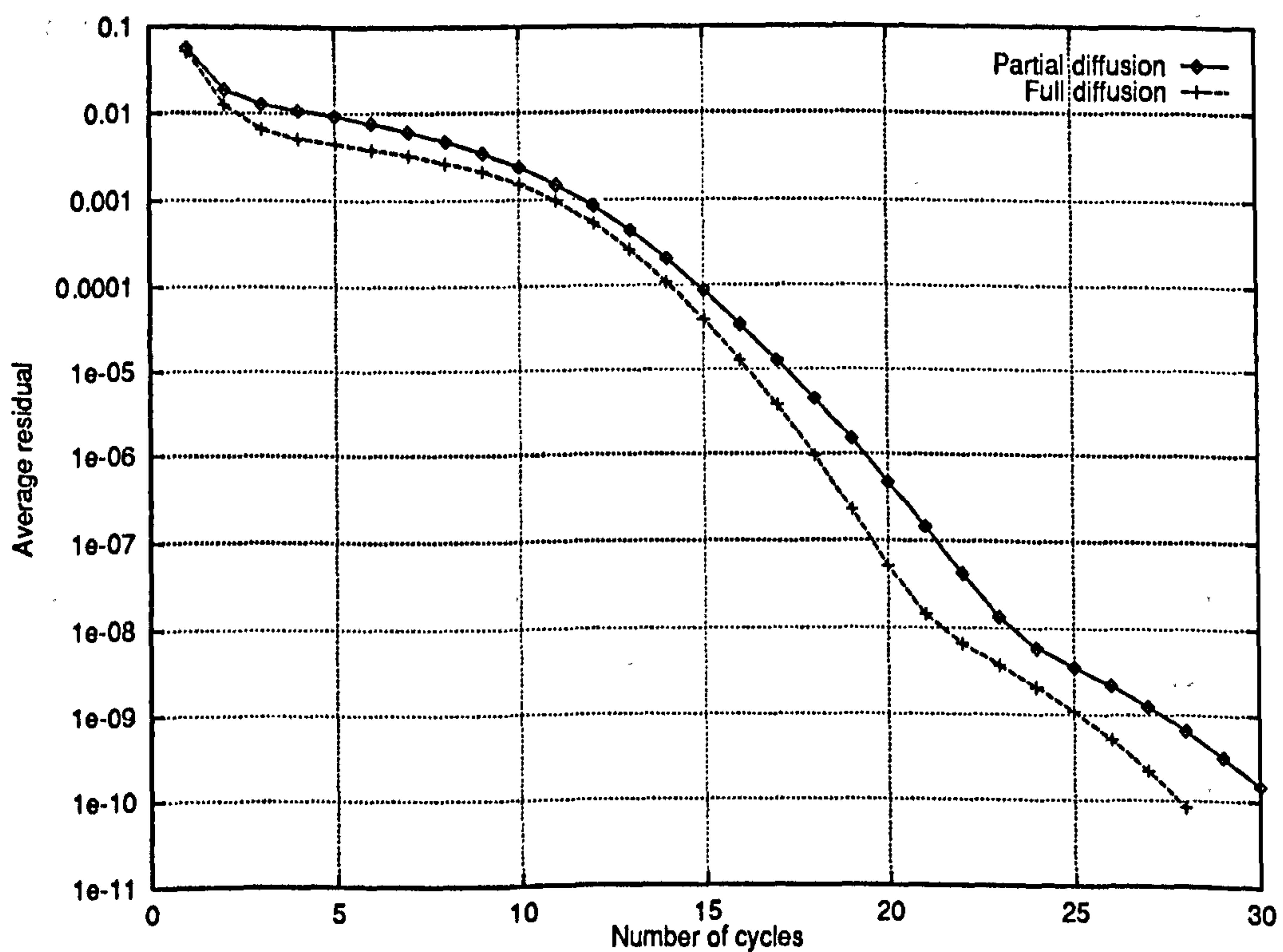


Figure 5.201: Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 3 grid

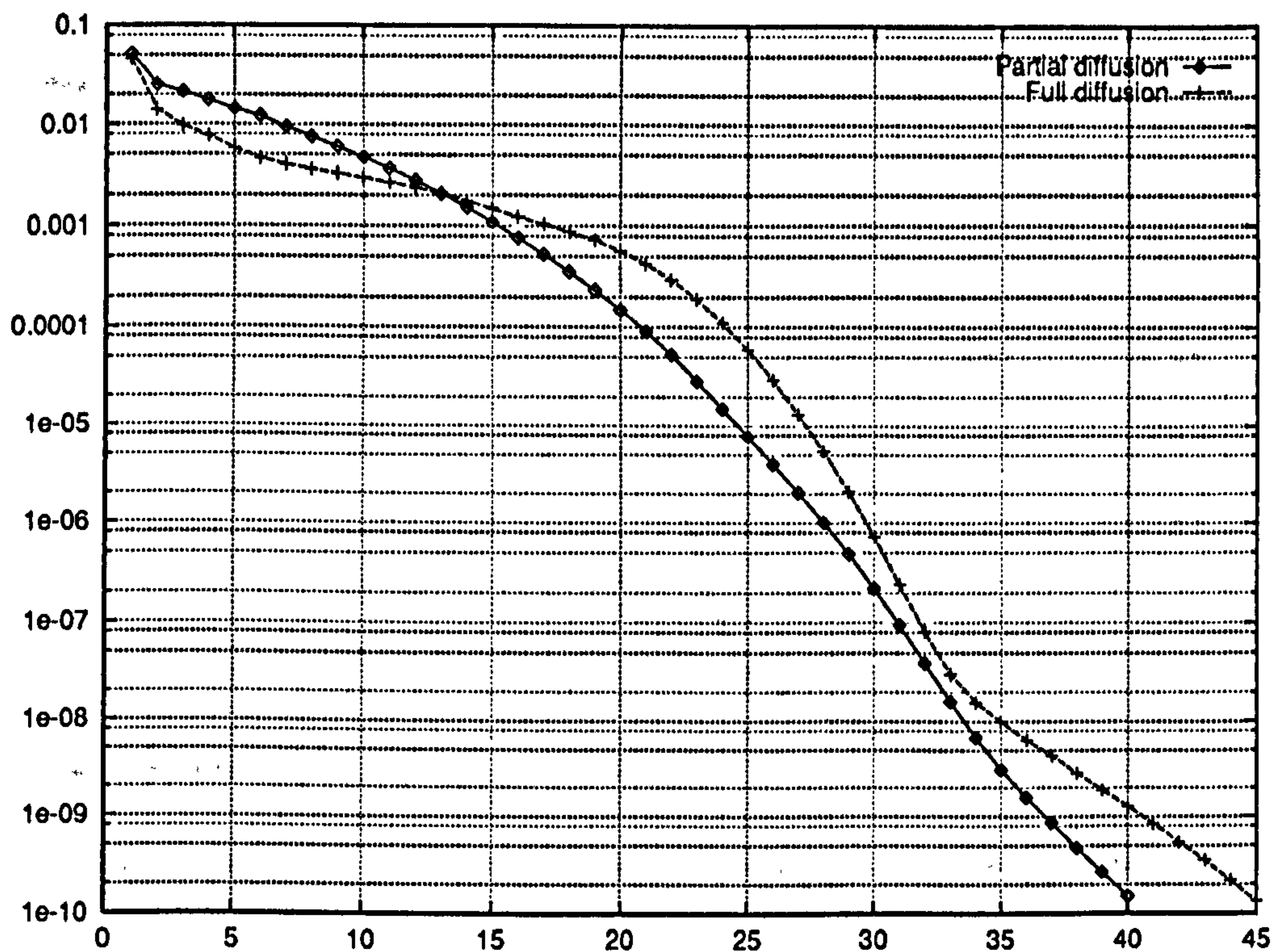


Figure 5.202: Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\tau \mathbf{T})$ on a level 4 grid

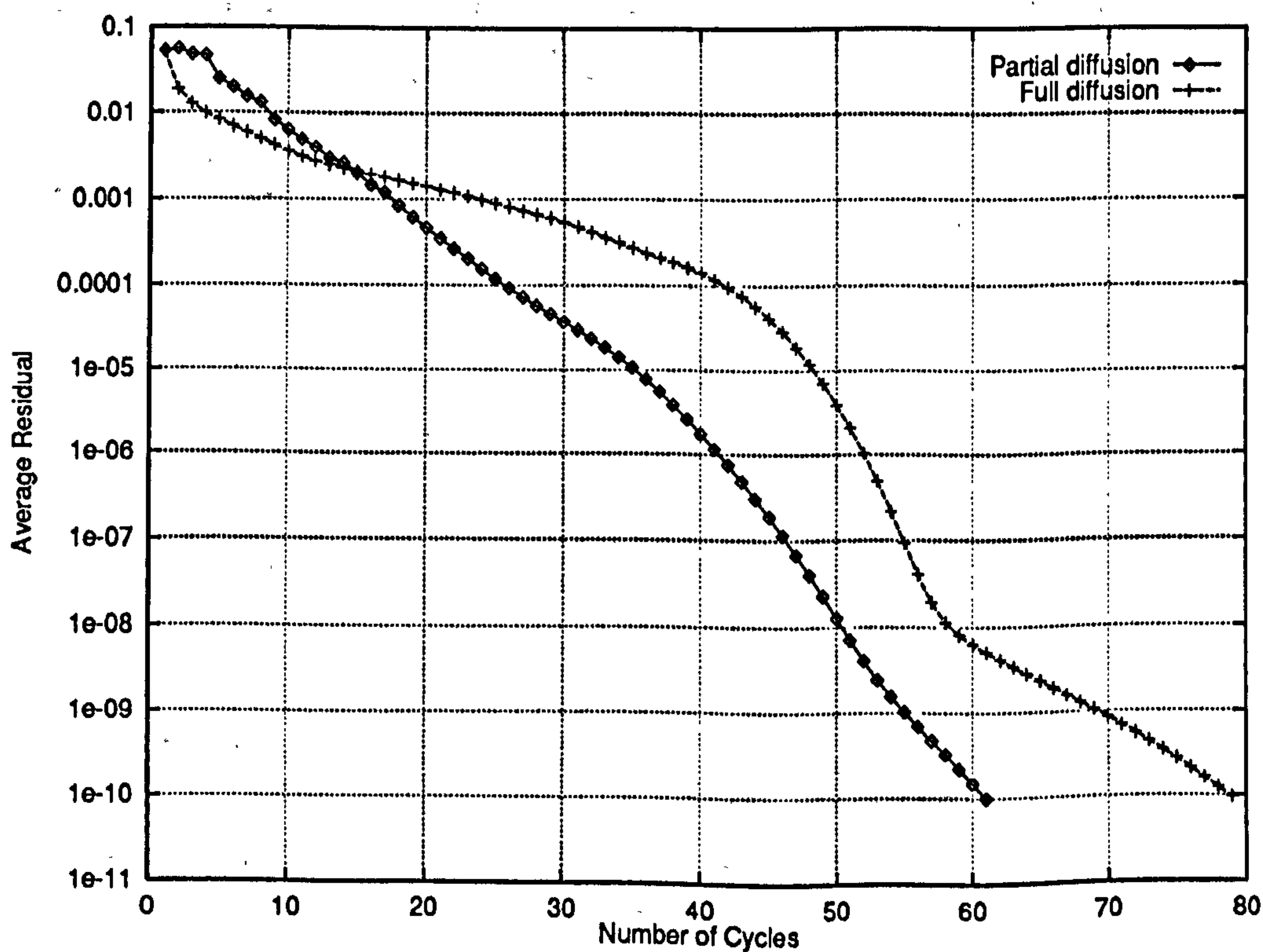


Figure 5.203: Two-phase channel flow – Comparison of the convergence histories for different formulations of the viscous shear stress tensor flux $\nabla \cdot (\tau \mathbf{T})$ on a level 5 grid

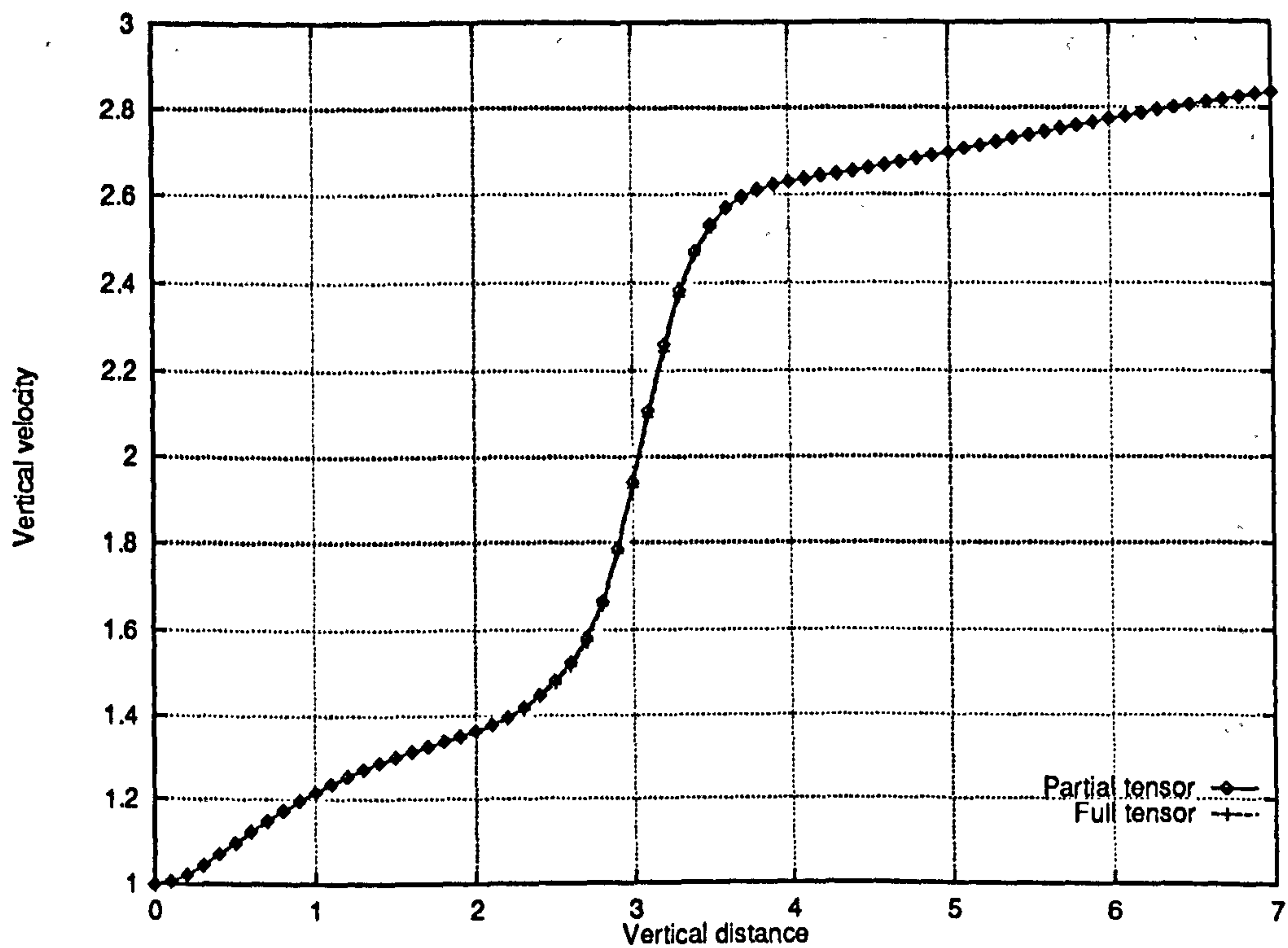


Figure 5.204: Multiphase T-junction problem – Comparison of horizontal velocity profiles along the line $y = 0.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid

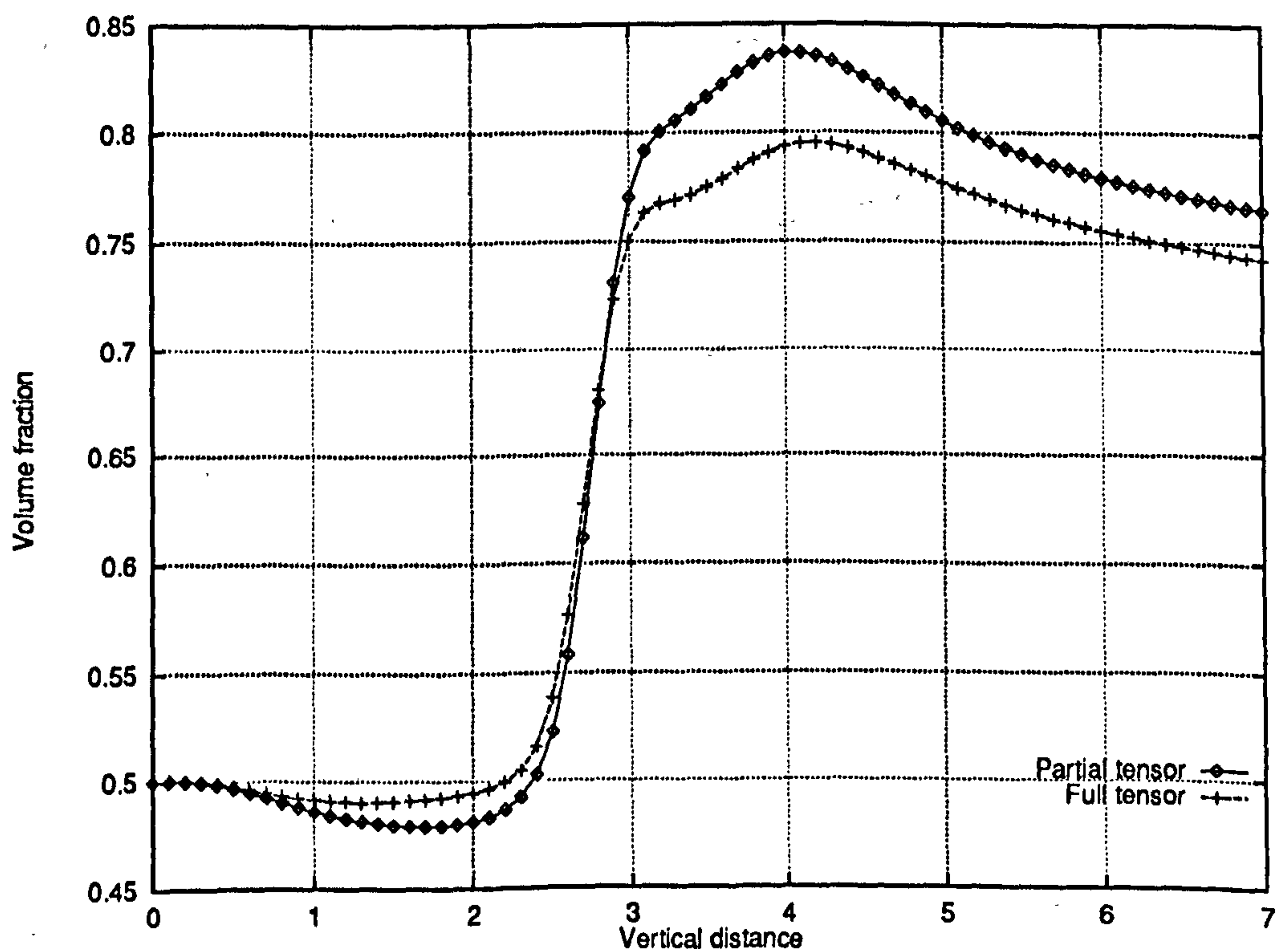


Figure 5.205: Multiphase T-junction problem – Comparison of volume fraction profiles along the line $y = 0.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid

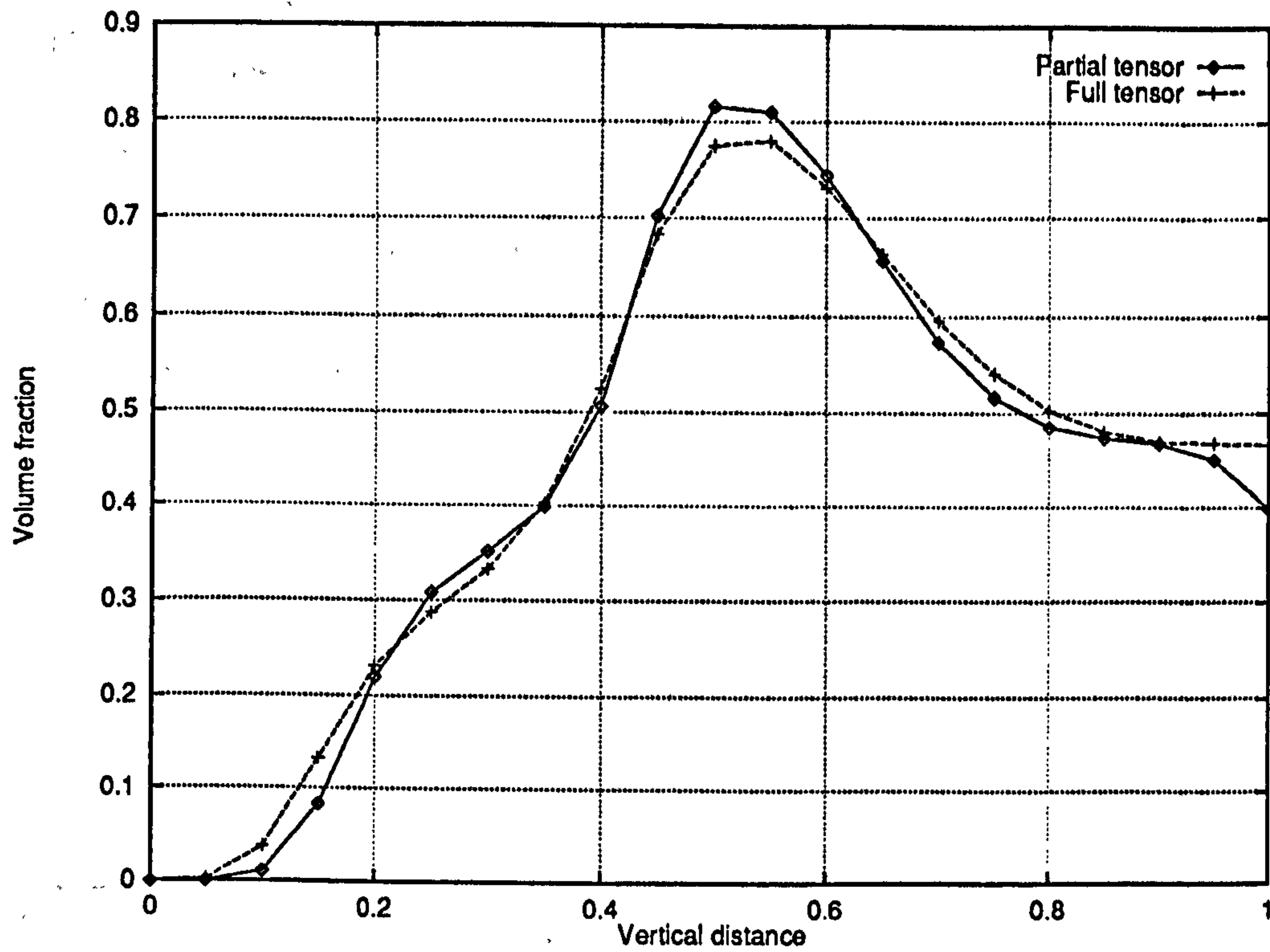


Figure 5.206: Multiphase T-junction problem – Comparison of volume fraction profiles along the line $x = 3.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid

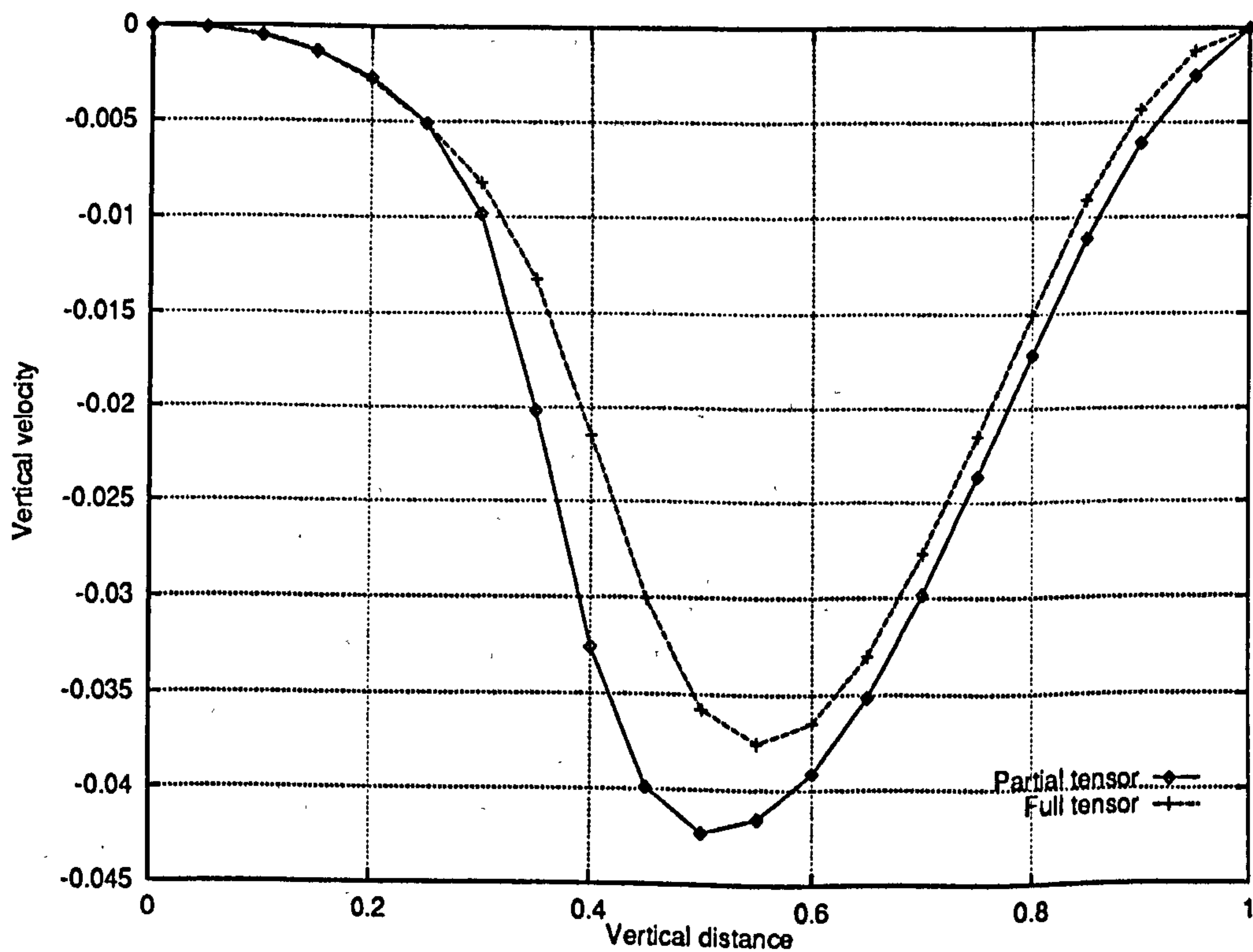


Figure 5.207: Multiphase T-junction problem – Comparison of vertical velocity profiles along the line $x = 6.5$ for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ on a level 4 grid

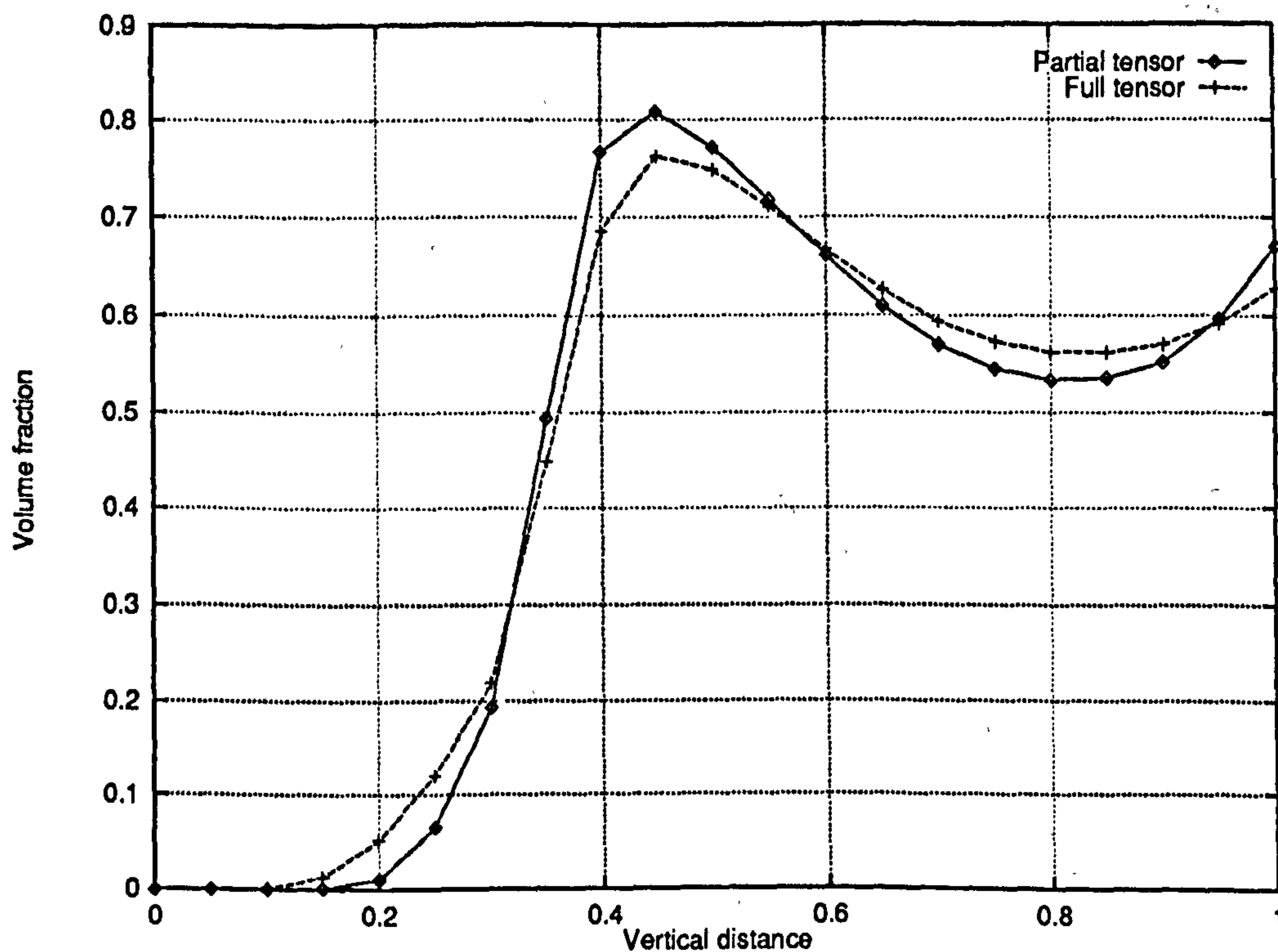


Figure 5.208: Multiphase T-junction problem – Comparison of the volume fraction profiles along the line $x = 6.5$ for different formulations of the stress tensor flux $\nabla \cdot (rT)$ on a level 4 grid

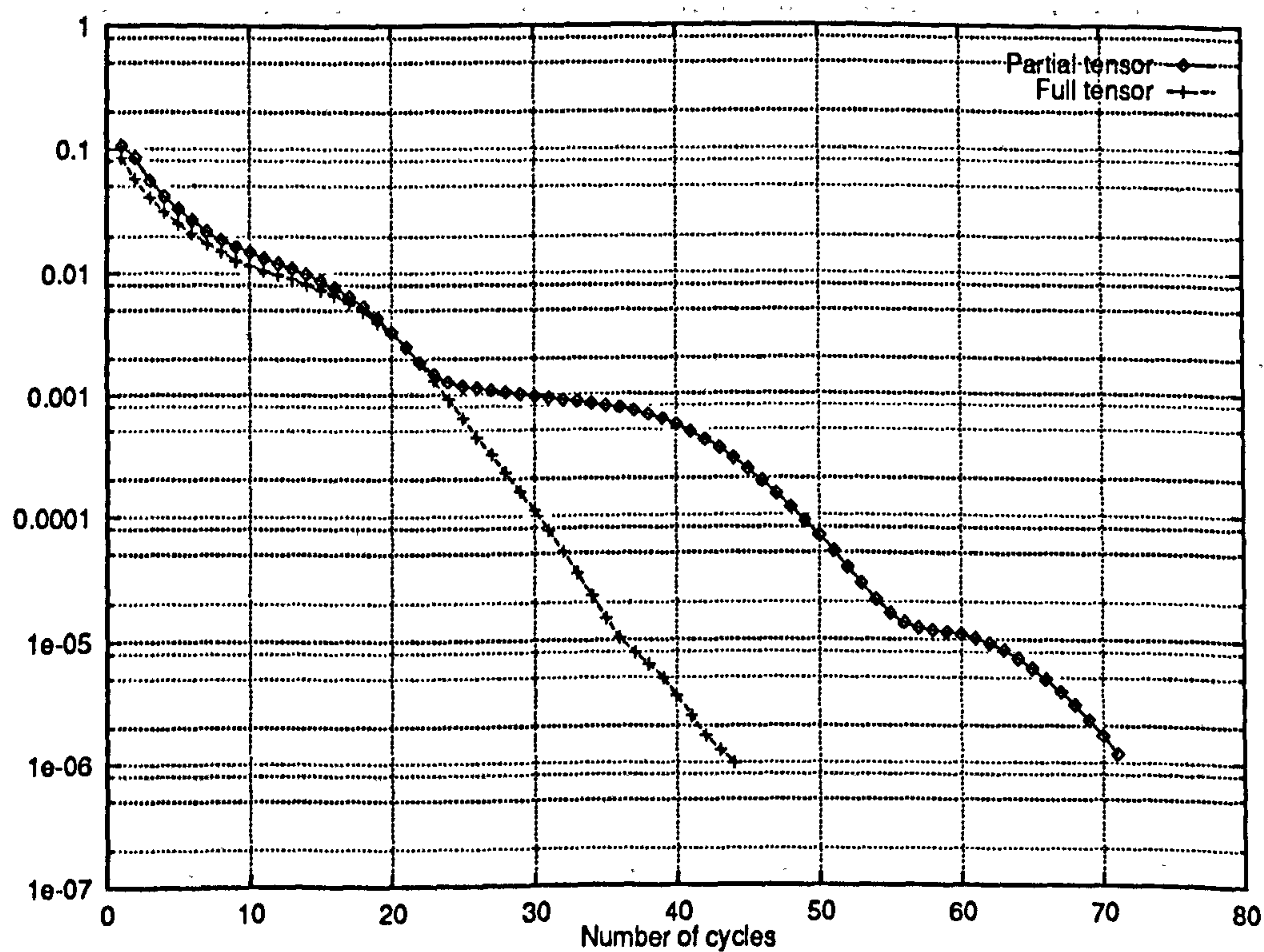


Figure 5.209: Multiphase T-junction problem – Convergence history for different formulations of the stress tensor flux $\nabla \cdot (rT)$ on a level 4 grid, showing the stabilising effect of the extra diffusive terms

5.4.14 More on the Viscous Tensor

In Section 5.4.13, it has been shown that in the case of a two phase flow, cross derivative terms can have a very detrimental effect on convergence factors and grid independence. In this section, the aim is to illustrate the fact that the same terms also affect convergence factors for single phase flows.

Firstly, a pseudo-two-phase entry flow is considered on the computational domain shown in Figures 5.1 and 5.3. The two phases have the same physical properties:

Reynolds Number	Viscosity	Density
$Re_1 = 100$	$\mu_1 = 0.01$	$\rho_1 = 1.0$
$Re_2 = 100$	$\mu_2 = 0.01$	$\rho_2 = 1.0$

The same boundary conditions are applied for each phase (see Section 4.3.4). The initial guesses are as follows: $u_{i+1/2,j}^1 = 0$, $u_{i+1/2,j}^2 = 1$, $v_{i,j+1/2}^1 = v_{i,j+1/2}^2 = 0$, $p_{i,j}^1 = p_{i,j}^2 = 0$, $r_{i,j}^1 = r_{i,j}^2 = 0.5$. Appendix B gives solution profiles for different grids.

For this problem, the solution field for the volume fractions are uniform:

$$r_1(x, y) = r_2(x, y) = \frac{1}{2}.$$

Hence, the extra-diffusive terms cancel out by virtue of the continuity equation and *when the iteration has converged*, the partial and full formulations of the stress tensor flux are equivalent to each other. In the context of true single phase computations, where the volume fractions are fixed to the value 1, the partial formulation is referred to as the simplified formulation. This is to emphasise the fact that it results from an analytical simplification of the partial differential equations.

For pseudo-multiphase flows, the two formulations are not *numerically* equivalent. Since the initial guesses for each phase are different, the computation of the solution actually involves non-zero corrections on the volume fraction field. Therefore, the terms (5.1) are not identically zero during the course of the computations. It is verified (Figures 5.210 to 5.212) (a) that the solutions are very close to each other and (b) that they are consistent with the single phase flow solutions. The fact that the differential equations are equivalent does not imply that the discrete equations are. Indeed, very small differences in the solution flow may be observed (see Table 5.16). The proximity of the solutions seems to indicate that the extra terms in the viscous tensor have been correctly implemented.

Here, it is the convergence factors which are of interest. Figure 5.213 clearly shows that the adoption of the partial stress tensor results in faster asymptotic convergence factors (see Table 5.17). In particular, it is clear that with the simplified formulation

System of equations	Maximum vertical velocity
Single phase (Simplified tensor)	0.165365
Multi phase (Full tensor)	0.165365
Multi phase (Partial tensor)	0.164197

Table 5.16: Comparison of the maximum modulus of the vertical velocity for phase 1 along the line $x = 0.15$ on level 4 grids, for different system of equations

System of equations	Asymptotic convergence factor
Single phase (simplified tensor)	0.8369
Multi-phase (Full tensor)	0.8501
Multi-phase (Partial tensor)	0.8369

Table 5.17: Asymptotic convergence factor for the single phase entry flow problem on level 4 grids – Comparison of different system of equations.

of the stress tensor flux, the multigrid convergence factors do not deteriorate as the residuals get smaller.

Secondly, a pseudo-multiphase (parabolic) flow through a channel is considered. Both phases are assumed to have the same physical properties (see Section 5.2.1) but here again different initial guesses are set. Convergence histories for different grid levels and for the partial form of the stress tensor are shown in Figure 5.214. This should be compared with Figure 5.215 which refers to the full formulation of the diffusive flux. For fine grids, the simplified flux leads to improved and more grid-independent convergence factors, as was the case for the pseudo two-phase entry flow problem. Table 5.18 gives the asymptotic convergence factors:

Grid level	Full tensor	Partial tensor
2	0.8977	0.9078
3	0.8818	0.8818
4	0.8554	0.8452
5	0.8603	0.8416

Table 5.18: Pseudo-two-phase (parabolic) channel flow problem – Comparison of asymptotic convergence factors for different formulations of the stress tensor flux $\nabla \cdot (r\mathbf{T})$

This experiment also establishes the fact that the effects of cross derivative terms are not specific to an entry flow but rather seem to be a general phenomenon.

Finally, a true single phase channel flow is considered. Figure 5.130 and 5.216 show the convergence history on different grid levels for the full and partial diffusive fluxes respectively. Asymptotic convergence factors are improved if the simplified formulation is used (see Table 5.19) and the benefit is more significant for fine grids.

Since the degradation of the convergence factors for finer grids is also observed for true single phase flows (for which the magnitude of the terms (5.1) should always be small because the volume fraction is not allowed to change), it is likely that the terms associated with the discretisation of (5.1) in the discrete Jacobians play at least a secondary role in the degradation of the multigrid convergence factors.

Grid level	Full tensor	Simplified tensor
3	0.8817	0.8771
5	0.8603	0.8189

Table 5.19: Single phase channel flow problem – Comparison of asymptotic convergence factors for different formulations of the diffusive flux

The single phase results were obtained with the automatic differentiation Jacobians and the Newton corrections were under-relaxed by taking $\lambda_r = 0.6$. If the automatic differentiation Jacobian is used, under-relaxation is necessary on fine grids to obtain reasonably grid independent convergence factors. This was discussed in Section 5.4.4. In Section 5.4.6, it is also shown that under-relaxation is needed for multiphase flows. It has been verified that the choice of the partial formulation for the stress tensor flux over the full formulation does not remove the need for under-relaxing the Newton correction if the automatic Jacobian is used. The need to add under-relaxation on fine grids arises from another reason.

In Section 5.4.13, we have established that the cross derivative terms which appear in the multiphase diffusive flux are responsible for the degradation of multigrid convergence factors. They are primarily associated with spatial variations of the volume fractions. By considering the case of pseudo multiphase and true single phase flows, in this section, it was possible to prove that the cross derivative terms further affect the convergence factors because the corresponding terms in the Jacobians affect the quality of the relaxation provided by the quasi-Newton solver. These terms are responsible for some of the discrepancies in convergence recorded between `pamg` and `pamg-multiphase`. Furthermore, the results presented in this section confirm that the performance of the quasi-Newton solver is strongly dependent on the detail of the approximation of the Jacobians.

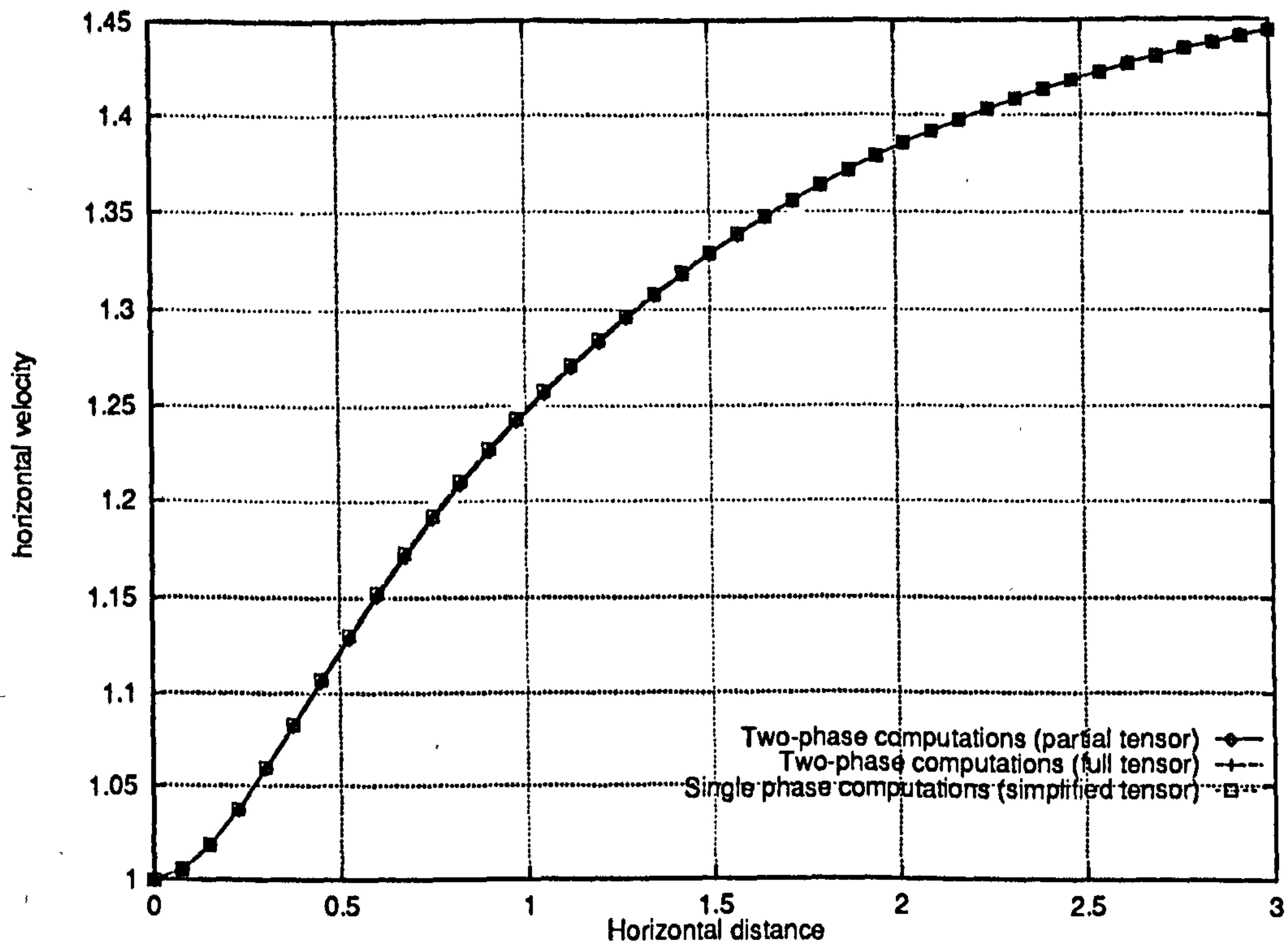


Figure 5.210: Pseudo two-phase entry flow – Horizontal velocity along $y = 0.5$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$

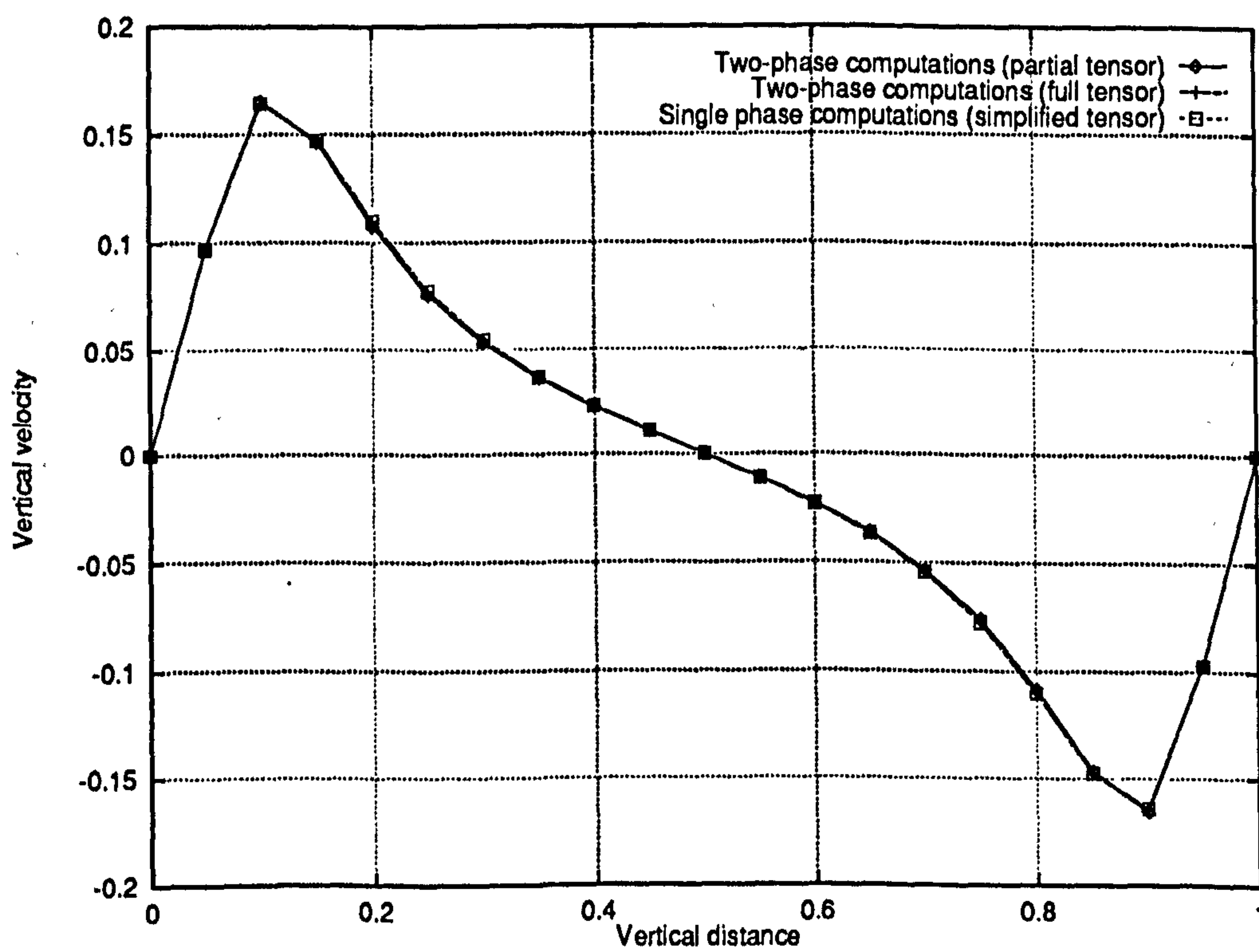


Figure 5.211: Pseudo two-phase entry flow – Horizontal velocity along $x = 0.15$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$

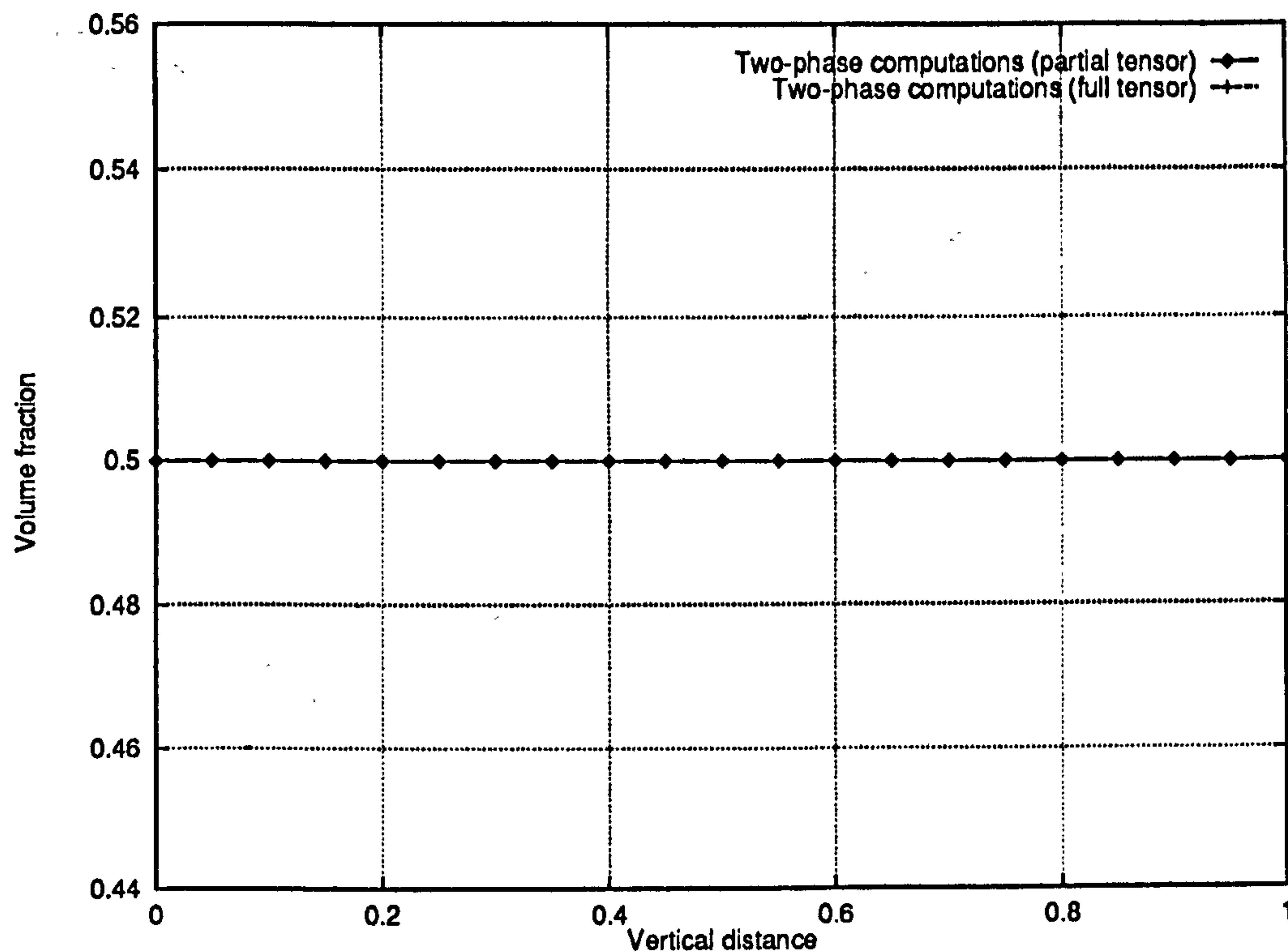


Figure 5.212: Pseudo two-phase entry flow – Horizontal velocity along $x = 0.15$ – Comparison of several formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$

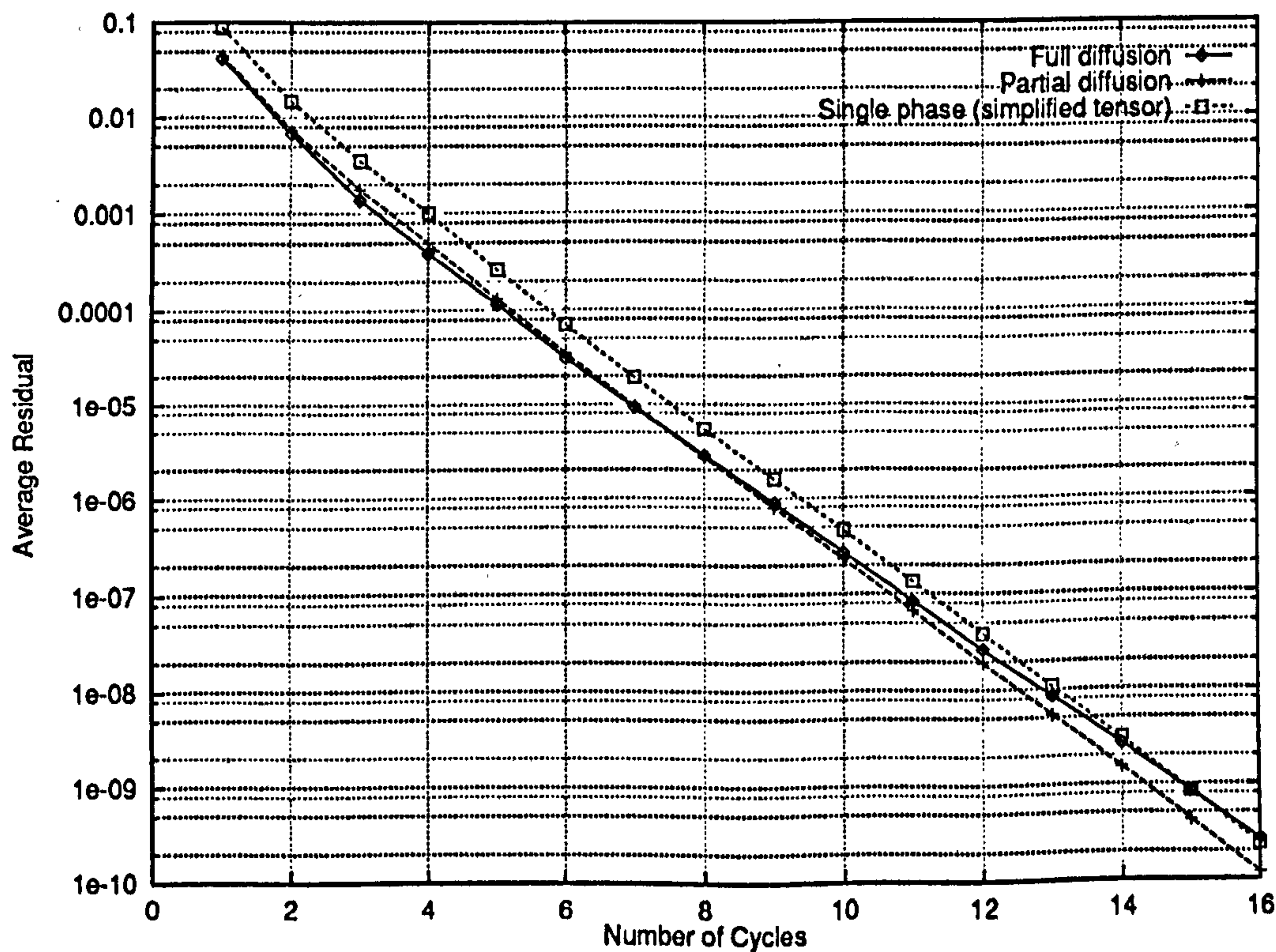


Figure 5.213: Pseudo two-phase entry flow – Convergence histories on a level 4 grid for different formulations for the stress tensor flux $\nabla \cdot (r\mathbf{T})$

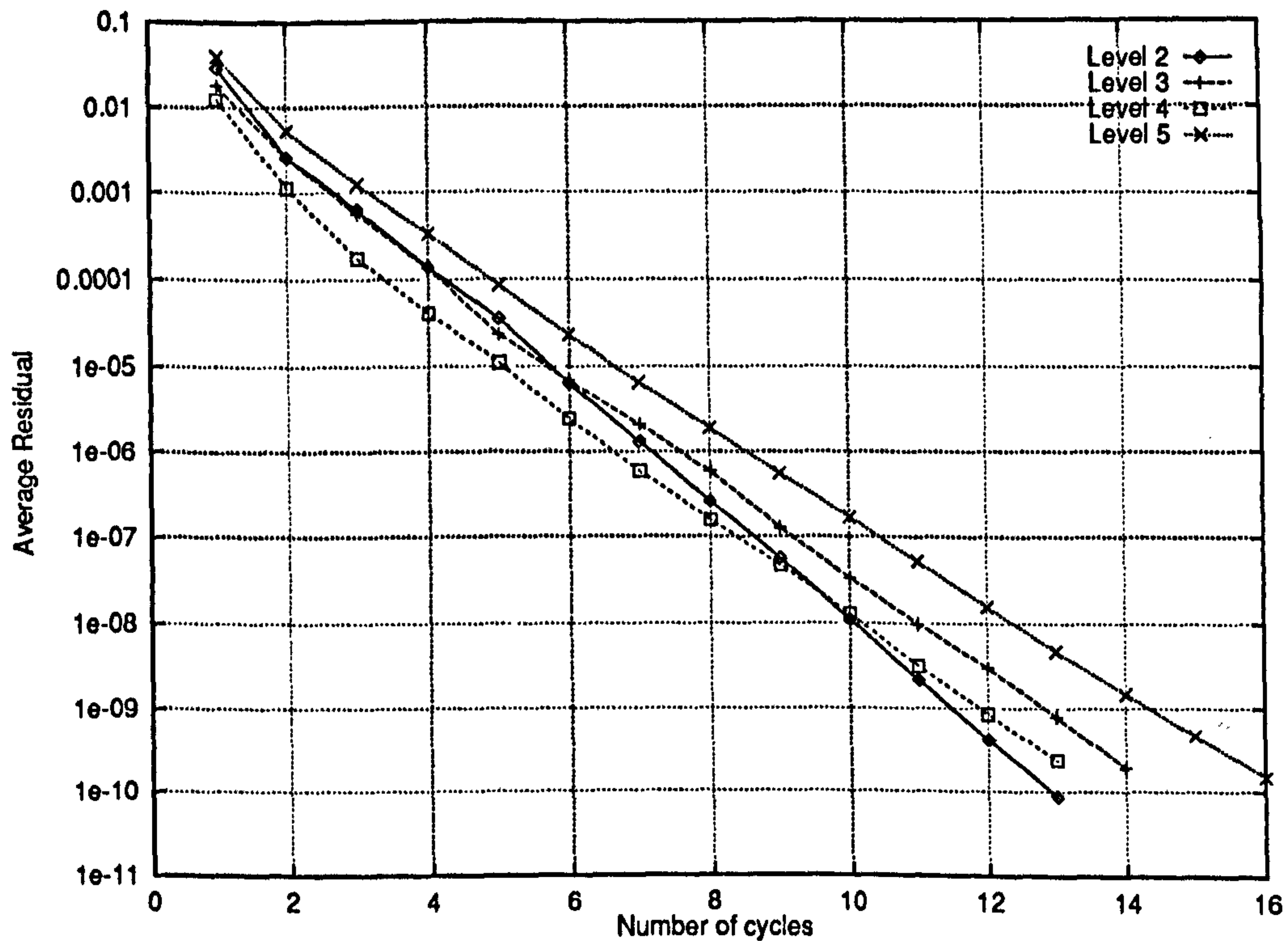


Figure 5.214: Pseudo two-phase channel flow with the partial formulation of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Convergence history for grid levels 2, 3, 4 and 5

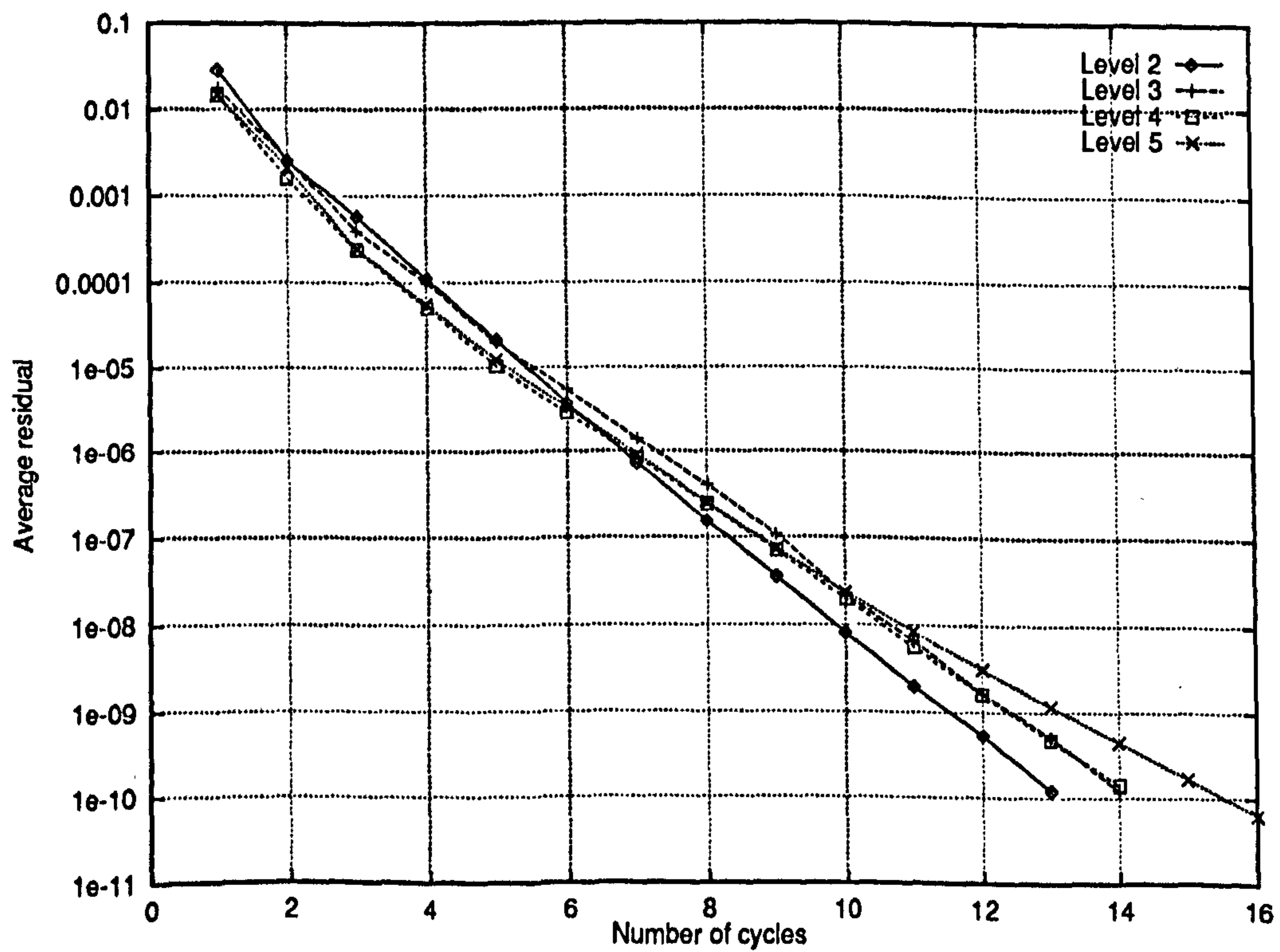


Figure 5.215: Pseudo two-phase phase channel flow with the full formulation of the stress tensor flux $\nabla \cdot (r\mathbf{T})$ – Convergence history for grid levels 2, 3, 4 and 5

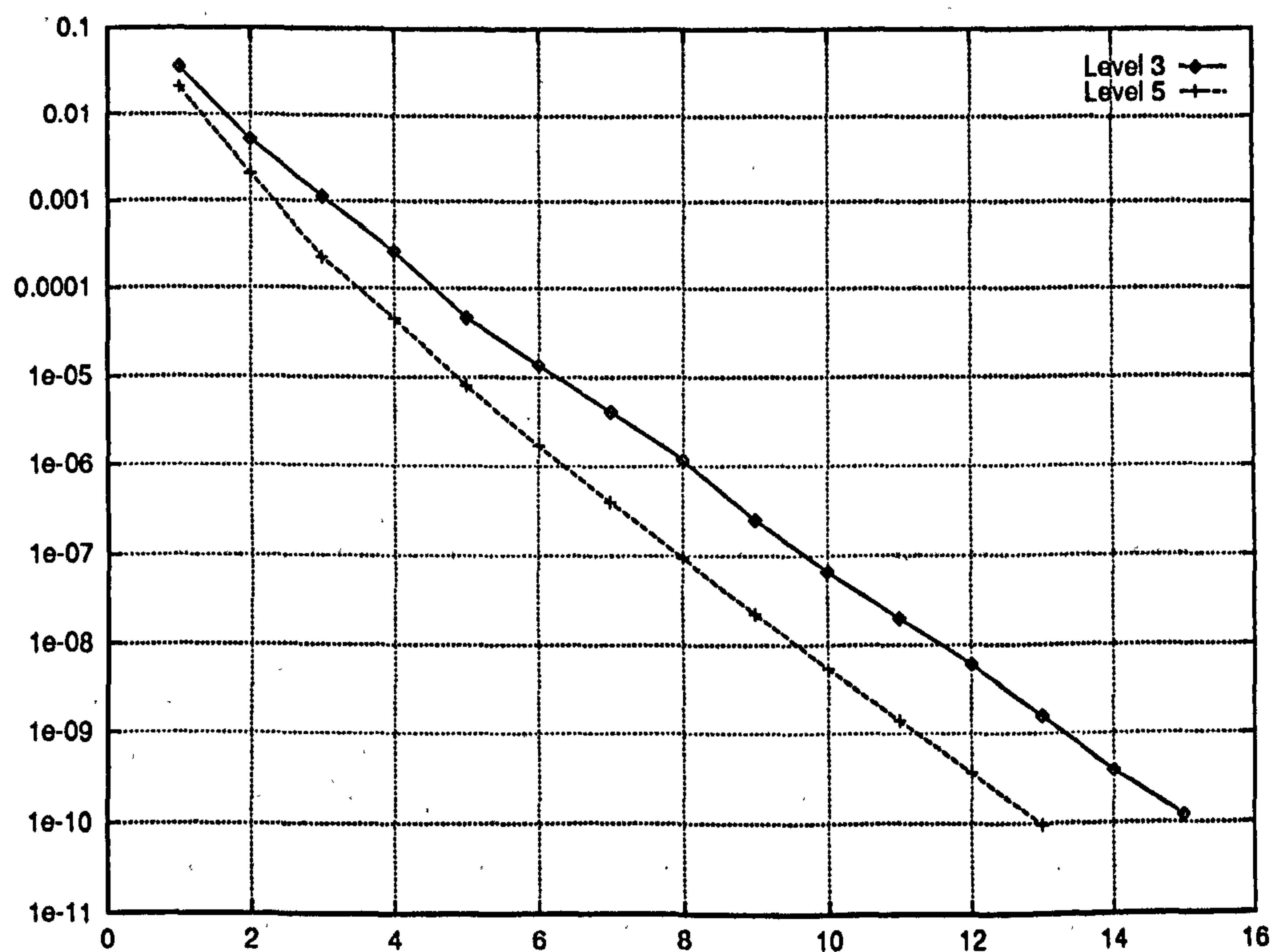


Figure 5.216: Single phase pipe flow with simplified stress tensor – Convergence history for grid levels 3 and 5

5.5 Conclusions

In this chapter, we have investigated the accuracy and efficiency of the `pang-multiphase` solution algorithm, which is based on a local quasi-Newton coupled multigrid approach. Although our solver builds on the single-phase code `pang`, it incorporates a large number of novel elements, as described in Chapter Four, which are necessary to successfully handle the increased mathematical complexity of the governing equations for multiphase flows. The main algorithmic developments are as follows:

- the use of upwind interpolation in the continuity equations;
- the use of automatic differentiation to obtain accurate expressions for the Jacobians matrices which are required by our local quasi-Newton single grid solver;
- the use of line-searching to remedy the non-linearities and ensure that the solutions satisfy the constraints of realisability.

The main conclusions which can be drawn are as follows:

- The `pang-multiphase` implementation of the discretised equations 3.89 to 3.92 derived in Section 3.3 has been validated by:
 1. Comparing the solutions for a range of important test cases with those provided by a commercial CFD code, `CFX 4.1`. `CFX 4.1` solves the same differential equations, but on a different type of grid and with a different, segregated, solution algorithm. Where applicable, the solutions have also been compared with those provided by the single phase solver `pang`.
 2. Establishing the degree of grid independence of the `pang-multiphase` solutions.

Both tests have given essentially positive results so that the `pang-multiphase` implementation can be considered to be correct.

- The quasi-Newton solver approach in general yields a good quality smoother, although line-searching and under relaxation on fine grids are sometimes necessary. This is not surprising since (i) the system of equations is strongly non-linear and (ii) line-searching allows the elimination of physically non-realizable solutions. We have also established that the performance of the solver is strongly dependent on the quality of the approximation for the local Jacobians. In the case of multiphase flows, very accurate expressions are needed to handle the strong non-linear coupling between the flow variables. In this respect, the use of automatic differentiation has been invaluable, as it has greatly simplified the development of the software.

- A FAS multigrid method built around the quasi-Newton solver coupled solver can be successfully applied to complex multiphase flows, such as the flow through a T-junction. Very good speed-ups have been observed when compared with the equivalent single grid computations. In terms of the number of fine grid iterations, it was also shown that *pamg-multiphase* can perform significantly better than a commercial CFD code based on the IPSA segregated solver.
- Complex multiphase flows can be efficiently simulated using a quasi-Newton coupled multigrid approach. In particular, the ability of the method to handle “dry-out” regions where a phase is almost absent has been demonstrated, at least for coarse grids. This is despite the fact that when the volume fraction for one phase approaches zero, the local Jacobians become badly conditioned since the volume fraction appears as a factor in most entries. As grids become finer, computations on adaptive grids show that the potential for generating non-physical corrections of the volume fraction in such “dry-out” zones increases. It is therefore important to implement tests which ensure that the volume fraction corrections are always physical, but convergence rates can be affected to a small extent. Furthermore, the bad-conditioning of the local Jacobian worsens on finer grids. One solution may be to reformulate the governing equations.
- Recirculation zones are another difficult flow pattern. There are indications that in certain cases, such problems are not well-posed in the steady case. When only one phase is recirculating, the steady problem is well-posed and the multigrid solver is successful. When both phases are recirculating in the same region, the problem is mathematically ill-posed and the multigrid solver tends to diverge. By contrast, computations are still possible with the single grid solver.

This suggests that the divergence of the multigrid method does indeed have a mathematical origin which affects the multigrid transfers. This is consistent with our hypothesis that the problem is ill-posed: different solutions are approximated on different grids.

- We have observed very good multigrid acceleration but convergence factors are not always grid-independent, particularly for simpler problems where multiphase effects dominate the solution. Our multiphase solver, unlike the single phase *pamg* solver, is therefore not optimal order-wise. The consequences of this fact need to be put in perspective: multigrid procedures are still very efficient and provide a large degree of robustness. In contrast to segregated solvers, the convergence of the *pamg-multiphase* solver does not depend on the quality of the initial estimate.
- We have investigated in detail various hypotheses to explain the dependence of the convergence factor on the grid size. The situation is quite complex (as would be expected from the nature of the operator); our main conclusions are:

1. The equations are highly non-linear. Although FAS is directly applicable to non-linear problems, the degree to which the multi-fluid equations are non-linear is important: the discrete operators can have different characteristics on different grids. The fact that the volume fraction fields are sensitive to the grid size is an indication that this is indeed the case. Hence, it will require time for the solutions on different grids to “adjust” to each other (via the defect). Moreover (and unusually) operators also need to adjust (through the solution estimate). In the single phase case, this happens quickly. It can be said with confidence that the non-optimality of the multigrid procedure reflects the difficulty of the problem. However, the non-linearities do not preclude the successful application of FAS.
2. The system of equations being solved contain terms which hinder the performance of multigrid solvers. The chief difficulties observed are linked with the diffusive fluxes $\nabla \cdot (r\mathbf{T})$ in the momentum equations (see Section 3.3.2). For the constitutive relationship used in this study, $\nabla \cdot (r\mathbf{T})$ contains cross-derivative terms which are non-zero if there are spatial variations of the volume fractions. These terms are “hyperbolic-like” and cannot be handled well by multigrid methods. We have given evidence to suggest that this might be a case of strong-alignment (in channel flows!). In problems where the dominant errors are due to the geometry of the flow, such as a two-phase flow through a T-junction, the effect of these terms remains largely hidden if the cell-Reynolds number is large enough and good grid independent (asymptotic) convergence factors have been observed. If the problem is simpler, such as the channel flow, the cross derivatives are more significant and have been shown to be responsible for degraded convergence factors both at large and small values of the residuals. When the cross derivative terms are ignored, grid independent asymptotic convergence rates are indeed observed. The grid dependent is then confined to higher residuals and can be attributed, with confidence, to the non-linearity of the equations.
3. A large body of evidence accumulated during this project indicates that the `pang-multiphase` solver is less efficient for low cell-Reynolds numbers. It was also observed that for very diffusive flows, convergence is harder to achieve. This does not appear to be confined to the multigrid algorithm as single grid computations also encounter difficulties. Although some hypotheses have been formulated, so far it has not been possible to isolate the root cause of this phenomenon.

Essentially, we conclude the `pang-multiphase` solver has been shown to be correct, robust and (intrinsically) fast. Convergence does not strongly depend on the quality of the initial guess and is many times faster than for equivalent single grid calculations. Approximate comparisons with segregated solvers also indicate that `pang-multiphase` is more efficient (at least in terms of the number of iterations required to achieve a certain accuracy). Furthermore, as the problem being solved

is very difficult and its solutions can have widely differing characteristics, fine tuning of certain elements in the algorithm is necessary for optimal performance.

We now focus our attention on adaptive computations which introduce further complications to the algorithms but allow large savings in computational cost, together with the possibility of controlling errors automatically.

... ..

Chapter 6

pamg-multiphase Computations on Adaptive Grids: Accuracy and Benefits

6.1 Introduction

In many flows, the solution is very strongly influenced by essentially local phenomena, which often require fine grids to be properly resolved. Uniform fine grids are computationally very expensive and an adaptive code, which *automatically* refines the computational grid in regions where it is sensible to do so (e.g. in regions of high truncation errors) is very beneficial in terms of performance as well as accuracy and robustness. The fact that grid refinement is automatic is a very important feature since it reduces the need for expertise and physical insight at the stage of grid generation.

Here again, the single phase solver pamg, which is our starting point has been proved to be a good adaptive code [12] and an important goal for this research was to incorporate similar capabilities for adaption in pamg-multiphase.

Refinement algorithms for both solvers are described in Sections 4.2.4 and 4.3.5. We note that our implementation of adaption is closely connected to the FAS multigrid procedure (see Section 2.5.8). This chapter presents the results of multiphase simulations on adaptive grids and assesses the benefits of adaption.

In the author's opinion, the question of error estimation has largely been settled by single-phase studies. A basic assumption was therefore that the defect computed by the FAS multigrid algorithm was a good refinement indicator for multiphase as well as single phase flows. This assumption is verified by the results obtained. The basic data structures developed for the single phase code pamg have also been retained.

Their extension to handle multiphase flows required careful thought (see Section 4.4) but did not pose any real difficulty in theory.

The main issue that needs to be faced when implementing adaption is that of the mass flux conservation across grid interfaces (see Section 2.6). The good performance recorded with *pamg* is partly due to an intelligent choice for the multigrid prolongation operator which conserves mass fluxes while being second order accurate (see Section 4.2.2). It is significantly harder, but not impossible, to design conservative interpolation procedures for multiphase flows and such a procedure, which again we believe to be original, is described in Section 4.3.3. It is based on explicitly transferring mass fluxes instead of the velocities.

First, we consider the accuracy of the refinement algorithm. Conservative and non-conservative interpolations are compared. It is concluded that:

- Non-conservative grid transfers do not necessarily prevent convergence;
- The errors associated with the non-conservation of mass are usually small;
- Non-conservative grid transfers are preferable to the conservative alternatives since the conditioning of the latter is worse in “dry-out” regions.

Secondly, results are given for adaptive computations for both the two-phase channel flow and two-phase T-junction problems, using the non-conservative transfers, and lead to the conclusion that the refinement algorithm is:

- *Accurate*: the solutions are close to those obtained on uniform grids
- *Robust*: the pattern of grid refinement is correct in the sense that refinement is never performed where it would be inaccurate

Finally, we assess the saving in terms of computational costs which adaption allows compared with computations on uniform grids. The two criteria considered are traditional: computing time and memory. Not surprisingly, the benefits of adaption are shown to be very significant indeed.

6.2 Adaptive Computation of the Two-Phase Channel Flow Problem

The two phase channel flow problem described in Section 5.3.1 is considered again but we now wish to obtain a solution on composite instead of uniform grids, using the refinement algorithm described in Section 4.2.4.

We use this problem as our model problem for analysing the behaviour of the refinement algorithm. Results for the more complex T-junction problem are then given to confirm the conclusions established for the channel flow problem. Firstly, the merits of transfer procedures which conserve mass across grid boundaries are ascertained with respect to those which do not¹. Comparisons are made on uniform as well as composite grids.

Some results on composite grids with an increasing number of levels are then presented. We illustrate the well known fact that the initial uniform grid should be fine enough for the features of the flow to be visible. We then show that provided this last condition is met, computations on adaptive grids are as accurate as the equivalent computations on uniform grids.

6.2.1 Effect of the Different Transfer Procedures on Uniform Grids

The transfer procedures should not affect the solution on uniform grids and this has been verified. See Figures 6.1 and 6.2 for an illustration of identical solutions on a level 3 grid.

However, convergence rates may be affected because the transfer operators introduce interpolation errors which need to be eliminated by the relaxation procedure. It has been consistently observed that the non-conservative formulation provides slightly better convergence rates than the conservative procedure on uniform grids (Figure 6.3 gives an example).

¹The different options for transfer operators between grids are described in Section 4.3.3 and the associated formulae can be found there.

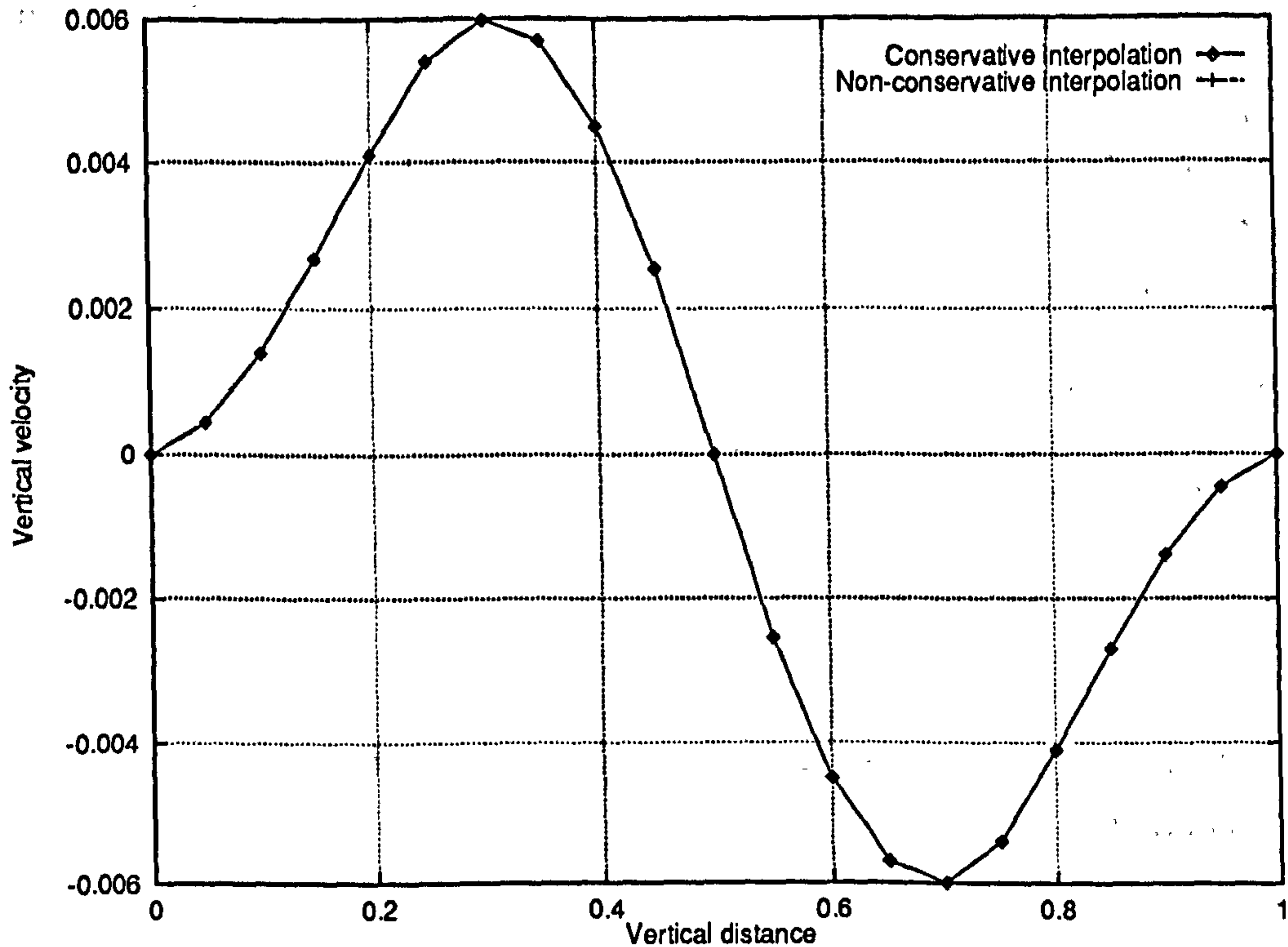


Figure 6.1: Multiphase channel flow problem – Comparison of the vertical velocity profiles (for phase 1) along the line $x = 2.5$ obtained on a level 3 uniform grid, for different transfer operators between grids

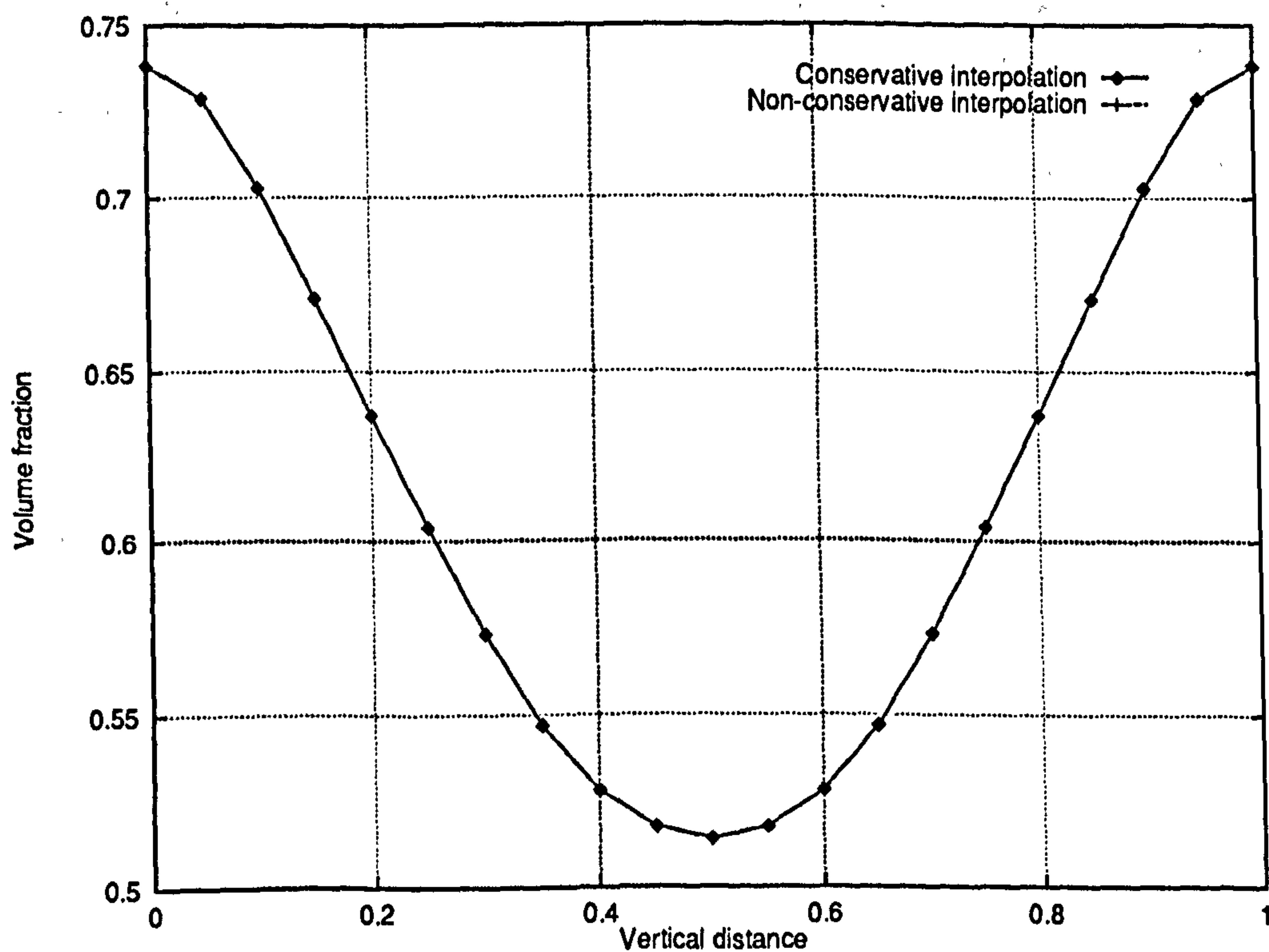


Figure 6.2: Multiphase channel flow problem – Comparison of the volume fraction profiles (for phase 1) along the line $x = 2.5$ obtained on a level 3 uniform grid, for different transfer operators between grids, showing that the fine grid solution is independent from the transfer operators

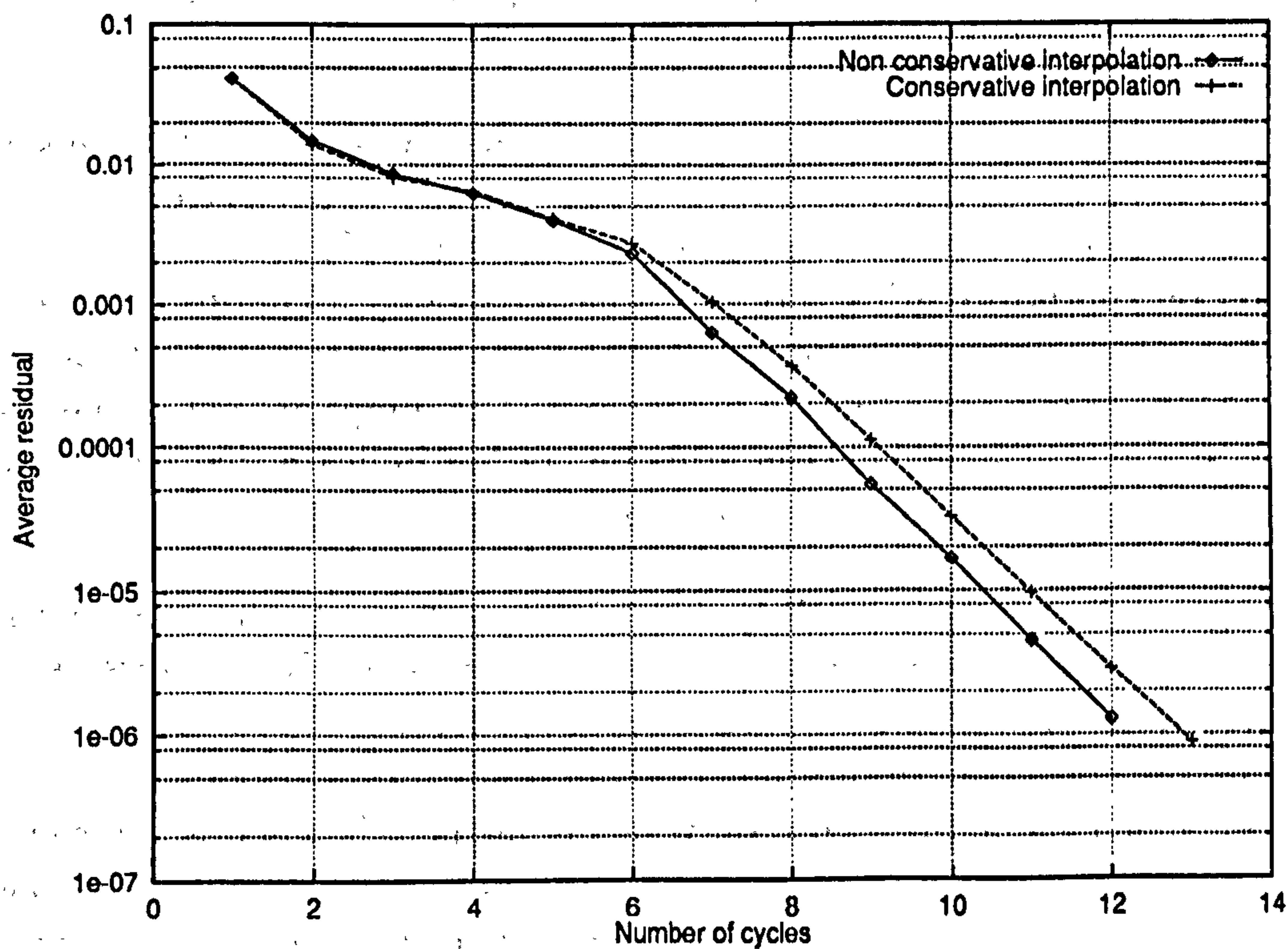


Figure 6.3: Multiphase channel flow problem – Comparison of the convergence rates of computations on a level 3 uniform grid for different transfer operators between grids

6.2.2 Adaptive A(2,3) Results

In this section, some results obtained for an A(2,3) computation² of the multiphase channel flow problem are presented. The goal is mainly to illustrate the point that adaption is not fool-proof and requires a reasonably representative approximation of the solution on a uniform grid in order to proceed efficiently. This is not new but still deserves to be borne in mind.

In this test case, the solution on the uniform grid at level 2 is refined locally to level 3. Figures 6.49 and 6.50 show the adaptive grids produced by the refinement algorithm for the two different sets of transfer operators, conservative and non-conservative. The solution obtained on these composite grids exhibits jumps at *some*, but not all, interfaces between grids (see Figures 6.4 and 6.5).

This phenomenon does not come as a surprise since in Section 5.3.1, we saw that for uniform grids, the solution at level 2 is significantly different from the solution at level 3. In regions where the solution is more grid independent *at grid interfaces*, the solution on composite grids is of much better quality as jumps are not readily distinguishable (see Figures 6.7 and 6.8) It is also important to note that conservative transfers do not change the situation in any way. The jumps are not connected with inaccuracy at the interfaces between grids but rather indicate that grid refinement is performed at the wrong places, i.e. places where the solution is changing rapidly.

The convergence histories for both types of interpolation are shown in Figure 6.9. It can be seen that enforcing conservation at the grids can have a dramatic effect on the convergence rates. Some evidence suggests that the slower smoothing factor observed for the conservative interpolation is due to the behaviour of the solution near jumps. When the uniform grid on which refinement is based is sufficiently fine for jumps in the solution to be eliminated, a second (slower) convergence regime is not present. Observation of the solution after 5 and 9 cycles, i.e. just before the establishment of the second slower convergence regime, shows that the jumps are present in the solutions (see Figure 6.6). Their effects on the convergence history are obviously masked by some other error terms. Another possible explanation is that slow convergence is associated with the volume fraction “kink” near the inlet which appears to develop more suddenly.

Furthermore, Figure 6.9 examined in connection with the grid refinement patterns (Figures 6.49 and 6.50) reveals that the plateau phase discussed in Chapter Five in the context of uniform grids, is clearly associated with the regions which are close to solid walls *and* where the flow is significantly separated, i.e. in the half of the

²In this chapter, the following notation will be adopted for an easy specification of computations:

- The letter “A” stands for adaptive computations.
- The first number denotes the level of the finest *uniform* grid.
- The second number denotes the final degree of refinement.

channel which is closest to the outlet.

We conclude by noting that in contrast to single phase flow computations, the fact that the interpolation operators do not conserve mass between grids does not prevent the solution algorithm from converging. Section 6.4 examines this question in detail.

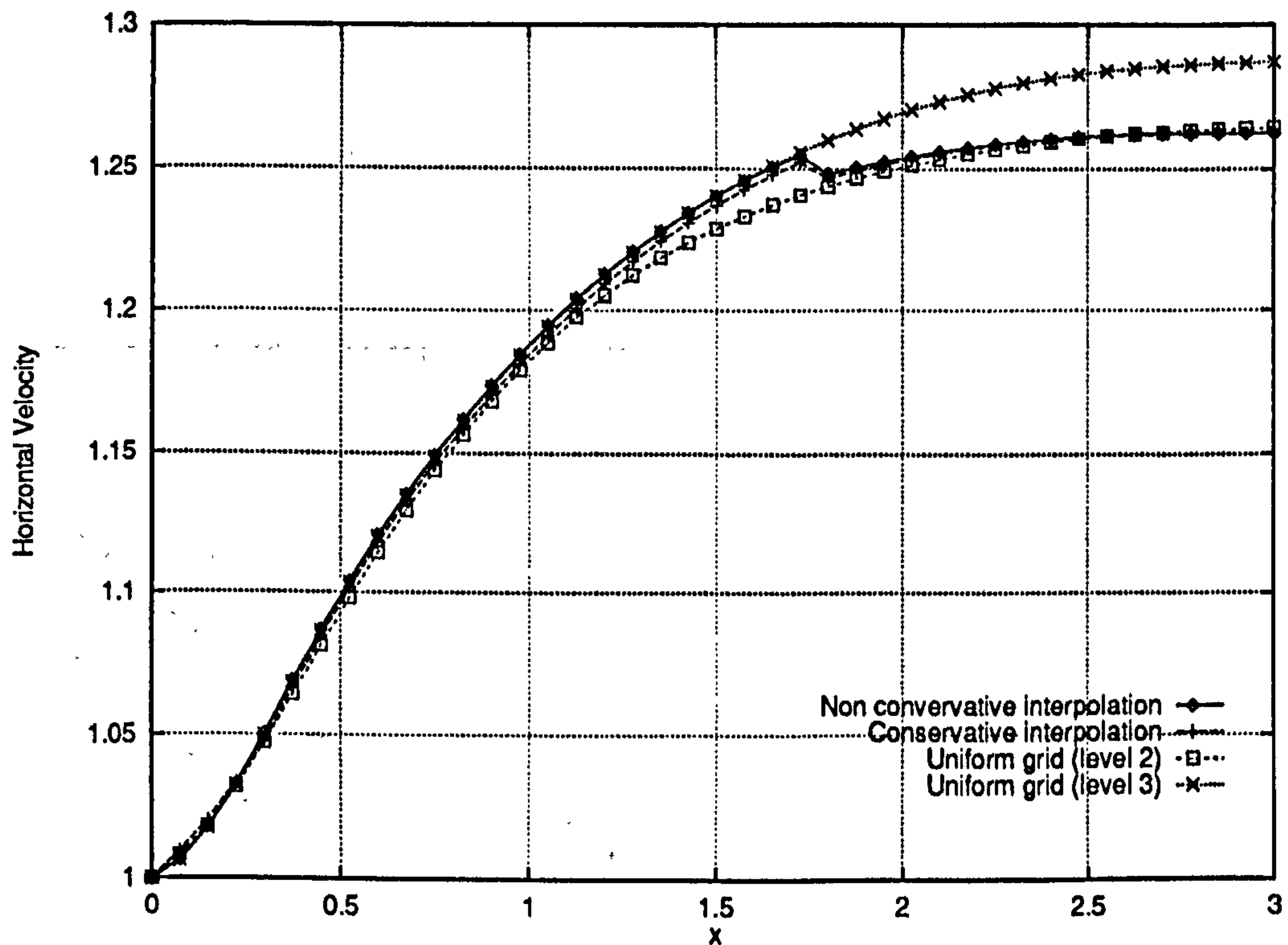


Figure 6.4: Multiphase channel flow problem – Comparison of the horizontal velocity profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids

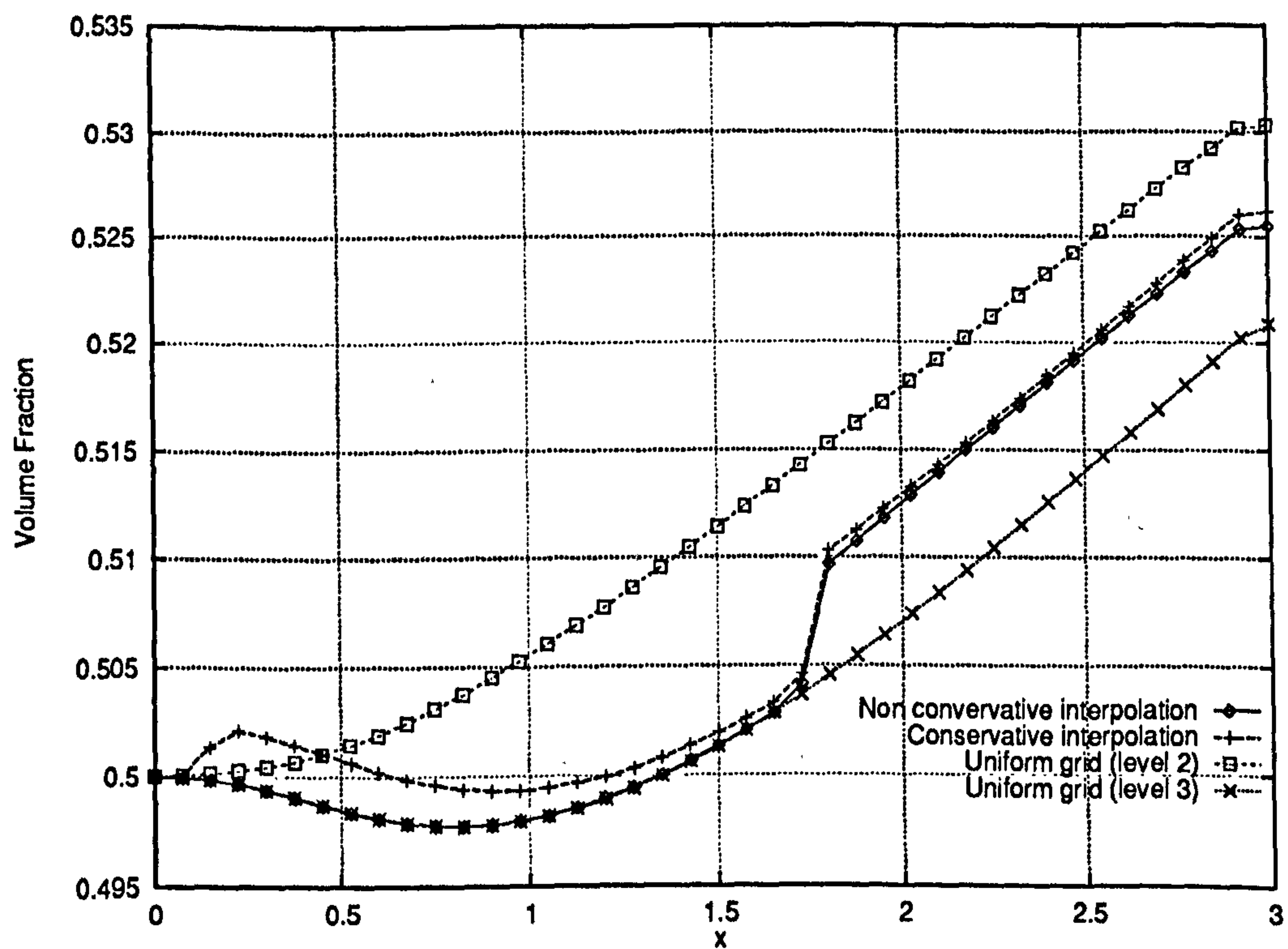


Figure 6.5: Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids showing the error caused by misplaced grid interfaces

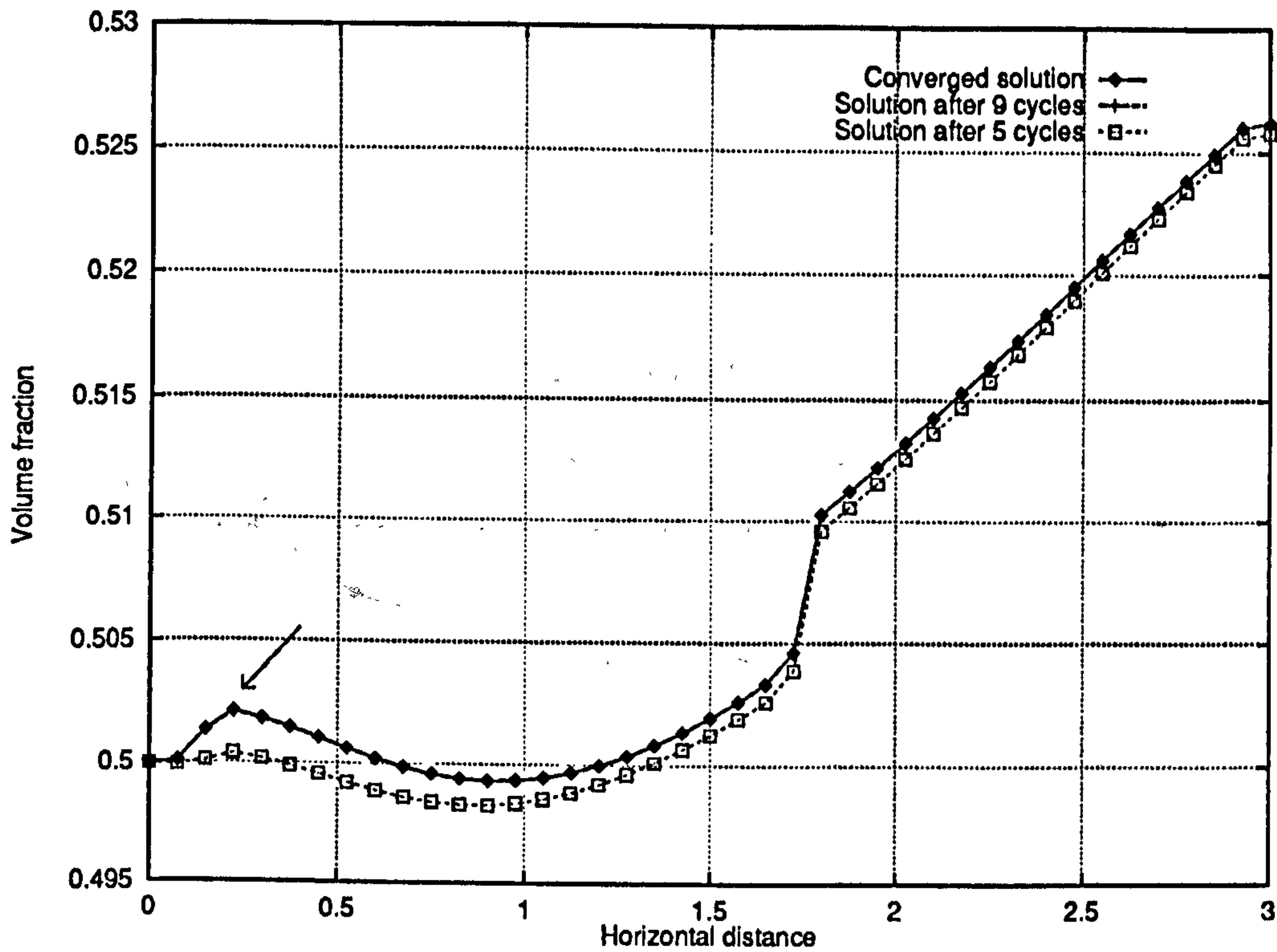


Figure 6.6: Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $y = 0.5$ obtained on a (2,3) adaptive grid, for different transfer operators between grids at different stages of the solution algorithm, showing that the development of solution jumps – Note the volume fraction “kink” near the inlet associated with a coarse/fine grid interface

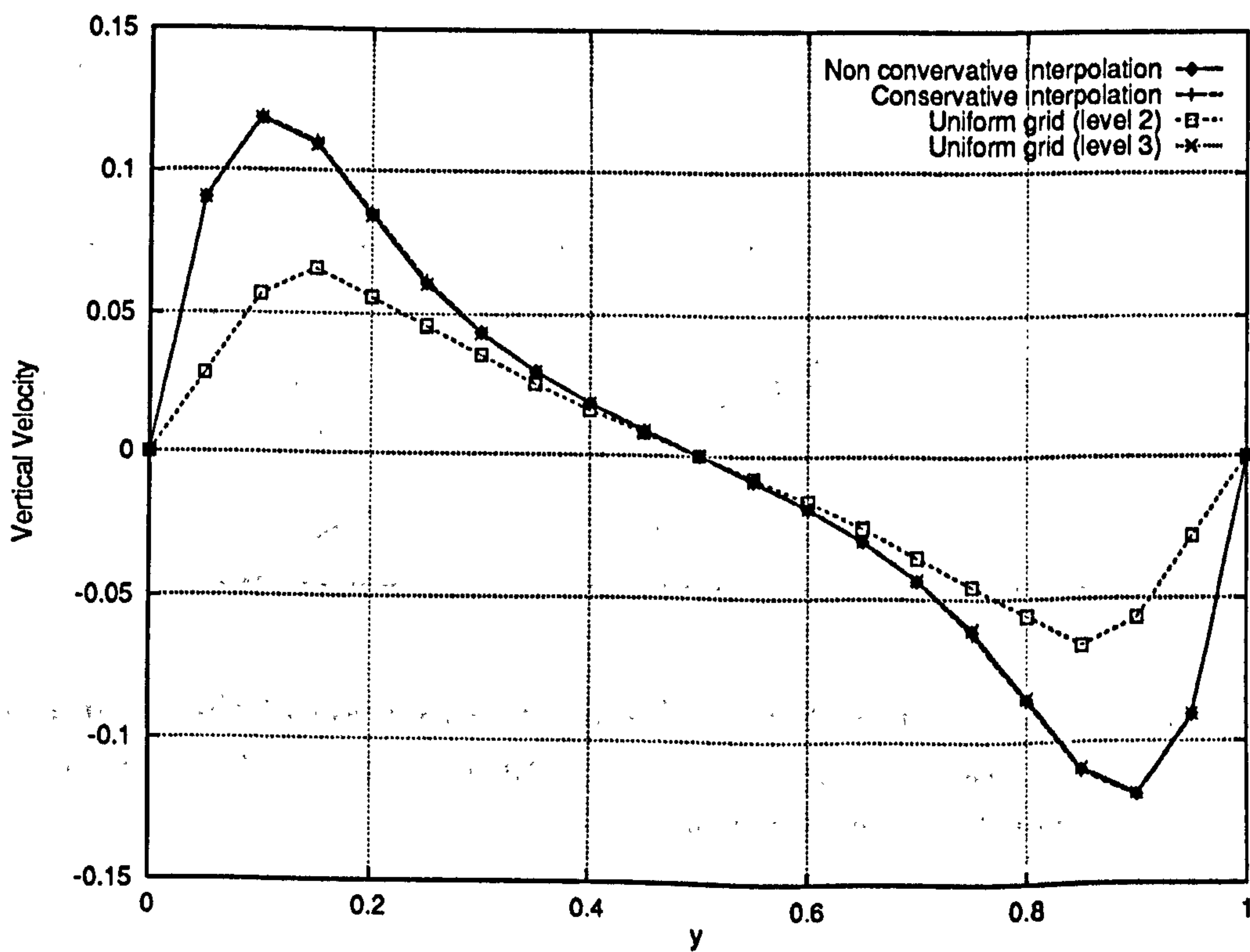


Figure 6.7: Multiphase channel flow problem – Comparison of the vertical velocity profile (for phase 1) along the line $x = 0.15$ obtained on a (2,3) adaptive grid, for different transfer operators between grids

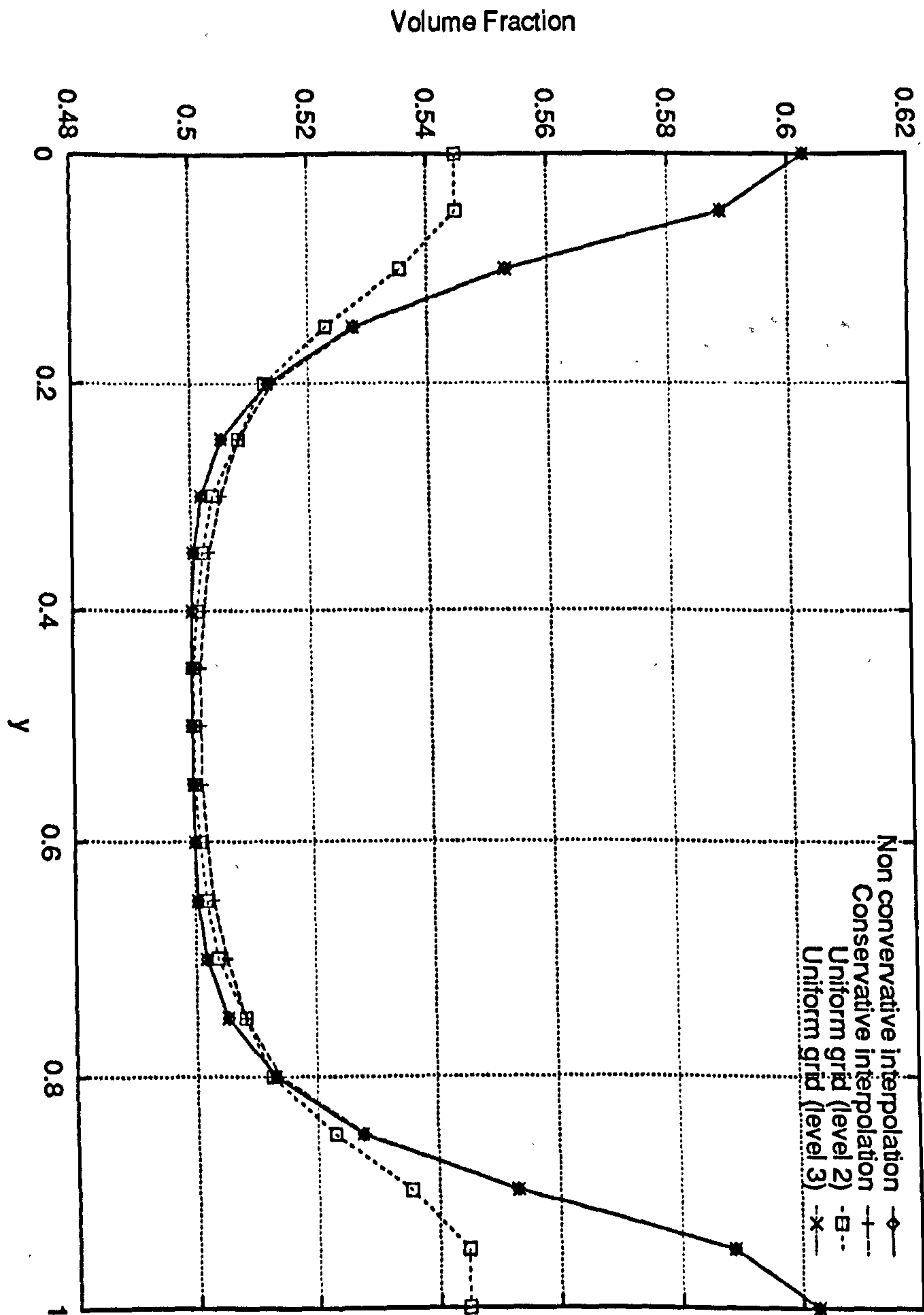


Figure 6.8: Multiphase channel flow problem – Comparison of the volume fraction profile (for phase 1) along the line $x = 0.15$ obtained on a (2,3) adaptive grid, for different transfer operators between grids

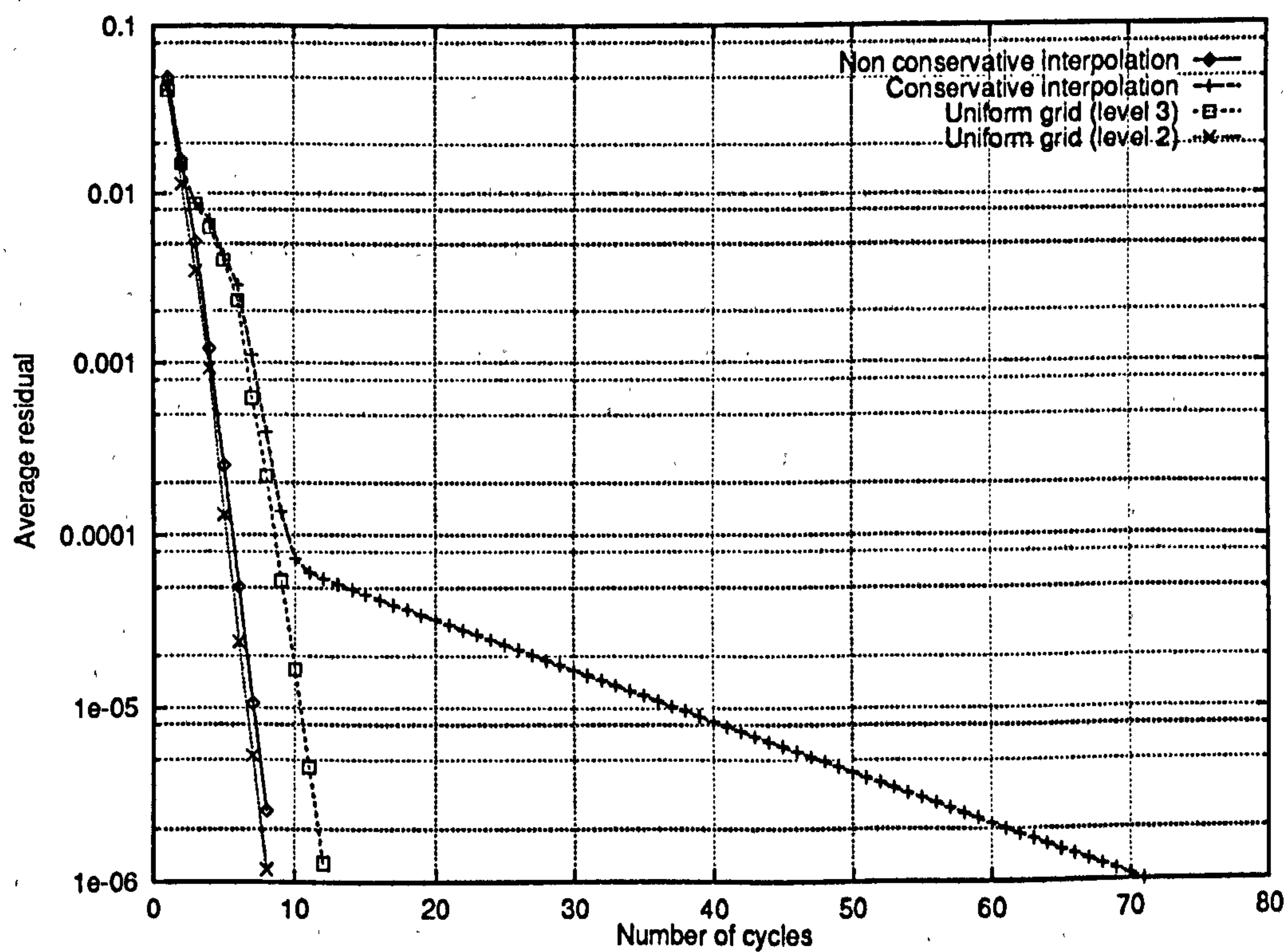


Figure 6.9: Multiphase channel flow problem – Convergence histories for adaptive A(2,3) computations and for different transfer operators between grids – See text for details.

6.2.3 Adaptive A(3,4) Results

Having established that it is important to have a reasonable uniform grid approximation of the solution in order for the refinement algorithm of Section 4.2.4 to work correctly, uniform grids computations are performed up to level 3 instead of level 2 as previously. In Section 5.3.1, we saw that between levels 3 and 5, the solution for the two-phase channel flow problem is reasonably grid independent except for the cross channel variation of the volume fraction. An approximation of the solution on a uniform grid at level 3 should therefore provide a good starting point for grid refinement.

Results are summarised in Figures 6.10 to 6.12. They confirm that it is possible to have convergent adaptive computations with transfer operators which do not conserve multiphase mass fluxes. This question is examined in Section 6.4.

If grid refinement is performed in the correct areas, the magnitude of the jumps in the solution is sharply reduced so that they no longer appear on the solution profiles. Provided that the uniform grid on which the initial refinement is based is fine enough to meaningfully (as opposed to accurately) represent the flow solution, then the refinement algorithm is robust. The user does not need to indicate where refinement is desirable. The refinement algorithm will refine the grid where necessary – in this case near solid walls – without losing accuracy at the interface. This pleasing behaviour follows from the fact that the error estimation, based on the FAS defect which approximates the truncation error, is good. Admittedly, the refinement pattern also depends on the value of the parameter γ ³. If γ is set too high, the refinement algorithm will be too selective and accuracy will suffer.

The conservative interpolation seems to give better results, particularly for the volume fraction – see Figures 6.11 and 6.12 – but the convergence rate significantly deteriorates for some values of the under-relaxation parameter λ_r (Figure 6.13) while for other values, the convergence rates are very similar (Figure 6.14). The convergence rates for adaptive and uniform grids are also compared, but this requires care since domains are not equivalent. It is sufficient to note that convergence rates for adaptive grids are comparable to those for uniform grids, which makes adaption very attractive because the cost of each multigrid cycle can be greatly reduced (see Section 6.5).

³ γ controls the selectivity of grid refinement: the higher the value of γ , the fewer quadrants will be refined. See Section 4.2.4.

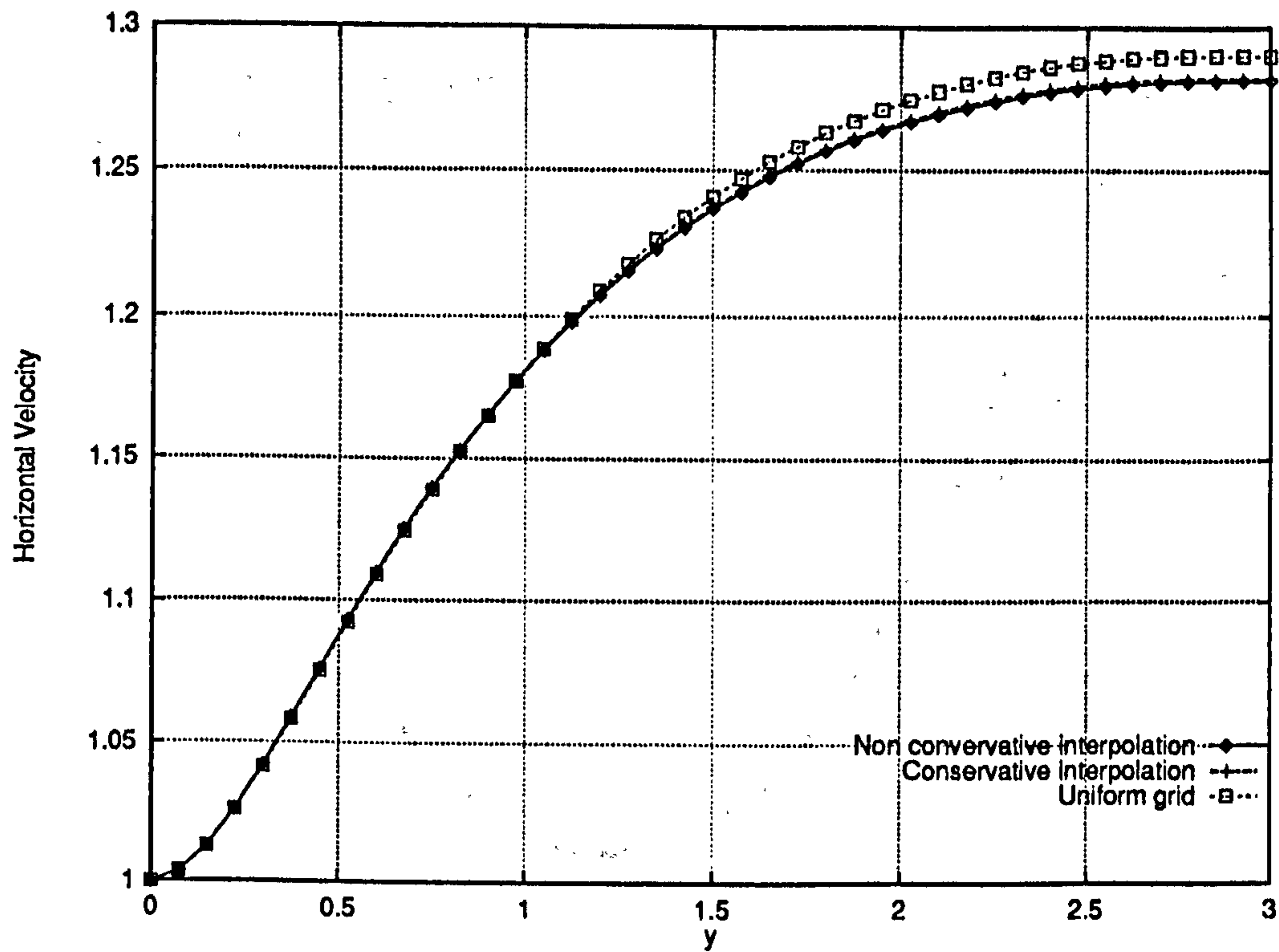


Figure 6.10: Multiphase channel flow problem – Horizontal velocity profile along the line $y = 0.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures

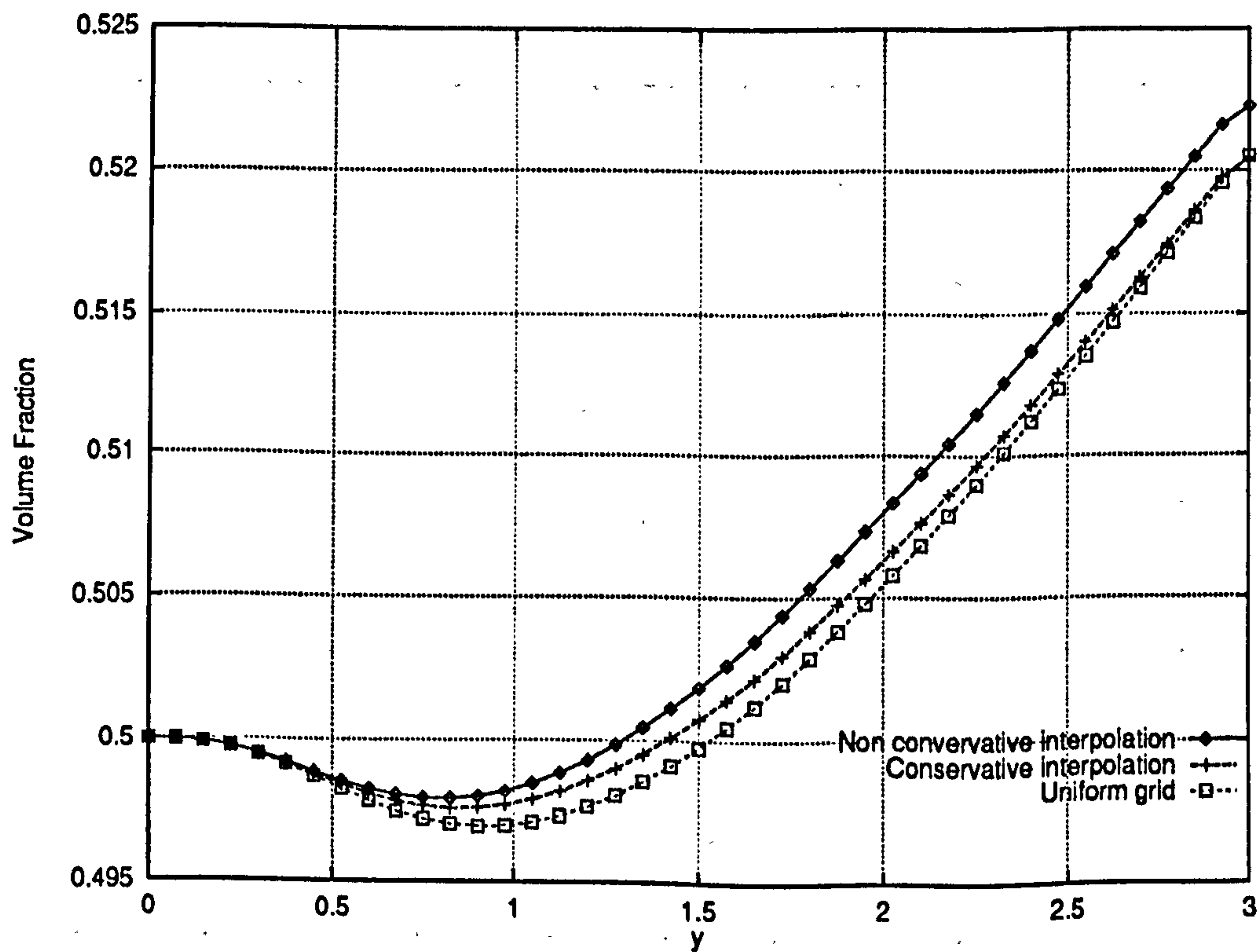


Figure 6.11: Multiphase channel flow problem – Volume fraction profile along the line $y = 0.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures

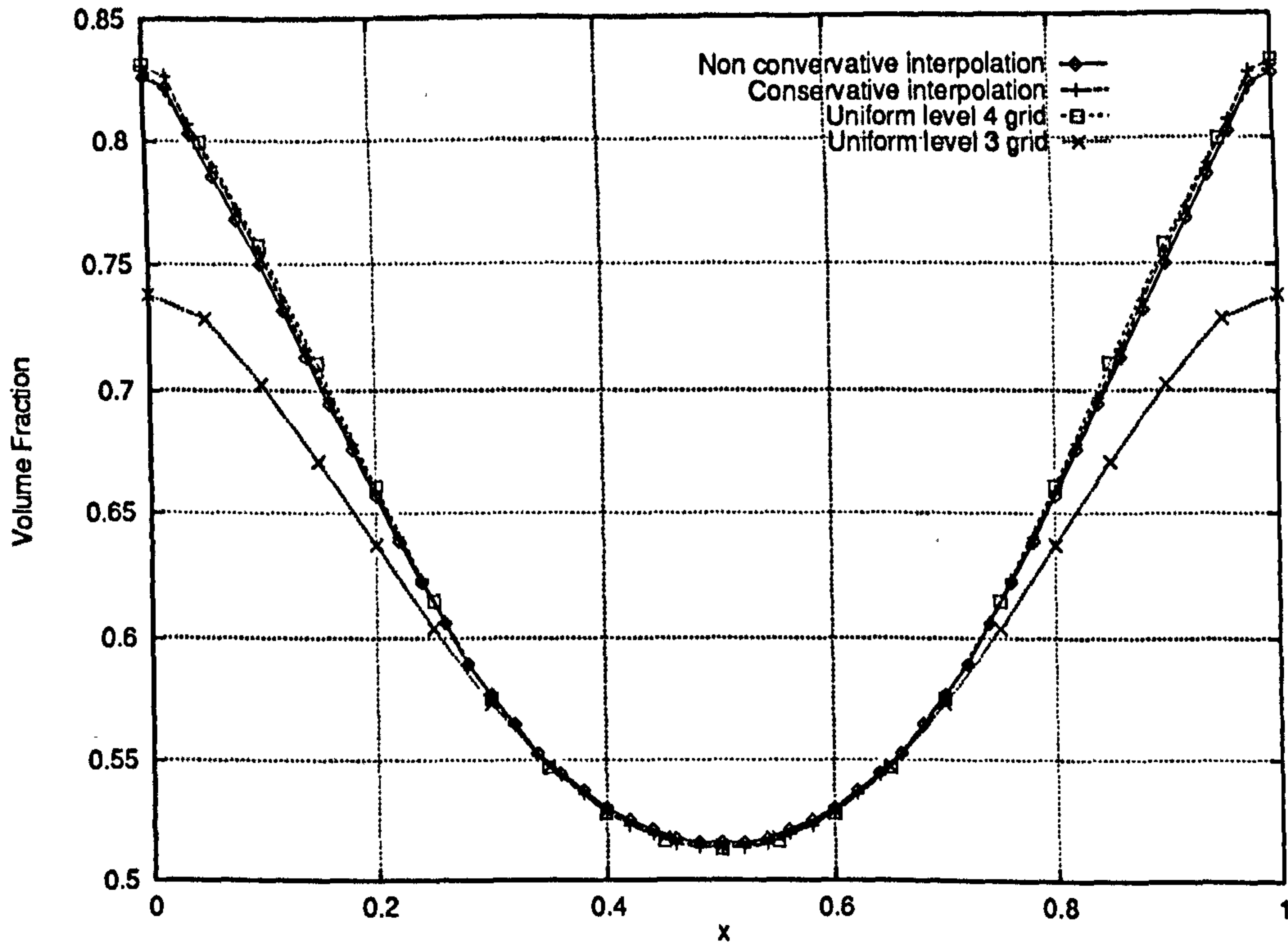


Figure 6.12: Multiphase channel flow problem – Volume fraction profile along the line $x = 2.5$ – comparison of uniform (4,4) and adaptive (3,4) solutions with conservative and non-conservative interpolation procedures – Note that the number of interpolation points is not related to the grid size

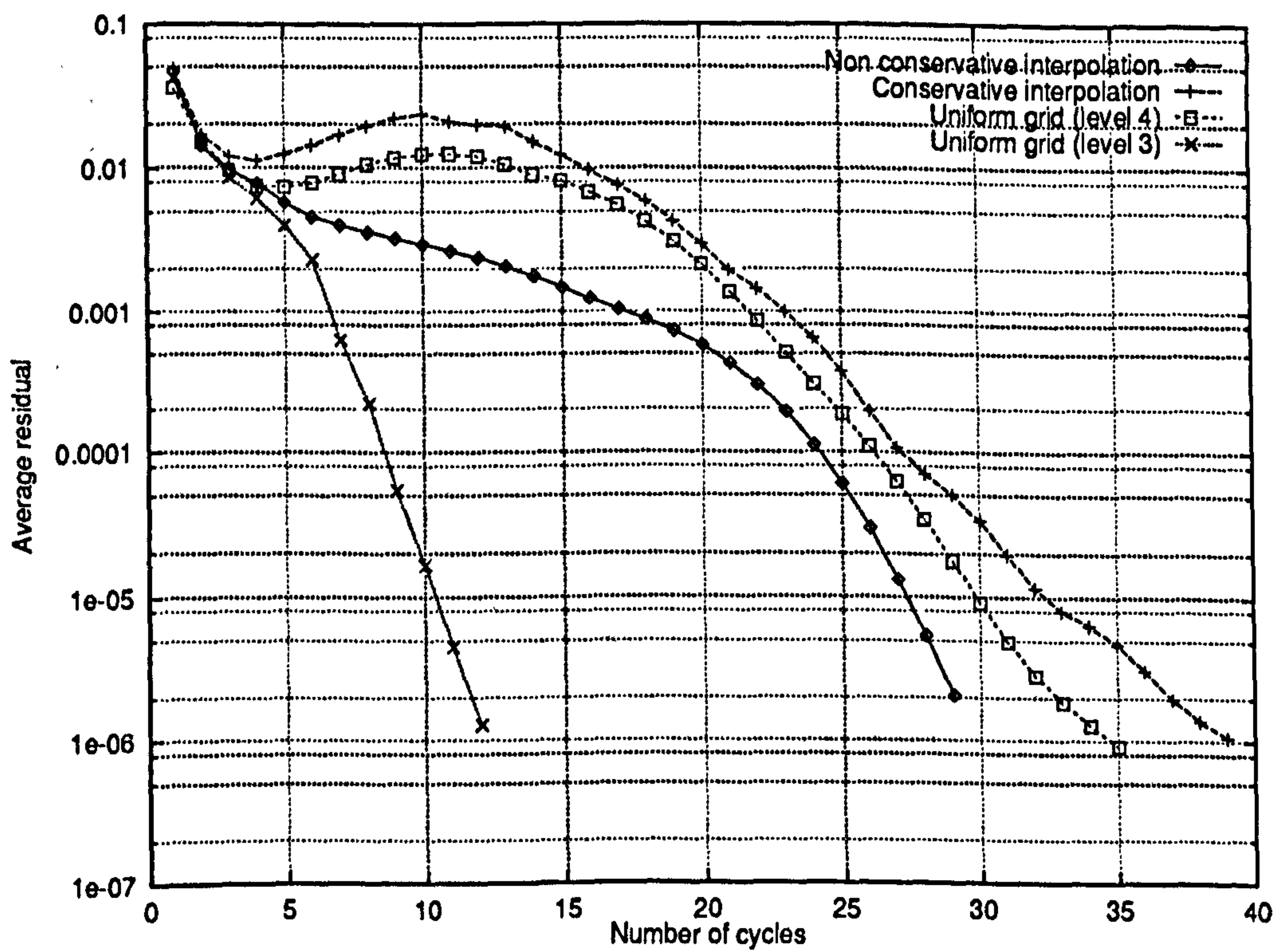


Figure 6.13: Multiphase channel flow problem – Convergence Rates for adaptive (3,4) computations with $\lambda_r = 0.8$

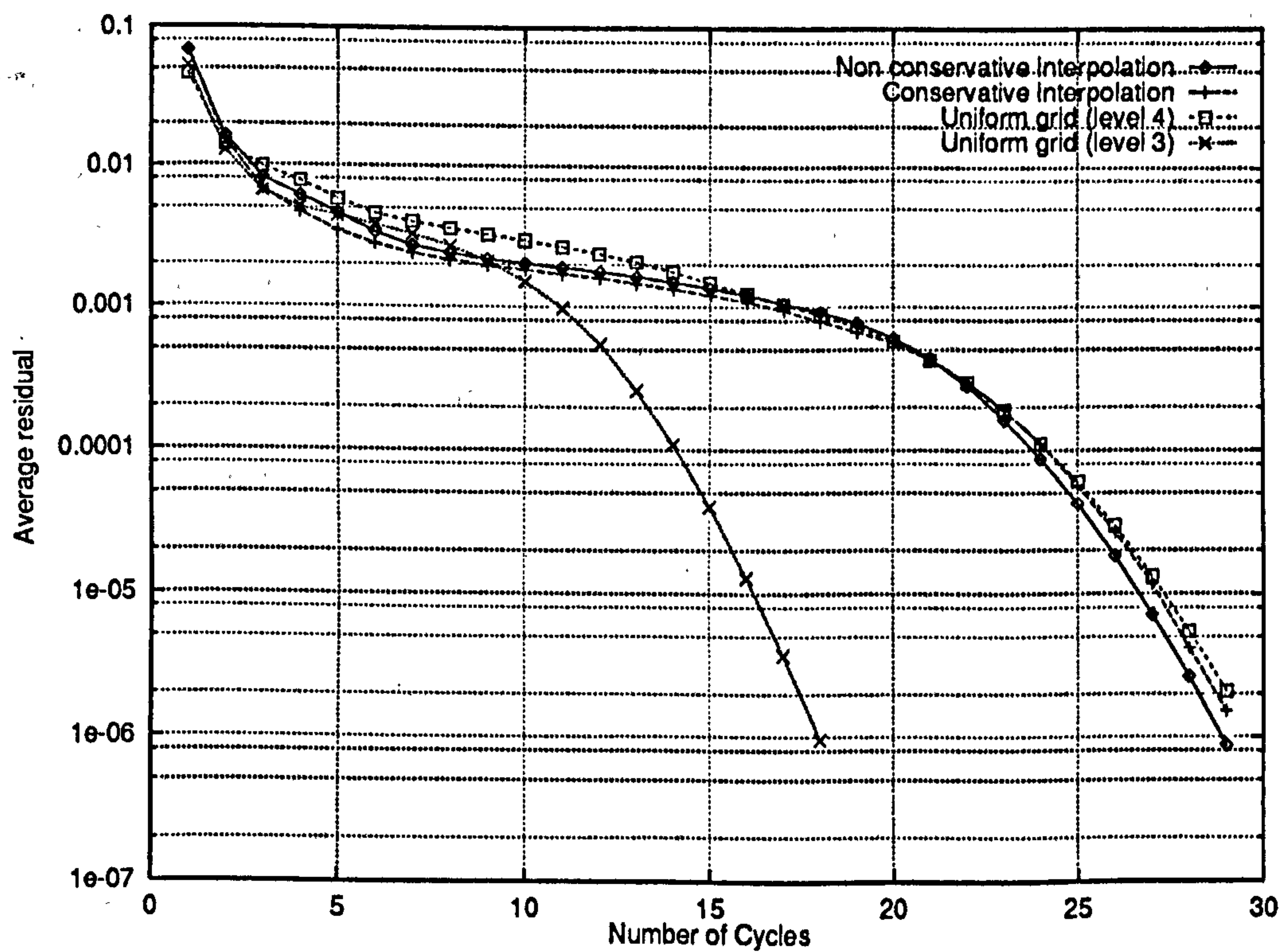


Figure 6.14: Multiphase channel flow problem – Convergence Rates for adaptive (3,4) computations with $\lambda_r = 0.6$

6.2.4 Adaptive A(3,5) and A(3,6) Results

In this section, further results on progressively finer grids are presented for the same multiphase channel flow. Here, non-conservative interpolation procedures are used as the conservative procedures have proven unsatisfactory for fine grids (see Section 6.4). The value of γ is set to 0.35 for both the A(3,5) and A(3,6) computations. These fine grid results both establish the accuracy of grid refinement and to a large extent validate the implementation of adaption in pang-multiphase.

In particular, the robustness of the solver is confirmed for multi-stage refinement i.e. the case when a refined grid is itself refined. By definition, there will be significant variation (relative to the grid size) in the solution in the regions where the grid is refined. Hence, it may be that further grid refinement introduces inaccuracies of the same type as those observed in Section 6.2.2. In pang-multiphase, as in pang, this is avoided by forcing refinement so that the grid aspect ratio for all eight neighbours of a cell is 2 at most ⁴. On the one hand, it greatly facilitates the implementation of adaption, on the other, it also ensures that as the level of refinement at a point increases, the width of the refined region around that point is enlarged. As a result, the boundaries between grid levels are continually being pushed away from the regions of high truncation error (see Figures 6.15 and 6.16).

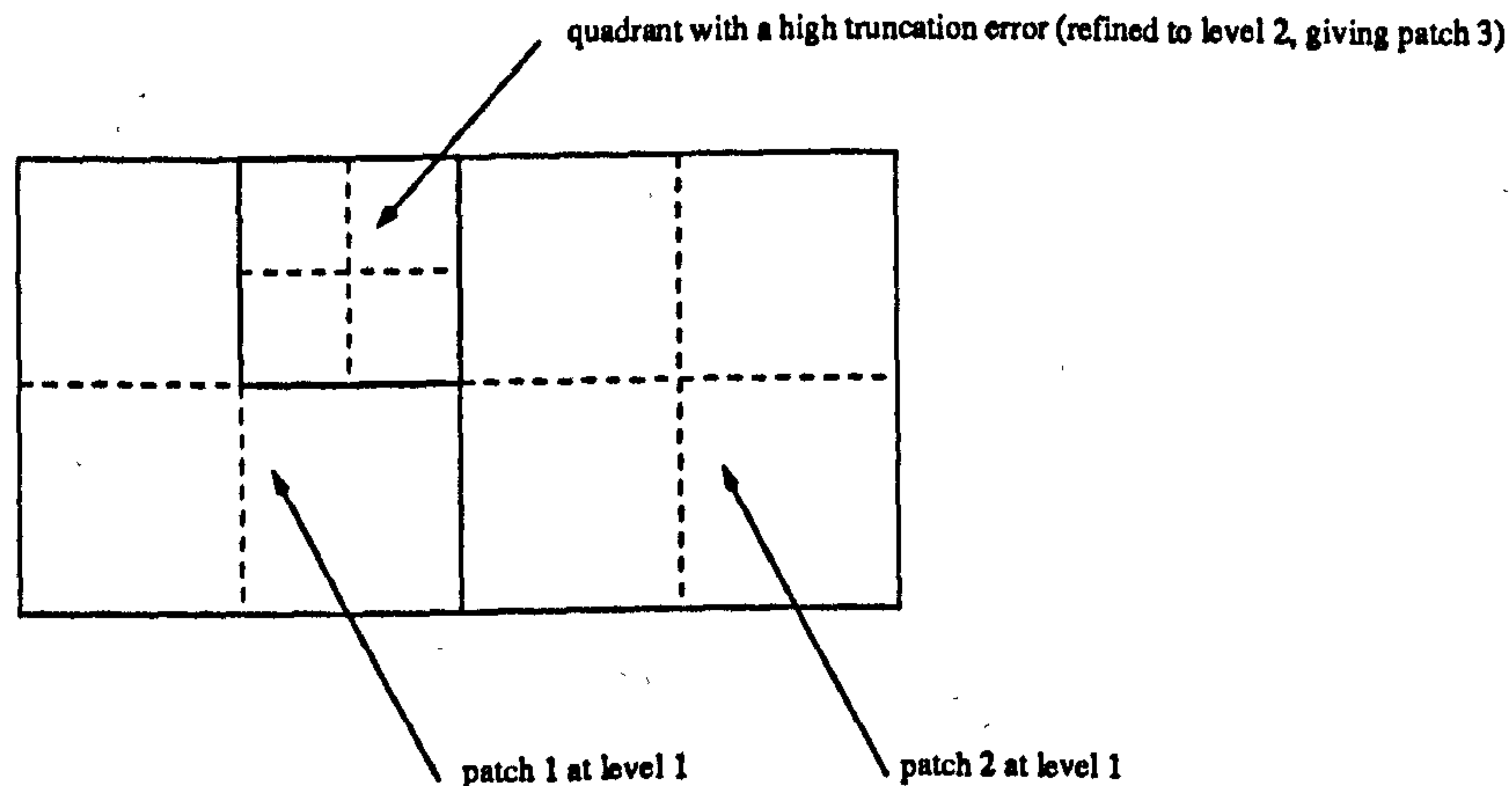


Figure 6.15: Refinement pattern based on truncation error alone

Solutions are shown in Figures 6.17 to 6.23. In particular, Figures 6.21 and 6.23 clearly demonstrate that by using adaptive grids, it is possible to accurately predict the variation of the volume fraction across the channel, at a fraction of the cost incurred in using uniform grids (see Section 6.5)

Figures 6.53 and 6.54 show the composite grids generated by pang-multiphase and Figure 6.24 presents the convergence history in each case.

⁴Furthermore, if a patch is refined at a re-entrant corner, then all the patches around the corner are refined.

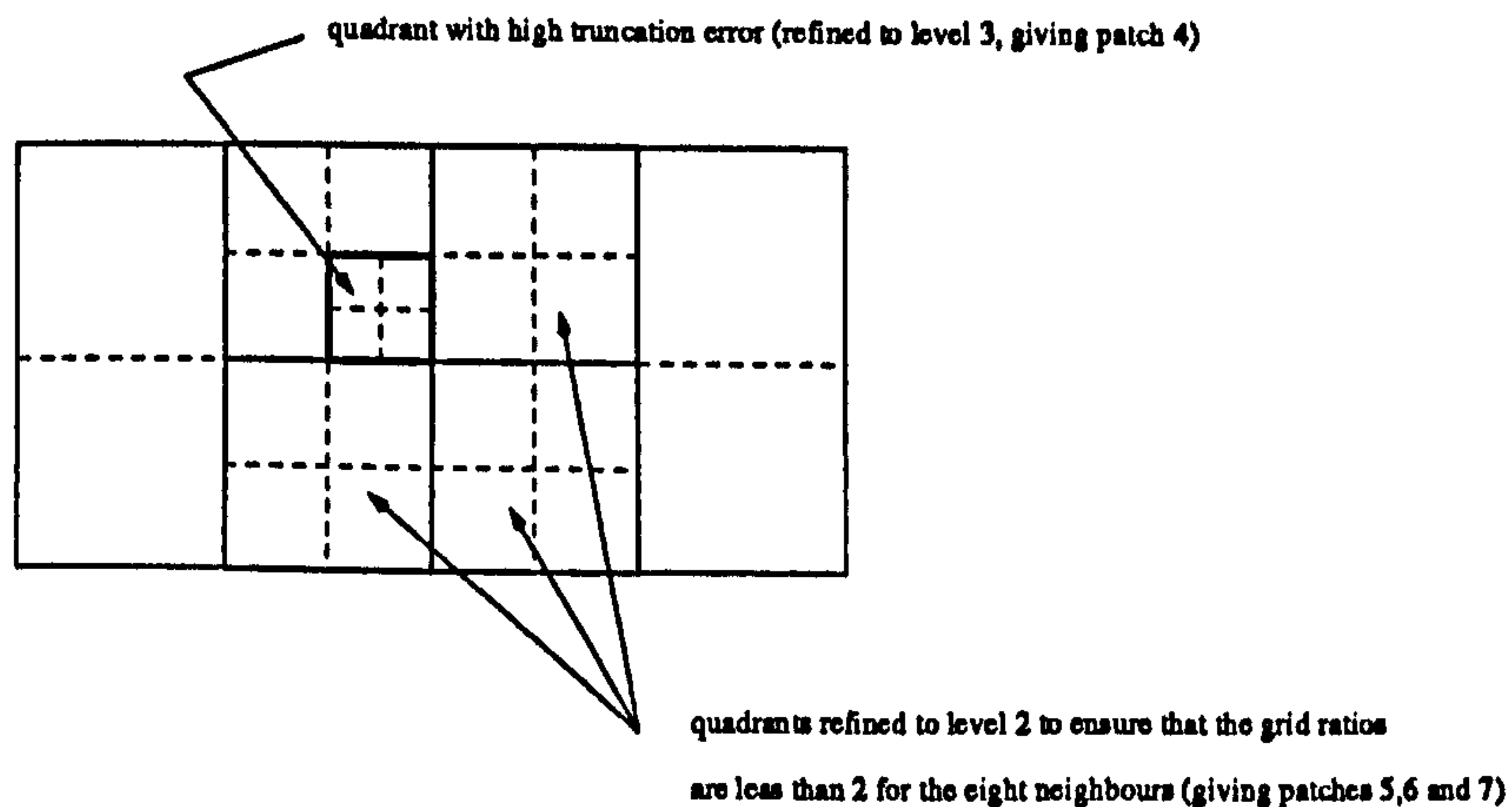


Figure 6.16: Refinement pattern based on truncation error plus forced refinement

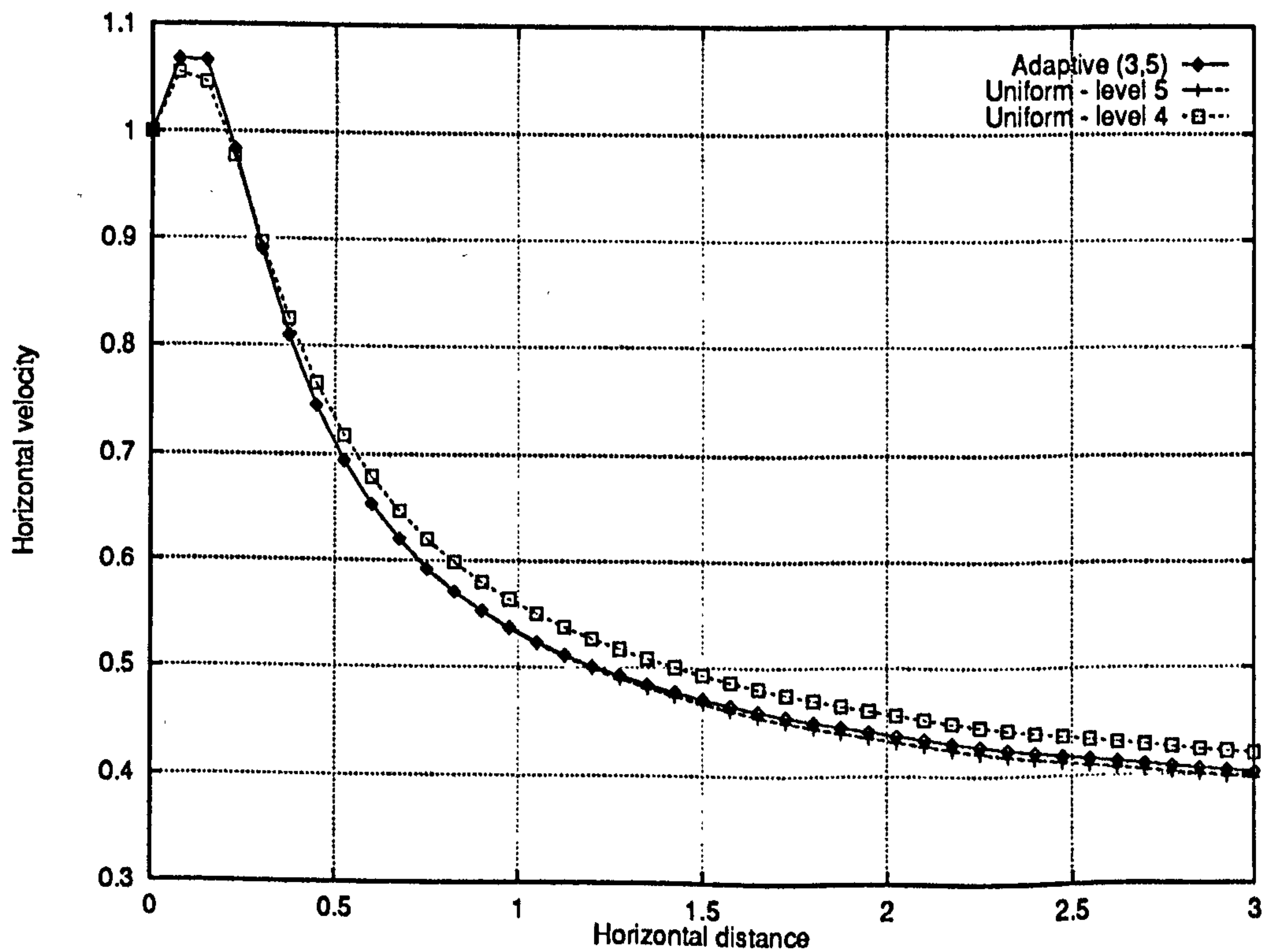


Figure 6.17: Multiphase channel flow problem – Horizontal velocity profiles along the line $y = 0.9$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels

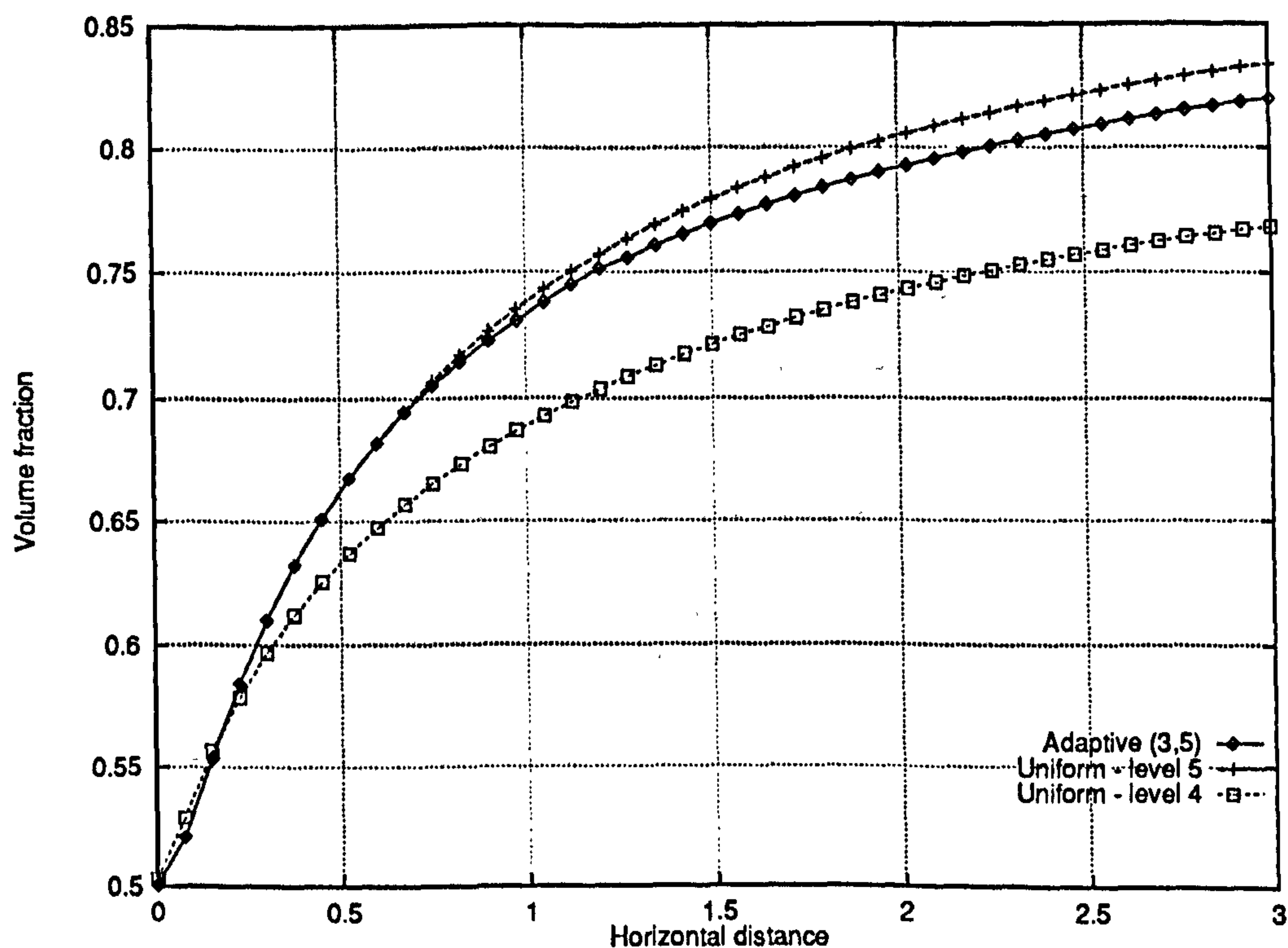


Figure 6.18: Multiphase channel flow problem – Volume fraction profiles along the line $y = 0.9$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels

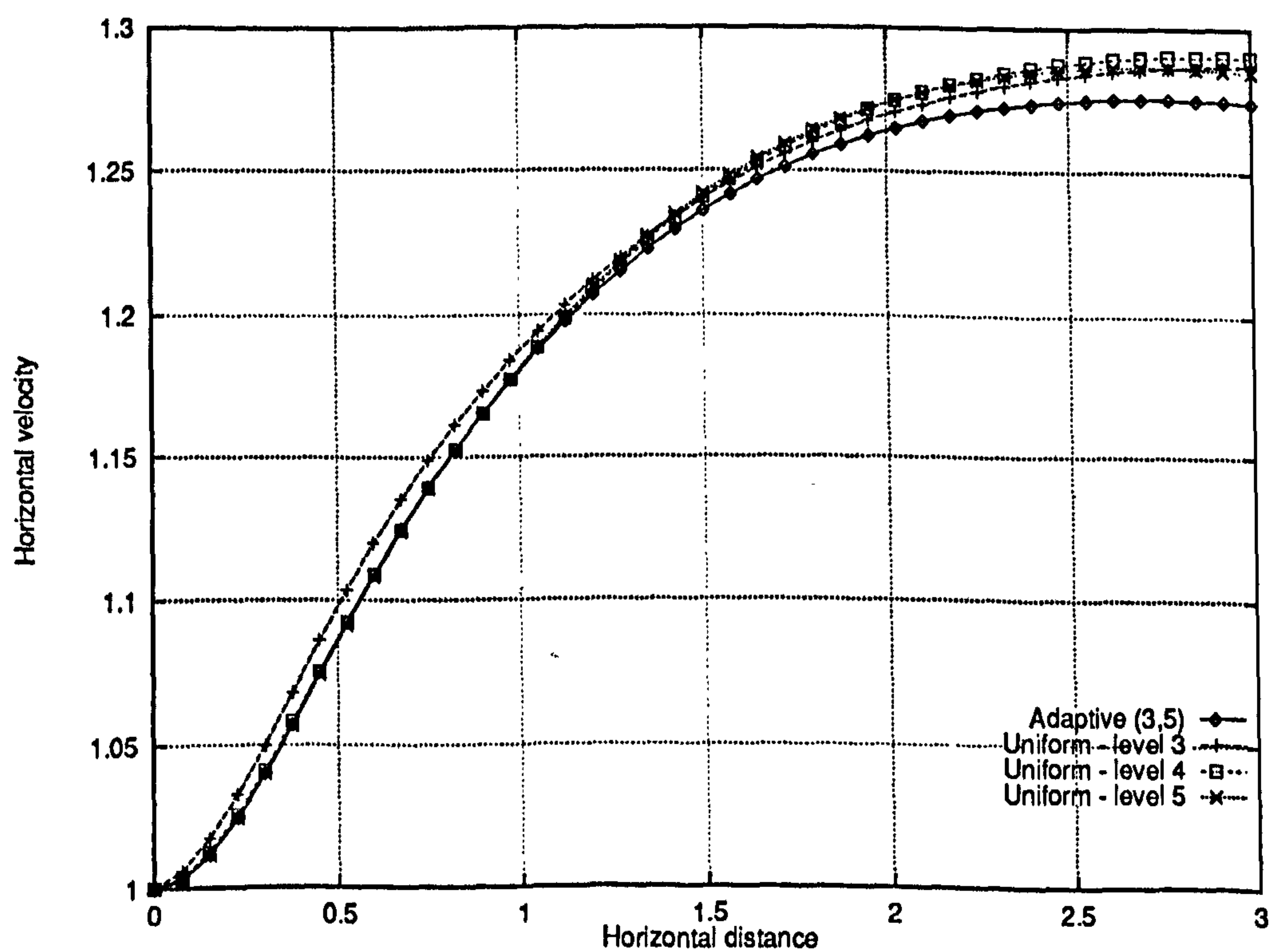


Figure 6.19: Multiphase channel flow problem – Horizontal velocity profiles along the line $y = 0.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels

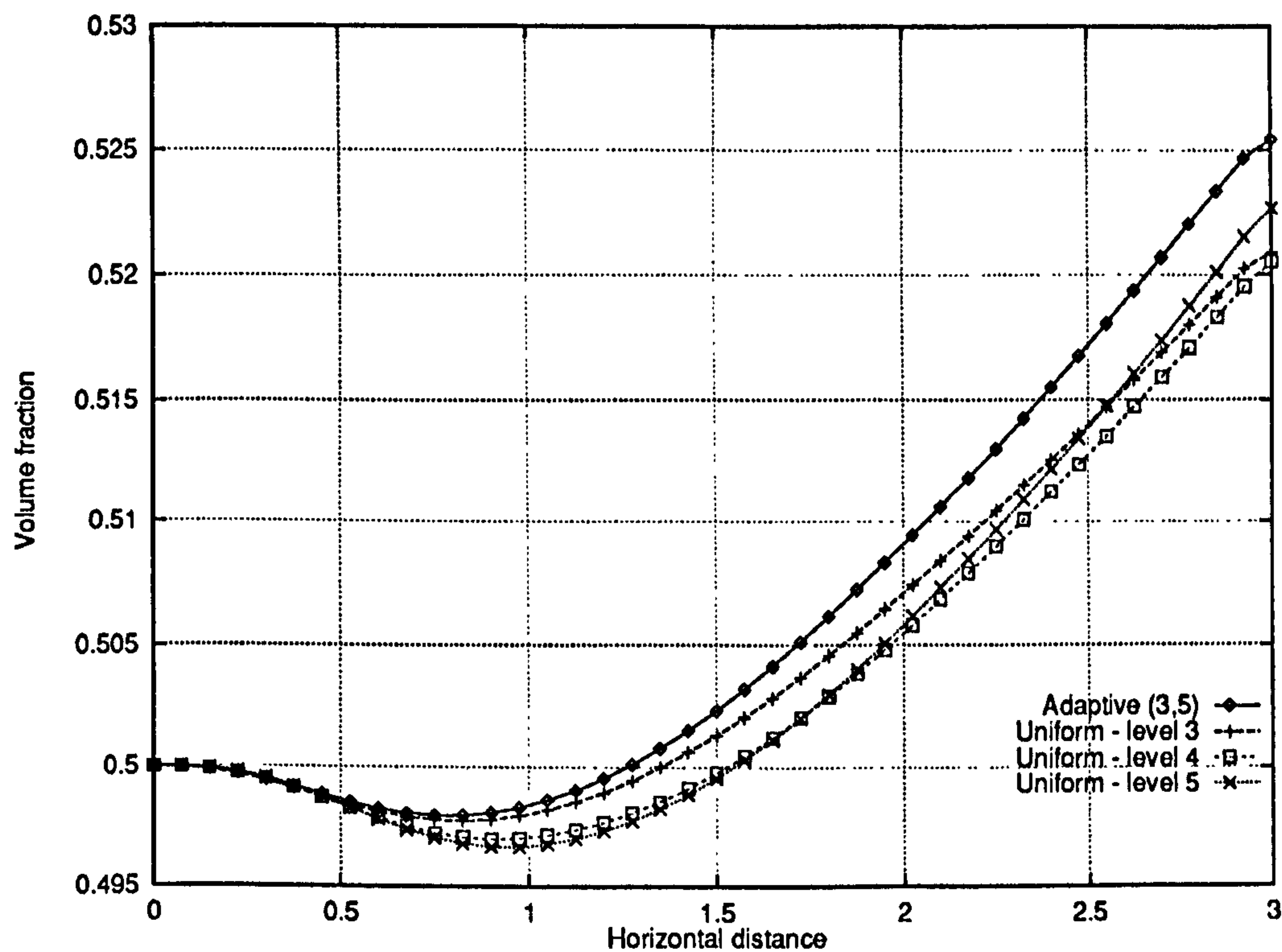


Figure 6.20: Multiphase channel flow problem – Volume fraction profiles along the line $x = 0.5$ – Comparison of the solutions for an adaptive $\Lambda(3,5)$ with uniform computations on different grid levels

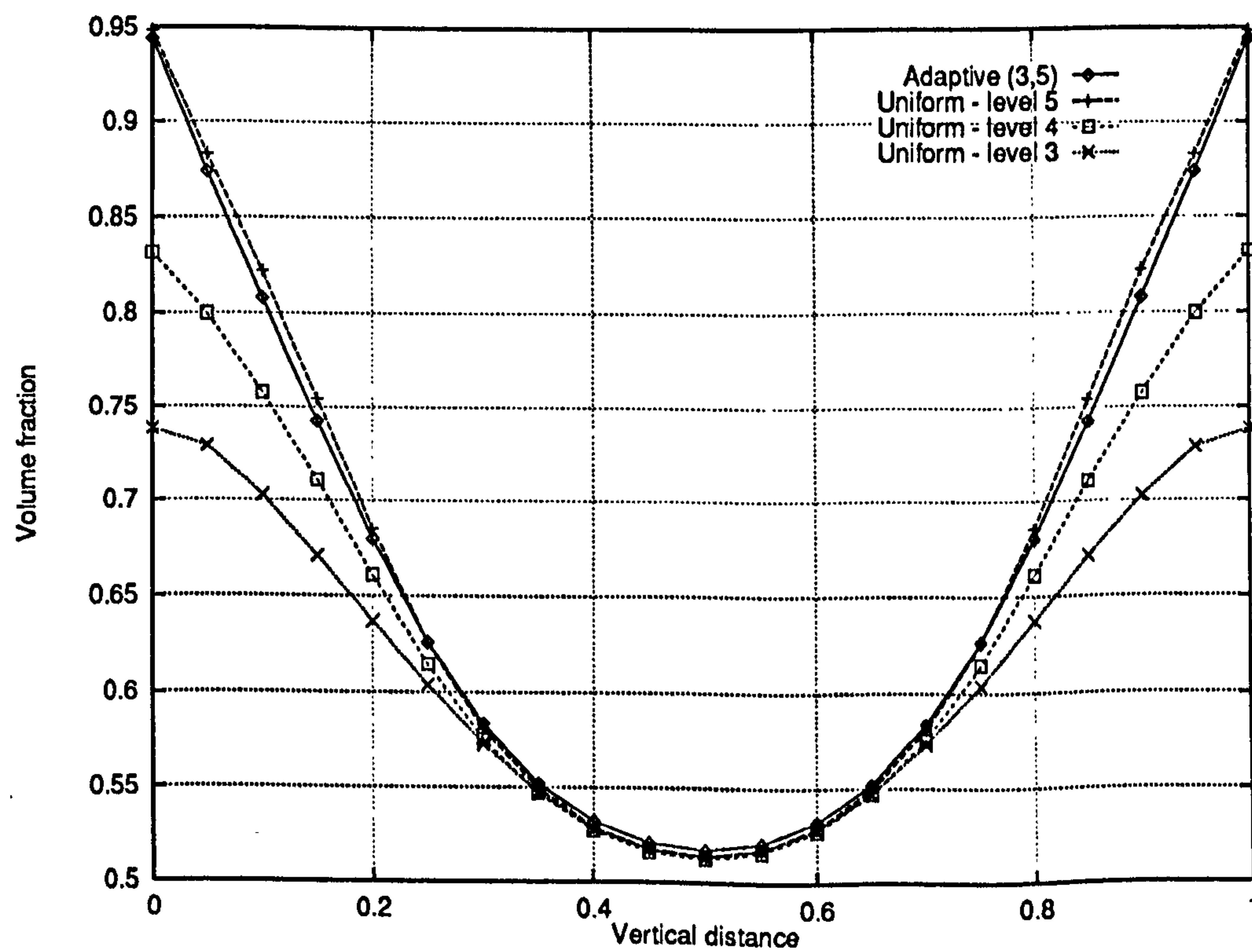


Figure 6.21: Multiphase channel flow problem – Volume fraction profiles along the line $x = 2.5$ – Comparison of the solutions for an adaptive $\Lambda(3,5)$ with uniform computations on different grid levels

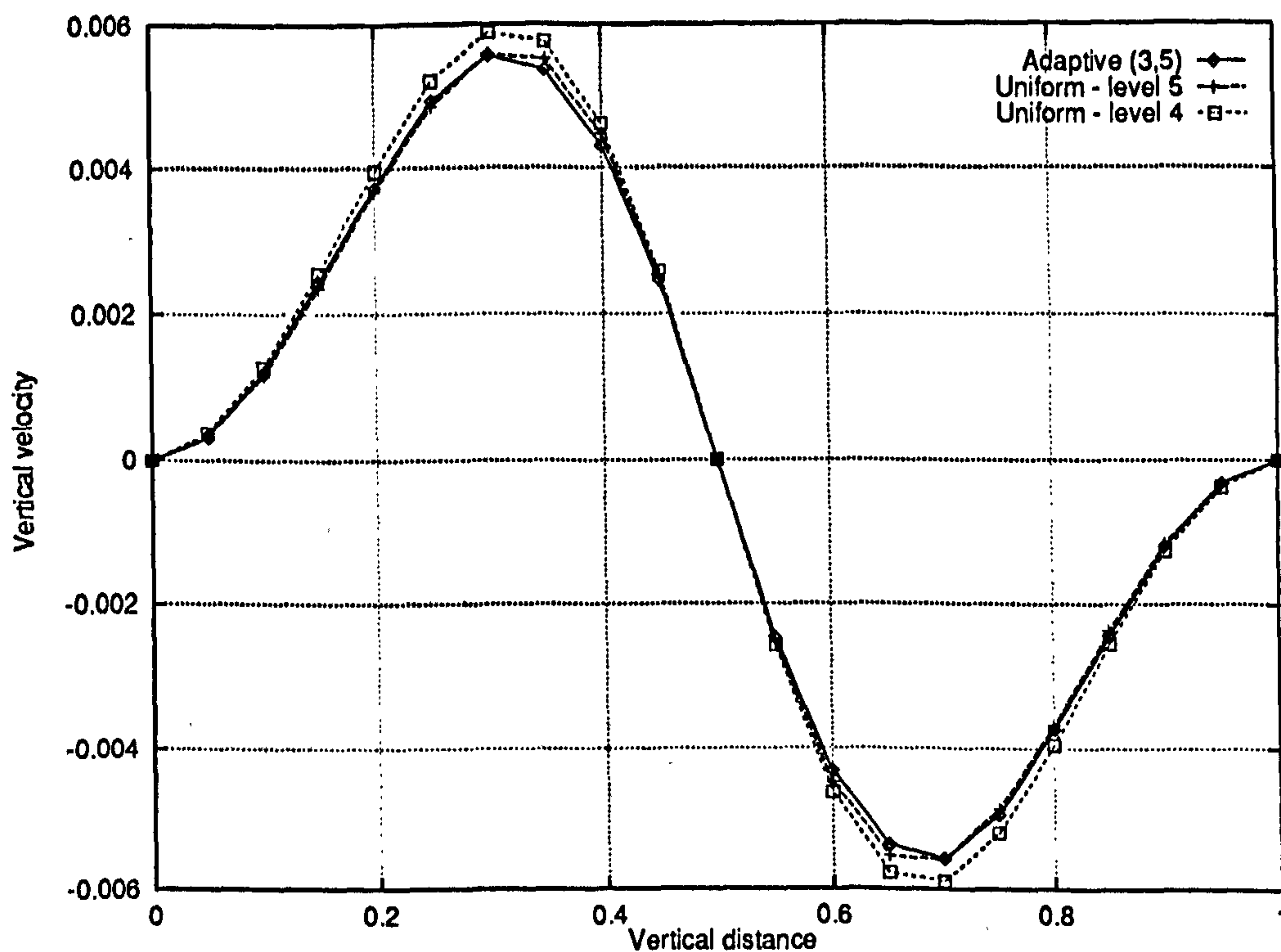


Figure 6.22: Multiphase channel flow problem – Vertical velocity profiles along the line $x = 2.5$ – Comparison of the solutions for an adaptive A(3,5) with uniform computations on different grid levels

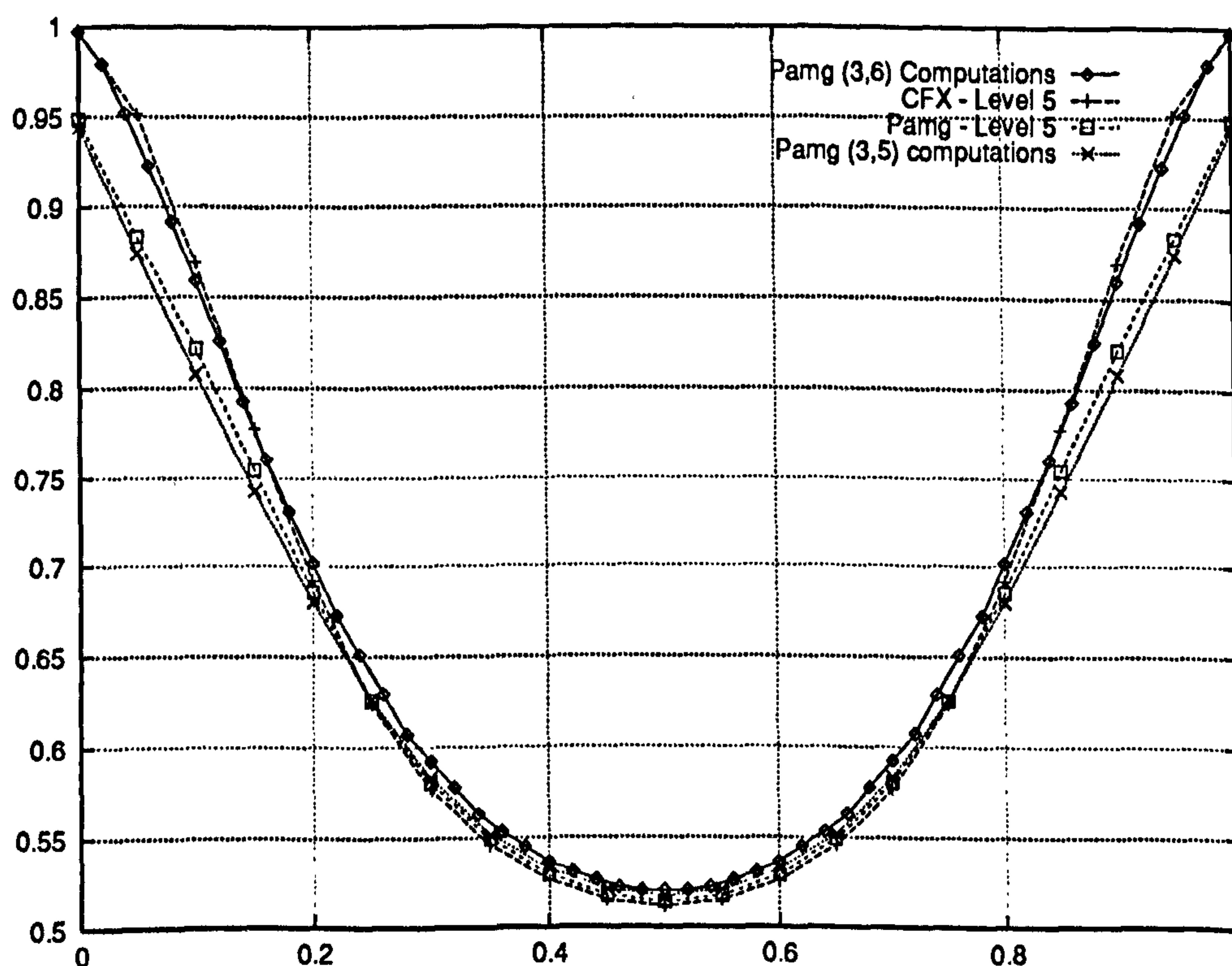


Figure 6.23: Multiphase channel flow problem – Comparison of the volume fraction profiles along the line $x = 2.5$ for adaptive A(3,6) and A(3,5) pamg-multiphase computations with solutions obtained on level 5 uniform grids with different solvers (pamg-multiphase and CFX 4.1) – Note that the number of interpolation points is not related to the grid size

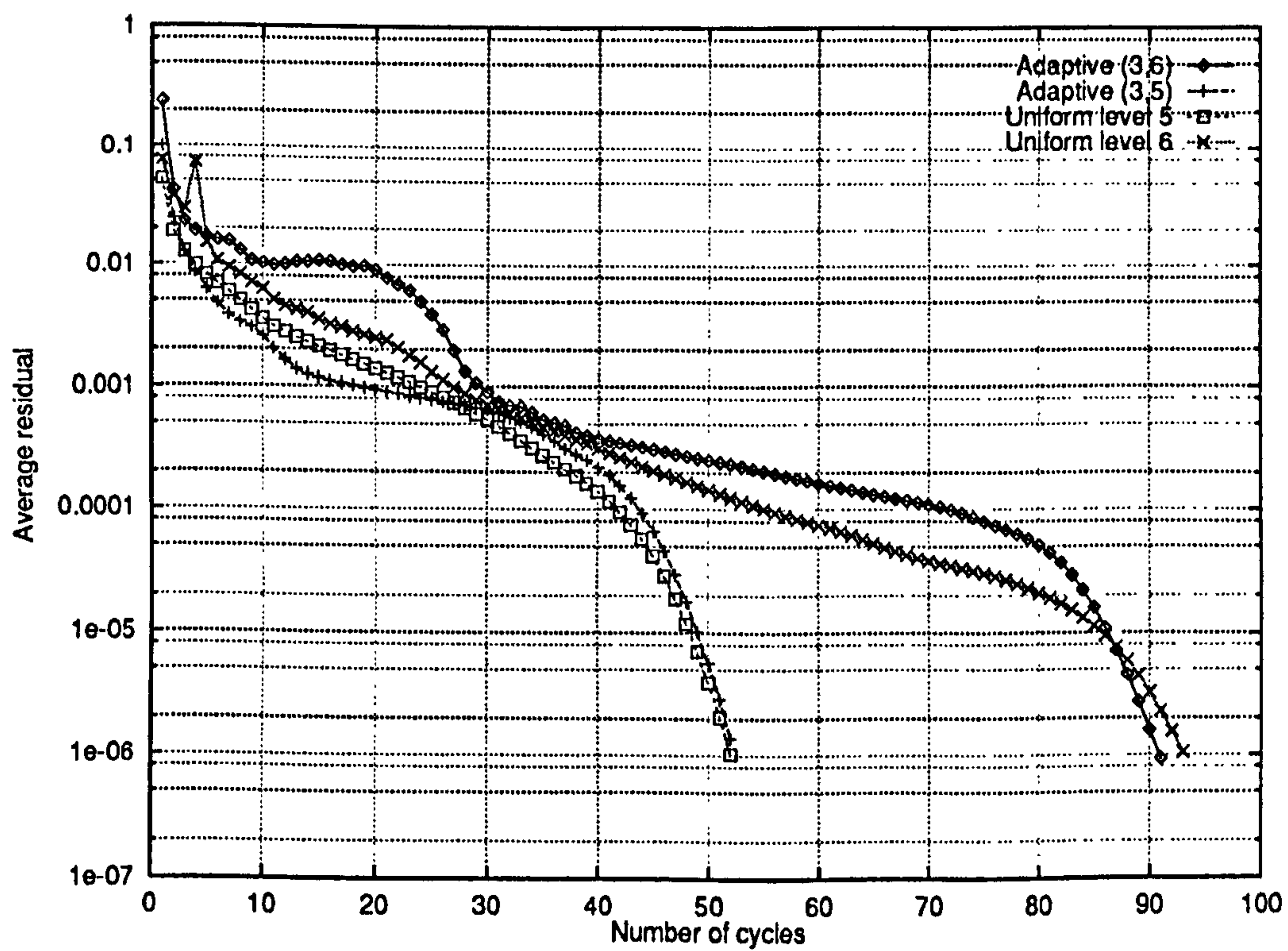


Figure 6.24: Multiphase channel flow problem – Comparison of convergence history for adaptive A(3,6) and adaptive A(3,5) pang-multiphase computations with uniform grid equivalents

6.3 Adaptive Computations of the Two-Phase T-Junction Problem

In this section, we consider the T-junction problem described in Section 5.3.2. Solutions on composite adaptive grids are again sought. Three problems have been attempted:

- an A(3,4) computation;
- an A(3,5) computation;
- an A(3,6) computation.

All tests concerning the multiphase flow through a T-junction were performed using the non-conservative interpolation operators. The reasons for this choice are given in Section 6.4. Furthermore, as the bulk of the truncation errors will be expected in a small region around the actual T-junction, the refinement parameter γ was set to a larger value than for the channel flow problem: $\gamma = 2$ instead of $\gamma = 0.35$. Hence, only the quadrants with large truncation error compared with the mean are refined (see Section 4.2.4).

Good results have been obtained for the first two computations. In particular, the accuracy of the solution is significantly improved in the neighbourhood of the junction (Figures 6.25, 6.26, 6.31 and 6.32). We also observe that at the outlet, the adaptive solution is quite close to the solution for a level 4 uniform grid (see Figures 6.28 to 6.30) even though most of domain is covered by a level 3 grid (see Figure 6.56). The fact that improving the accuracy in specific regions of the flow also has a beneficial effect on the accuracy in non-refined regions indicates that the flow solution is very dependent on local effects. This confirms the validity of the adaptive approach to simulating fluid flow.

Figure 6.33 shows the convergence histories for different grid levels. Convergence for the A(3,4) and A(3,5) computations was rapid and qualitatively similar to the convergence histories for uniform grid computations. By contrast, it was not possible to achieve convergence on the sixth level for the A(3,6) computation. Hypotheses to explain this failure have been formulated and tested (see below).

Firstly, as finer grids are used, a recirculation zone appears just downstream of the T-junction (see Figures 6.36, and 6.37). This feature also occurs for single phase flows (see Figure 6.38). While recirculation zones pose no significant problems in the single phase case (it was verified that `pang-multiphase` converged for an A(3,6) single phase computation ⁵), we have already mentioned that certain types of multiphase

⁵Figures 6.34 and 6.35 show the results recorded in the single phase case. Comparing the adaptive grid for the single phase simulation (Figure 6.61) with its equivalent for the multiphase

recirculation zones may prevent the convergence of multigrid computations because the problem is mathematically ill-posed. In other cases, however, computations converge without problems.

Some evidence suggests (see Section 5.4.10) that convergence is only hindered when both phases are recirculating. If only one phase is recirculating, the volume fraction of the other is determined by mass conservation, and the steady problem is still well-posed. In the case of the T-junction problem, only the lighter phase (phase 2) recirculates (compare Figures 6.39 and 6.40) and accordingly, the recirculation zone does not appear to affect the intrinsic efficiency of the solver. However, it does affect the performance of the refinement algorithm. This is demonstrated by the following argument.

It was observed that, in the neighbourhood of the recirculation region, the refinement algorithm fails. Figures 6.41 and 6.42 show that grids are refined in places where the flow changes a lot and consequently, the solution exhibits jumps. In fact, the value of γ has been set too high. If a lower value for γ is chosen, $\gamma = 1$, more quadrants are refined. It is still apparent that the errors are essentially of a local nature: by comparing Figures 6.58 and 6.59 which show level 6 adaptive grids for $\gamma = 1$ and $\gamma = 2$, it is obvious that only clearly delimited regions of high truncation errors are refined. The refinement pattern for $\gamma = 1$ proved to be more satisfactory as jumps disappear from the solutions (Figures 6.43 and 6.44).

Figure 6.45 shows the convergence histories for the computation of the solution on a level 6 adaptive grid (obtained with $\gamma = 1$), as well as the associated intermediate calculations aimed at estimating the truncation errors. Their analysis leads to two interesting conclusions:

- Firstly, computations on level 5 and 6 are not smooth. An explanation can be argued along the following lines. Level 5 computations are (almost) smooth for $\gamma = 2.0$ so that the oscillatory behaviour of the residual is associated with the region of the outlet where the flow is stratified, since this region is refined to level 5 for $\gamma = 1.0$ and not for $\gamma = 2.0$ (compare Figures 6.58 and 6.59 again). It is highly unlikely that these oscillations can be ascribed to the bad conditioning of local Jacobians in the stratified region since residuals are very significantly reduced in subsequent iterations.

The most likely cause is that in such “dry-out” zones, the probability of generating non-physical values for the volume fractions on fine grids after a coarse grid correction is increased. This explanation is consistent with the fact that oscillations appear for larger values of the residuals when coarse grid corrections are larger. In the author’s opinion, the same effect explains the oscillations observed on level 5 uniform grids (see Figure 5.155). Oscillations do not

problem (Figure 6.58), we also remark that the refinement near the inlets is due to real single phase effects, namely the strong pressure gradient generated by the rapid deceleration of the fluid(s) in the neighbourhood of solid walls.

appear for the same values of the residuals because the relative weight of the separation zone in the average residual is less for uniform grids.

The results discussed here were obtained with test (4.45) (the “no-change” strategy). This was changed to test (4.44) (the “cut-off” strategy) and the resulting convergence history is shown in Figure 6.46. Convergence rates for level 3 and 4 are not altered but with the “cut-off” strategy, the computations on level 5 fail to converge. The comparison with Figure 6.45 demonstrates that careful handling of the volume fraction is necessary in “dry-out” regions and that the method chosen can greatly affect the convergence rates on fine grids when there are regions of almost complete phase separation.

- Secondly, for $\gamma = 1$, the residuals for the level 6 grid are reduced until they reach a value just above 10^{-6} . From that point on, the residuals are not reduced smoothly and we observe a very oscillatory pattern. The fact that the residuals are alternatively decreased and increased, and that the amplitude of the oscillation is small would tend to indicate that the cause is the bad conditioning of the system. Table 6.1 certainly shows that the local Jacobians can be ill-conditioned. When condition numbers reach values of the order 10^6 , an accuracy better than 10^{-8} is difficult to achieve for double precision (64 bits) computations. Of course, the conditioning of the global Jacobian (which is the one actually solved by the symmetrical coupled Gauss-Seidel procedure) will be at least as large as that of the local systems. It should be noted, finally, that the trend of the residual is still slightly downward.

It seems difficult to attribute the failure of the A(3,6) computations with $\gamma = 2.0$ to large condition numbers for the local Jacobians, since the oscillation pattern is different. In the light of Figures 6.41 and 6.42, the most probable explanation is that non-convergence is related to the grid refinement being too selective. If the grid pattern is wrong, then the solutions on grids 5 and 6 will be very different, as attested by the presence of jumps in the solution. Therefore, the quality of the coarse grid correction is degraded and convergence on the finer grid hindered. This hypothesis is confirmed by the observation that reducing γ to produce a less selective grid refinement solves the problem. Another possible avenue would be to use operator based transfers.

Level	Average condition Number	Note
1	110	
2	369	
3	1819	
4	15 469	Maximum value: 141 161 (in stratified region)
5	35 688	Maximum value: 264 908 (in stratified region)
6	44 110	Maximum value: 502 437 (near the recirculation zone)

Table 6.1: Multiphase flow through a T-junction – Estimates of the Condition numbers at different levels for an A(3,6) adaptive computation with $\gamma = 2$

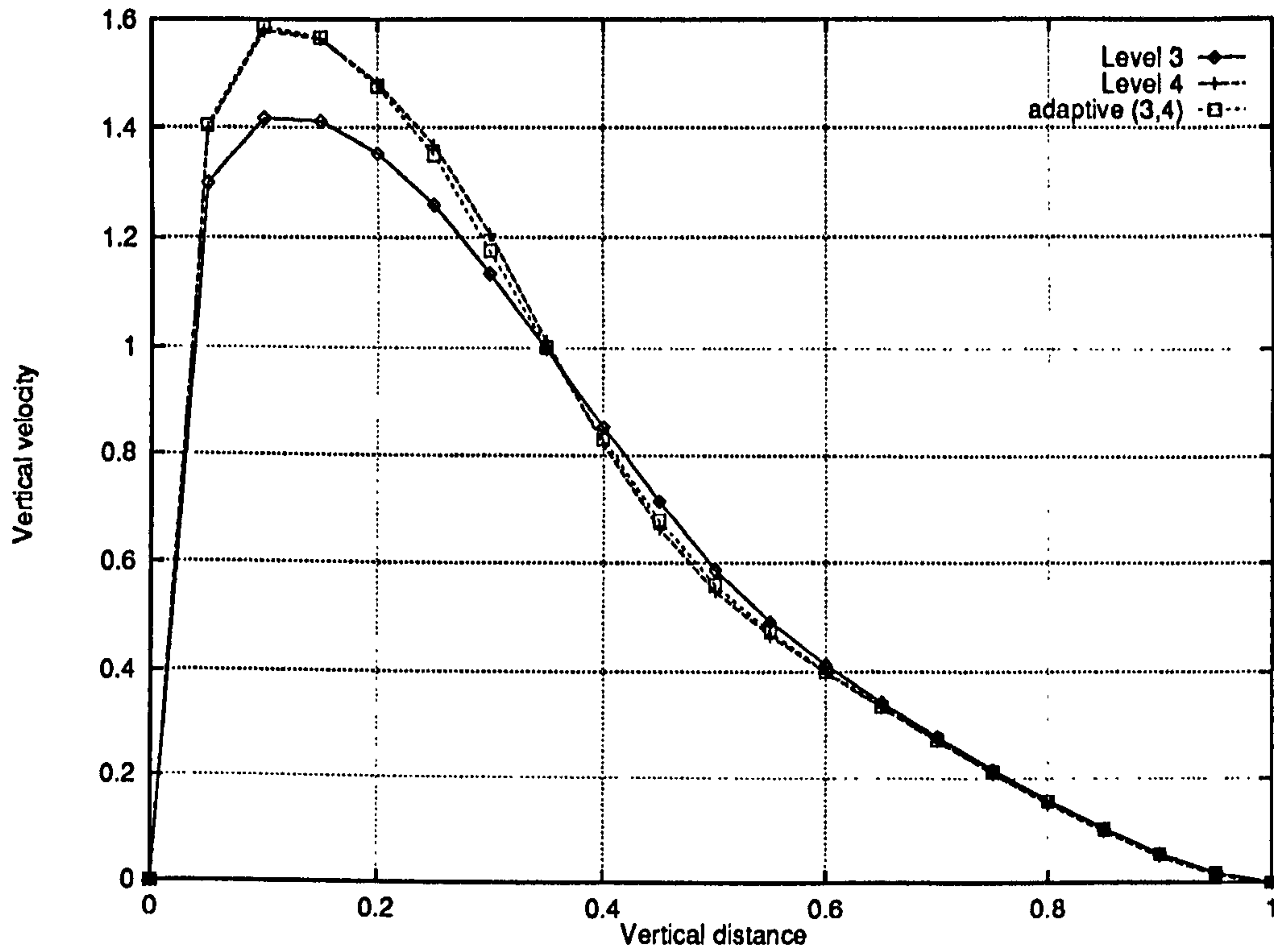


Figure 6.25: Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase I) on uniform and adaptive grids

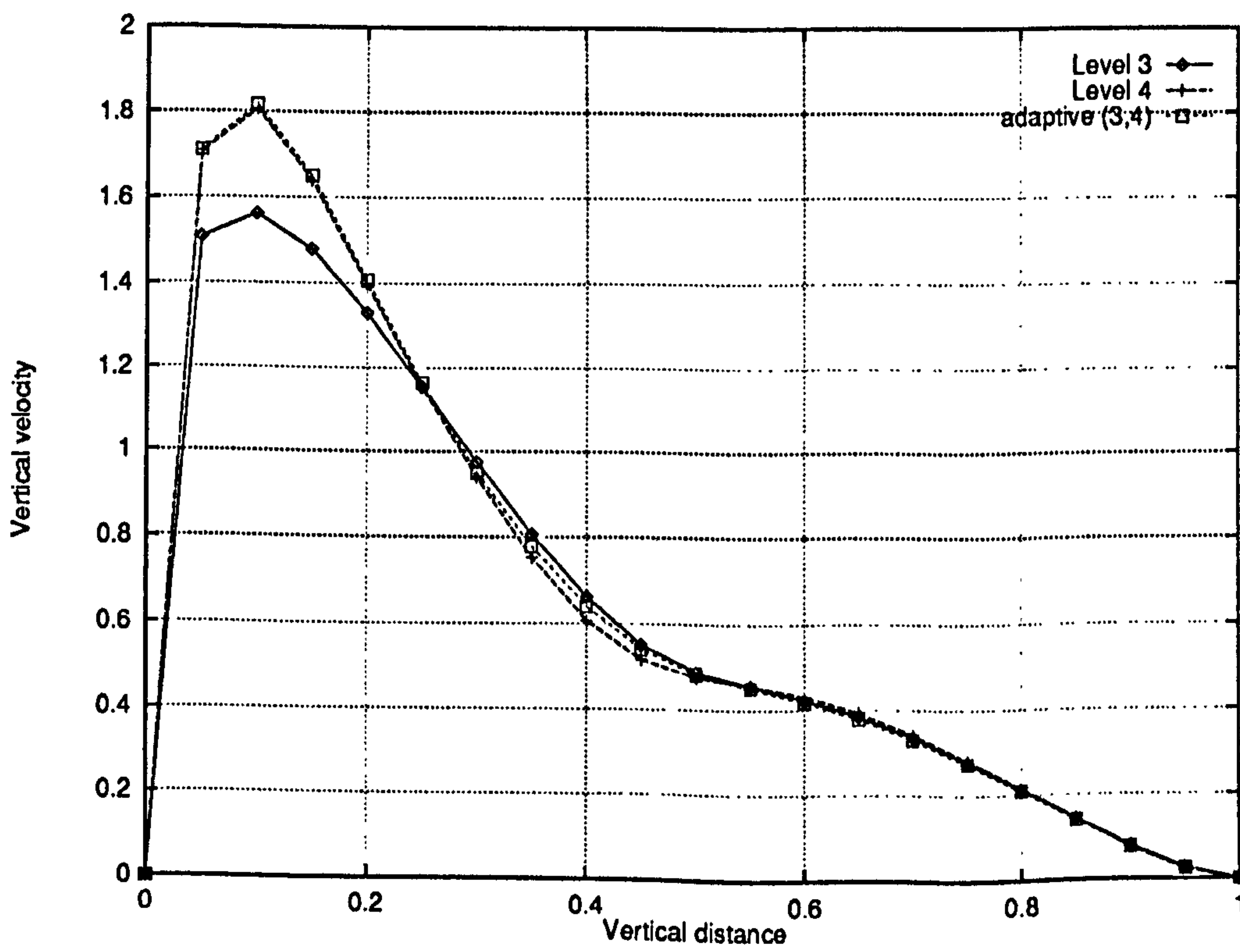


Figure 6.26: Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase II) on uniform and adaptive grids

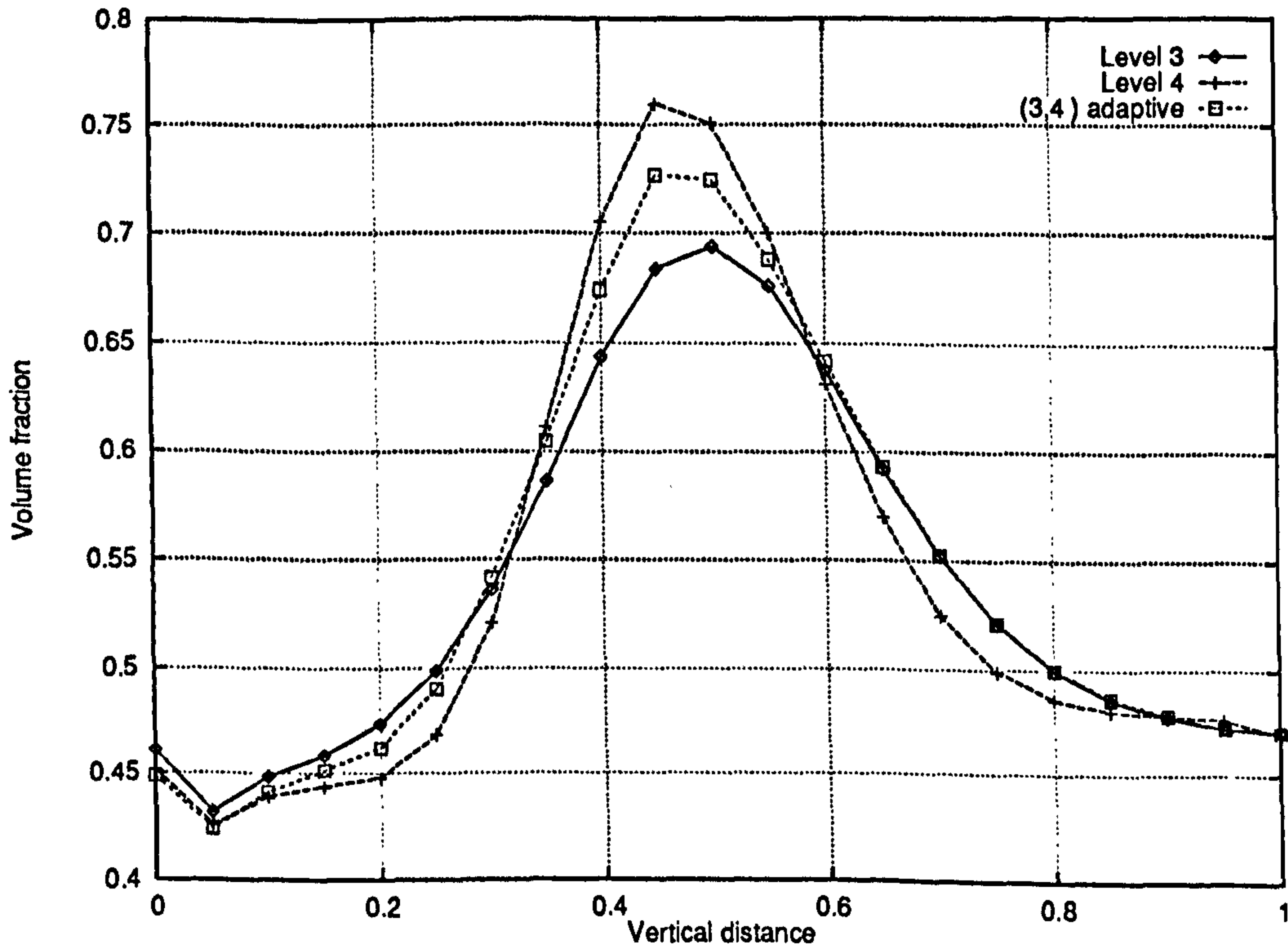


Figure 6.27: Two-phase flow through a T-junction – Volume fraction profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids

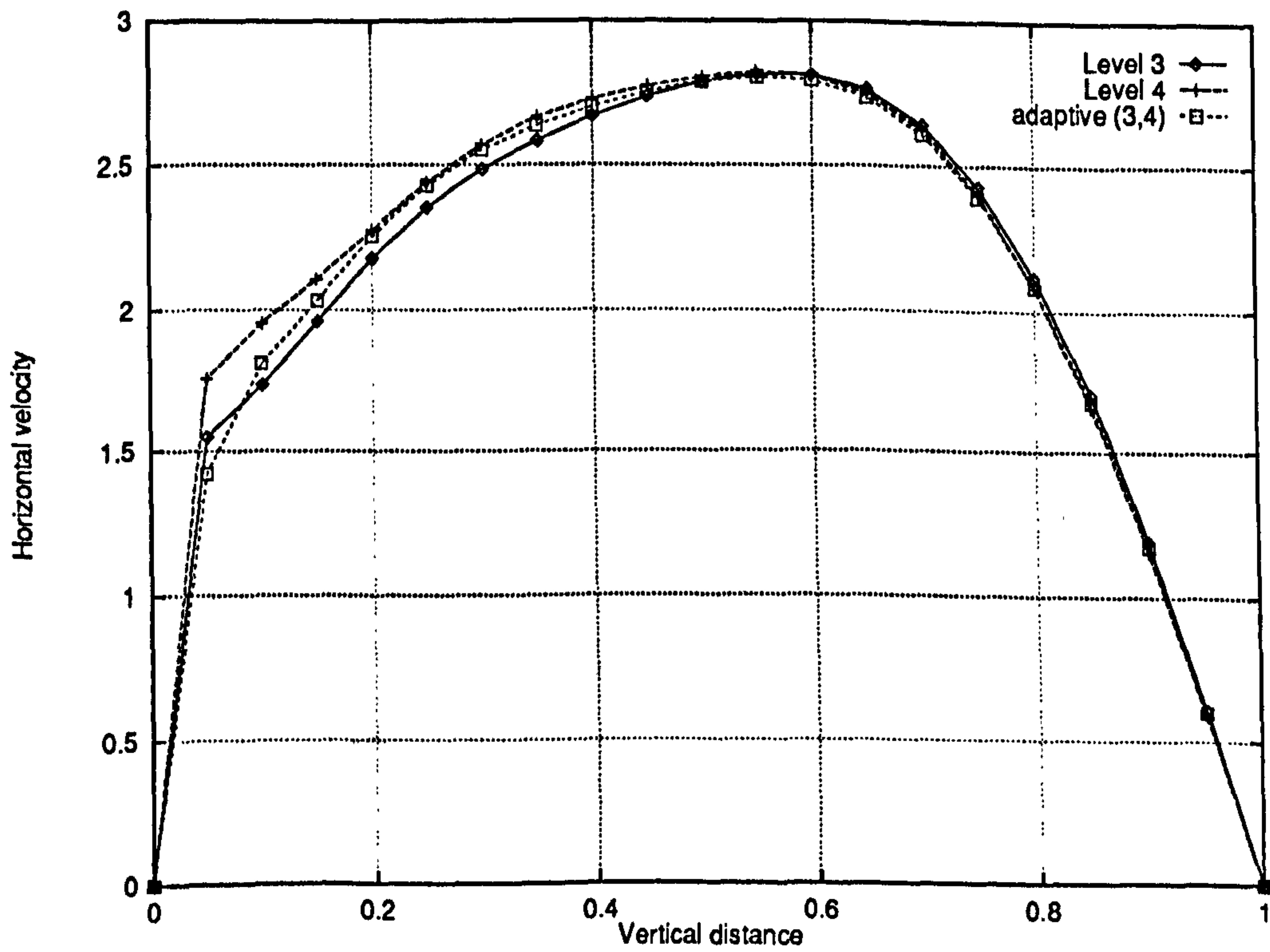


Figure 6.28: Two-phase flow through a T-junction – Horizontal velocity profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids

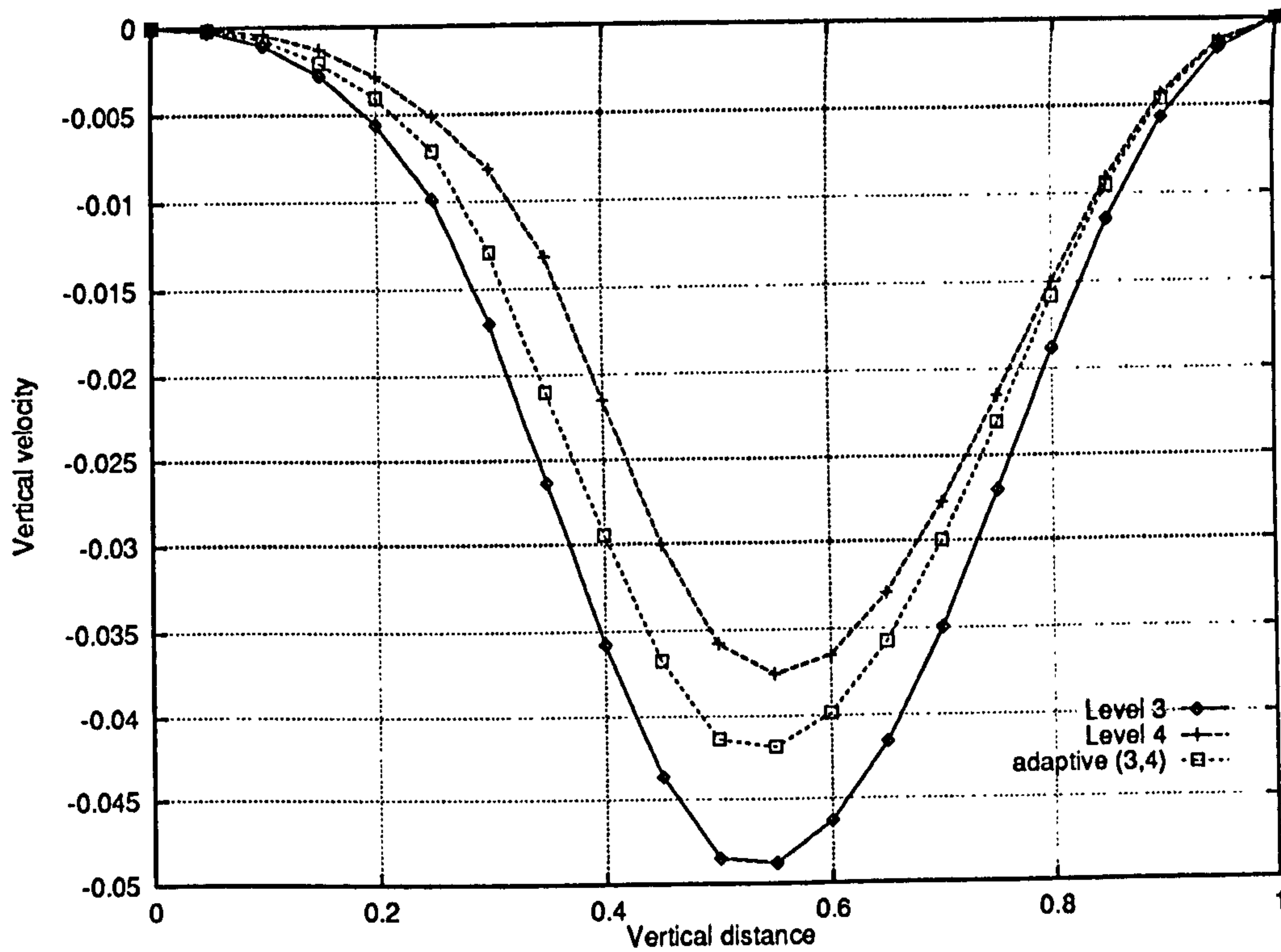


Figure 6.29: Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 6.5$ (Phase II) on uniform and adaptive grids

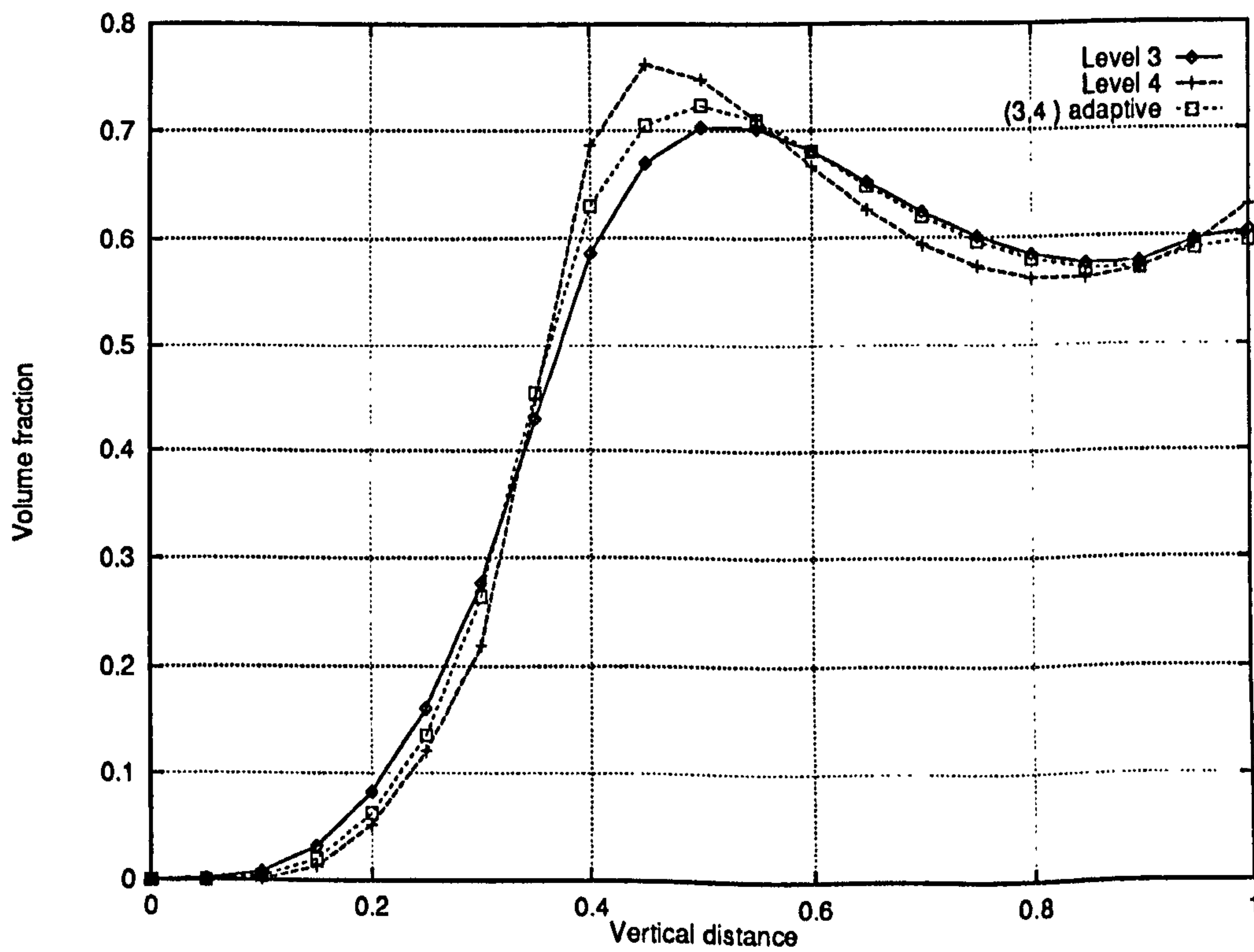


Figure 6.30: Two-phase flow through a T-junction – Volume fraction profiles along the line $x = 6.5$ (Phase I) on uniform and adaptive grids

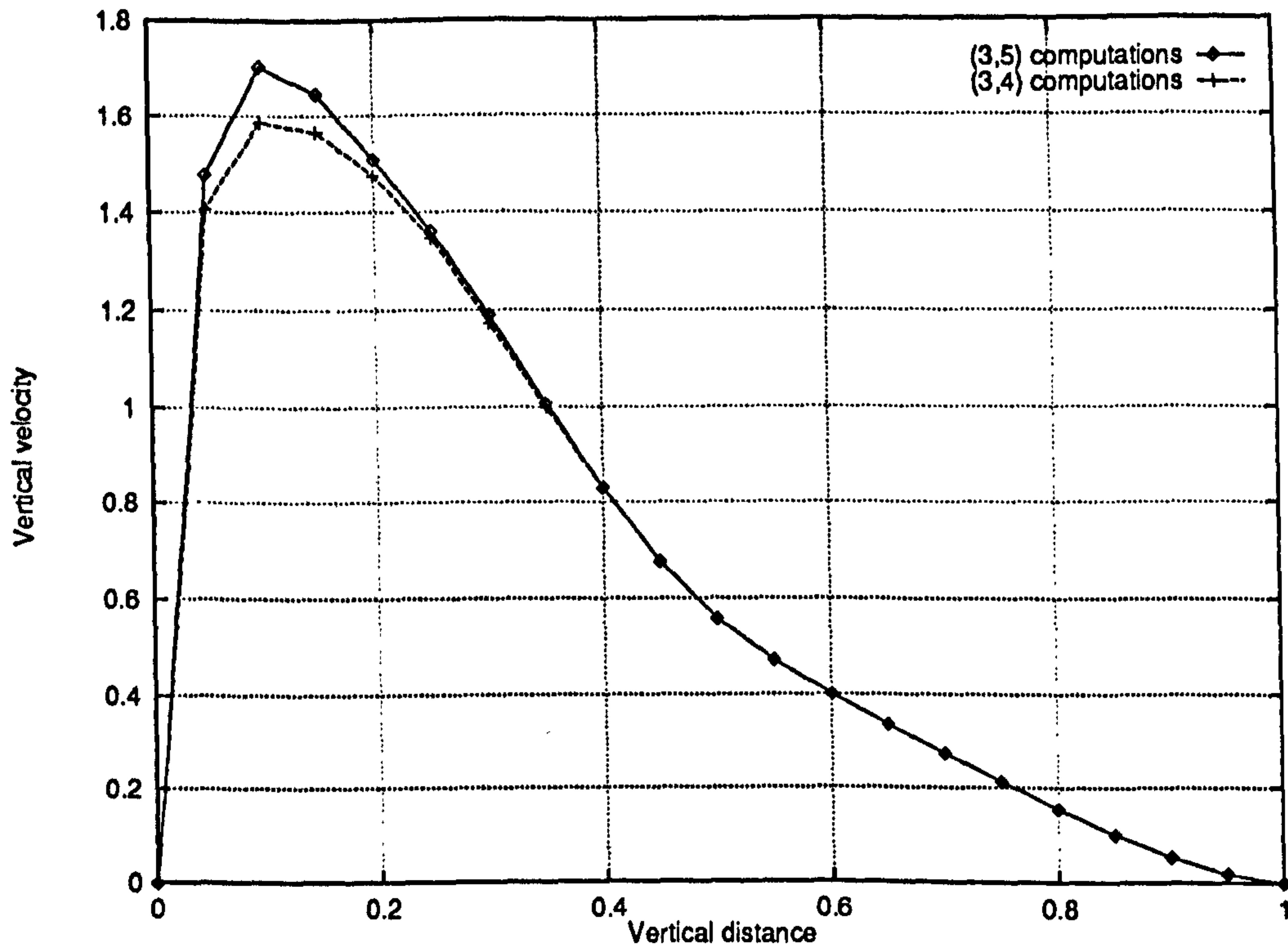


Figure 6.31: Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase I) on different adaptive grids

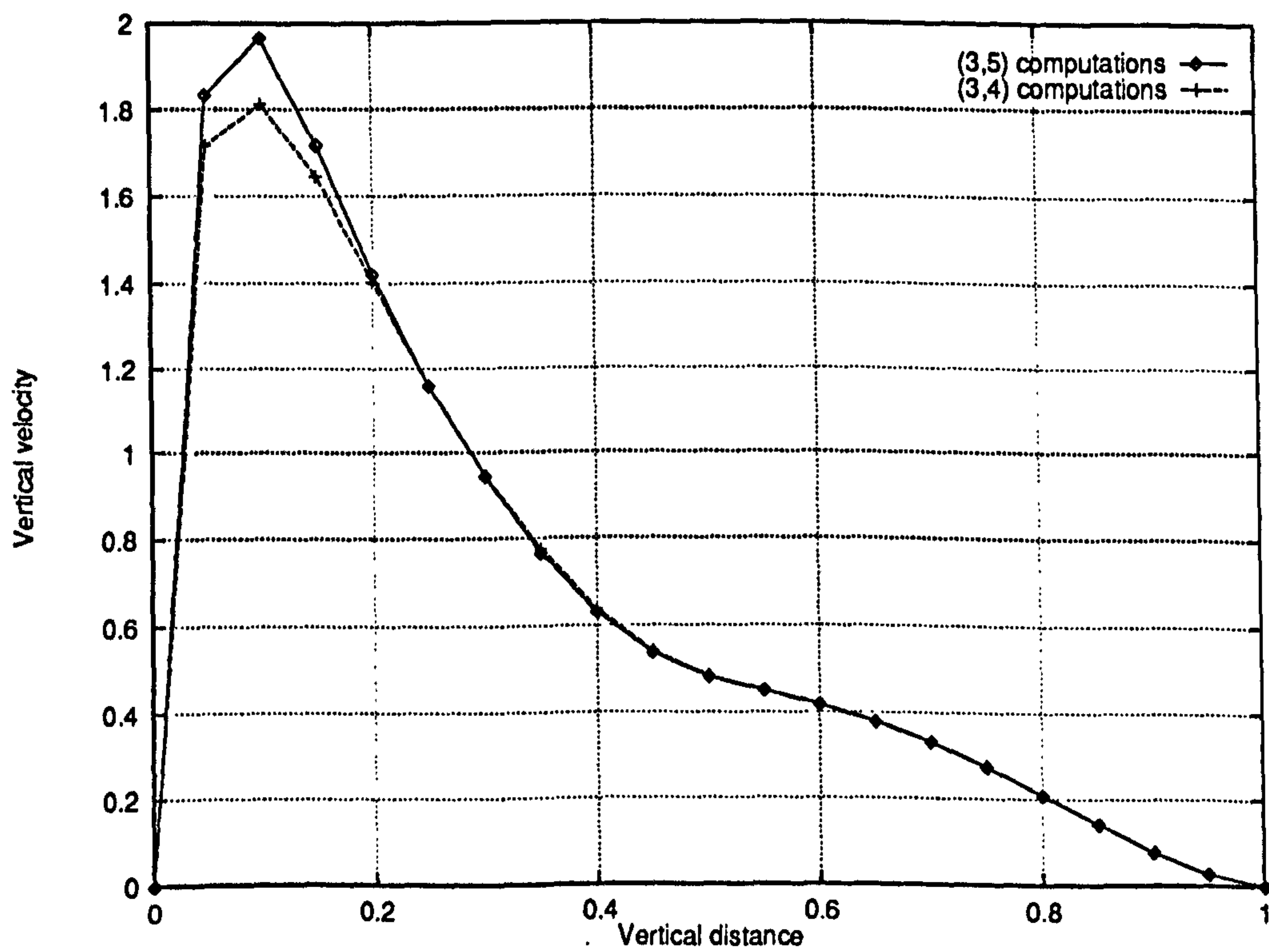


Figure 6.32: Two-phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ (Phase II on uniform and adaptive grids)

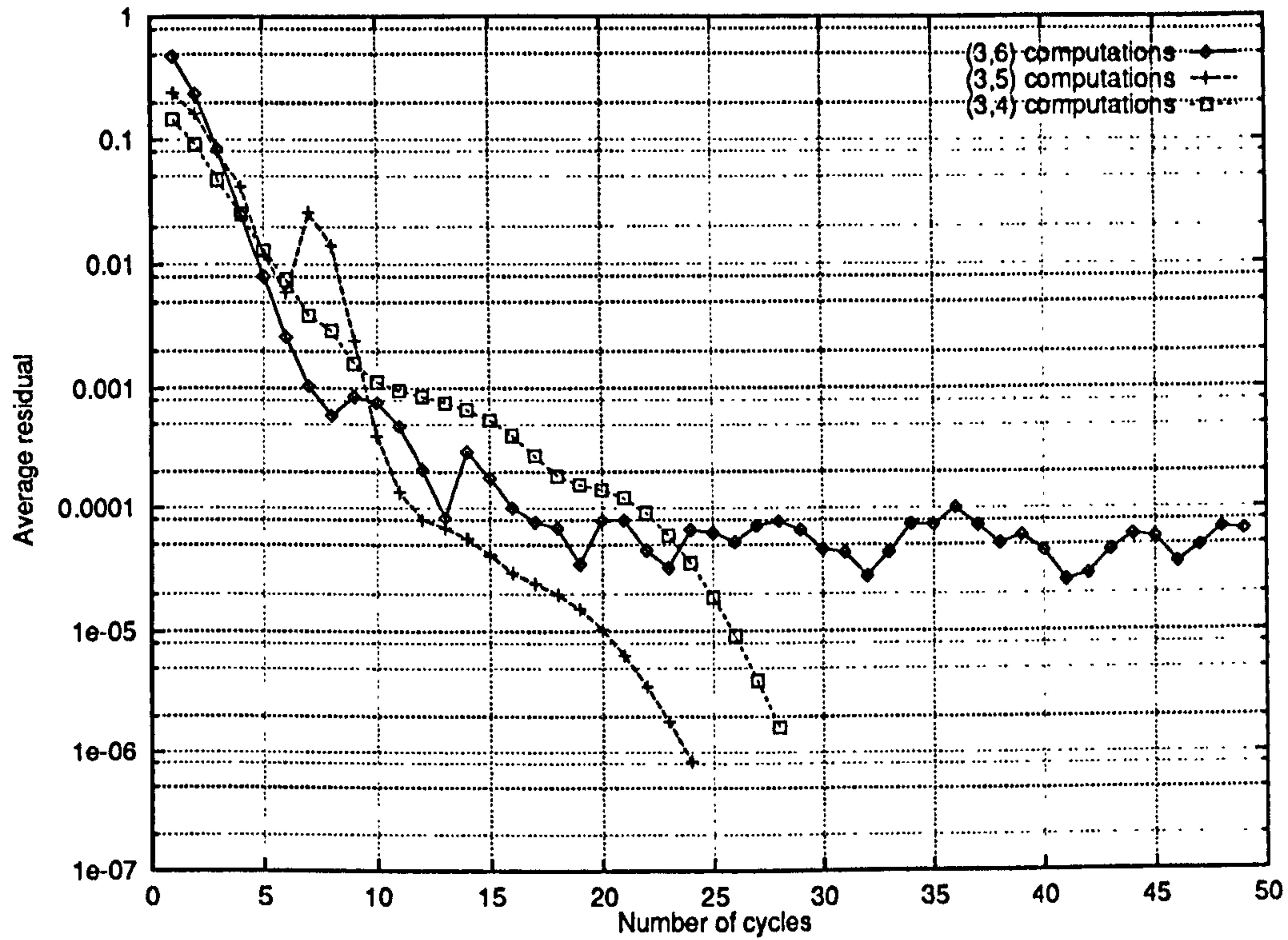


Figure 6.33: Two-phase flow through a T-junction – Convergence histories for different adaptive grid computations and for $\gamma = 2$ – See text for explanations about the convergence history of the (3,6) computation.

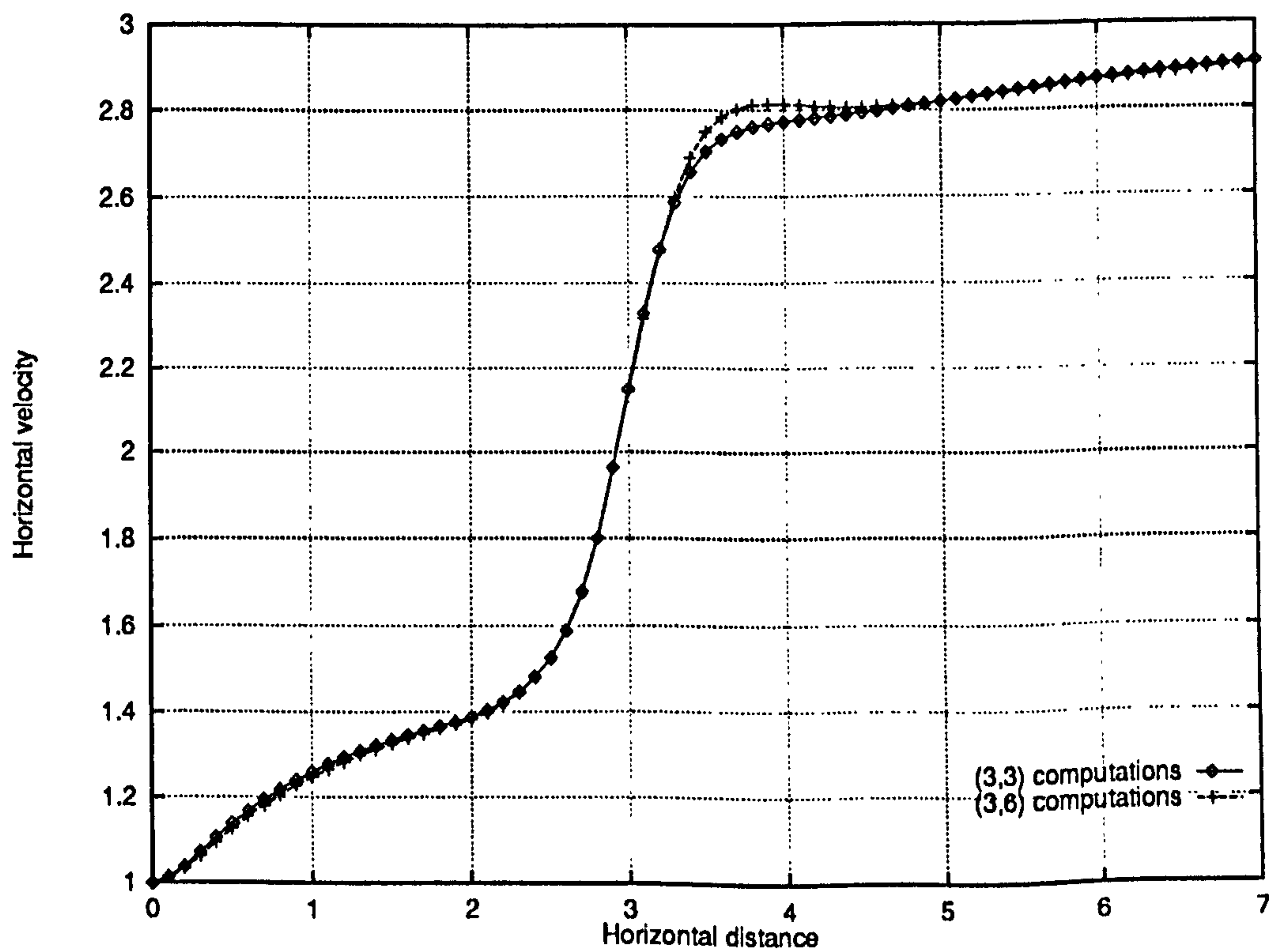


Figure 6.34: Single phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.5$ on uniform and adaptive grids ($\gamma = 2.0$)

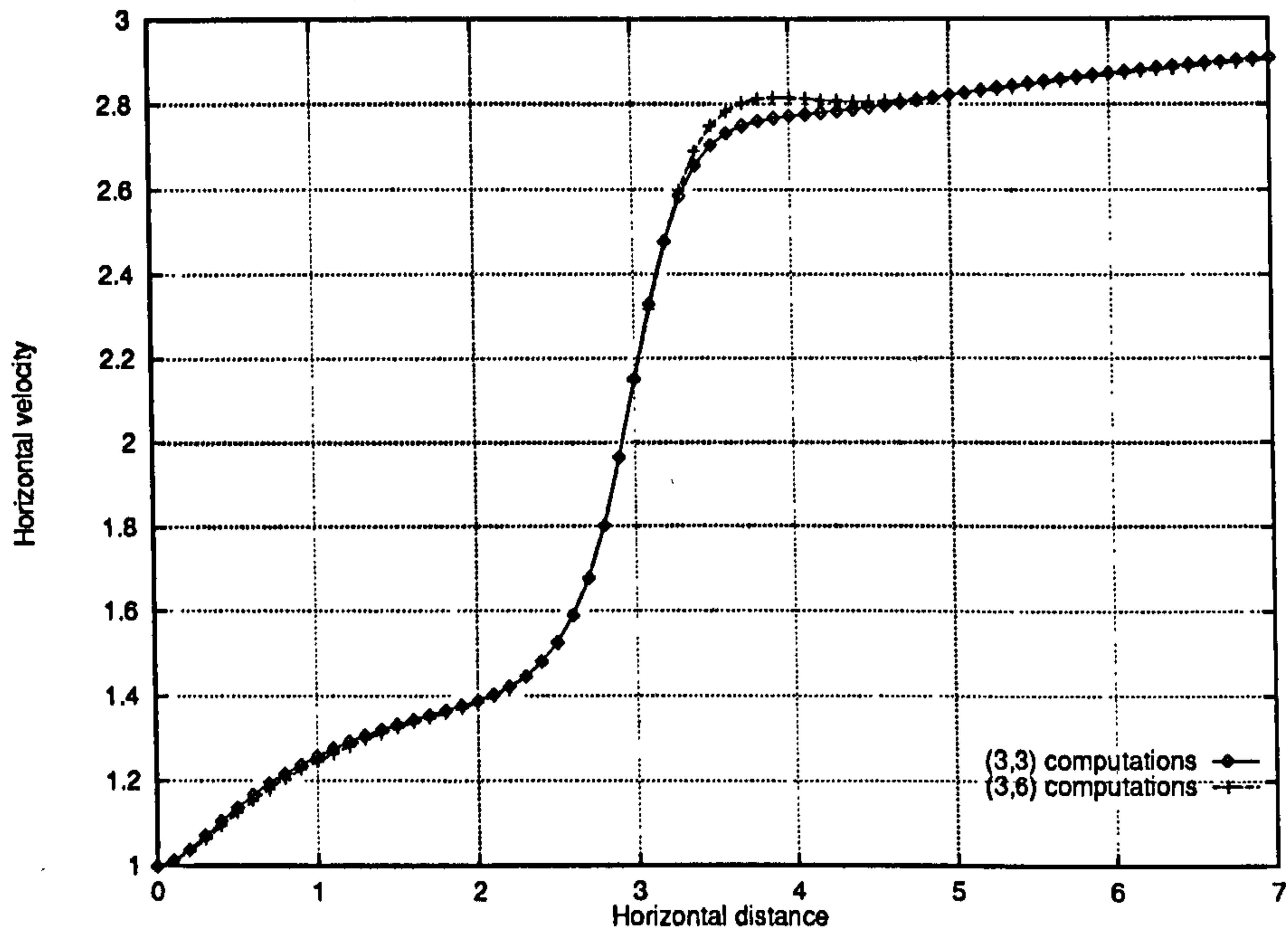


Figure 6.35: Single phase flow through a T-junction – Vertical velocity profiles along the line $x = 3.0$ on uniform and adaptive grids ($\gamma = 2.0$)

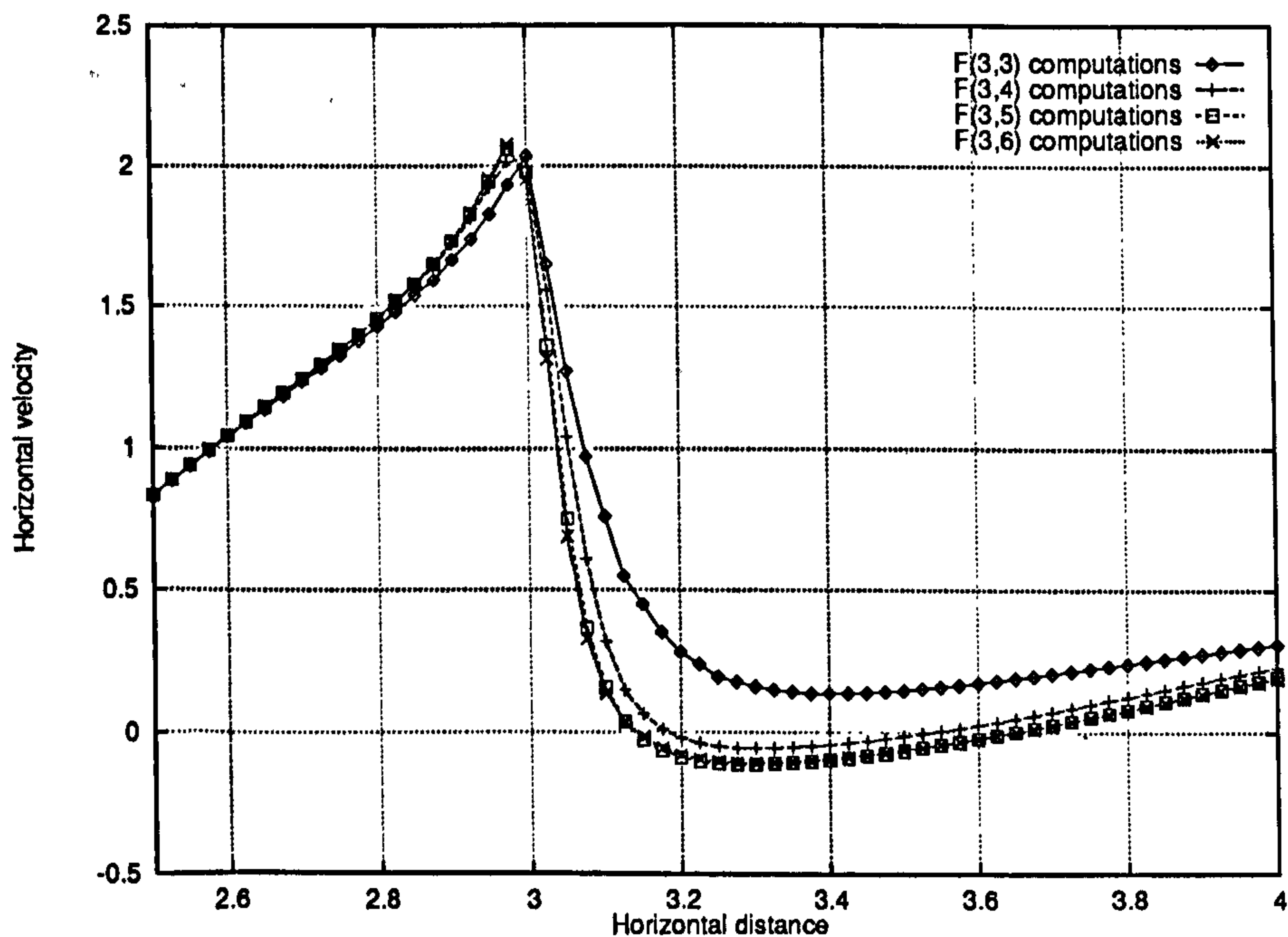


Figure 6.36: Two-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$)

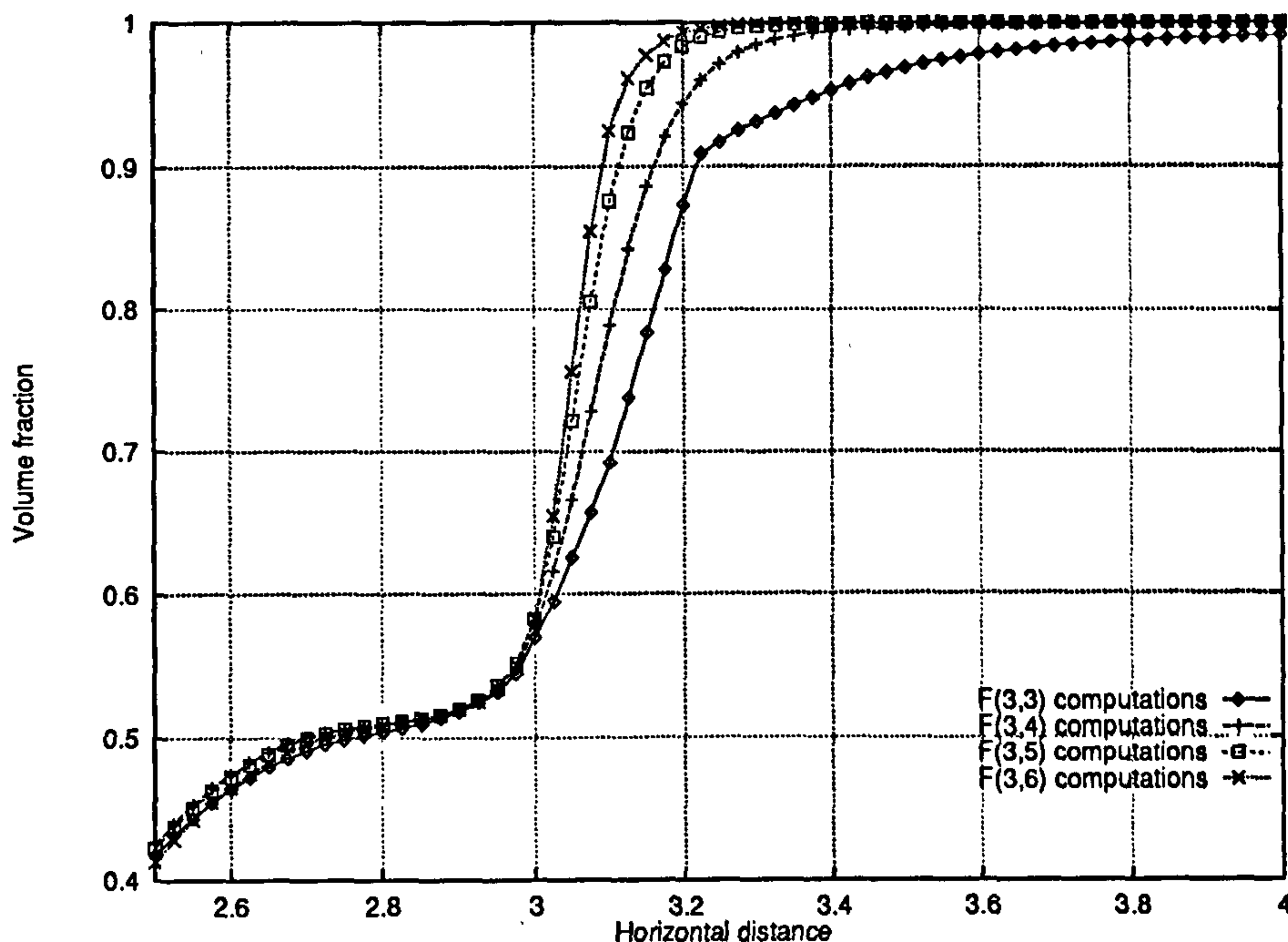


Figure 6.37: Two-phase phase flow through a T-junction - Vertical velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$)

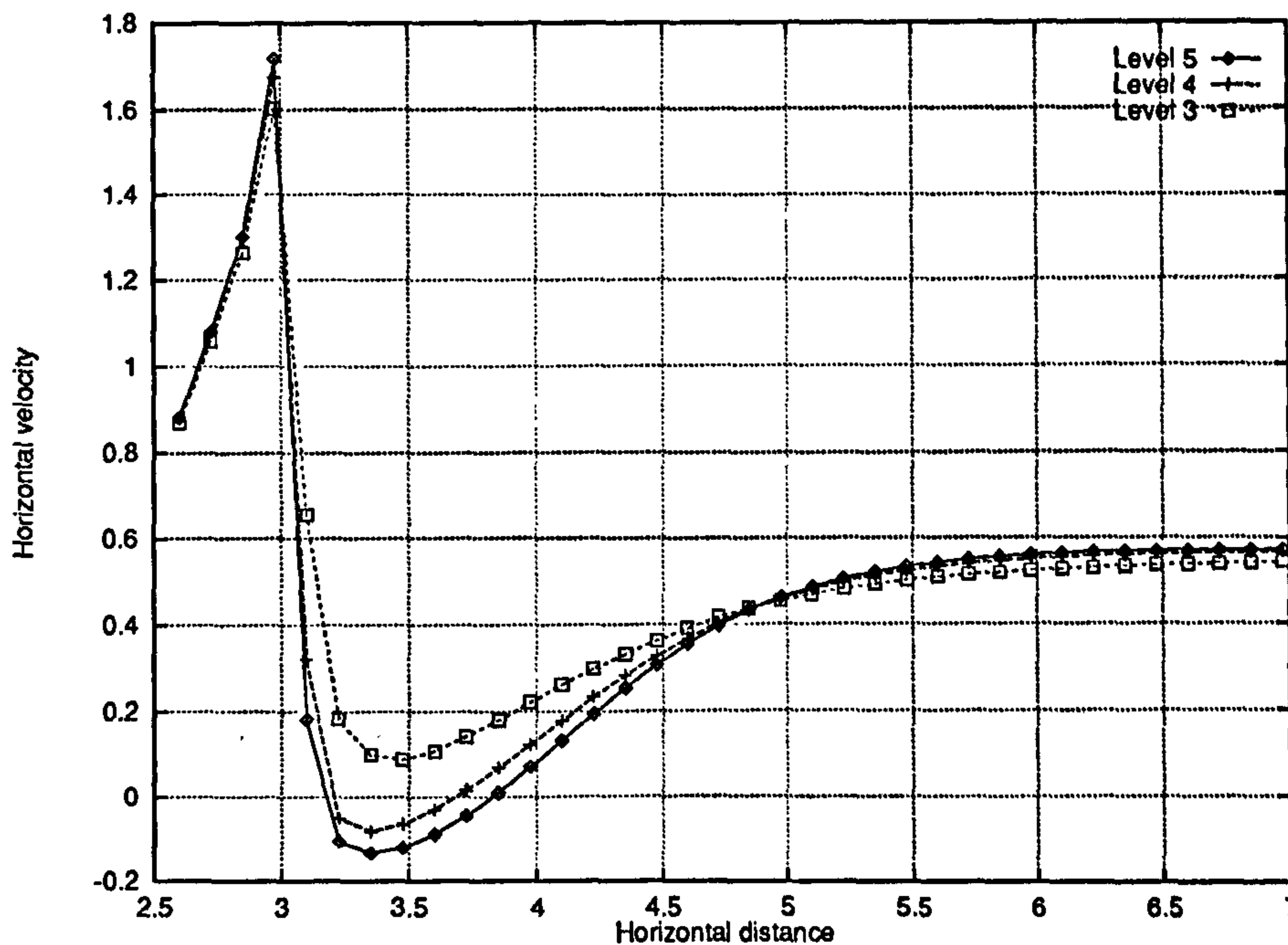


Figure 6.38: Single-phase flow through a T-junction - Horizontal velocity profiles along the line $y = 0.05$ for different levels of uniform grids

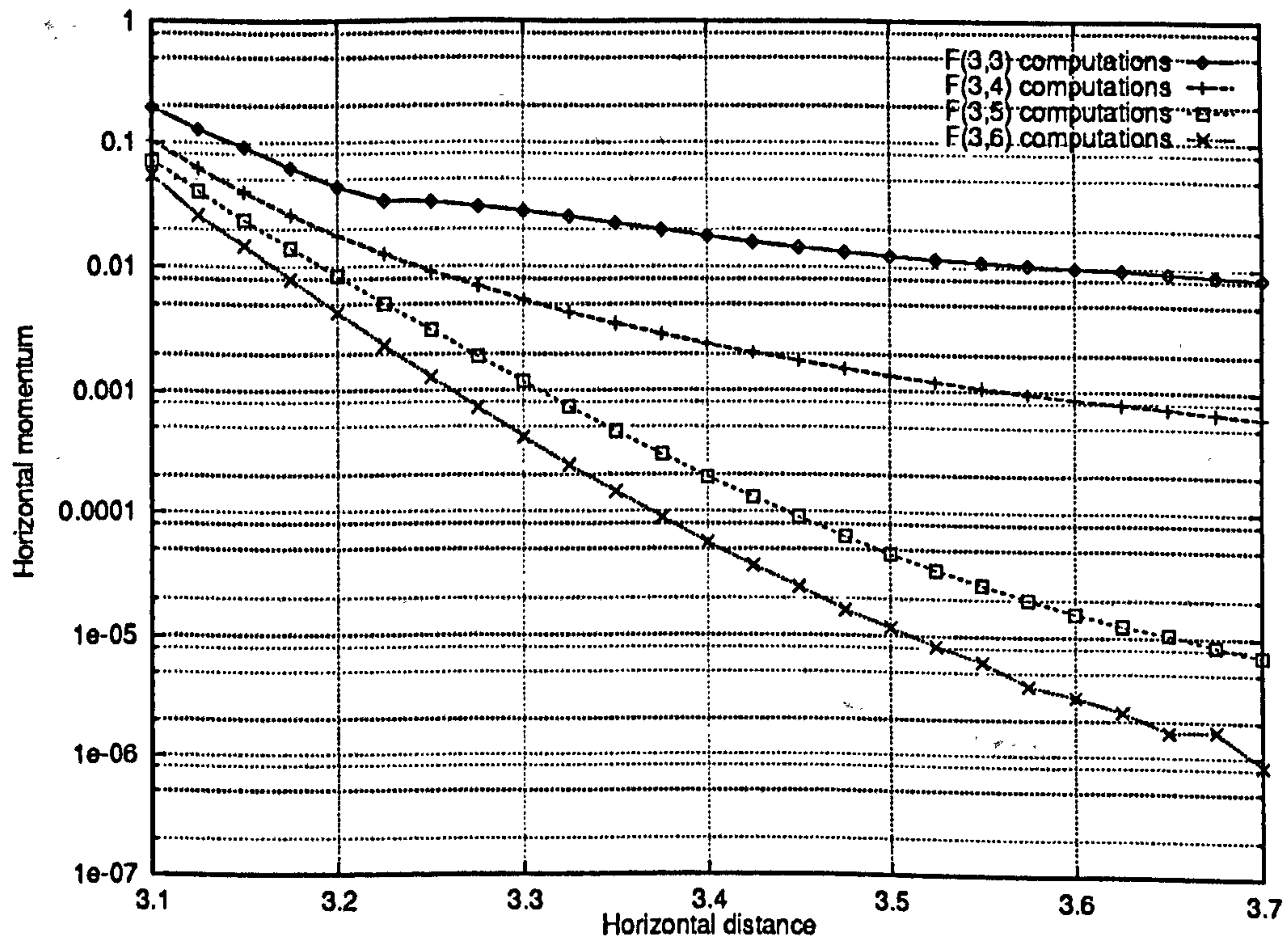


Figure 6.39: Two-phase flow through a T-junction – Horizontal momentum profiles along the line $y = 0.05$ (phase 1) for different adaptive grids ($\gamma = 1$), plotted on a logarithmic scale

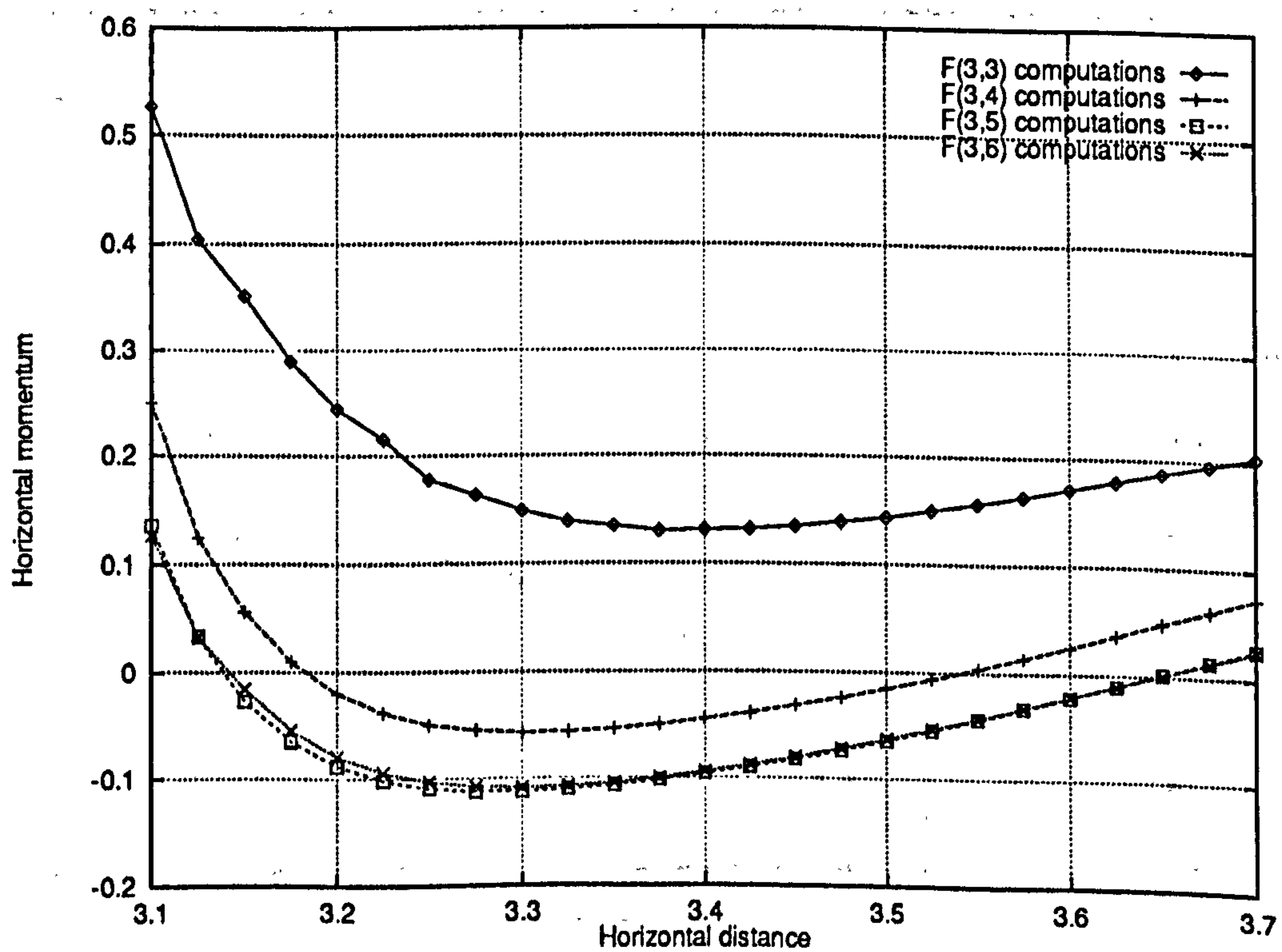


Figure 6.40: Two-phase flow through a T-junction – Horizontal momentum profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$)

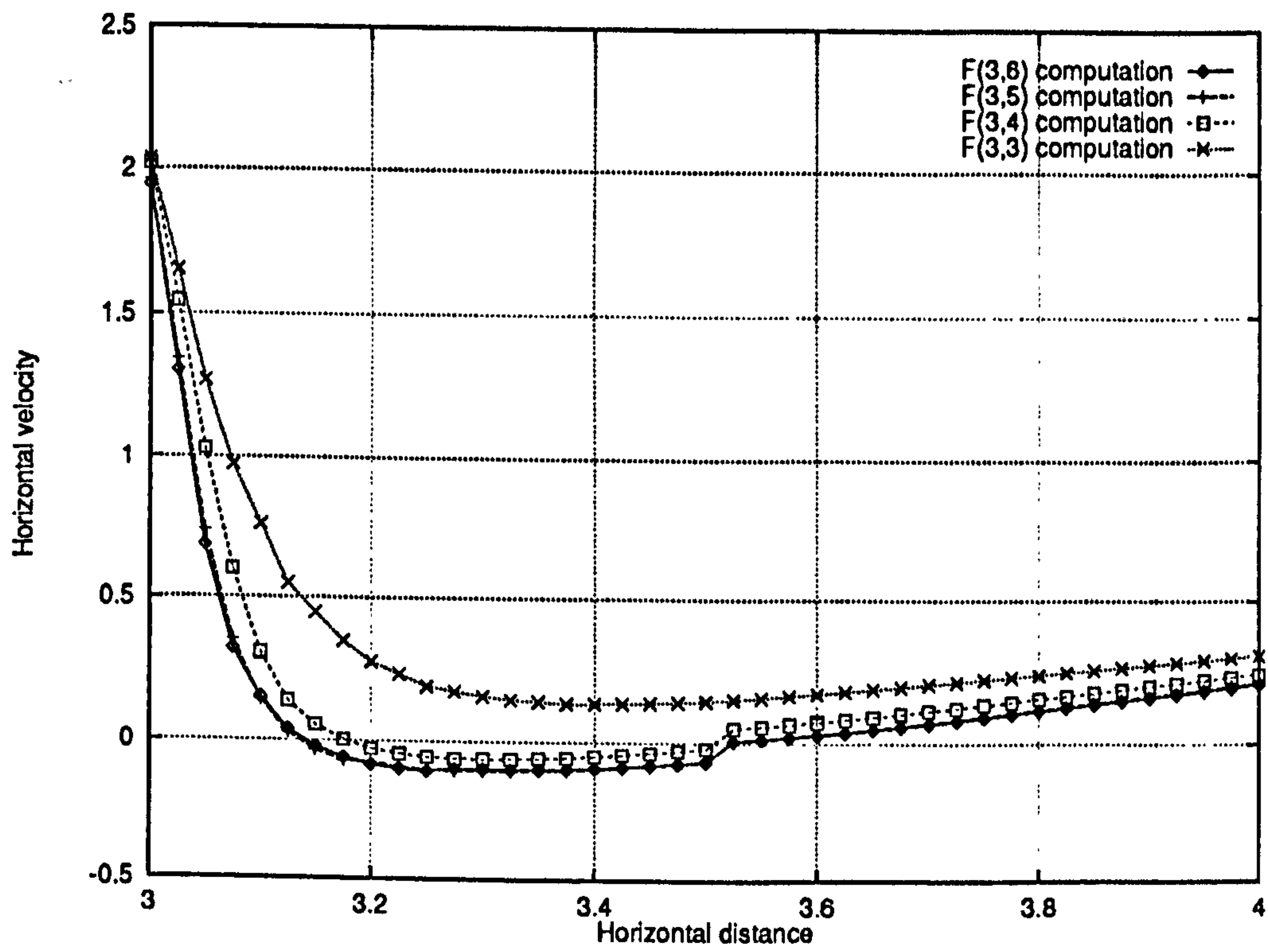


Figure 6.41: Two-phase flow through a T-junction – Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 2$)

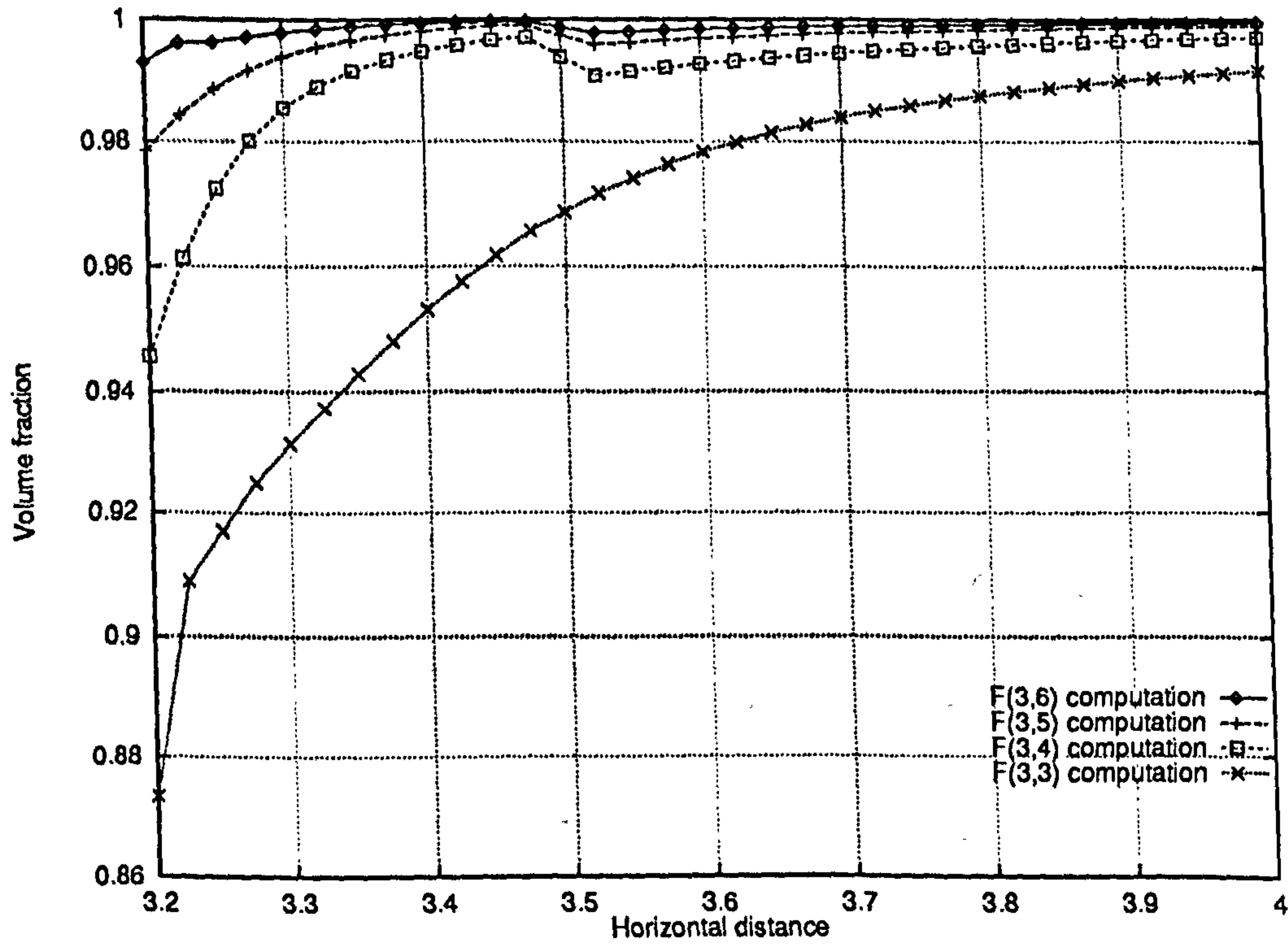


Figure 6.42: Single phase flow through a T-junction - Volume fraction profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 2$)

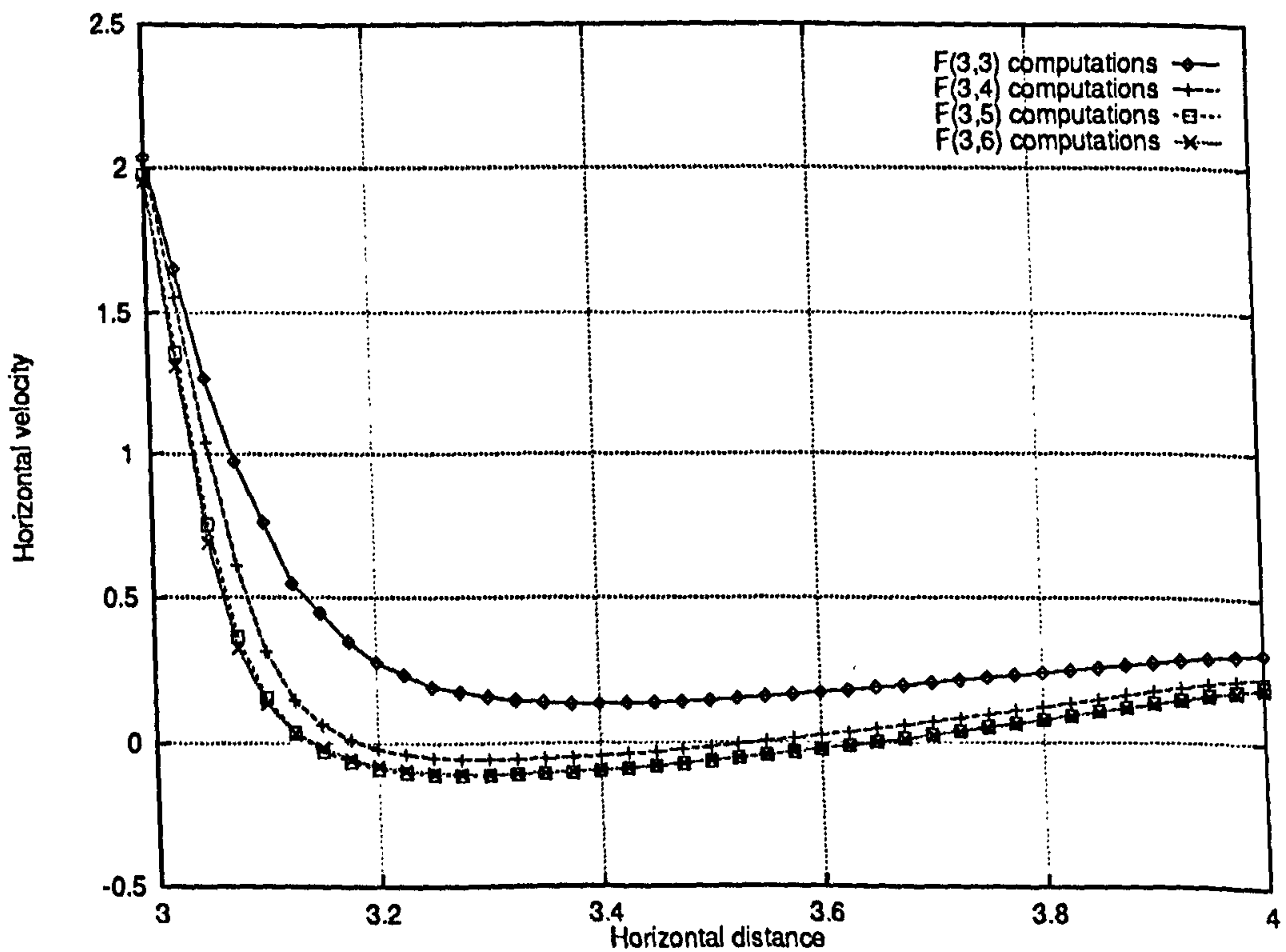


Figure 6.43: Two-phase flow through a T-junction - Horizontal velocity profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$) - Compare with Figure 6.41

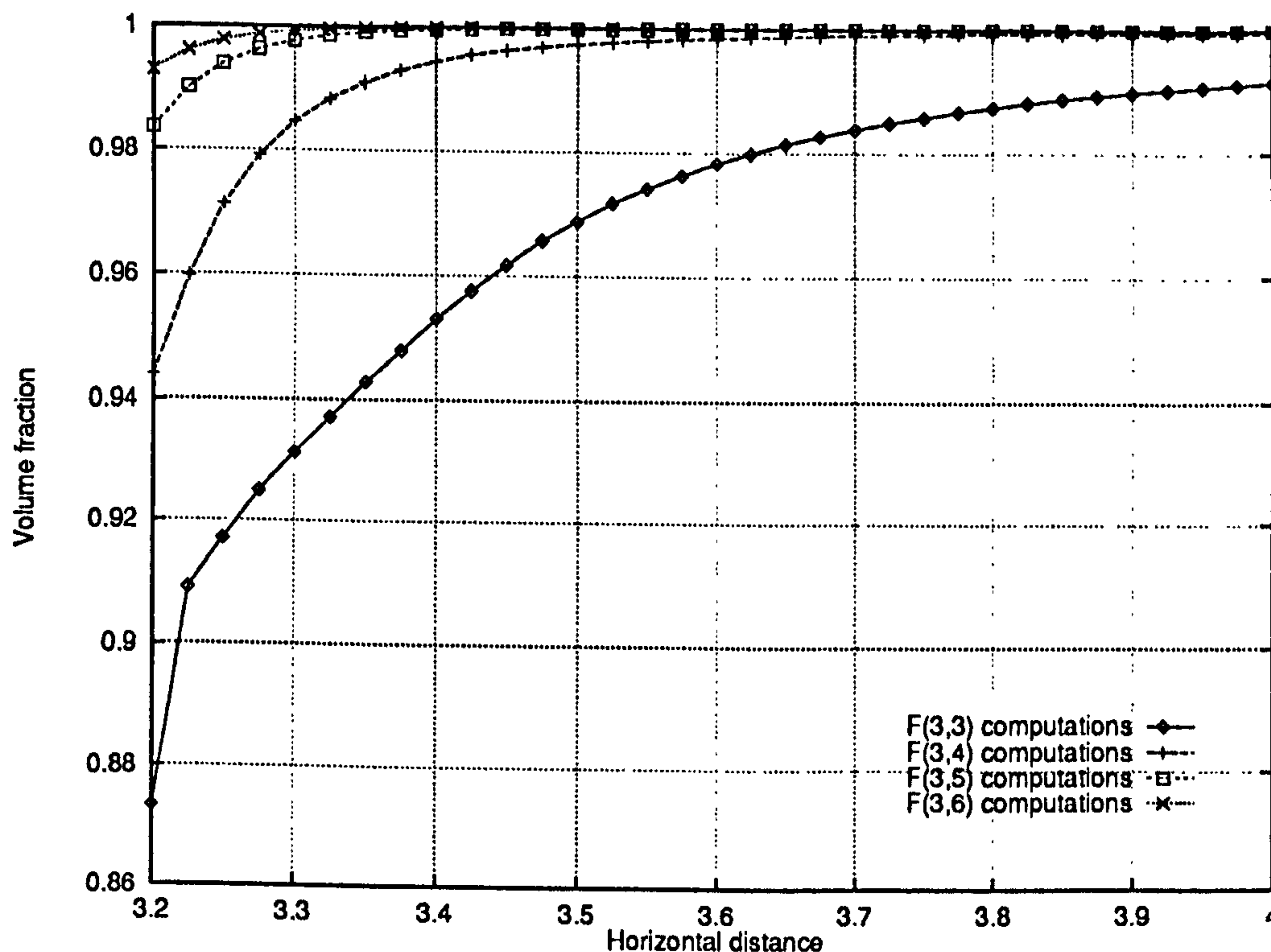


Figure 6.44: Single phase flow through a T-junction – Volume fraction profiles along the line $y = 0.05$ (phase 2) for different adaptive grids ($\gamma = 1$) – Compare with Figure 6.42

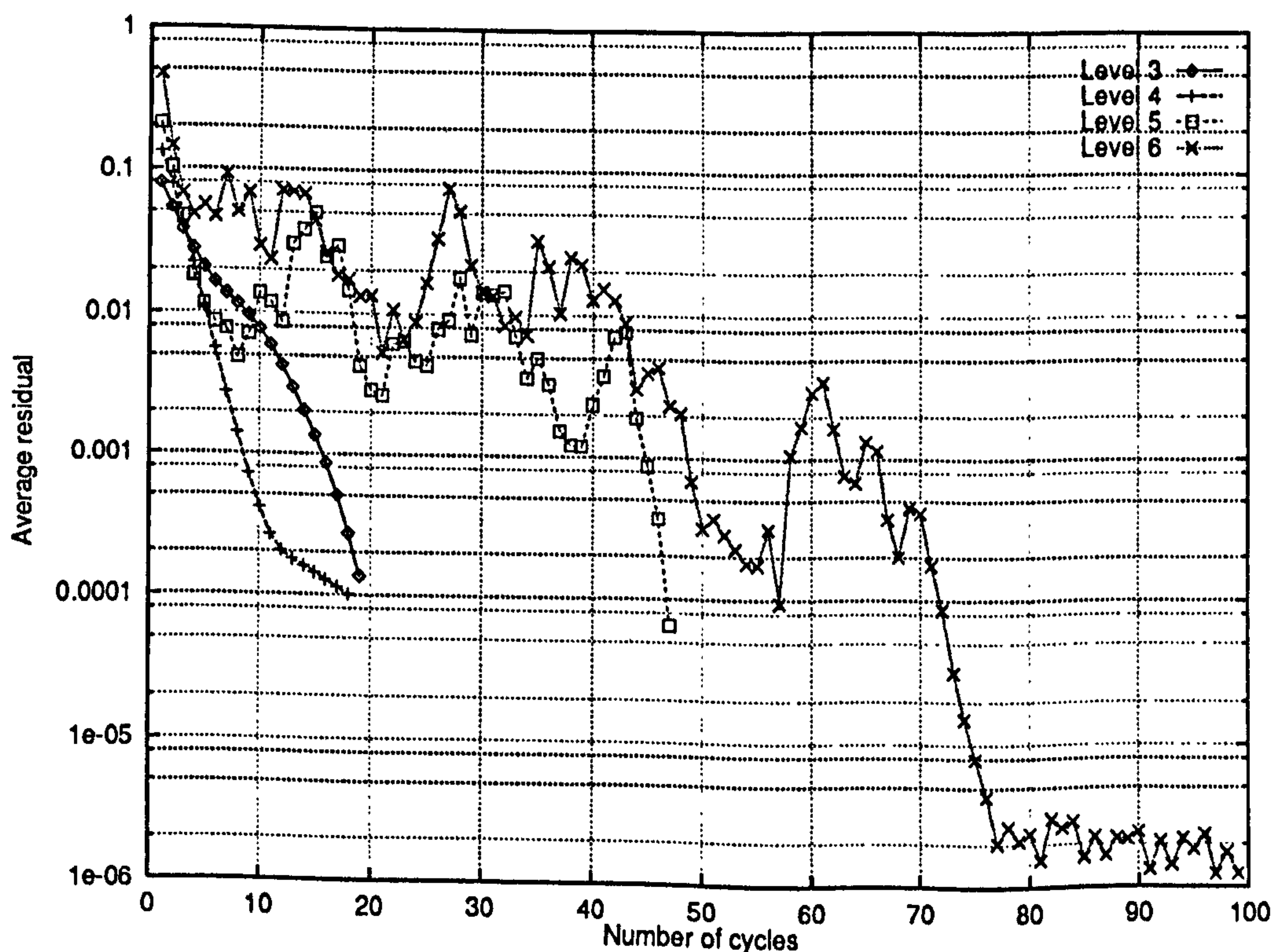


Figure 6.45: Two-phase flow through a T-junction – Convergence rates for an $\Lambda(3,6)$ computation (and intermediate computations) for $\gamma = 1.0$ with the “no-change” strategy for handling the fine grid volume fraction corrections – Compare with Figure 6.33

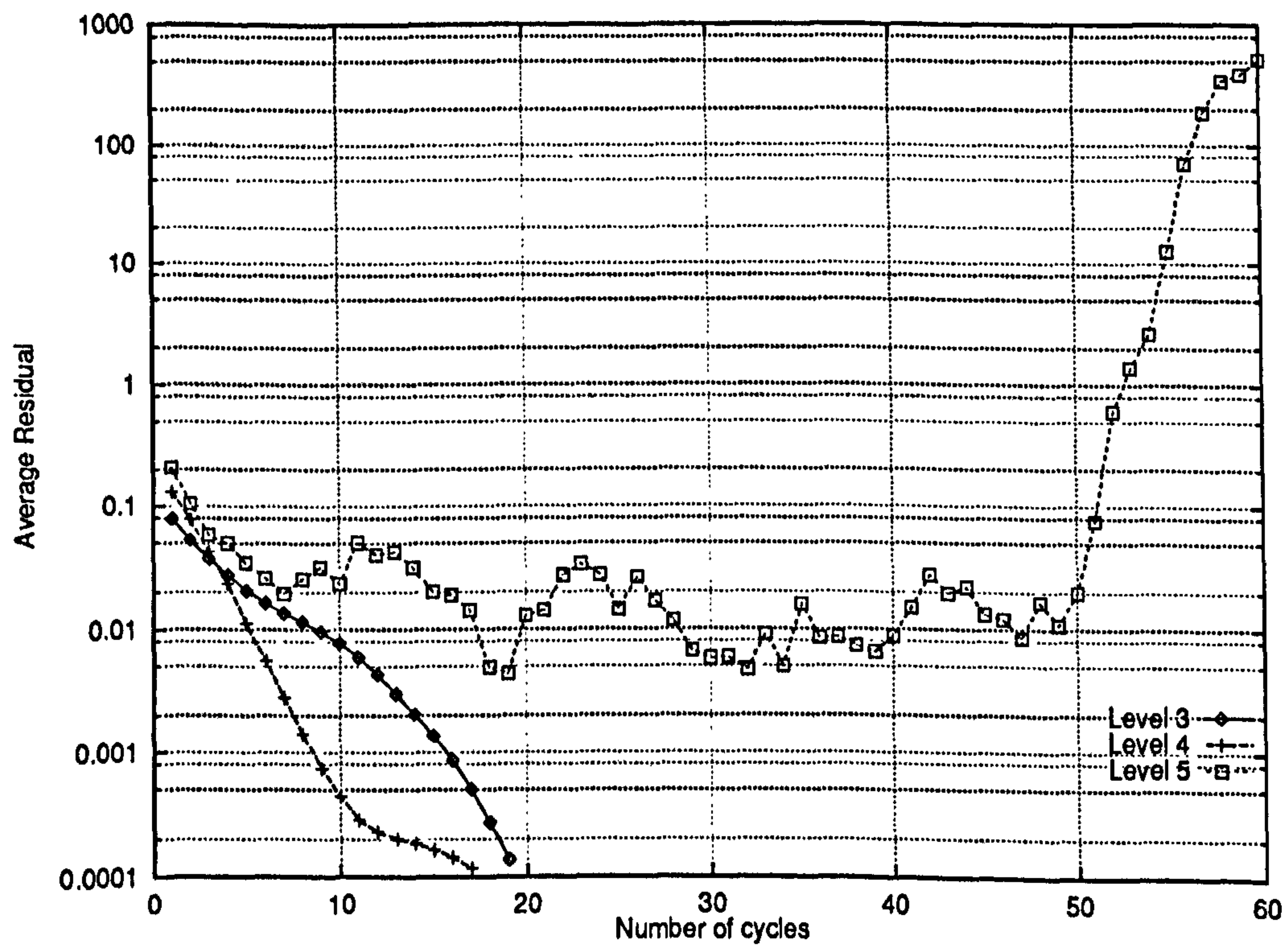


Figure 6.46: Two-phase flow through a T-junction – Convergence rates for an A(3,6) computation (and intermediate computations) for $\gamma = 1.0$ with the “cut-off” strategy for handling the fine grid volume fraction corrections – Compare with Figure 6.45

6.4 Interpolation Procedures, Conservation of Mass and Convergence

The discretisation of the governing equations in the single phase case leads to a strict conservation of numerical mass fluxes on any collection of cells. Furthermore, the usual restriction operators conserve mass fluxes at cell boundaries. Hence, if the prolongation operator is not conservative, mass will be created or destroyed at grid interfaces on adaptive grids and achieving convergence is impossible. This effect does not apply to uniform grids because here grid transfers only affect the quality of the coarse grid correction, not the fine grid solution itself.

For multiphase flows, the mass fluxes at cell edges are dependent on two variables, the velocities and the volume fractions. The latter are not available at cell edges. Interpolation is therefore necessary. As a result it is impossible to conserve mass fluxes across grids if the volume fractions and the velocities are transferred independently. This is the case for the pang operators and their direct extension in pang-multiphase. Neither the restriction nor the prolongation are conservative and yet adaptive computations based on these transfers have been shown to converge. The simplest explanation is that the errors introduced in the prolongation are balanced by the errors arising from the restriction and vice-versa. A direct consequence would be that a single phase solver with non-conservative restriction and prolongation could also converge on composite grids. This hypothesis has not yet been tested. We remark, however, that for multiphase flows, due to the presence of the volume fractions, there is an extra degree of freedom in determining the mass fluxes. This is further accentuated by the use of upwind interpolation for the volume fractions in the mass fluxes, so that the numerical coupling between mass fluxes on one cell is reduced (see Figure 6.47).

Interpolation procedures which conserve mass fluxes for multiphase flows (see Section 4.3.3, page 106) are therefore not necessary. They may be more accurate, but their use in multiphase simulation is limited by the fact that they become badly conditioned in regions of “dry-out”.

The conservative interpolation procedures rely on transferring the mass fluxes instead of the velocity fields in a conservative manner. The velocities, which are needed by the solver, are obtained from the mass flux field by dividing it by the suitably interpolated volume fraction field. See equations (4.48) and (4.49):

$$u_{i+1/2,j} = \frac{2q_{i+1/2,j}^u}{\left((1 + \text{sgn}(q_{i+1/2,j}^u))r_{i,j} + (1 - \text{sgn}(q_{i+1/2,j}^u))r_{i+1,j} \right)}$$

$$v_{i,j+1/2} = \frac{2q_{i,j+1/2}^v}{\left((1 + \text{sgn}(q_{i,j+1/2}^v))r_{i,j} + (1 - \text{sgn}(q_{i,j+1/2}^v))r_{i,j+1} \right)}$$

As soon as the volume fraction approaches zero, the procedure becomes badly conditioned and large velocities could be generated.

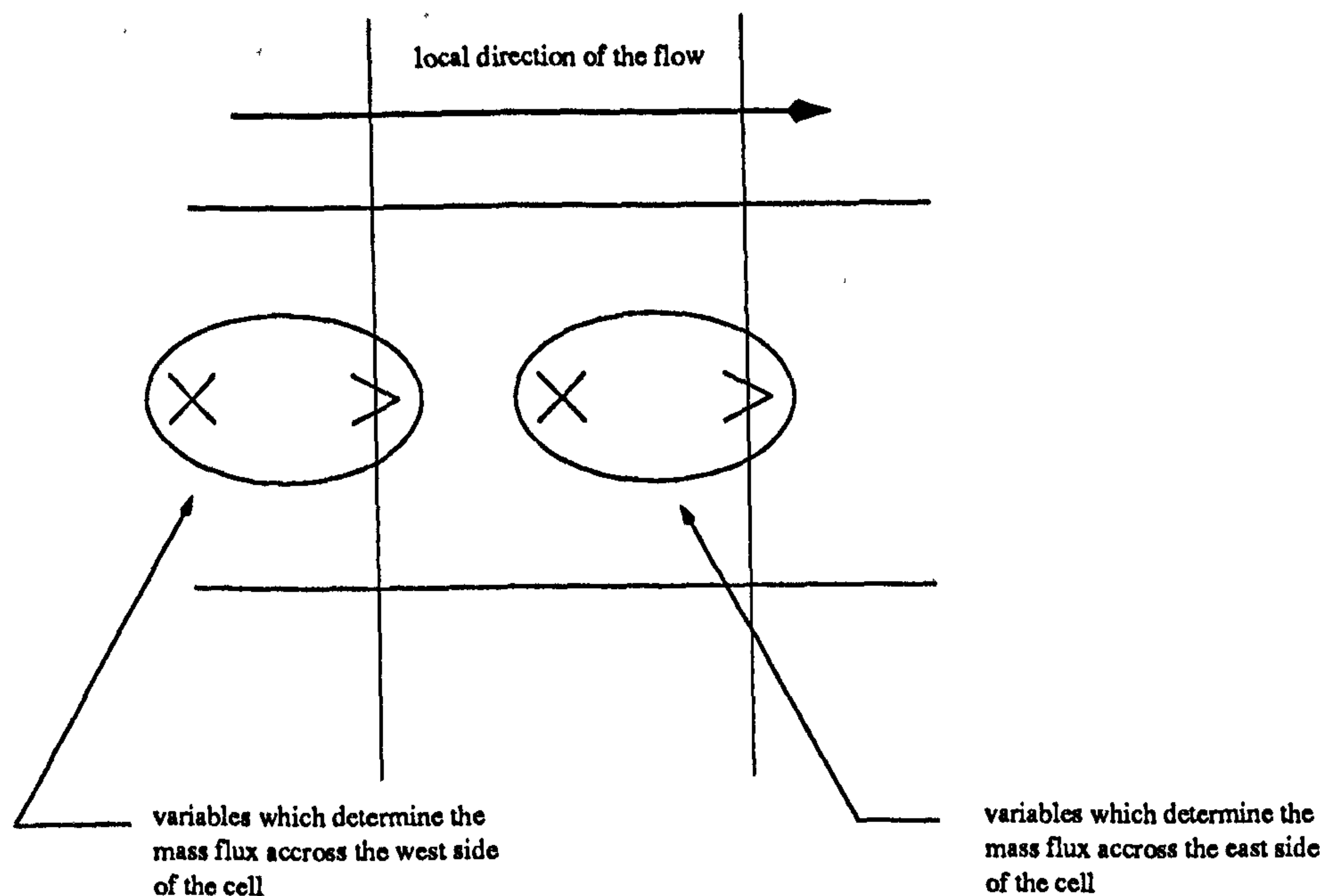


Figure 6.47: Numerical coupling introduced by the computation of horizontal mass fluxes through a computational cell

This is an effect that has indeed been encountered. For the channel flow problem, if conservative interpolation is used, adaptive (3,4) computations converge without problems but adaptive (3,5) computations do not. This occurs because the volume fraction field near the sides of the channels approaches zero for one of the phases at level 5, but not at level 4. Similarly, for the multiphase flow through a T-junction, there is a significant region where the volume fraction approaches zero on all levels, and consequently, it has not been possible to obtain convergence, even on uniform grids with the conservative interpolation procedures.

The solution to this last problem may be to rewrite the solver in terms of momentum instead of velocities and/or reformulate the equations in regions when the equations become badly conditioned.

6.5 Estimation of the Performance Gain from Adaptive Gridding

The previous sections have shown that provided some care is taken in obtaining a first sufficiently accurate uniform grid approximation of the solution, the refinement algorithm described in Section 4.2.4 works very well and maintains accuracy compared to the equivalent uniform grids. In this section, we want to quantify the savings in computational costs which can be achieved by using adaptation.

We consider two measures: (i) execution time and (ii) memory requirement. In order to measure execution time independently from the architecture and the level

of optimisation, we measure the cost of an algorithm in terms of work unit defined as before as the cost of one relaxation sweep on the *uniform* grid associated with the largest grid level considered in a given computation. This measure should be directly proportional to raw execution time⁶ We then define the *speed-up* as the ratio of the computational work on adaptive grid by the computational work on the “equivalent” adaptive grid. Table 6.2 summarises the results. The work unit considered at a given level is one relaxation sweep on the *uniform* grid. Note that in the computation of the adaptive work, the intermediate computations aimed at estimating the truncation errors have been included.

Problem	Level	Adaptive strategy	Computational Work		Speed-up
			Adaptive	Uniform	
Channel flow	4	(3,4) Non-conservative	168	222	1.32
Channel flow	5	(3,5) Non-conservative	173	366	2.12
T-junction	4	(3,4) Non-conservative	133	336	2.53
T-junction	5	(3,5) Non-conservative	68	612	9.00

Table 6.2: Comparison of computational work required to solve the channel problem and the T-junction problem on adaptive and uniform grids

We now consider memory requirements. The obvious measure to adopt here is the total number of patches created during the course of the computation for all levels, since the amount of information which needs to be stored varies linearly with the number of grid points defined. Results are summarised in Table 6.3.

Problem	Level	Adaptive strategy	Patches created		Savings
			Adaptive	Uniform	
Channel flow	4	(3,4) Non-conservative	129	255	49.4 %
Channel flow	5	(3,5) Non-conservative	345	1023	66.3 %
T-junction	4	(3,4) Non-conservative	237	765	69.0 %
T-junction	5	(3,5) Non-conservative	288	3069	90.6 %

Table 6.3: Comparison of memory required to solve the channel problem and the T-junction problem on adaptive and uniform grids

Both in terms of computational costs and memory requirements, the results shown above clearly indicate that adaption is of great benefit. We have observed speed-ups by one order of magnitude combined with a 90 % savings on memory requirements. Since the refinement strategy is automatic, these benefits are immediately available: no skilled user input is required to define the refinement pattern. Although a value for the parameter γ has to be supplied, it is not necessary to have detailed a-priori knowledge about the expected solution.

Admittedly, savings by one order of magnitude correspond to the T-junction problem which is very well suited to adaptive computations since regions of high truncation

⁶Reduced paging for adaptive computations may reduce the execution times further. It should be noted however that the most CPU intensive part of the algorithm – that is the local quasi-Newton solver – is local and therefore not very susceptible to paging delays.

error are highly localised. Grid refinement can be very selective, hence the high quality of the results reported here. The benefits of adaption are not as dramatic in the case of the channel flow. This is mainly due to the fact that the truncation errors are distributed more evenly across the computational domain. Although this is a challenging situation for a refinement algorithm, we conclude that adaption is still clearly beneficial.

It can further be noted that the gains in terms of computational costs are not as significant as the memory savings. In the author's opinion, this reflects the cost of estimating the truncation error. The intermediate computations are relatively costly since they are performed at levels of accuracy for which the convergence rates are in a phase of plateau. This limitation does not alter our conclusion that adaption is very cost-effective.

6.6 Conclusions

In Chapter 5, we assessed the accuracy and efficiency of the `pamg-multiphase` flow on a range of important multiphase test problems. We have now considered the same test cases, with the exception of the backward-facing problem, on adaptive grids.

Adaption greatly complicates the implementation of the solution algorithm (although far less so than for transient computations). Handling of complex data structures is required to perform even basic management grid tasks. We have built on the framework provided by the original `pamg` to implement adaptive solution for multiphase flows, but the extension was not trivial. The main difficulty was in (i) designing grid transfer operators which conserve multiphase mass fluxes and (ii) implementing them in the existing framework. This was carried out successfully. In particular, the algorithm for providing internal boundary conditions at grid interfaces had to be thoroughly overhauled.

It is a major conclusion from single phase studies that the simulation of incompressible flows on adaptive grids requires the multigrid transfers to conserve the mass fluxes between grids. Otherwise, the discrete equations will not admit any solution. On staggered grids, enforcing mass conservation for multiphase flows is difficult since they involve both velocities and volume fractions which are not defined at the same points. To overcome this difficulty, the multigrid transfers have been rewritten so that mass fluxes and volume fractions are interpolated rather than velocities and volume fractions. Mass conservation is automatically ensured (using the `pamg` operators) and the velocities fields are defined in a unique way. Hence, no modifications are necessary on the basic quasi-Newton solver. We have referred to these new interpolation operators as the *conservative* formulation while the original `pamg` operators written in terms of velocities and volume fraction are called the *non-conservative* formulation.

Firstly, we compared the conservative and non-conservative interpolation operators. Quite surprisingly, the non-conservative formulation did not prevent the convergence of the solution algorithm. Their success is due to the fact that neither the prolongation nor the restriction are conservative so that errors can cancel out. Furthermore, our discretisation of the continuity which makes use of upwind interpolation for the volume fractions, is such that the numerical mass fluxes on any given cell are strongly decoupled, i.e. the mass fluxes at different locations involve solution values defined at different grid locations. This may facilitate the compensation of the errors. We have actually favoured the non-conservative operators since the conservative formulation can become badly conditioned in “dry-out” zones, i.e. regions where the phases are almost completely separated.

Adaptive computations with non-conservative interpolation were shown to yield accurate results compared with solutions on uniform grids. It can be concluded that:

1. The pang-multiphase implementation of adaption is essentially correct;
2. As in single phase flows, the FAS defect which approximates the truncation error of the discretisation is a good refinement indicator: refinement is performed in physically correct locations
3. The errors associated with the non-conservation of mass at boundary interfaces are small.

Adaptive computations provide a similar level of accuracy at a much reduced computation cost, both in terms of execution time and in terms of memory requirement. Gains of one order of magnitude have been observed for both measures and for suitable problems i.e. problems where the truncation errors are relatively localised. Even for problems where they are more evenly distributed (e.g. the two-phase channel flow), adaption is very clearly beneficial.

Some care is needed in order to generate correct grid refinement. The preliminary uniform grid has to be fine enough to be representative of the exact solution. Furthermore, the parameter γ has to be given a value which is sufficiently small to avoid the refinement procedure becoming too selective. The correct choices depend on the characteristics of the problem and particularly on the spatial distribution of the truncation errors. However, the level of skill required from the user in order to successfully use adaption is minimal. In particular, detailed knowledge of the expected solution is unnecessary in order to perform the gridding. Hence the benefits of adaption in terms of computational cost are very easily available.

The gains in memory usage and computational time allow the study of more demanding problems than if the code is restricted to uniform grids. We have already made use of this facility and computed a fine grid solution for the multiphase T-junction and have gained further understanding of the behaviour of the multiphase solutions on fine grids.

6.7 Uniform and Adaptive Grids for the pamg and pamg-multiphase Solvers

The following figures show the computational grids used by the pamg and pamg-multiphase solvers for the different test cases presented in Chapters 5 and 6. They represent *quadrants*⁷, i.e. groups of 2×2 cells and *not* individual cells. Furthermore the aspect ratios between the vertical and horizontal distances are only approximately correct.

6.7.1 Channel Geometry

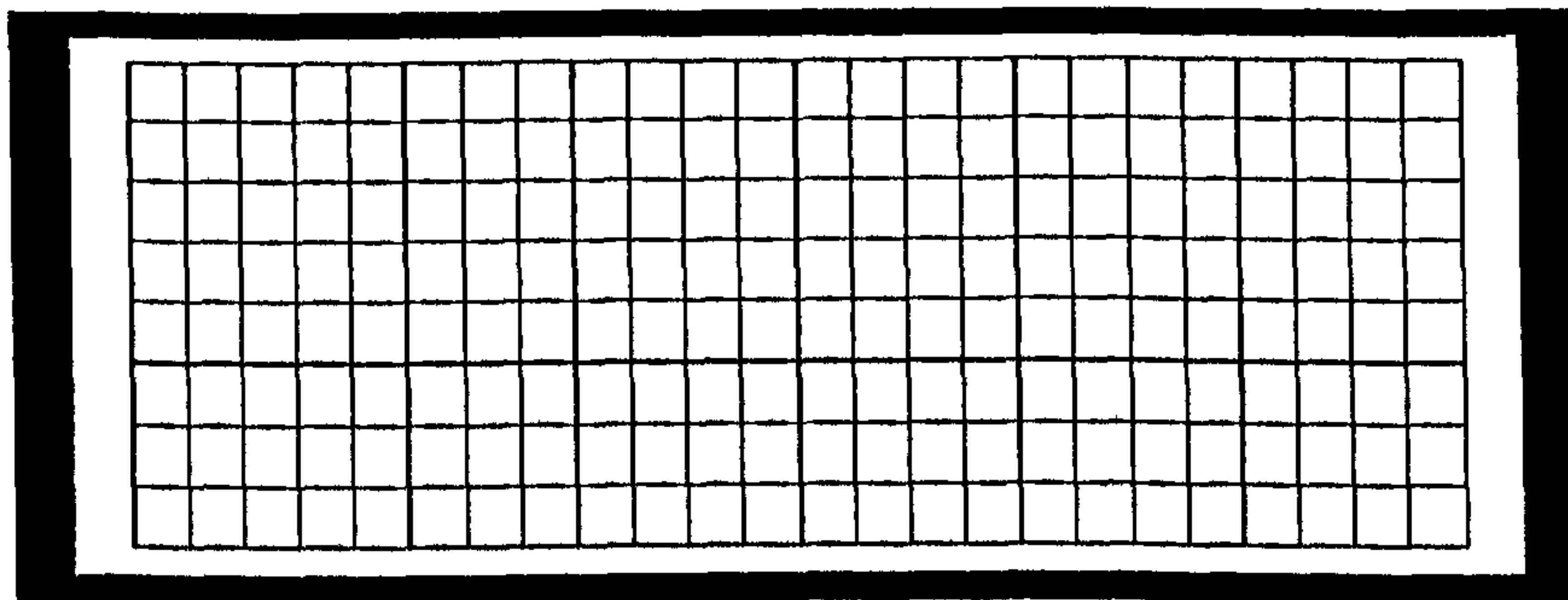


Figure 6.48: Uniform grid (level 3) for the channel problem

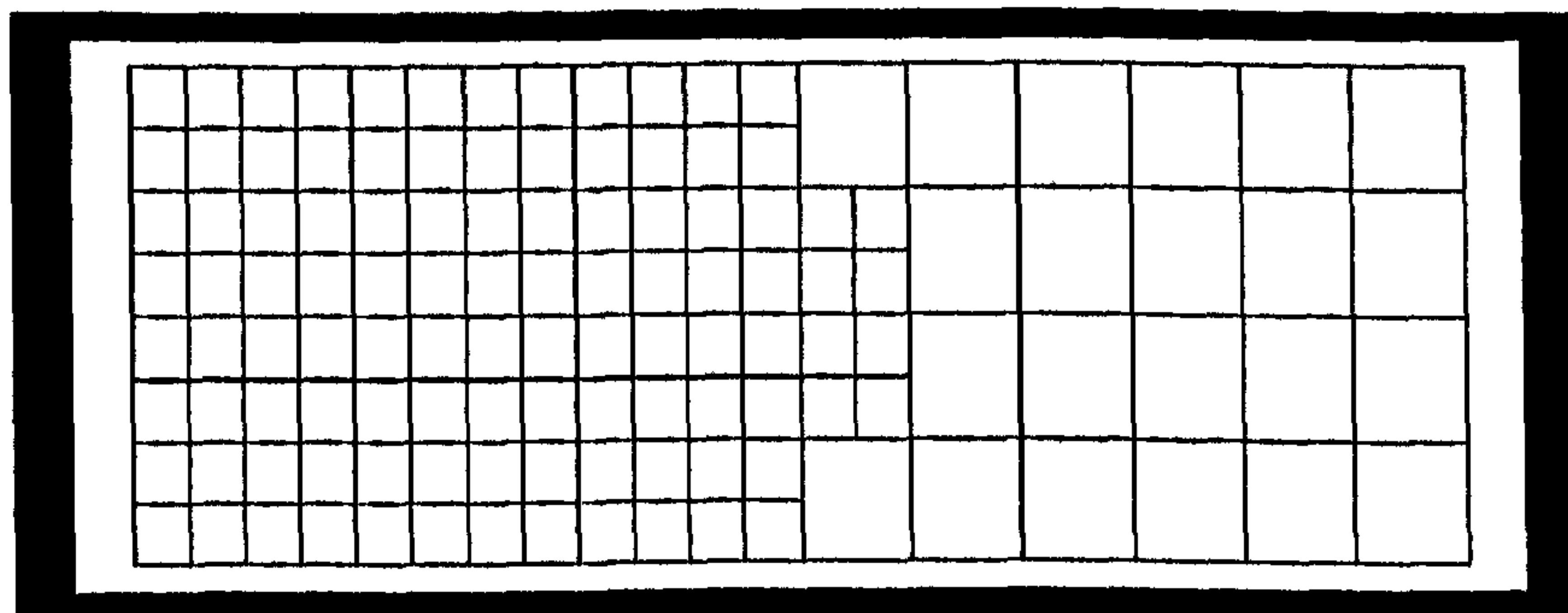


Figure 6.49: Adaptive (2,3) Grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$

⁷See Section 4.2.4.

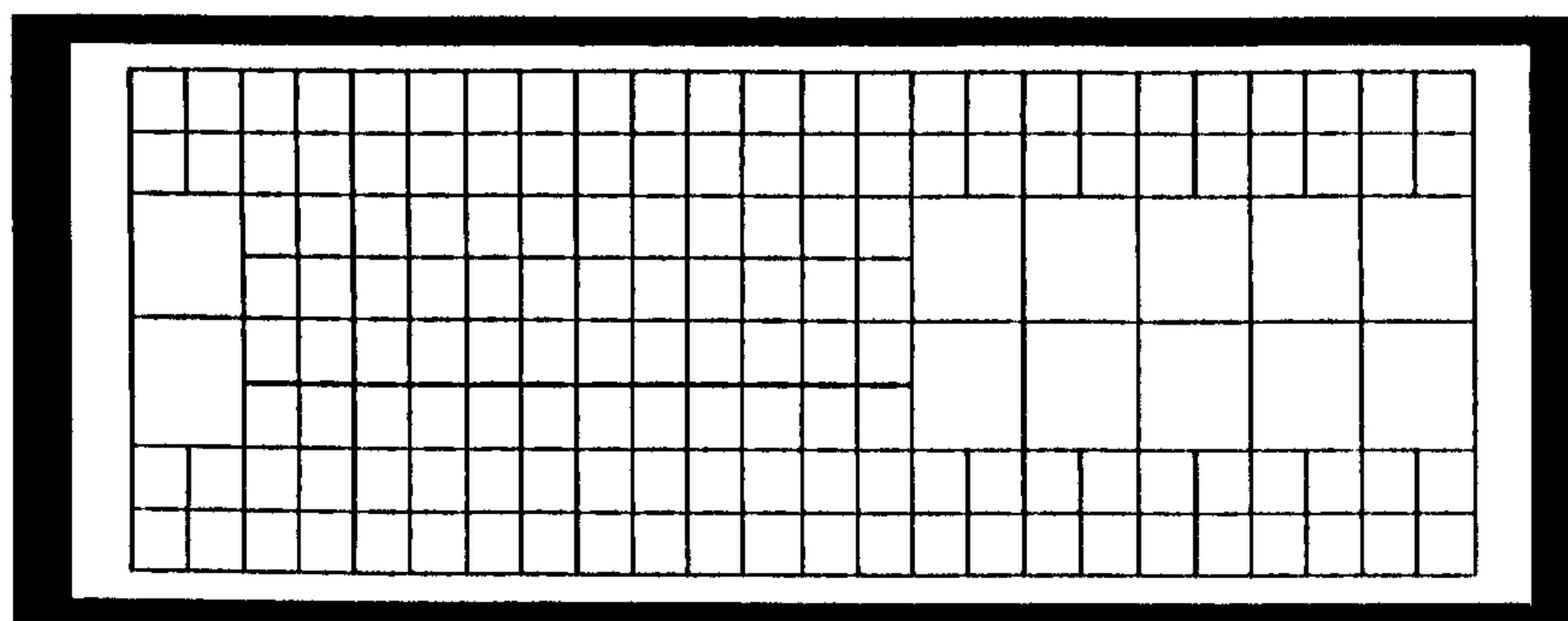


Figure 6.50: Adaptive (2,3) Grid for the multiphase channel problem – Conservative interpolation – $\gamma = 0.35$

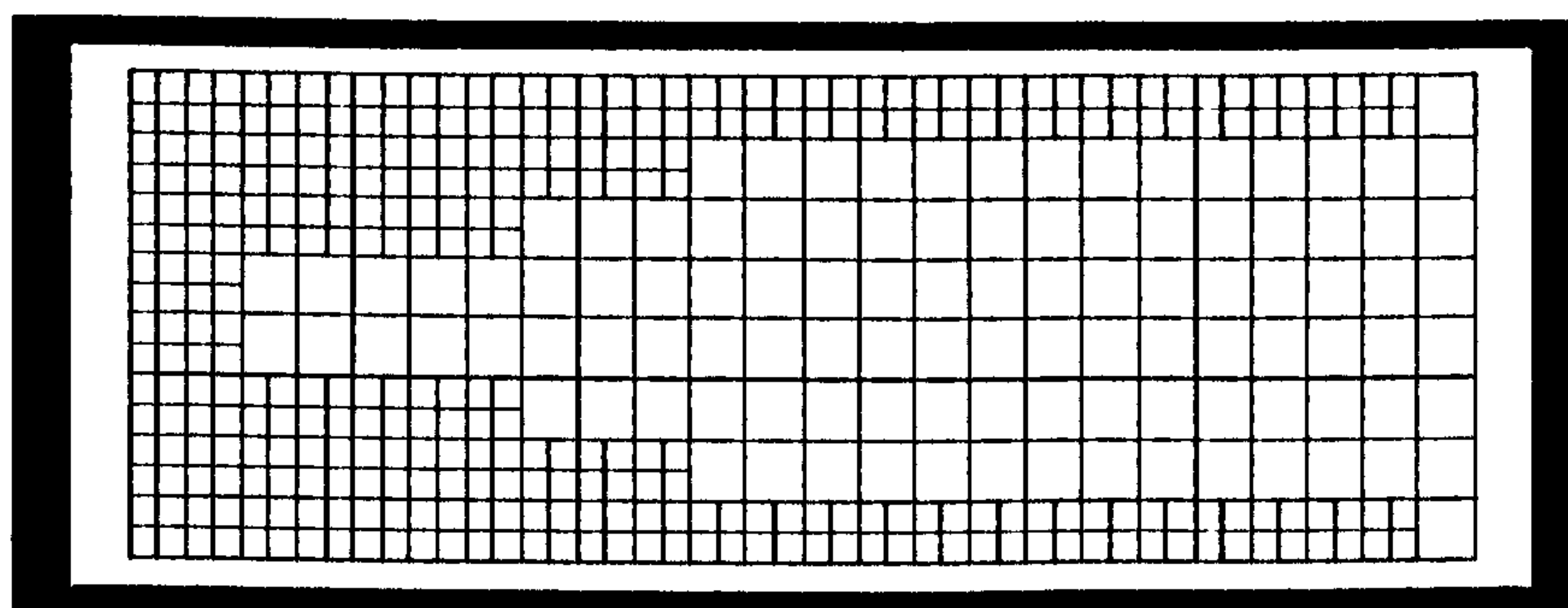


Figure 6.51: Adaptive (3,4) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$

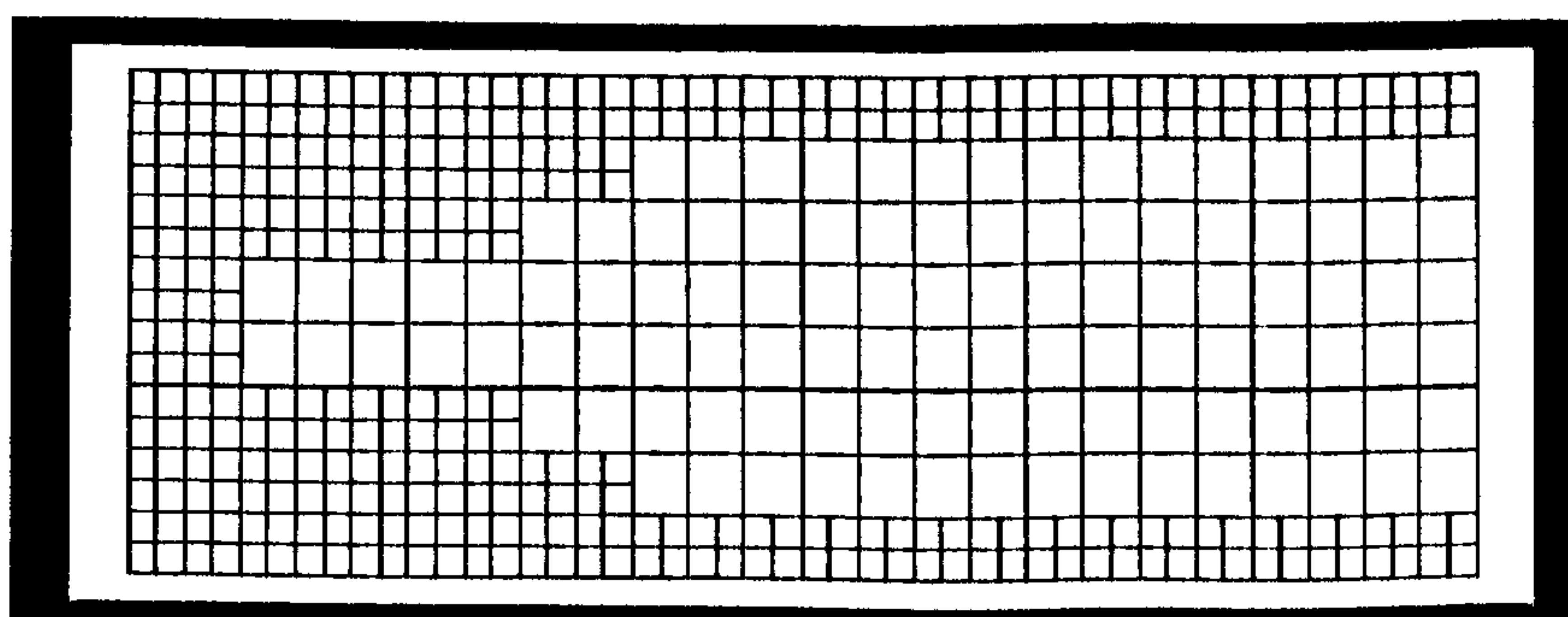


Figure 6.52: Adaptive (3,4) grid for the multiphase channel problem – Conservative interpolation – $\gamma = 0.35$

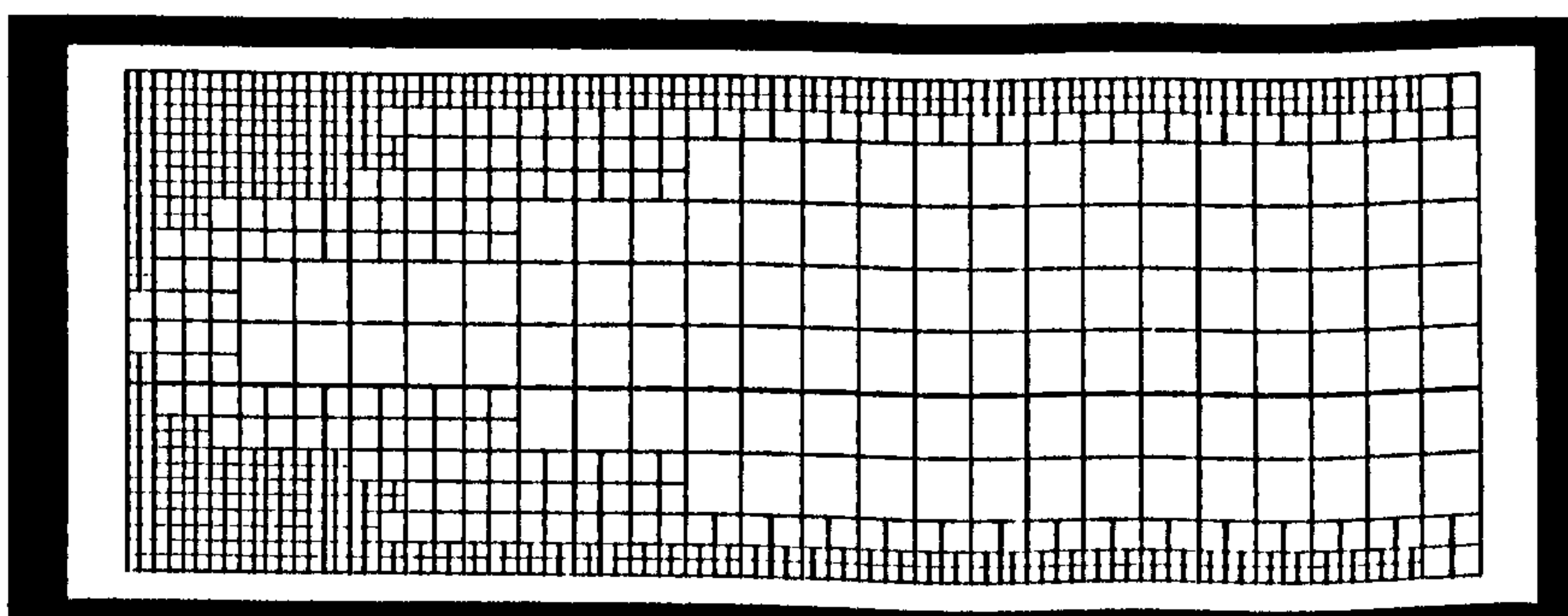


Figure 6.53: Adaptive (3,5) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$

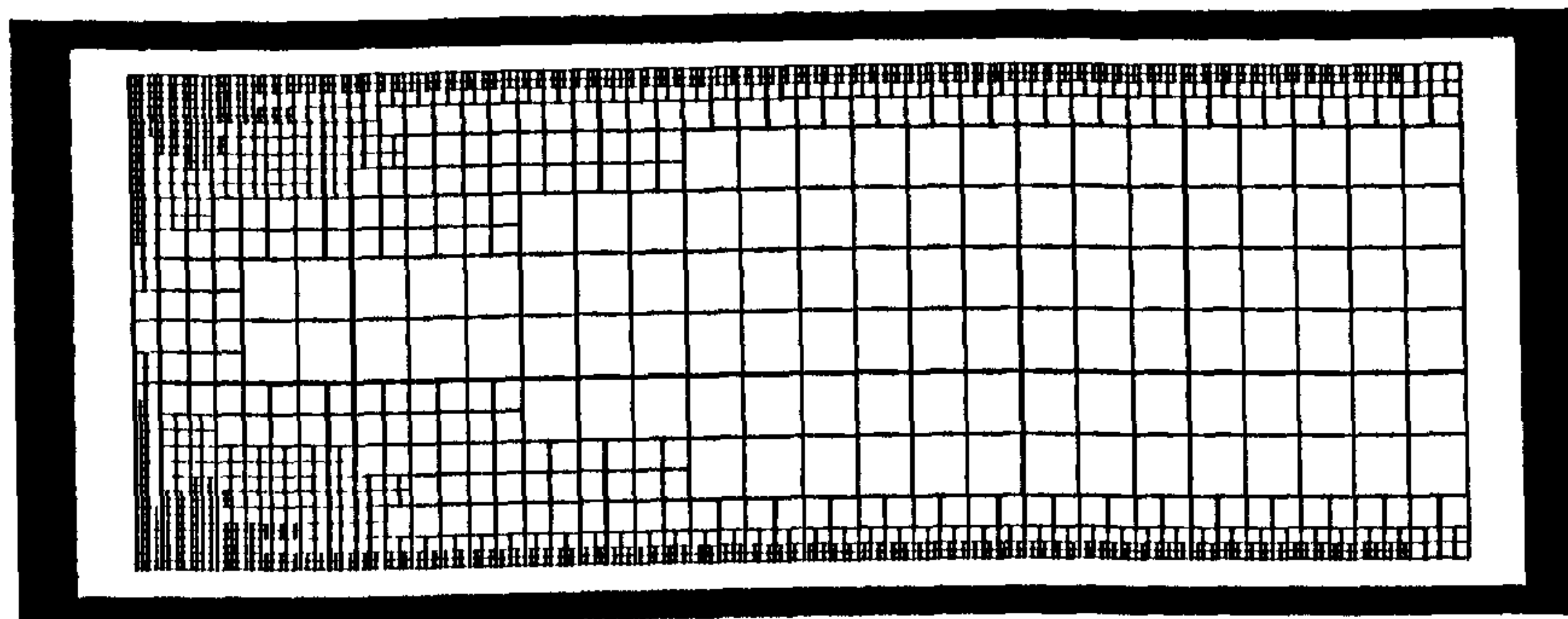


Figure 6.54: Adaptive (3,6) grid for the multiphase channel problem – Non-conservative interpolation – $\gamma = 0.35$

6.7.2 T-junction Geometry

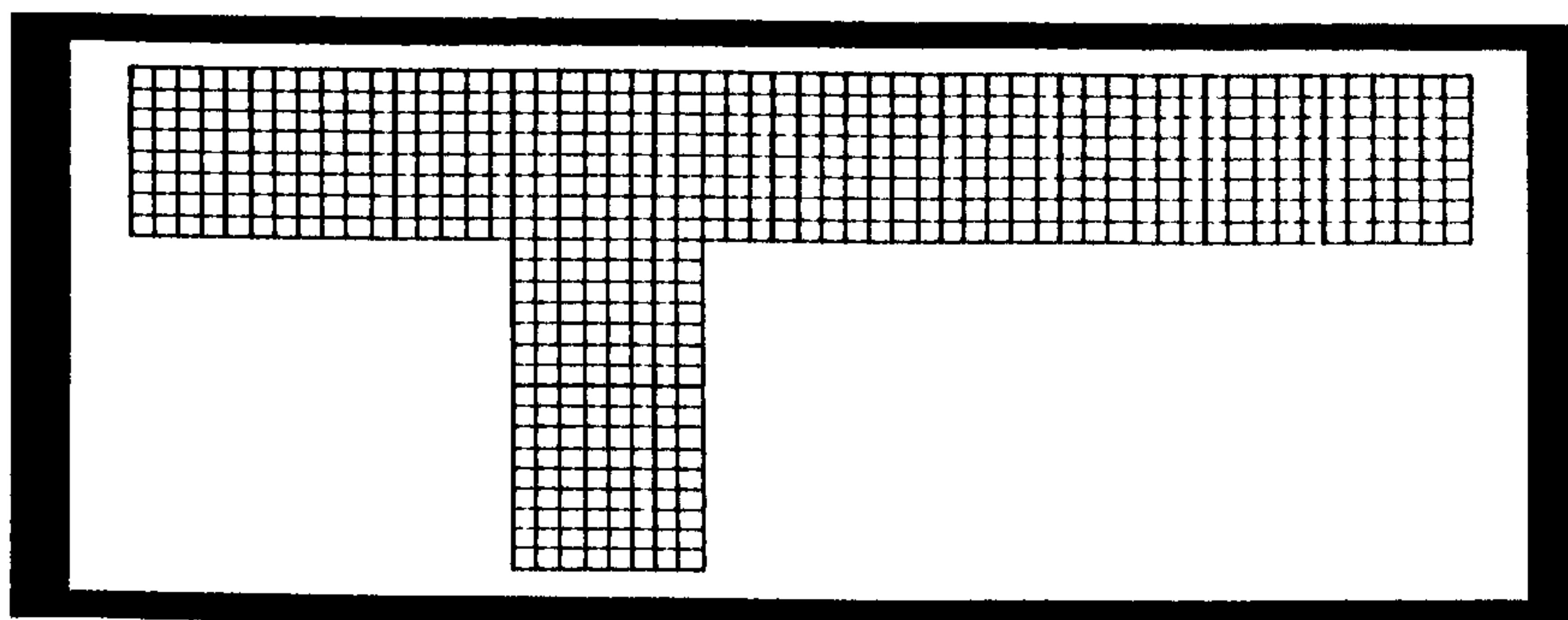


Figure 6.55: Uniform grid (level 3) for the T-junction problem

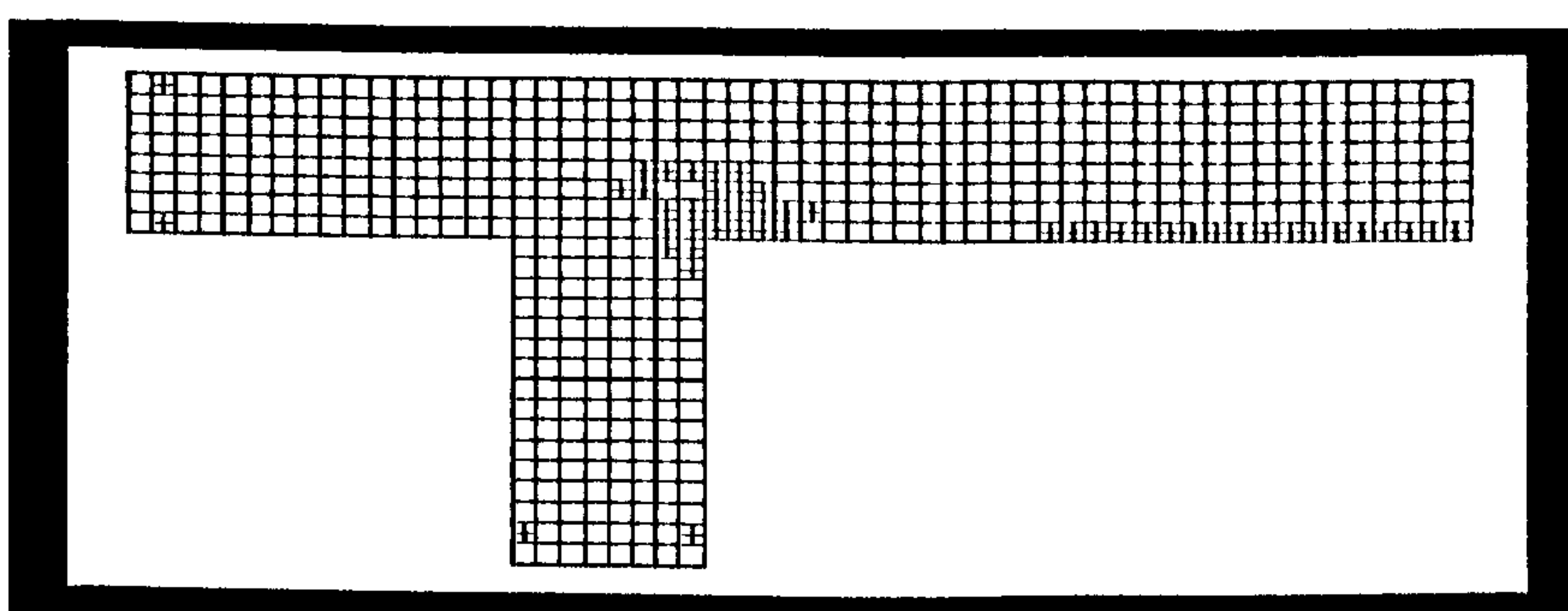


Figure 6.56: Adaptive (3,4) grid for the T-junction problem - $\gamma = 2.0$

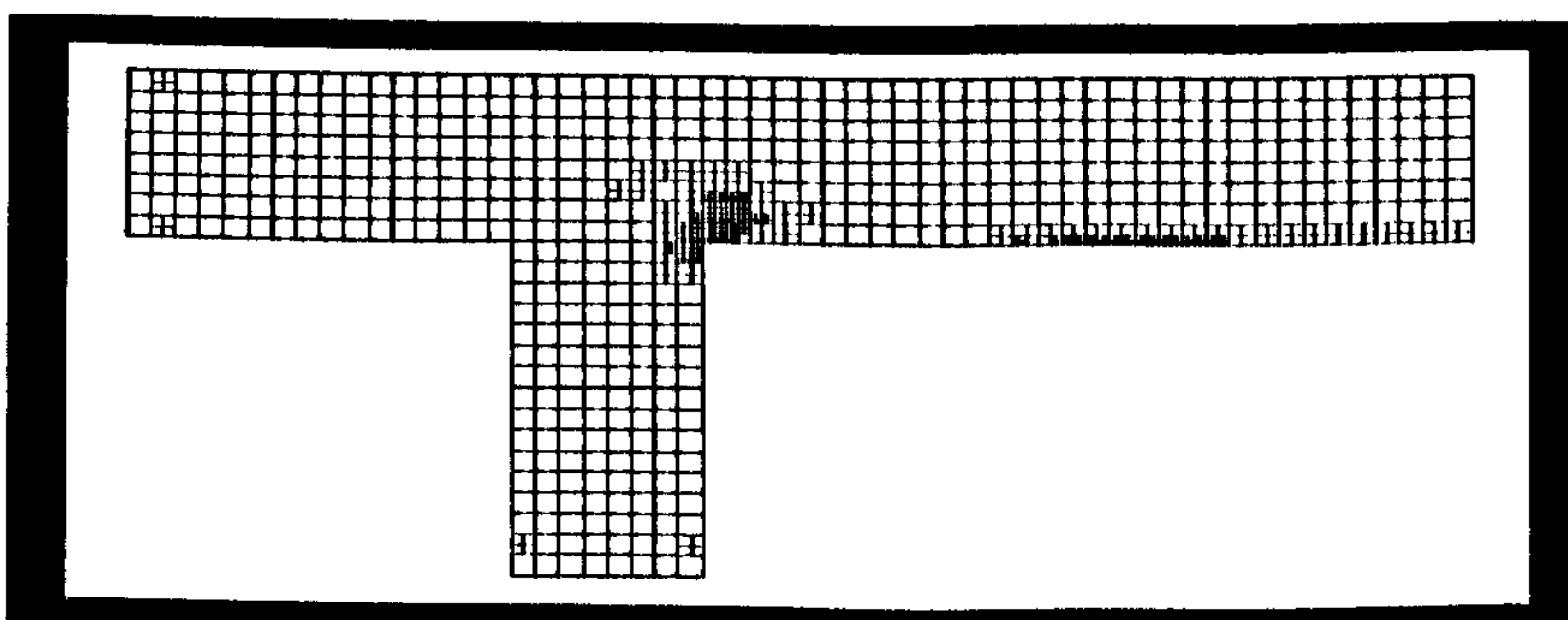


Figure 6.57: Adaptive (3,5) grid for the T-junction problem - $\gamma = 2.0$

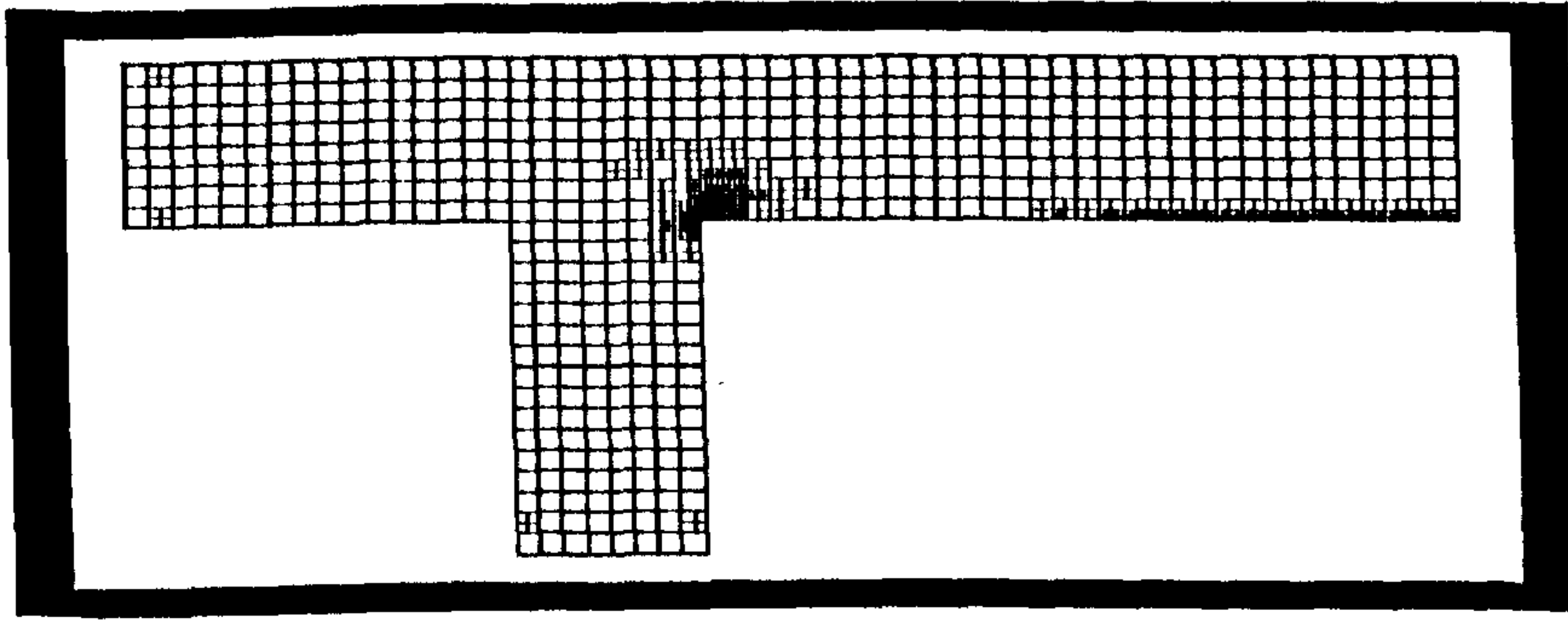


Figure 6.58: Adaptive (3,6) grid for the T-junction problem – $\gamma = 2.0$

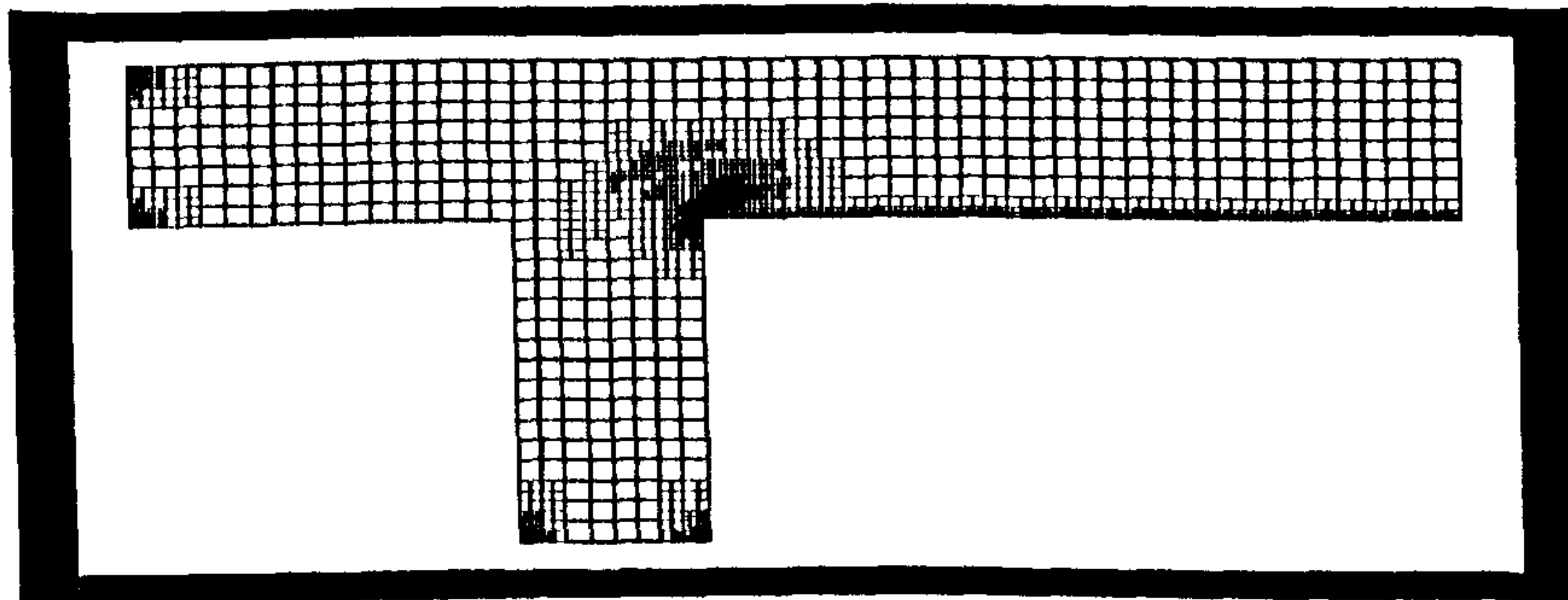


Figure 6.59: Adaptive (3,6) grid for the T-junction problem – $\gamma = 1.0$

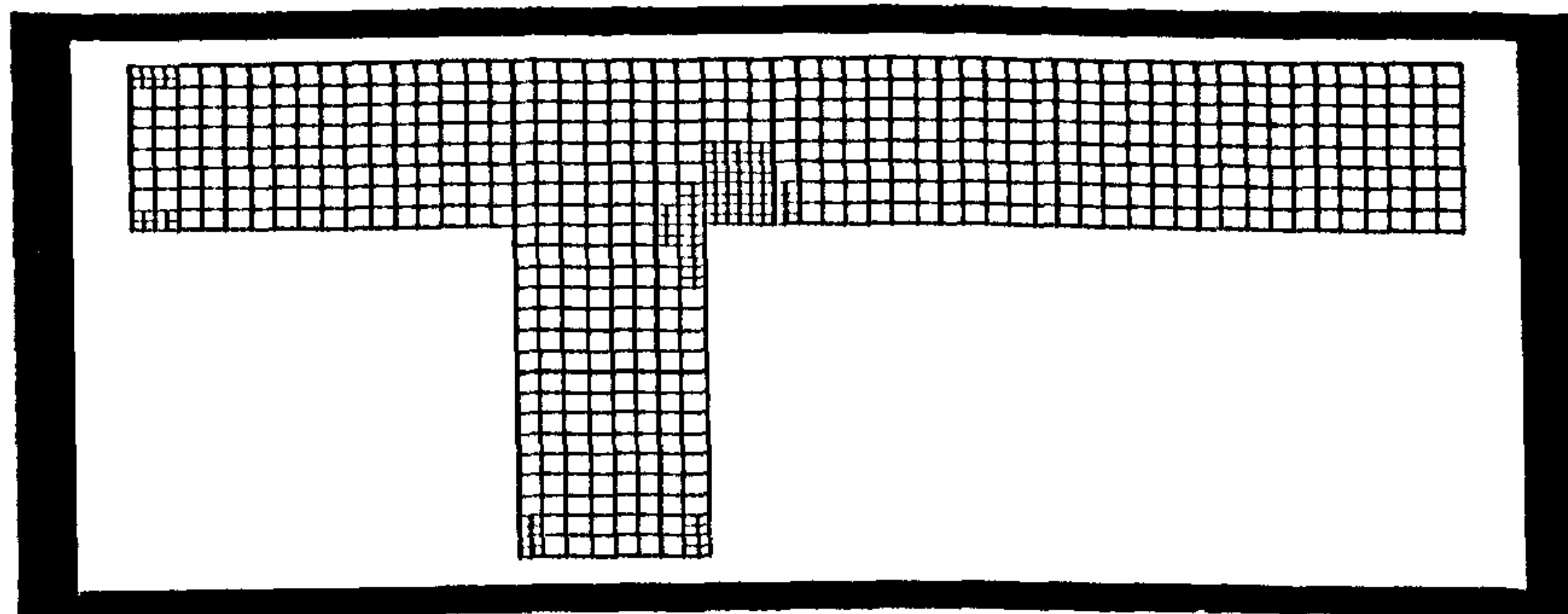


Figure 6.60: Adaptive (3,4) grid for the single phase T-junction problem – $\gamma = 2.0$

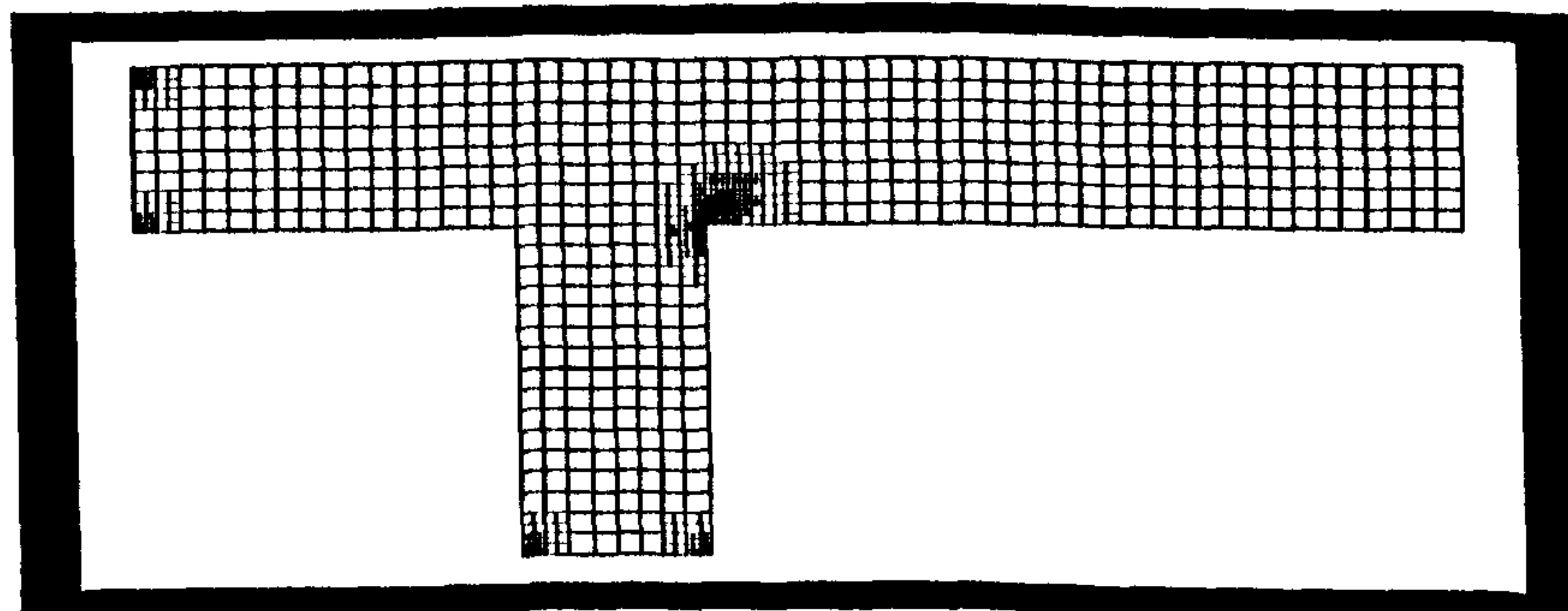


Figure 6.61: Adaptive (3,6) grid for the single phase T-junction problem – $\gamma = 2.0$

6.7.3 Backward-Facing Step Geometry

The adaptive grid shown in Figure 6.63 corresponds to a single phase computation. Some adaptive computations for the multiphase backward-facing problem have been successfully performed but are not reported here.

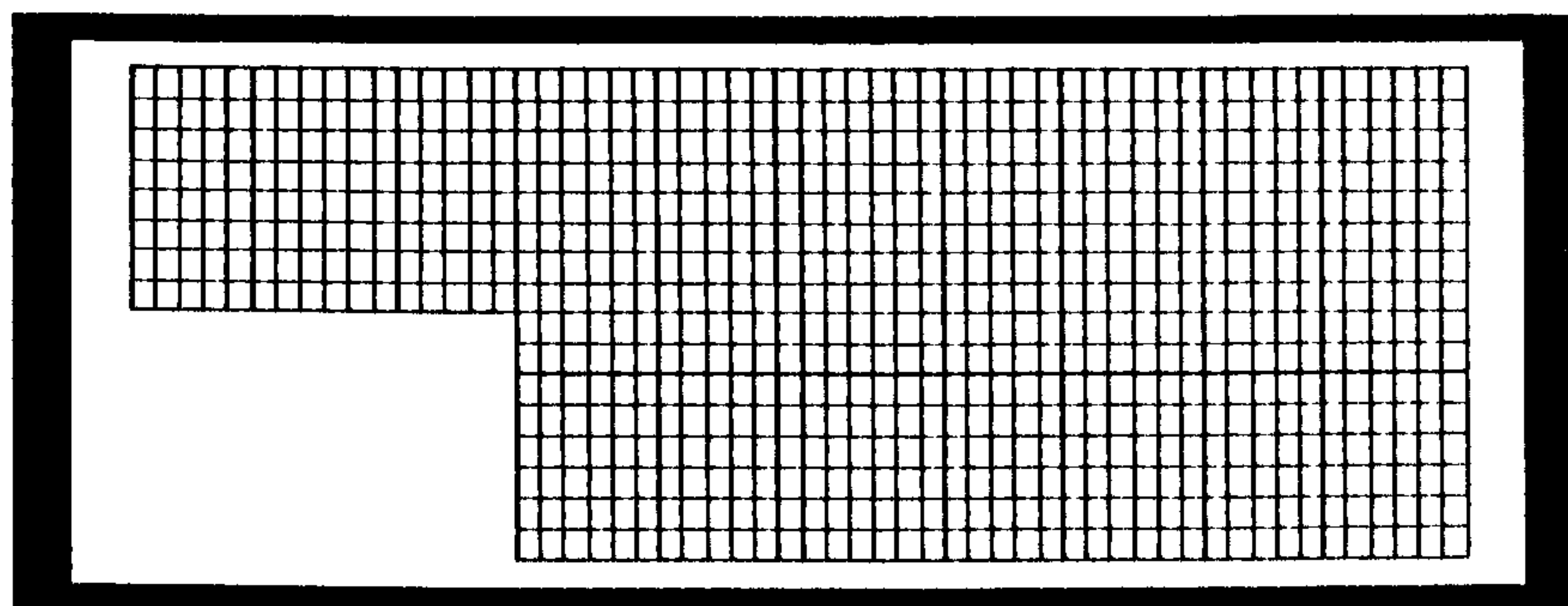


Figure 6.62: Uniform grid (level 3) for the multiphase backward-facing step problem

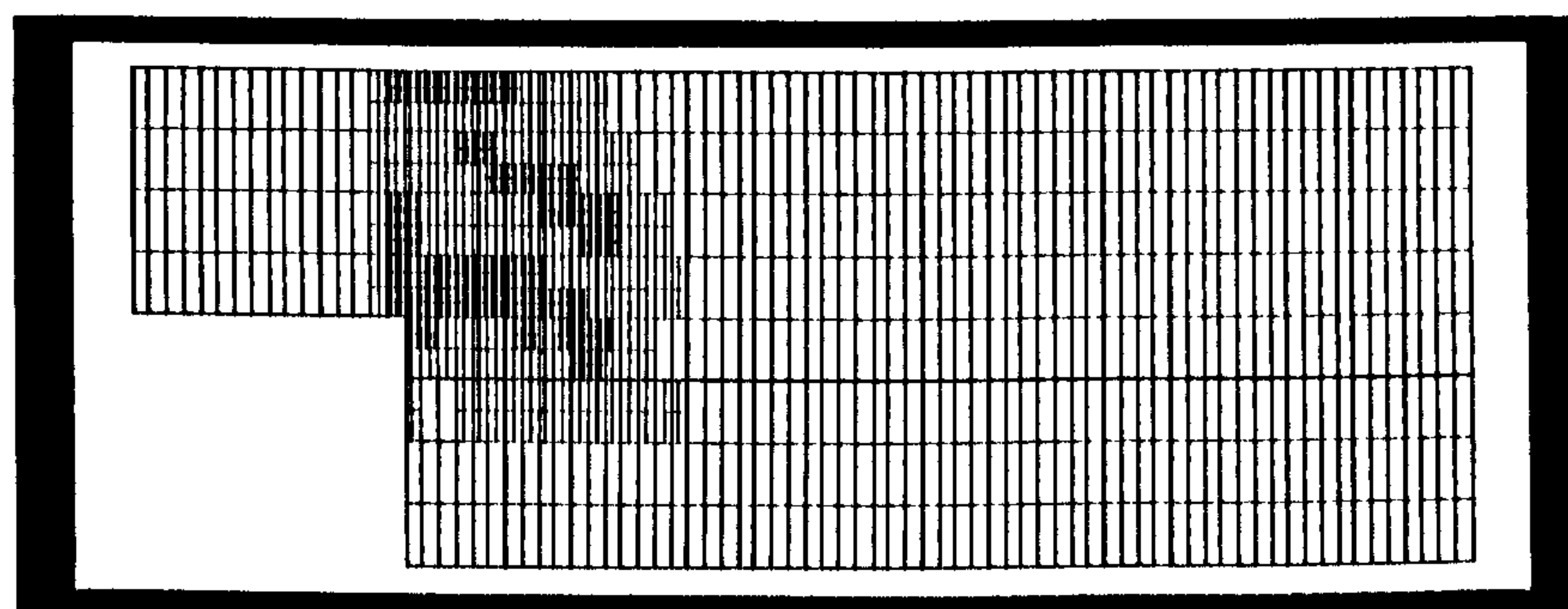


Figure 6.63: Adaptive (2,4) grid for the single phase backward-facing step problem
– $\gamma = 2.0$

Chapter 7

Future Work

The present thesis has demonstrated the potential of an adaptive quasi-Newton coupled multigrid approach to the numerical simulation of multiphase flows. There are three main avenues possible in the future development of the software and solution algorithm:

- *Performance enhancements:* run times could, in the author's opinion be reduced significantly. As previously mentioned, the code has no optimisation whatsoever. This was a design choice made in order to reduce both implementation errors and numerical inaccuracies, which is important in an initial development stage. Prime candidates for optimisation include: the removal of run-time automatic differentiation, possible simplifications of the Jacobians, the use of frozen Jacobians and the exploitation of the sparseness of the local Jacobians to design fast block inversion procedures. The author is also quite convinced that a number of computational bottlenecks are present in the software but are overshadowed by the weight of automatic differentiation.
- *Algorithmic improvements:* Frequent references have been made to the fact that due to the strong non-linearity of the discrete operators, solutions can vary on different grids. This is particularly true for the volume fractions. In such a case, it may be worth considering operator based interpolation to improve the quality of the grid transfers and the multigrid convergence rates, particularly in the non-asymptotic regime.

Other algorithmic work could include:

1. the development of better adaptive strategies in order to automatically select the refinement parameter γ ;
2. the study of the behaviour of the volume fractions (and of the truncation errors) near walls in order to improve the boundary conditions;
3. the implementation of alternate (possibly hybrid) formulations in terms of momentum instead of velocities.

- *Modelling options:* in certain regions, multiphase flow patterns are such that the Newton systems become badly conditioned locally. A typical example is the case where the phases are totally separated. It should be possible to improve the conditioning without losing accuracy by reformulating the equations. A related issue is to resolve the non-uniqueness of the solutions when the steady problem is not well-posed in a mathematical sense because several phases are recirculating at a point. This problem arises mainly for multigrid computations. Conceptually, it is necessary to fix in some physically acceptable way the mass of a recirculating phase across different grid levels.

Once run times have been optimised, studying the behaviour of the solver at low cell-Reynolds numbers should be greatly facilitated, and it will be possible to determine the exact cause for the observed deterioration in numerical performance. This is important from both an intellectual and practical viewpoint since this is in the author's opinion, a major aspect of the solution algorithm which is not adequately understood.

The ability of the software to handle general rather than Cartesian geometries would be a major advantage. This could be achieved by the use of body-fitted coordinates, but would require a large development effort. Alternatively, one could take advantage of the adaptive capabilities of the present software to replace curved boundaries by approximating Cartesian geometries, with increasing accuracy as grids are refined. Work is currently being undertaken in AMAC to implement this concept in the *pang* single phase solver [113]. However, it should be noted that in problems where the dominant effects are associated with the boundaries, the accuracy of such an approach is limited.

The extension of the software to the case of *transient* multiphase flows would lead to a large widening of its domain of applicability, as many two-phase flows are intrinsically transient in nature. Problems such as the study of droplet dynamics in an Eulerian framework or phase separation are just two examples of possible applications. Transport processes in multiphase pipelines (possibly in a network arrangement) are another domain of interest which is particularly amenable to adaptive computations.

Chapter 8

Conclusions

In the course of this thesis, the numerical simulation of multiphase flow has been considered. Our main goal was to develop a robust and efficient numerical solver for the two dimensional steady multi-fluid equations for incompressible viscous fluids.

Such high performance solvers already exist in the case of single phase flows: by combining a quasi-Newton coupled solver with a FAS multigrid method, it is possible to build a solution algorithm which is faster than traditional segregated solvers (in the line of the SIMPLE procedure) by at least one order of magnitude, and often more. Quasi-Newton solvers and multigrid methods are naturally complementary because the former allow an accurate treatment of the local coupling between the flow variables while the multigrid procedure quickly resolves the long wavelengths in the solution. The resulting solver is efficient – grid independent convergence rates and optimal complexity (order-wise) are observed for many non-linear problems – and robust since its convergence does not depend in any way on the quality of the initial guess. Furthermore, adaption is easily integrated in this framework. Here again, the FAS method is a cornerstone: (i) it provides accurate error estimation at no extra cost via the defect and (ii) it allows fast fine grid computations. Computational composite grids can be generated accurately and automatically without requiring detailed a-priori knowledge of the expected solution. Hence adaption not only allows vast savings in computational time and memory usage, but also provides the potential for controlling the discretisation error of the numerical solution.

These concepts have been implemented in a Navier-Stokes solver for steady viscous incompressible flows, called *pamg*, by Thompson and others [1]. The main focus of the present work is to extend these techniques to the case of multiphase flows. The simulation of such flows is particularly challenging, even compared with advanced single-phase CFD, for at least three reasons: (i) the physical phenomena are highly complex, (ii) the available mathematical models are not perfectly established since the fundamental issues pertaining to their well-posedness are still pending and (iii) the design of efficient numerical methods is complicated by the large degree of non-linearity of the governing equations as well as restrictions on the physically realisable

solutions. However, the basic assumption underlying the present work is that if an accurate treatment of the local coupling between the flow variables is crucial for efficient single phase solvers, then it is even more vital for multiphase flows due to the increased non-linearity and the tight coupling which necessarily exists between all phases in most multiphase phenomena.

While adaptive coupled multigrid methodologies have been very successfully applied to single phase fluid flows for the last fifteen years, this approach is, to the best of the author's knowledge, entirely original in the field of multiphase flow simulation. A multi-fluid solver, called `pamg-multiphase` has been developed, building on the single-phase solver `pamg`. This task has required the careful examination of many issues, some of which were almost impossible to foresee at the beginning of the project. A range of original algorithmic developments have been introduced to solve these problems. They include:

- *The derivation of consistent discretisations in order to have a stable quasi-Newton single-grid solver.* Attention has to be devoted to interpolation operators which were required because of the use of staggered grids. Otherwise, they may introduce numerical coupling between the variables which renders local Jacobians badly conditioned. In particular, it has been necessary to use upwind interpolation for the volume fractions which appear in the discrete continuity equations.
- *The use of automatic differentiation to obtain exact expressions for the numerical Jacobians.* The principle of Newton's method is very simple but its implementation can be difficult, particularly for complex systems. Unless the computed Jacobians and residuals are consistent with each other, Newton's method will not converge. Automatic differentiation ensures a high degree of consistency since only the code for the computation of the residuals needs to be developed, the computations of the Jacobians follow immediately. The fact that exact expressions for the numerical Jacobians are available is important: if simplified expressions are used, then convergence can be lost, due to an inaccurate treatment of the non-linearity. This also suggests that the flow variables in the multiphase case are indeed very tightly coupled together. Because software development is greatly facilitated, it was also possible to conduct easily experiments for different discretisation options.
- *The use of line-searching to obtain a globally convergent Newton's method.* Line-searching relies on the fact that the Newton correction is a descent direction for the Euclidean norm of the residual. Unless line searching is implemented, Newton's method applied to multiphase flow diverges very quickly. This again reflects the non-linearity of the equations. A further benefit of line-searching is that it allows a efficient monitoring of the corrected solution to ensure that they are always physically realisable.
- *The modification of the multigrid grid transfers.* This is so that (i) the possibility of generating non-physical approximations is eliminated and (ii) mass

fluxes can be conserved between grids. This last property is important in the context of adaptation. For single phase flows, the discrete problem does not admit any solution unless the mass fluxes are conserved. For multiphase flows, conserving mass fluxes on a staggered grid is slightly more complicated since the mass fluxes depend on both the velocities and the volume fractions which are not defined at the same grid points.

Besides these algorithmic issues, important computational questions needed to be examined. In this respect, the development of `pamg-multiphase` was challenging for two main reasons: (i) the governing equations and their discretisations are not thoroughly understood, (ii) the solution algorithm proposed based on multigrid methods is by its very nature composite, and all its many parts have to be correctly implemented. The availability of the single phase solver was an advantage, but its full exploitation has required a careful methodology.

In the present thesis, the discrete equations for both the single phase and the multiphase solvers, have been derived in the finite volume framework and on staggered grids. Use was made of hybrid first order/second order schemes whose property of monotonicity is useful in connection with multigrid methods. The implications of discretisation choices upon the performance of the solver have been discussed. The details of the implementations have been also been given.

Having completely defined our solver for the multi-fluid equations, its performance has been assessed on the following criteria: (i) accuracy (ii) robustness and (iii) efficiency.

Firstly, solutions obtained with `pamg-multiphase` for a wide range of important test problems, of varying complexity, have been described, discussed in some detail and compared, not with experimental data, but with the solutions provided by an independent software. We have chosen a widely used and validated commercial CFD package: CFX 4.1. CFX 4.1 solves the same set of governing equations as `pamg-multiphase` but its solution algorithm is based on the IPSA procedure, which is segregated. This approach to validating the software presents the benefit that experimental errors are totally eliminated. A very good degree of agreement is obtained between the `pamg-multiphase` and CFX 4.1 solutions, for the *totality* of the test cases studied. It can therefore be considered that the implementation of the `pamg-multiphase` solver for the multi-fluid equations is essentially correct and accurate.

It was also concluded that `pamg-multiphase` is capable of handling difficult multiphase flow patterns such as complete phase separation and recirculation zones, at least for sufficiently coarse grids. In the first case, the local Jacobians become more and more badly conditioned as the grid size is reduced. As far as recirculation zones are concerned, the issue of the well-posedness of the steady problem is raised. Since there are no mass transfers between the main flow and recirculation zones, it is not possible to fix the volume fraction of each phase in the recirculation zone so there

may be several equivalent solutions. `pamg-multiphase` has nevertheless successfully computed a range of recirculation zones. This is because the problem is only ill-posed when both phases are recirculating at any given point. In many cases, only one phase is recirculating (note that the multi-fluid model allows different velocity fields for each phase) and the problem is therefore well-posed because the volume fractions are determined by the conservation of mass for the non-recirculating phase. When a recirculation problem is ill-posed, the multigrid solver usually fails. It was verified that the single grid solver is often more stable in such cases.

Secondly, the robustness of the `pamg-multiphase` solver was established by considering that its convergence is not strongly dependent on the quality of the initial guess for the solution. For all the results presented in this thesis, uniform initial distribution for all flow variables have been used. This is due to the fact that the use of hybrid schemes in a multigrid framework defines an implicit continuation method. By contrast, the performance of segregated solver is very dependent on the quality of the initial guess.

Thirdly, the question of efficiency is considered. Our first conclusion is that the FAS multigrid procedure is a very successful acceleration technique for the basic single quasi-Newton solver applied to the multi-fluid equations. Speed-ups by a factor of 20 have been observed for complex multiphase flows. It is harder to compare the performance of `pamg-multiphase` with that of a segregated solver. Rough estimates nevertheless indicate that `pamg-multiphase` is more efficient than CFX 4.1, our reference commercial CFD package. Given that the solution procedure is a pure implementation of FAS which offers the potential for grid independent convergence rates, the ideal would be to observe a linear complexity, i.e. the convergence of the solution algorithm in $\mathcal{O}(N)$ operations where N is the number of discrete unknowns. This ideal is not achieved with the `pamg-multiphase` solver. Reasons for this slight limitation are investigated in detail. The main conclusions are as follows:

- The dependence of the convergence rates on the grid size reflects the non-linearity of the problem. Of course, the FAS procedure is directly applicable to non-linear problems. Indeed, in some cases, grid independent convergence rates are sometimes observed for such problems, the single phase `pamg` being a prime example. However, non-linear problems require that the operators as well as the solutions be approximated. Moreover, both the operators and the solutions may have different characteristics on different grids. The FAS procedure will take these differences into account. Indeed, the FAS defect can be considered as a correction applied to make the solution on different grids fit together. However, the more non-linear the problem, the more effort will have to be devoted to the process of adjustment.
- Standard choices for the diffusive tensor in the momentum equation introduce cross derivative terms which are “hyperbolic-like”. It has been shown that these terms are partly responsible for the deterioration of multigrid convergence factors on sufficiently fine grids. If these terms are not included, then

convergence rates which are asymptotically grid-independent are sometimes observed. However, the cross derivative terms have a diffusive effect on the volume fractions. They therefore exert a stabilising influence on the computations.

- A large body of evidence suggests that the performance of both the single grid quasi-Newton coupled solver and the multigrid procedure deteriorates as the (cell) Reynolds number decreases. This contrasts with single flows for which multigrid methods work better at low Reynolds numbers since the degree of non-linearity of the equations is then reduced. So far, we have been unable to isolate the precise cause for this low (cell) Reynolds number effect, but several hypotheses have been formulated which could be investigated once the solutions have been optimised.

Due to the nature of the equations, it is not altogether surprising that convergence factors should depart somewhat from the ideal. It remains that a FAS multigrid method still leads to a solution algorithm which is very efficient and very robust for a complex and very non-linear problem.

Computations on adaptive grids have been successfully performed for multiphase flows even though it is difficult to derive grid transfer operators which are stable and conserve mass fluxes between grids. We have derived and implemented an interpolation procedure which is conservative. It is based on explicitly transferring mass fluxes. Unfortunately, the procedure is unstable in “dry-out” zones, i.e. regions where the flow is completely separated and may require the equations to be reformulated. However, it was also shown that the property that the grid transfer operators are mass conservative is not necessary for the existence of a discrete solution. The grid transfers defined for the single phase solver *pang* do not conserve mass when they are extended to multiphase flows but still allow the multigrid method to converge on adaptive grids. We explain this behaviour in terms of error compensation with the volume fractions. Unlike the traditional grid transfer operators applied to single phase flows on staggered grids, neither the prolongation nor the restriction are conservative for multiphase flows. We also note that the discretisation procedure adopted for the continuity equation renders the numerical mass fluxes strongly decoupled, which may facilitate the process of error compensation. Finally, the errors introduced by the non-conservative mass fluxes are small and can be neglected compared with the truncation errors.

Adaptive computations — based on the non-conservative grid transfers — have been performed for two test cases: the two-phase channel flow problem and the two-phase flow through a T-junction. Comparisons are made with equivalent results on uniform grids. As in the single phase case, grids were generated automatically by refining in regions of high truncation error, which is estimated by the multigrid defect. Adaptive solutions closely agree with those obtained on uniform grids. The refinement algorithm is therefore proved to be accurate and robust in the sense that detailed a-priori information about the expected solution is not required. This is

crucially important because it renders adaption easy to use. Its benefits in terms of computational cost and memory usage are therefore directly available. These savings can be very large. We have observed speed-up and reductions in memory usage by one order of magnitude compared with equivalently accurate uniform grid computations. Such good results were obtained for suitable problems, i.e. problems for which the truncation errors are relatively localised. Even when the truncation errors are more evenly distributed, adaption is still very beneficial.

At the close of this project, it can reasonably be argued that the initial goals of designing a robust, efficient and accurate solver for the simulation of relatively complex multiphase flows have been attained. It has been demonstrated that the adaptive quasi-Newton coupled multigrid approach to the simulation of fluid flow is as applicable to multiphase flows as it is to single phase flows. The quasi-Newton coupled solver handles the tight local coupling of the flow variables sufficiently well to be a good multigrid smoother. Multigrid provides very significant acceleration and finally adaption greatly reduces the computational cost of obtaining a solution to a given accuracy. However, the solver is not optimal order-wise. This slight limitation reflects (i) the strong degree of non-linearity of the equations and (ii) the presence of terms in the governing equations which are not handled well by the multigrid procedure.

The system of equations which governs multiphase flows is highly complex. Hence, it is very difficult to fully understand the numerical behaviour of a discrete solver based on these equations, particularly one which has many parts and aspects, such as multigrid methods. It remains that efficient numerical computations can be obtained. In many cases, poor numerical performance is usually the limiting factor in the development of advanced physical models. The methodology developed in this thesis may therefore allow the study of a wide range of multiphase phenomena.

Appendix A

One-Dimensional Steady Two-Phase Flow

A.1 Model Equations

We wish here to ascertain the usefulness of hybrid schemes for the simulation of steady multiphase flows at high cell-Reynolds numbers, and consider the following, very simple, one dimensional problem:

$$\frac{\partial r_\alpha u_\alpha}{\partial x} = 0 \quad (\text{A.1})$$

$$\frac{\partial r_\alpha u_\alpha^2}{\partial x} + \nu_\alpha \frac{\partial}{\partial x} \left(r_\alpha \frac{\partial u_\alpha}{\partial x} \right) = -\frac{1}{\rho_\alpha} r_\alpha \frac{\partial p}{\partial x} + r_\alpha k_\alpha |u_\alpha| u_\alpha \quad (\text{A.2})$$

with $0 < x < l$. The associated boundary conditions are:

$$\begin{aligned} u_\alpha(0) &= 1 \\ r_\alpha(0) &= 0.5 \\ \left. \frac{\partial u_\alpha}{\partial x} \right]_{x=l} &= 0 \\ \left. \frac{\partial r_\alpha}{\partial x} \right]_{x=l} &= 0 \end{aligned}$$

A.2 Discretisation and Hybrid Schemes

A finite volume approach together with an upwind interpolation for the volume fractions leads in the case of the continuity equation, to:

$$\frac{1}{\Delta x} \left(r_{i+1/2} u_{i+1/2} - r_{i-1/2} u_{i-1/2} \right)$$

with:

$$[r]_{i+1/2,j} \equiv \frac{1}{2} \left[(1 - \text{sgn}(u_{i+1/2,j}))r_{i+1,j} + (\text{sgn}(u_{i+1/2,j} - 1))r_{i,j} \right]$$

$$[r]_{i-1/2,j} \equiv \frac{1}{2} \left[(1 - \text{sgn}(u_{i-1/2,j}))r_{i,j} + (\text{sgn}(u_{i-1/2,j} - 1))r_{i-1,j} \right]$$

The discretisation of the momentum equation is:

$$A_c u_{i+1/2} - A_m u_{i-1/2} - A_p u_{i+1/2} = Q$$

where Q is the discretisation of the right-hand-side:

$$Q_{i+1/2} = -\frac{1}{2\rho} (r_{i+1} + r_{i-1}) (p_{i+1} - p_i) - k \frac{1}{2} (r_{i+1} + r_{i-1}) |u_{i+1/2}|u_{i-1/2} \quad (\text{A.3})$$

and $A_c = A_m + A_u$.

If we introduce the following quantities:

$$C_m = \frac{1}{4\Delta x} r_i (u_{i-1/2} + u_{i+1/2})$$

$$D_m = \frac{\nu}{\Delta x^2} r_i$$

$$C_p = \frac{1}{4\Delta x} r_{i+1} (u_{i+1/2} + u_{i+3/2})$$

$$D_p = \frac{\nu}{\Delta x^2} r_{i+1},$$

central differencing is equivalent to defining:

$$A_m = C_m + D_m$$

$$A_p = C_p - D_p$$

while hybrid differencing requires:

$$A_m = (\max(D_m, |C_m|) + C_m)$$

$$A_p = -(\max(D_p, |C_p|) - C_p)$$

We impose the following Dirichlet conditions at the inlet:

$$u_1 = u_2 = 1$$

$$r_1 = r_2 = 1/2$$

while at the outlet, Neumann conditions exact to first order, are specified:

$$u_{n+3/2} = u_{n+1/2}$$

$$r_{n+1/2} = r_{n-1/2}$$

for both phases. Finally, for all computations, we fix the numbers of cells to $n = 50$ and alter the length of the pipe in order to vary the cell-Reynolds numbers. This approach is (i) valid because equilibrium is reached very close to the inlet and (ii) convenient because a large number of cells is needed on coarse grids to observe instabilities. Otherwise, the Neumann boundary conditions may have the effect of damping the oscillations.

First, we consider first the case where the fluids have the same “physical” properties. Then, the system should behave similarly to the equations for single phase flows (assuming both phases have the same starting guesses) and we should confirm the well known result that central differencing is unstable if:

$$Re_c > 2.$$

We set:

$$\rho_1 = 1.0 \quad \rho_2 = 1.0,$$

$$\nu_1 = \nu_2 = 0.05.$$

$$k = 0.5$$

The analytical solution for this system is trivial: we have $u(x) \equiv 1$ and therefore, the pressure decreases linearly along the pipe:

$$p(x) = p(0) - \rho k x$$

for $l = 4.9$, i.e. $Re_c = 1.96$, central differencing yields stable computations (and correct answers!) (see Figure A.1). For $l = 5.1$, i.e. $Re_c = 2.04$, the same computations are unstable (see Figure A.2). For this last case, if central differencing is replaced by hybrid differencing, we achieve stable computations and correct answers (see Figure A.3).

We now consider a “real” multiphase problem with the following parameters

$$\rho_1 = 1.25 \quad \rho_2 = 0.75,$$

$$\nu_1 = \nu_2 = 0.05.$$

$$k = 1.0$$

See Figures A.4 to A.6 for the solution to the equations for $l = 4.0$ using hybrid differencing (the grids used to generate this solution contained 256 cells). With central differencing, computations are stable for $l = 4.0$ where $(Re_c)_{\max} \approx 1.83$ (Figure A.7), and unstable for $l = 4.1$ where $(Re_c)_{\max} \approx 1.87$ (Figure A.8). The latter case becomes stable if hybrid differencing is used (Figure A.9).

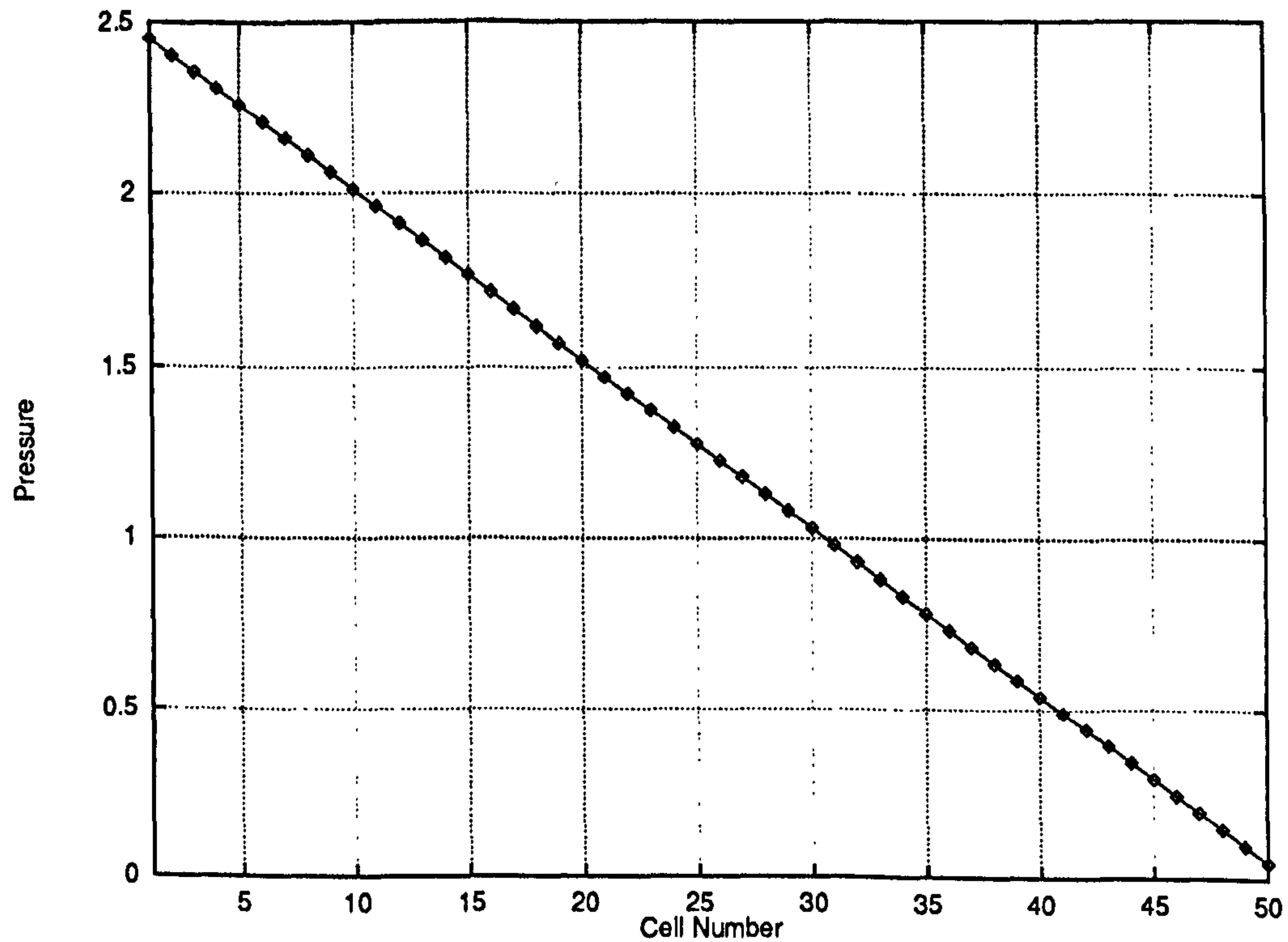


Figure A.1: One-dimensional pseudo-two-phase flow problem with $l = 4.9$ – Converged pressure field obtained with central differencing

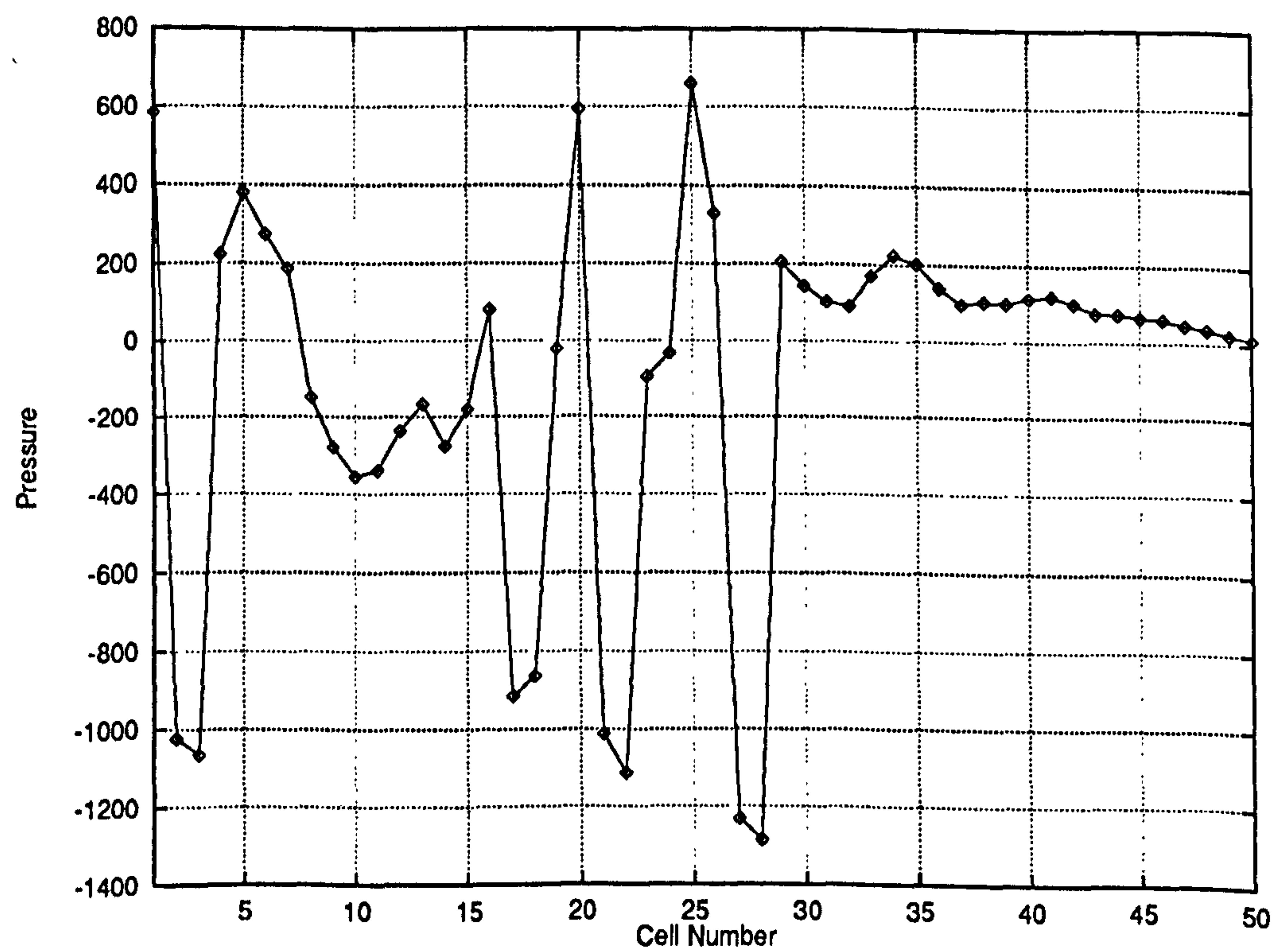


Figure A.2: One-dimensional pseudo-two-phase flow problem with $l = 5.1$ – Converged pressure field obtained with central differencing

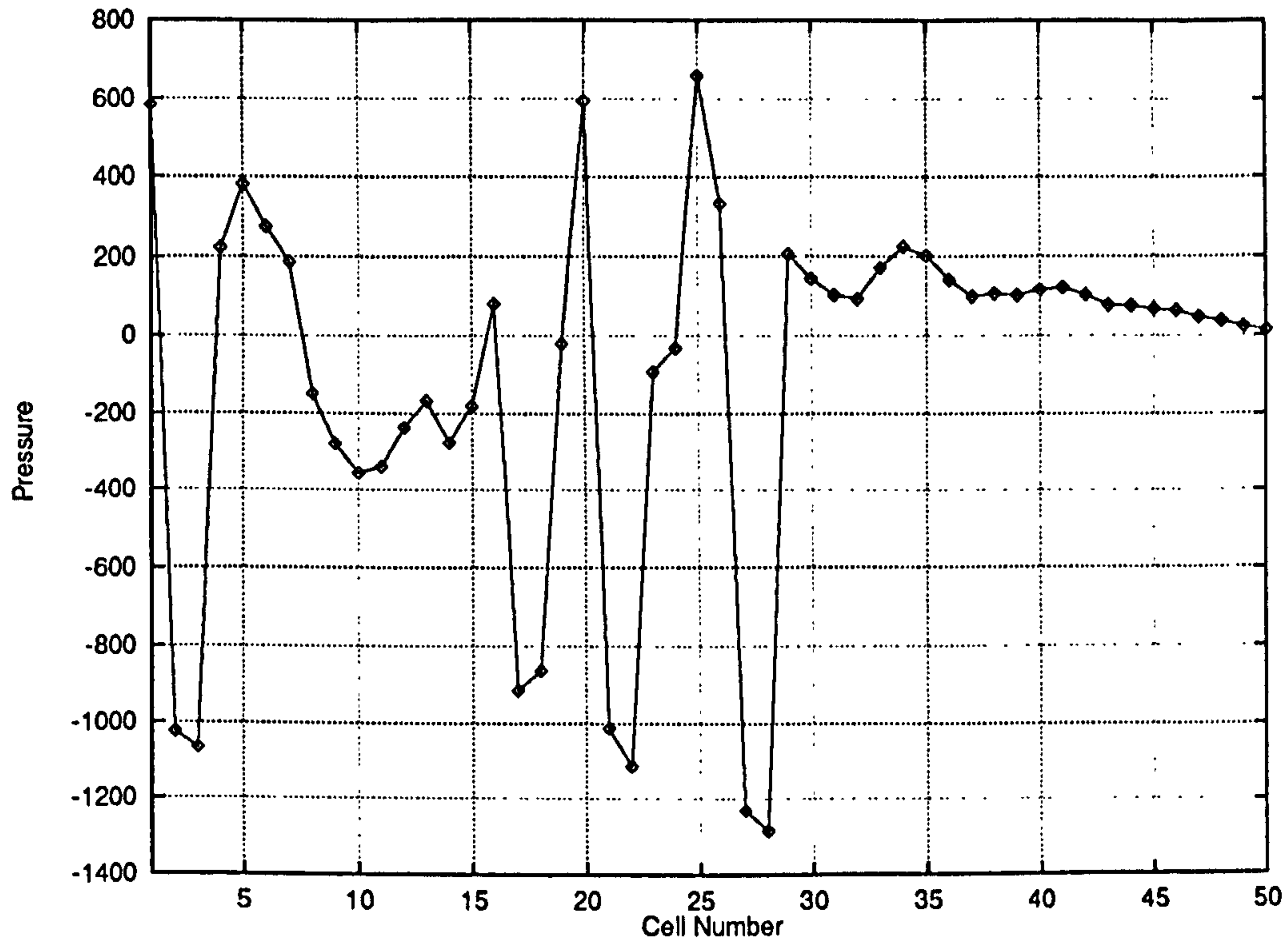


Figure A.3: One-dimensional pseudo-two-phase flow problem with $l = 5.1$ – Converged pressure field obtained with hybrid differencing

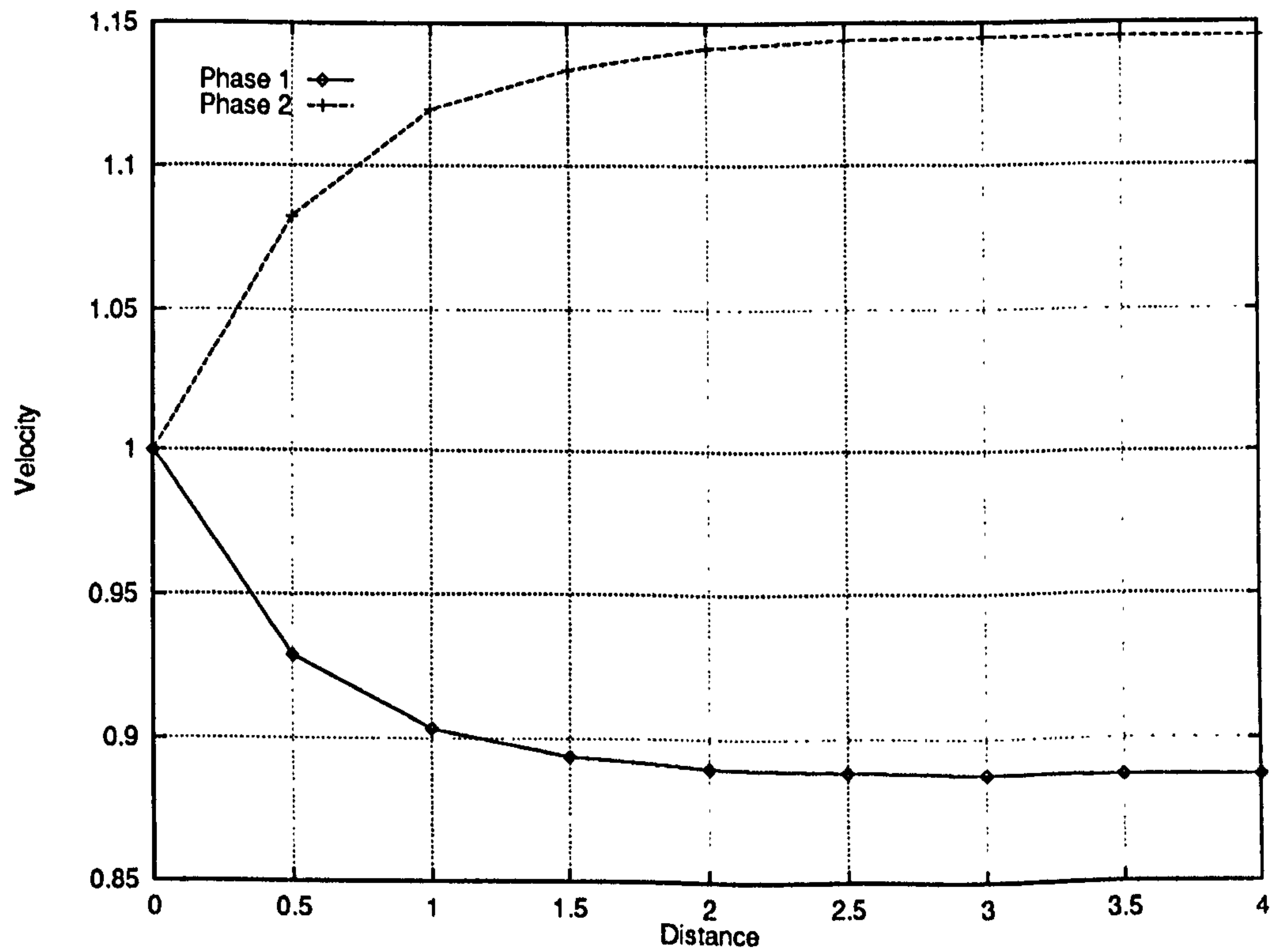


Figure A.4: Velocity solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing

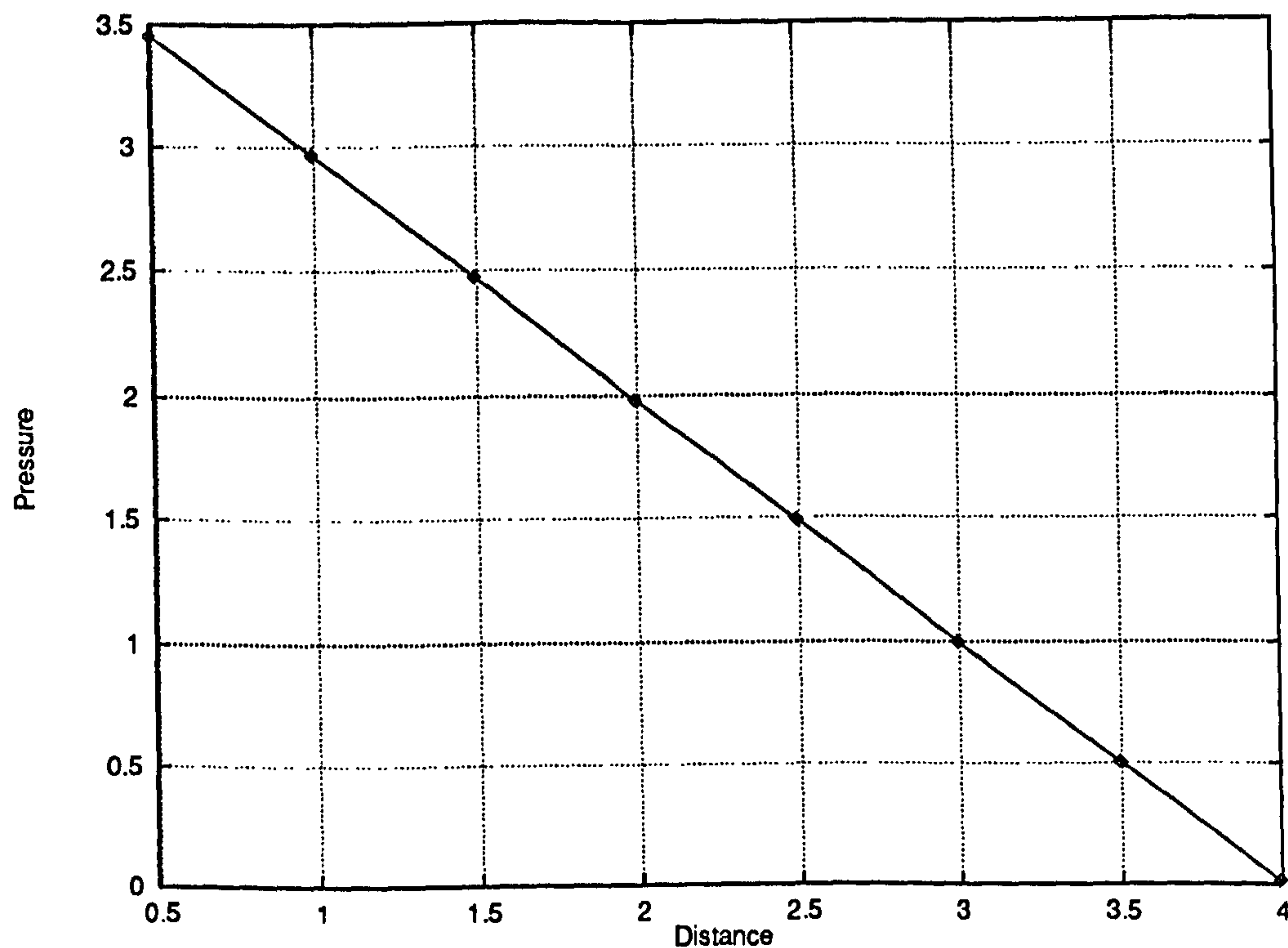


Figure A.5: Pressure solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing

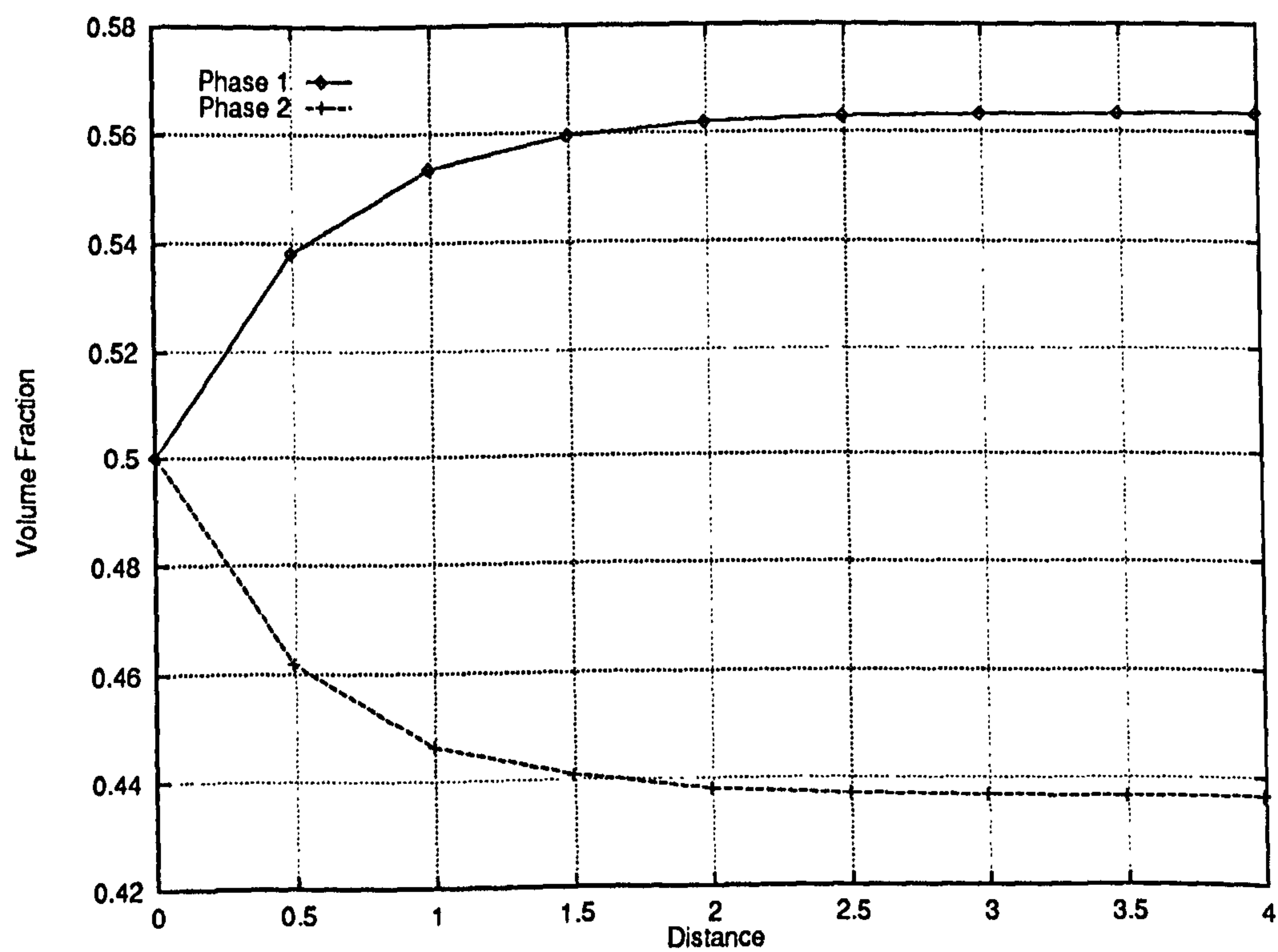


Figure A.6: Volume fraction solution for the one-dimensional multiphase flow problem for $l=4.0$, using 256 cells and hybrid differencing

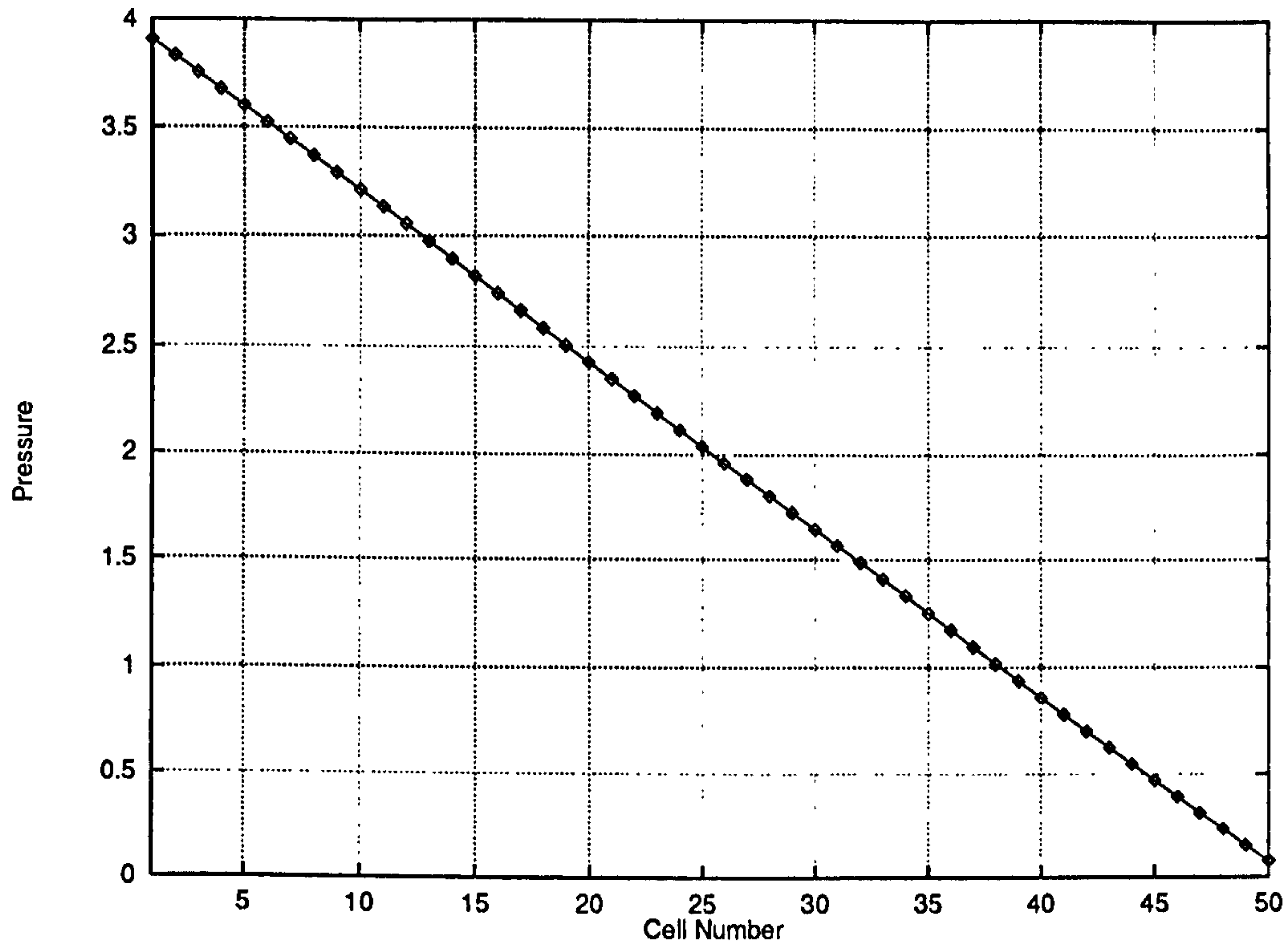


Figure A.7: Converged pressure solution for the one-dimensional multiphase flow problem - $l=4.0$ - Central differencing

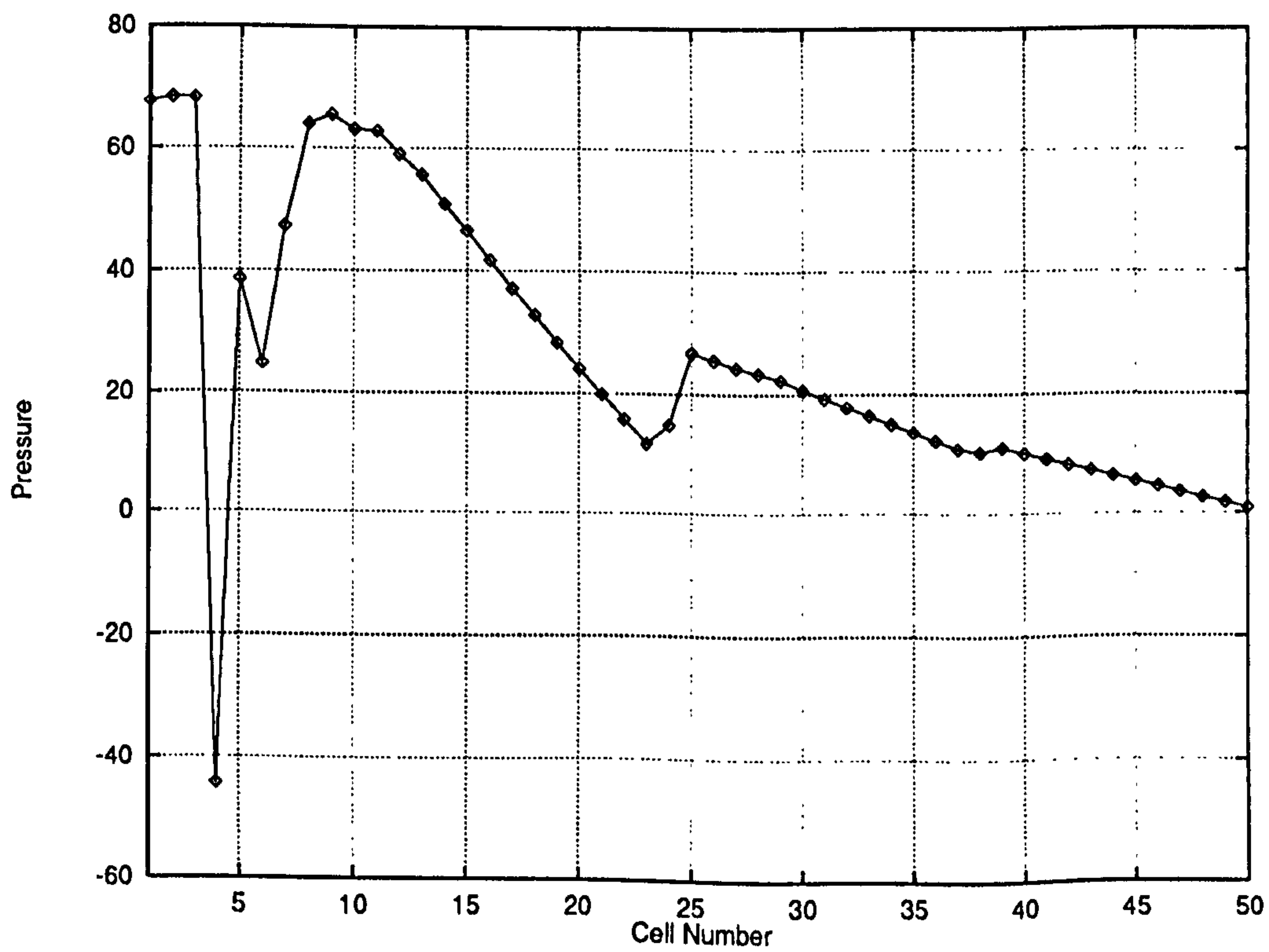


Figure A.8: Approximation of the pressure solution for the one-dimensional multiphase flow problem after 600 iterations - $l=4.1$ - Central differencing

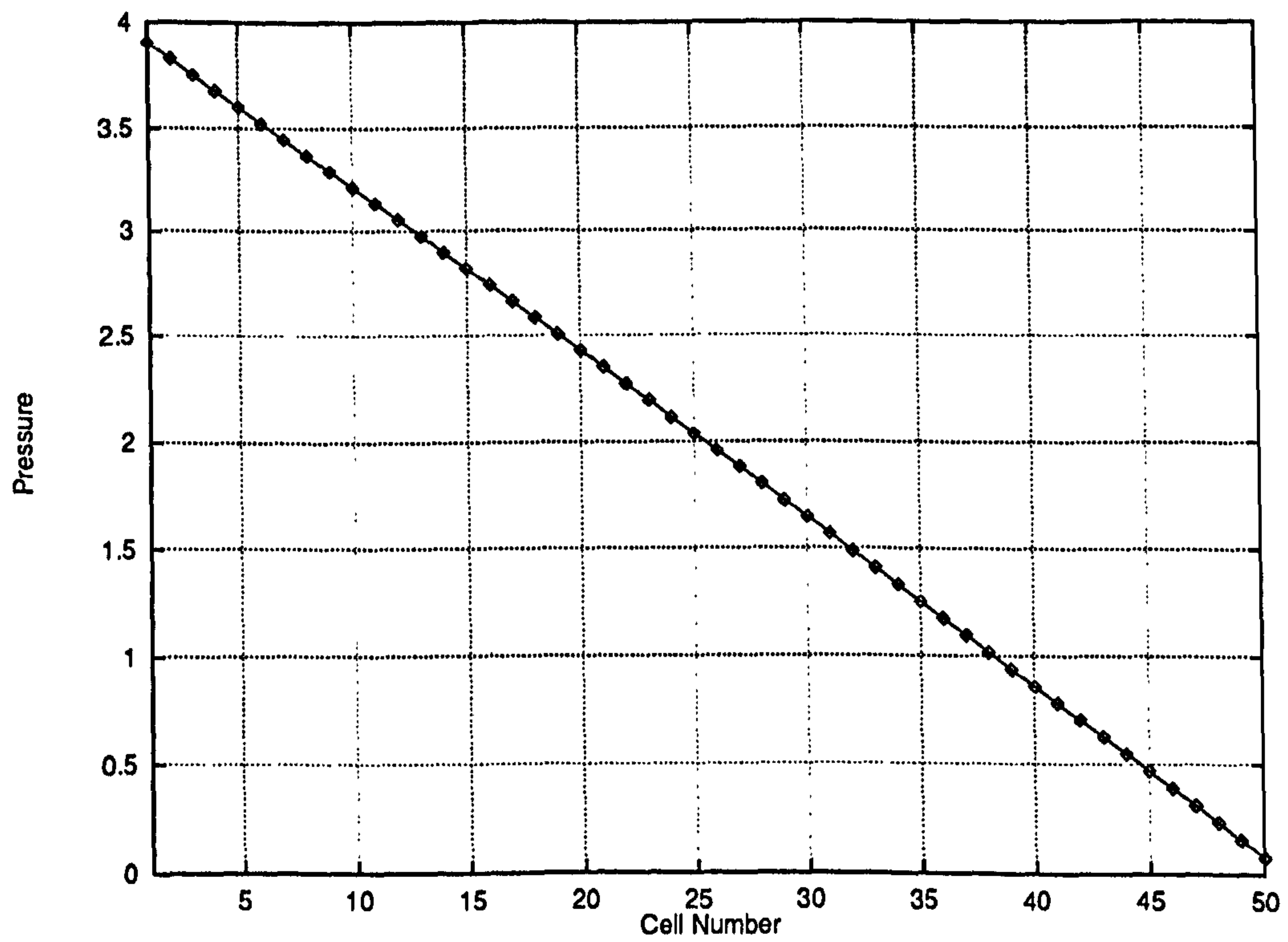


Figure A.9: Converged pressure solution for the one-dimensional multiphase flow problem – $l=4.1$ – Hybrid differencing

A.3 Multigrid Solutions

We consider the same test case as above, with $l = 4.0$, and wish to obtain multi-grid solutions (using a F(2,2) cycling strategy). Table A.1 and Figure A.10 summarise the results observed: multigrid methods work very well on this example. Furthermore, the convergence rates are very largely grid-independent. Finally, we have observed that the 4 grids computations were eight time faster than the equivalent single grid computations (Figure A.11).

Number of grids	Number of cells (fine grid)	Convergence factor (per relaxation work unit)
3	32	0.928
4	64	0.916
5	128	0.908
6	256	0.878

Table A.1: One-dimensional multiphase flow problem – Convergence factors at different grid levels

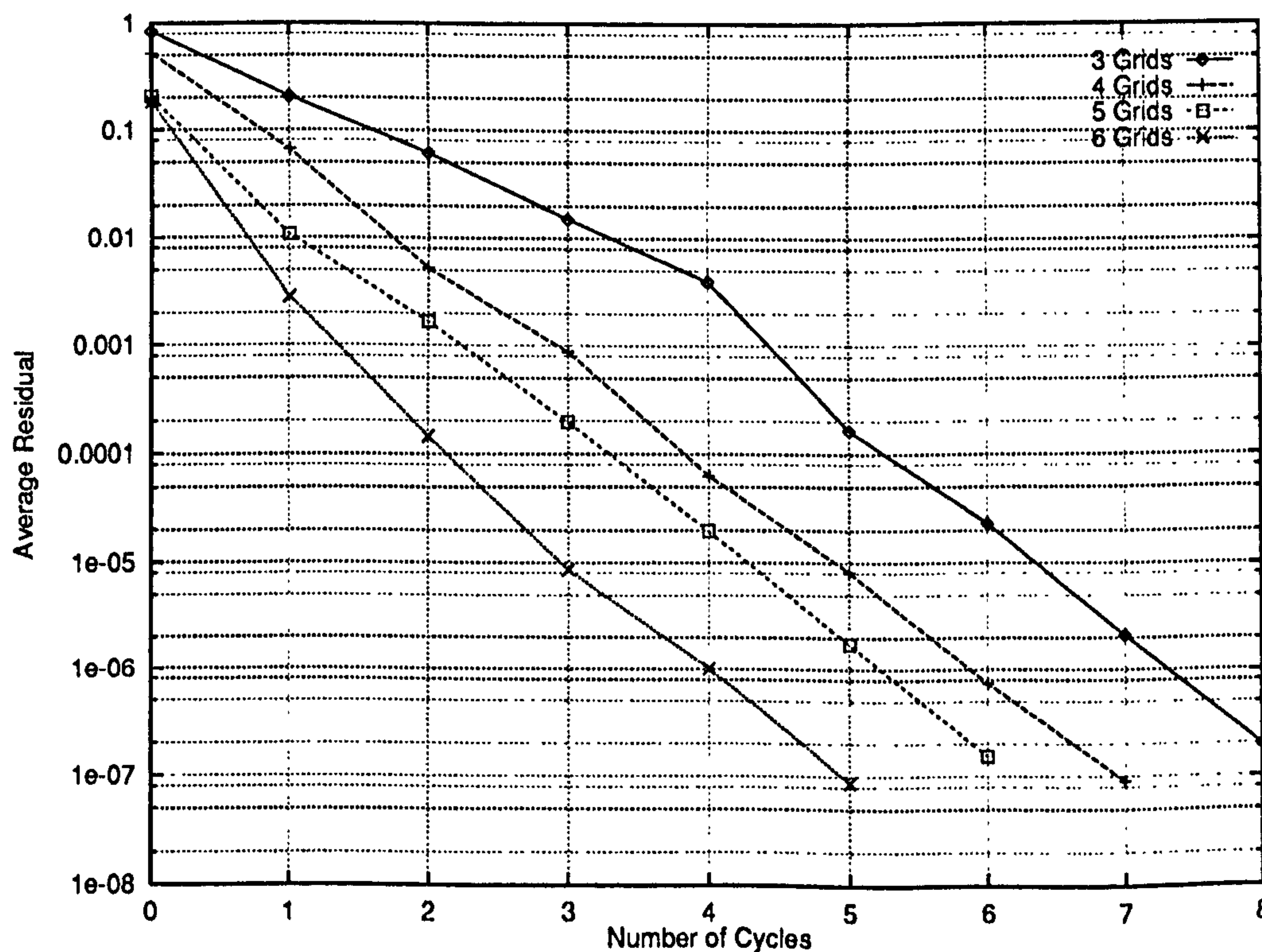


Figure A.10: One-dimensional multiphase flow problem – Multi-grid convergence rates for different grid levels

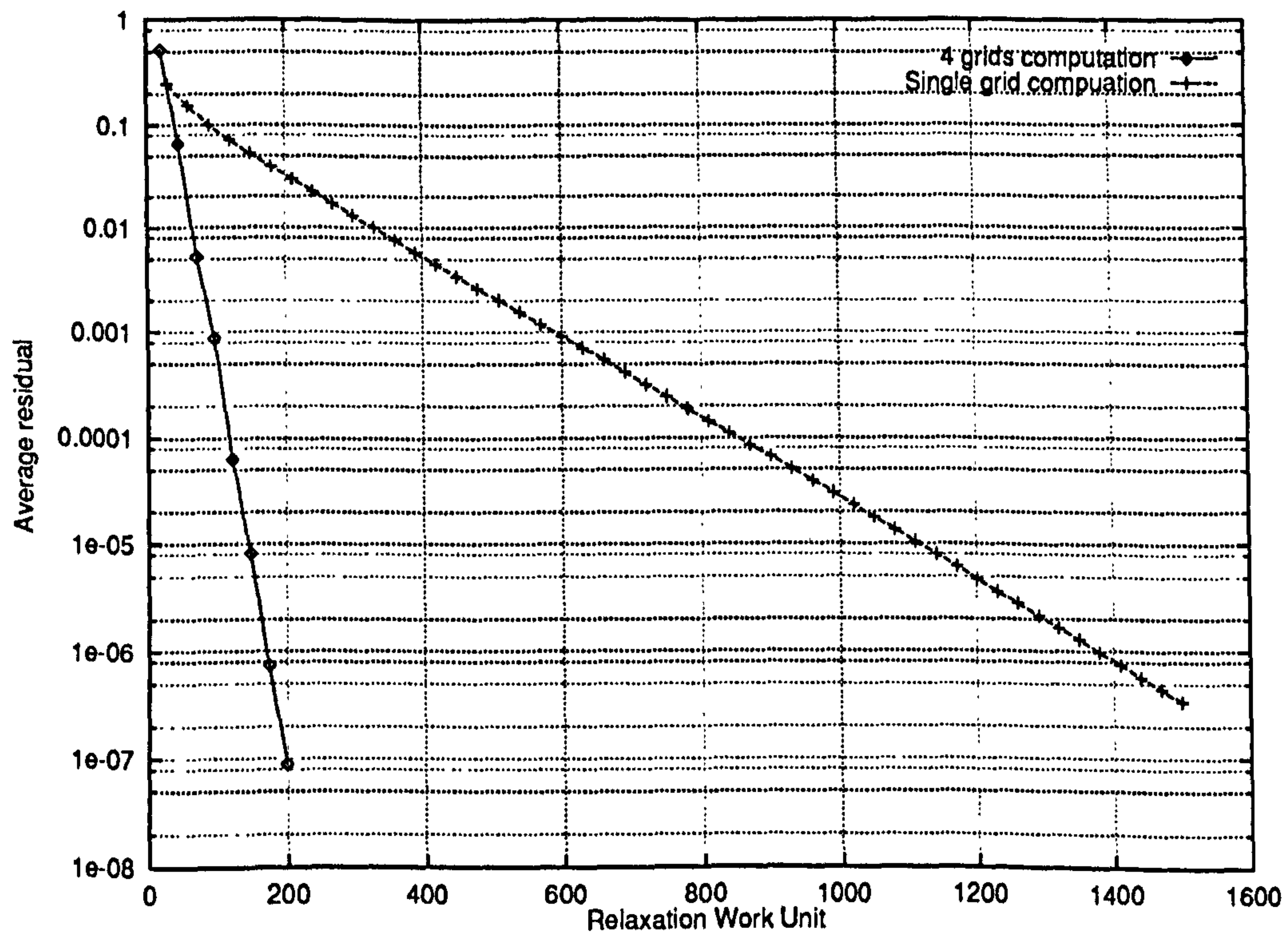


Figure A.11: One-dimensional multiphase flow problem – Comparison of the convergence rates for multi-grid and single grid computations

Appendix B

Solution Profiles for a Single Phase Entry Flow

Here, we give some solution profiles for the entry flow problem referred to in Section 5.4.14.

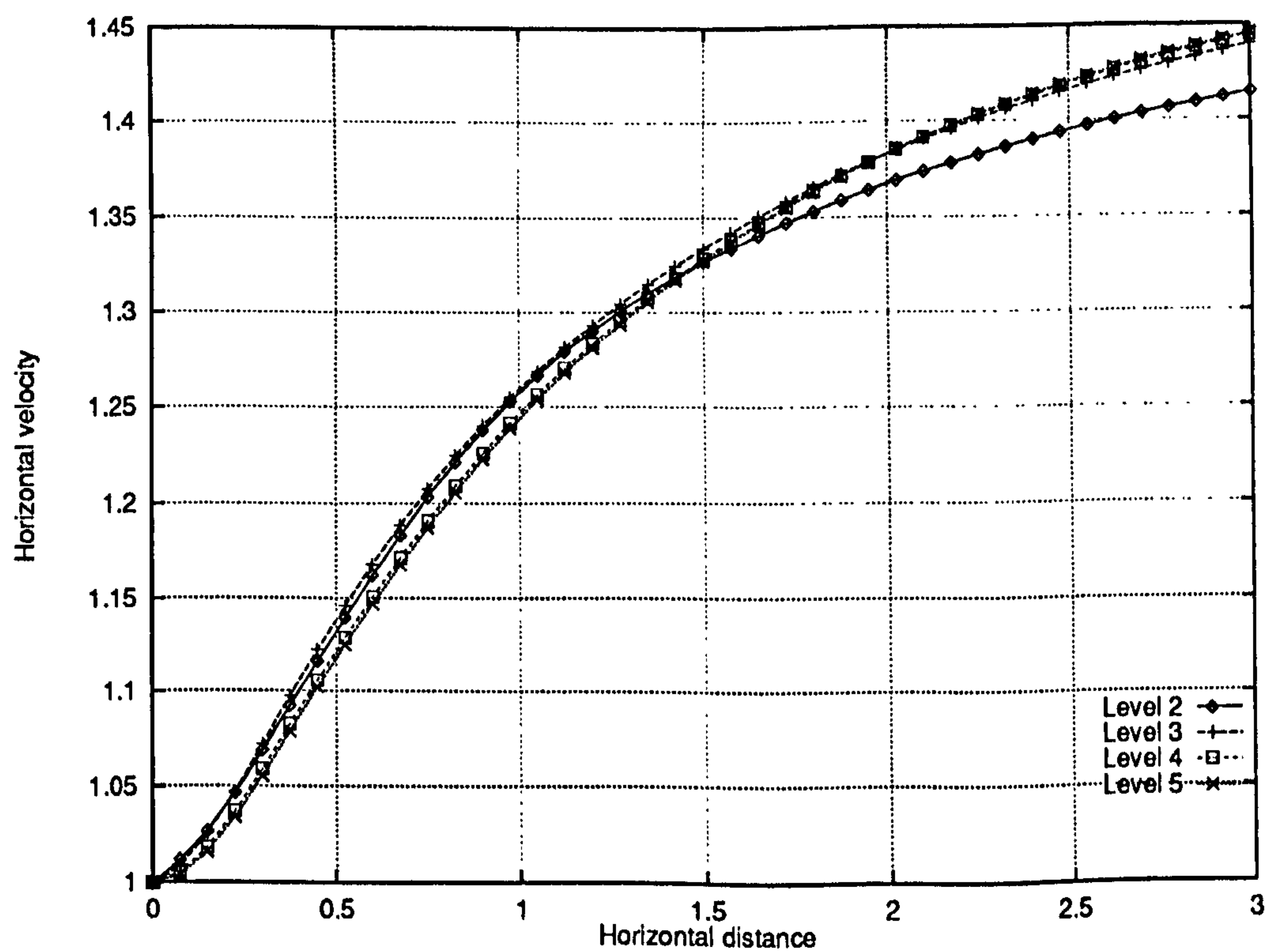


Figure B.1: Pseudo two-phase entry flow – Horizontal velocity along $y = 0.5$ for different finest grid levels

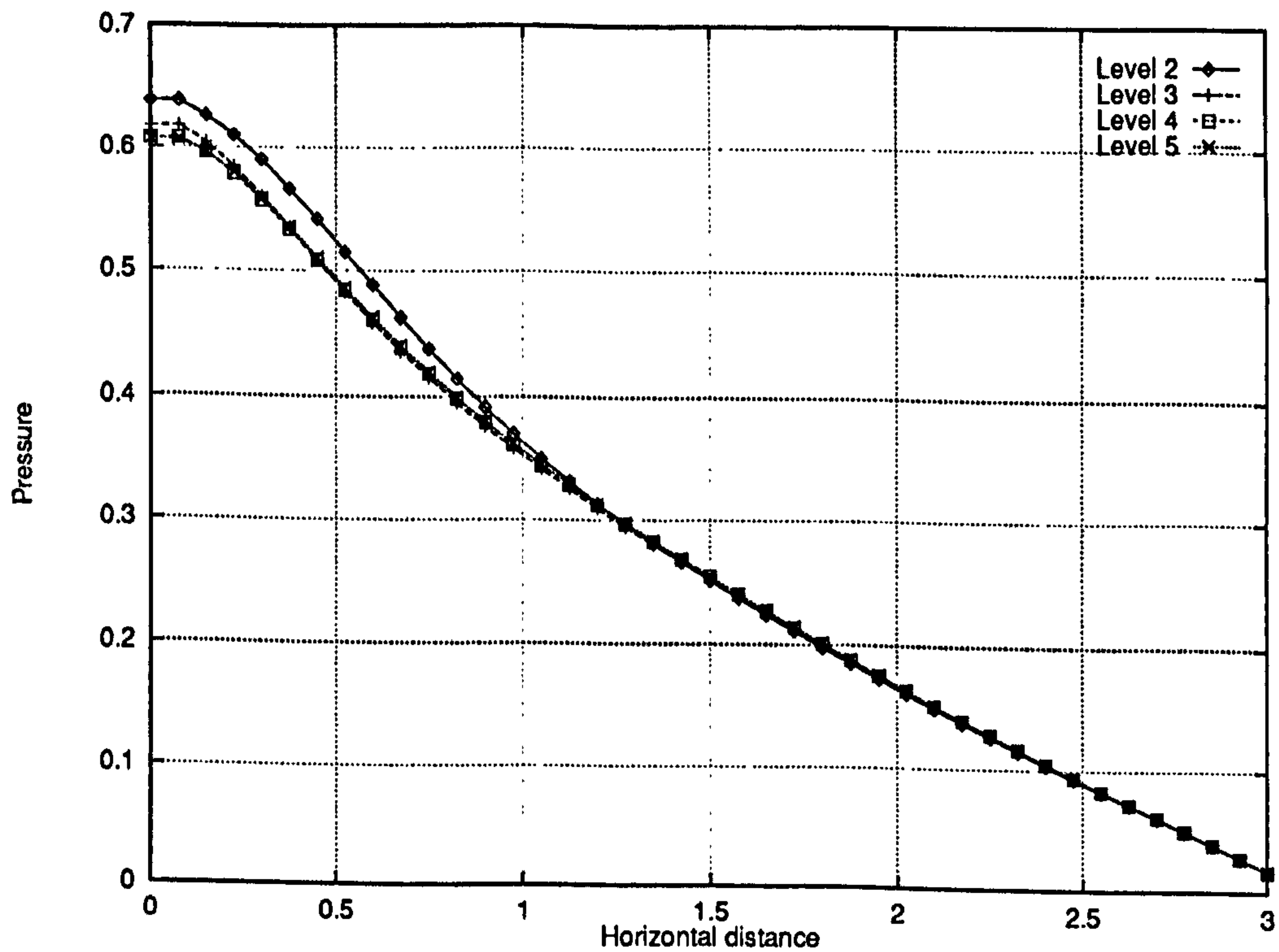


Figure B.2: Pseudo two-phase entry flow – Pressure along $y = 0.5$ for different finest grid levels

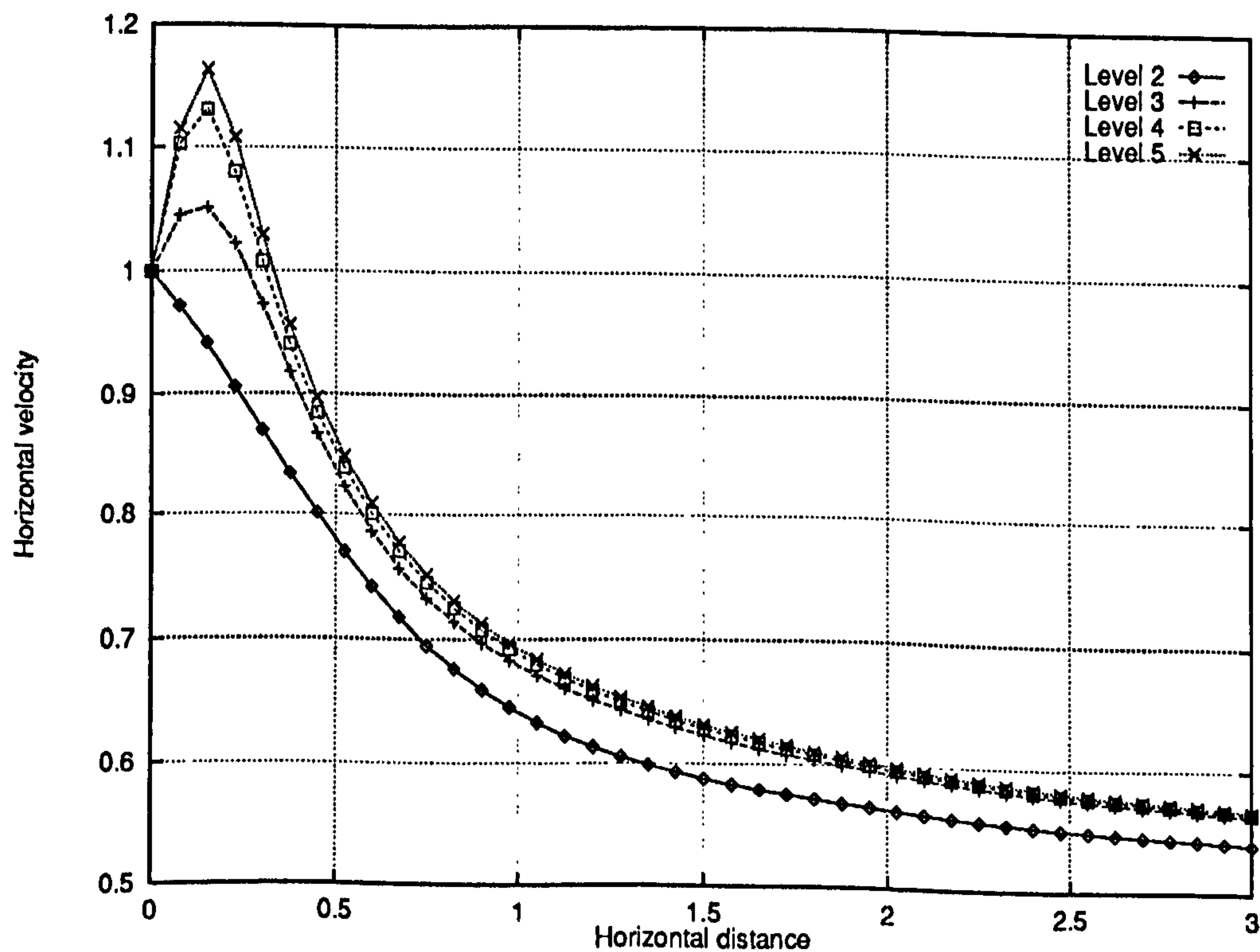


Figure B.3: Pseudo two-phase entry flow – Horizontal velocity along $y = 0.1$ for different finest grid levels

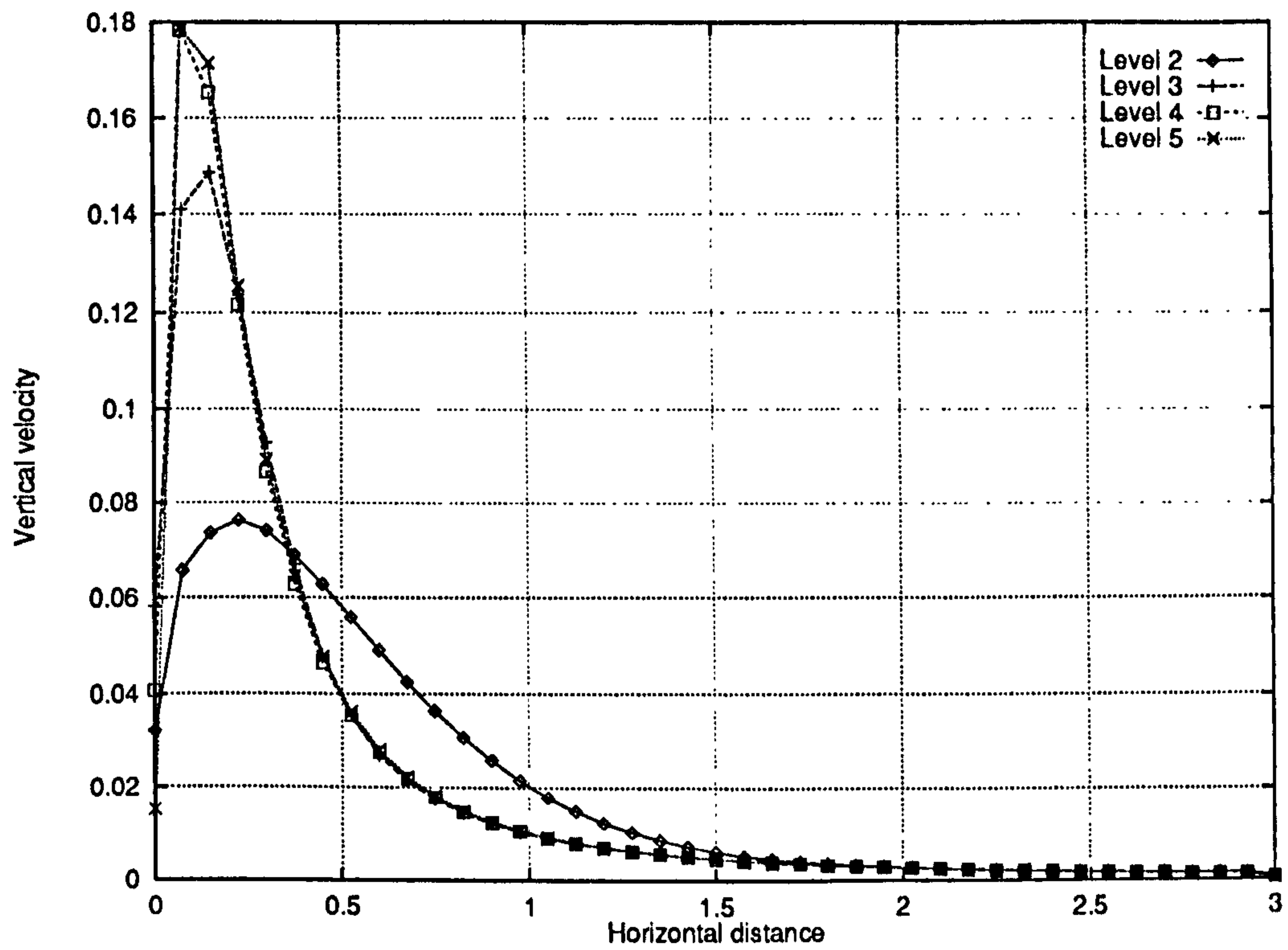


Figure B.4: Pseudo two-phase entry flow – Vertical velocity along $y = 0.1$ for different finest grid levels

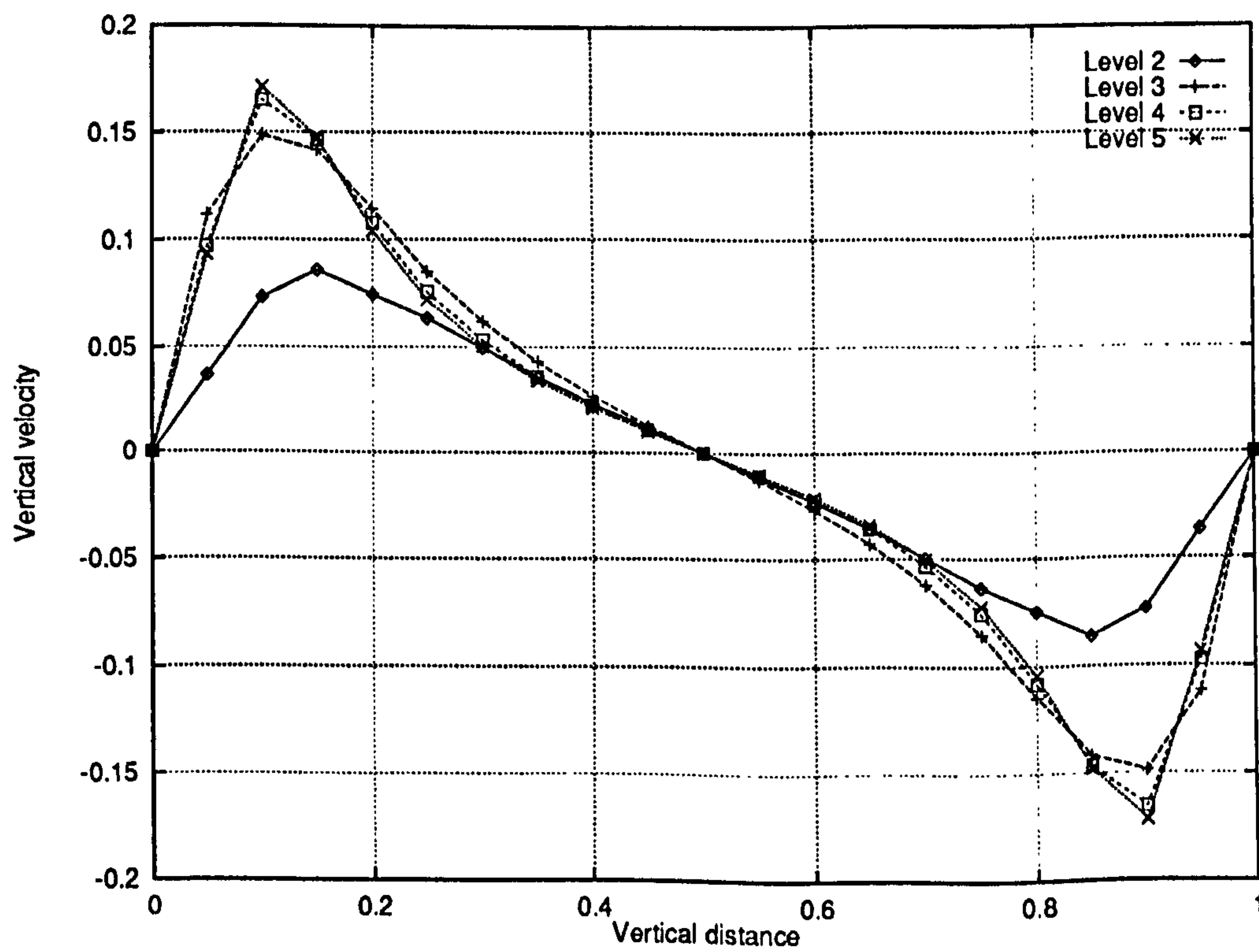


Figure B.5: Pseudo two-phase entry flow – Vertical velocity along $x = 0.15$ for different finest grid levels

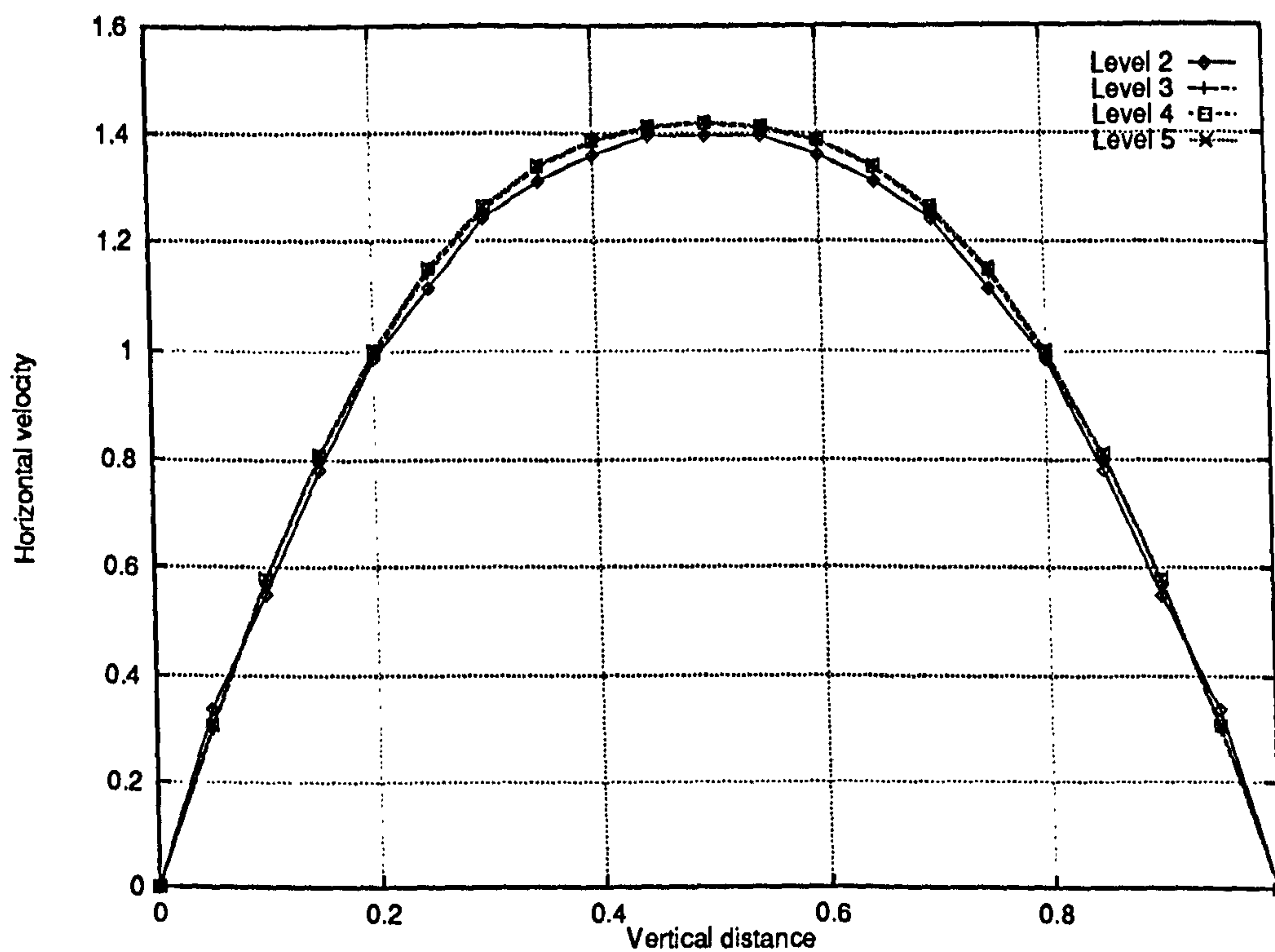


Figure B.6: Pseudo two-phase entry flow - Horizontal velocity along $x = 2.5$ for different finest grid levels

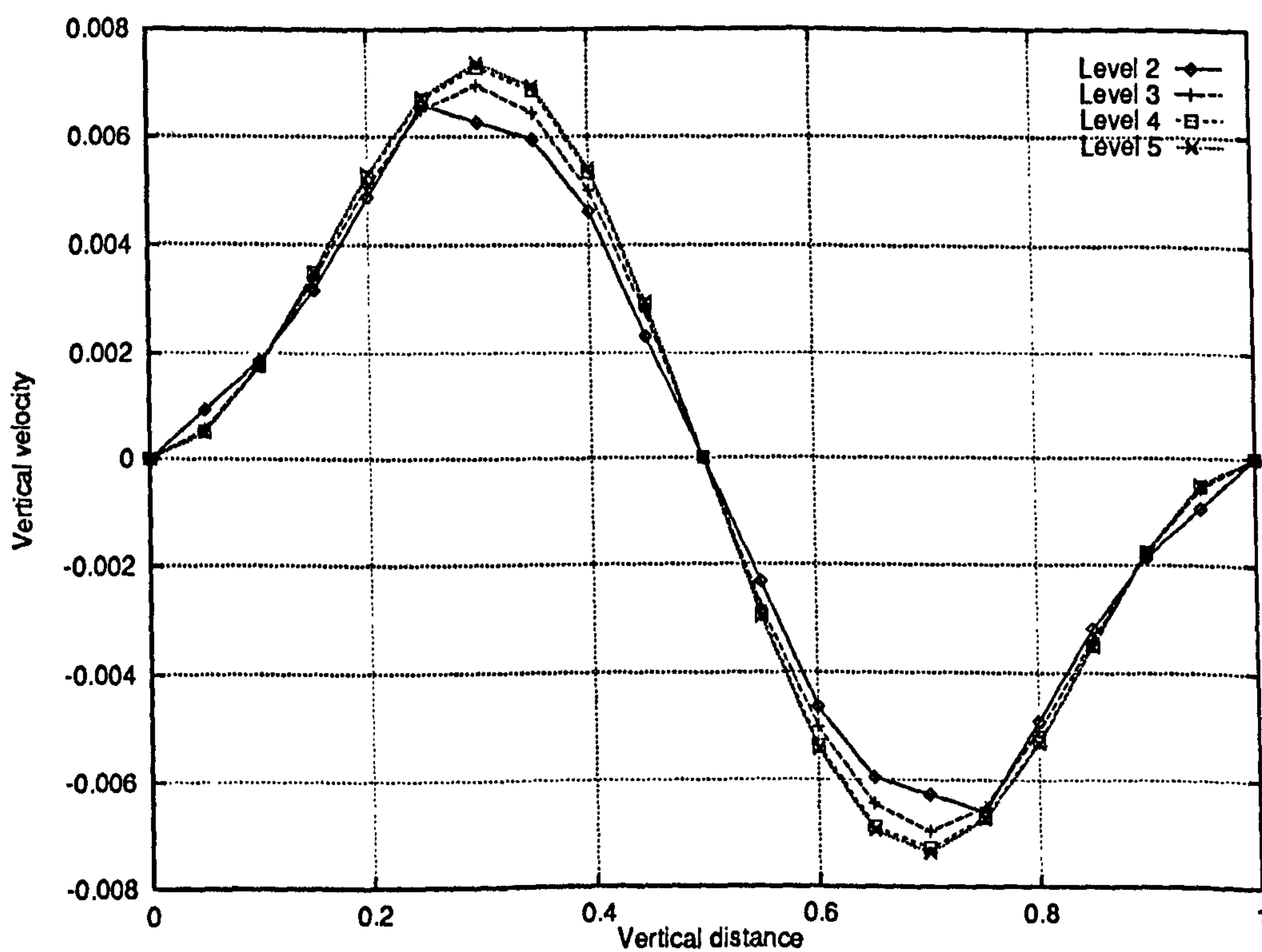


Figure B.7: Pseudo two-phase entry flow - Vertical velocity along $x = 2.5$ for different finest grid levels

Bibliography

- [1] C. P. Thompson, "A parallel adaptive multi-grid algorithm for the incompressible Navier-Stokes equations," in *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters (Proceedings of the NATO Advanced Workshop on Asymptotic-Induced Numerical Methods for Partial Differential Equations)*, pp. 293–309, Kluwer Academic Publishers, 1993.
- [2] G. K. Batchelor, *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.
- [3] P. Garabedian, *Partial Differential Equations*. New-York: Wiley, 1964.
- [4] G. A. Sod, *Numerical Methods in Fluid Dynamics, Initial and Initial Boundary-Value Problems*. Cambridge University Press, 1985.
- [5] R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial Value Problems*. New York: Wiley, 2nd ed., 1967.
- [6] P. J. Roache, *Computational Fluid Dynamics*. Albuquerque, N.M.: Hermosa, 1976.
- [7] R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*. Berlin, Heidelberg: Springer Verlag, 1983.
- [8] M. Ishii, *Thermo-Fluid Dynamic Theory of Two-Phase Flow*. Eyrolles, 1975.
- [9] H. B. Stewart and B. Wendroff, "Two-phase flow: Models and methods," *Journal of Computational Physics*, vol. 56, pp. 363–409, 1984.
- [10] A. Brandt and N. Dinar, "Multigrid solutions to elliptic flow problems," in *Numerical Methods for Partial Differential Equations* (S. V. Parter, ed.), pp. 53–147, Academic Press, 1979.
- [11] C. P. Thompson, W. R. Cowell, and G. K. Leaf, "On the parallelization of an adaptive multigrid algorithm for a class of flow problems," *Parallel Computing*, vol. 18, 1992.

- [12] C. P. Thompson, G. K. Leaf, and J. van Rosendale, "A dynamically adaptive multigrid algorithm for the incompressible navier-stokes equations – validation and model problems," *Applied Numerical Mathematics*, vol. 9, no. 3, pp. 511–532, 1992.
- [13] C. P. Thompson, G. K. Leaf, and S. P. Vanka, "The application of a multi-grid method to a buoyancy-induced flow problem," in *Multi-grid Methods* (S. F. McCormick, ed.), pp. 605–629, Dekker, 1988.
- [14] W. A. Mulder, "Analysis of a multigrid method for the Euler equations of gas dynamics in two dimensions," in *Multigrid Methods: Theory, Applications and Super-Computing* (S. McCormick, ed.), New-York: Dekker, 1988.
- [15] A. Brandt, *Guide to Multigrid Development*, pp. 220–312. Vol. 960 of Hack-bush [80], 1982.
- [16] F. H. Harlow and A. A. Amsden, "Numerical calculation of multiphase flows," *Journal of Computational Physics*, vol. 17, pp. 19–52, 1975.
- [17] W. Wulff, "Computational methods for multiphase flow," in *Multiphase Science and Technology* (G. F. Hewitt, J. M. Delhaye, and N. Zuber, eds.), vol. 5, ch. 3, Hemisphere Publishing Corporation, 1990.
- [18] I. Toumi and A. Kumbaro, "An approximate linearised Riemann solver for a two-fluid model," *Journal of Computational Physics*, vol. 124, pp. 286–300, 1996.
- [19] L. Sainsaulieu, "Finite volume approximation of two phase fluid flows based on approximate Roe-type Riemann solvers," *Journal of Computational Physics*, vol. 121, pp. 1–28, 1995.
- [20] F. Berger and J.-F. Colombeau, "Numerical solutions of one-pressure models in multifluid flows," *SIAM Journal on Numerical Analysis*, vol. 32, pp. 1139–1154, 1995.
- [21] K. H. Bendiksen, D. Malnes, R. Moe, and S. Nuland, "The dynamic two-fluid model OLGA: Theory and applications," *SPE Production Engineer*, May 1991.
- [22] M. Sussman, P. Smereka, and S. Osher, "A level-set approach for computing solutions to incompressible two-phase flow," *Journal of Computational Physics*, vol. 114, pp. 146–159, 1994.
- [23] W. J. Rider, D. B. Kothe, S. J. Mosso, J. H. Cerutti, and J. I. Hochstein, "Accurate solution algorithms for incompressible multiphase flows," *AIAA Paper 95/699*, 1995.
- [24] R. Saurel, A. Forrestier, D. Veyret, and J.-C. Loraud, "A finite volume scheme for two-phase compressible flows," *International Journal for Numerical Methods in Fluids*, vol. 18, pp. 803–819, 1994.

- [25] M. J. Andrew, "Accurate computation of convective transport in transient two-phase flow," *International Journal for Numerical Methods in Fluids*, vol. 21, pp. 205–222, 1995.
- [26] D. A. Drew and L. A. Segel, "Analysis of fluidized beds and foams using averaged equations," *Studies in Applied Mathematics*, vol. 50, pp. 233–257, 1971.
- [27] J. H. Mahaffy, "A stability-enhancing two-step method for fluid flow calculations," *Journal of Computational Physics*, vol. 46, pp. 329–341, 1982.
- [28] J. R. T. Lahey, Jr., M. L. de Bertodano, and O. C. Jones, Jr., "Phase distribution in complex geometry conduits," *Nuclear Engineering and Design*, vol. 141, pp. 177–201, 1993.
- [29] T. C. Chawla and M. Ishii, "Equations of motions for two-phase flow in a pin bundle of a nuclear reactor," *International Journal of Heat and Mass Transfer*, vol. 21, pp. 1057–1068, 1978.
- [30] T. C. Chawla and M. Ishii, "Two-fluid model of two-phase flow in a pin bundle of a nuclear reactor," *International Journal of Heat and Mass Transfer*, vol. 23, pp. 991–1001, 1980.
- [31] J. Fabre, A. Line, and L. Peresson, "Two-fluid/two-flow-pattern model for transient gas-liquid flow in pipes," in *Multiphase Flow – Proceedings of the Fourth International Conference*, pp. 269–284, BHRA, 1989.
- [32] R. Moe and K. H. Bendiksen, "Transient simulation of 2-3D stratified and intermittent two-phase flows. Part I. Theory," *International Journal for Numerical Methods in Fluids*, vol. 16, pp. 989–1005, 1993.
- [33] M. Watson, "Instabilities, waves and slugs in pipeline two-phase flows," in *Proceedings of the Third Conference on Hyperbolic Problems (Uppsala, Sweden, 11-15 June, 1990)*, Studentlitteratur, 1990. Lund.
- [34] A. D. Fitt, "The character of two-phase gas/particulate flow equations," *Applied Mathematical Modelling*, vol. 17, pp. 339–354, 1993.
- [35] P. S. Gough and F. J. Zwarts, "Modelling heterogeneous two-phase reacting flow," *AIAA Journal*, vol. 17, pp. 17–25, 1979.
- [36] J. I. Ramos, "One-dimensional, time-dependent, homogeneous, two-phase flow in volcanic conduits," *International Journal for Numerical Methods in Fluids*, vol. 21, pp. 253–278, 1995.
- [37] P. Lezeau and C. P. Thompson, "MTD research proposal (second phase)," Tech. Rep. MTDP5/PROP/CRAN/PL960918, AMAC, Cranfield University, 1996.

- [38] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti, "Modelling merging and fragmentation in multiphase flows with SURFER," *Journal of Computational Physics*, vol. 113, pp. 134–147, 1994.
- [39] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*. Elsevier Applied Science Publishers, 1979.
- [40] H. d' Arcy, *Les fontaines publiques de la Ville de Dijon*. Paris: Dalmont, 1856.
- [41] G. B. Wallis, *One-Dimensional Two-Phase flow*. McGraw-Hill, 1969.
- [42] G. F. Hewitt, J. M. Delhay, and N. Zuber, eds., *Multiphase Science and Technology*, vol. 1–5. Hemisphere Publishing Corporation, 1990.
- [43] S. L. Soo, *Multiphase Fluid Dynamics*. Beijing: Science Press, 1990.
- [44] I. G. Currie, *Fundamental Mechanics of Fluids*. New-York: McGraw-Hill, 1974.
- [45] J. O. Hinze, *Turbulence*. McGraw-Hill, 2nd ed., 1975.
- [46] M. Ishii, "Two-fluid model for two-phase flow," in *Multiphase Science and Technology* (G. F. Hewitt, J. M. Delhay, and N. Zuber, eds.), vol. 5, ch. 1, Hemisphere Publishing Corporation, 1990.
- [47] D. A. Drew, "Averaged field equations for two-phase media," *Studies in Applied Mathematics*, vol. 50, pp. 133–166, 1971.
- [48] D. A. Drew and L. A. Segel, "Averaged equations for two-phase flows," *Studies in Applied Mathematics*, vol. 50, pp. 205–231, 1971.
- [49] D. A. Drew, "Mathematical modelling of two-phase flow," in *Annual Review of Fluid Mechanics*, vol. 15, pp. 261–291, Annual Reviews Inc., 1983.
- [50] B. Mohammadi and O. Pironneau, *Analysis of the K- ϵ Turbulence Model*. John Wiley, 1994.
- [51] R. F. Hoskins, *Generalised Functions*. Ellis Horwood Limited, 1979.
- [52] M. Ishii and G. Kocamystafaogullari, "Two-phase flow models and their limitations," in *Proceedings of the NATO Advanced Research Workshop on the Advances in Two-Phase Flow and Heat Transfer* (S. Kakac and M. Ishii, eds.), 1982.
- [53] C. W. Hirt and B. D. Nichols, "Volume Of Fluid (VOF) method for the dynamics of free boundaries," *Journal of Computational Physics*, vol. 39, p. 201, 1981.
- [54] F. H. Harlow and J. E. Welch, "The MAC method: a computing technique for solving visous incompressible transient fluid flow problems involving free surfaces," *Physics of Fluids series A*, vol. 8, p. 2182, 1965.

- [55] P. D. Lax, *Hyperbolic Conservation Laws and the Mathematical Theory of Shock Waves*. Philadelphia: SIAM, 1973.
- [56] G. B. Whitham, *Linear and Non-Linear Waves*. Wiley-Interscience, 1974.
- [57] C. Hirsch, *Numerical Computation of Internal and External Flows*, vol. 1. John Wiley & sons, 1990.
- [58] C. Hirsch, *Numerical Computation of Internal and External Flows*, vol. 2. John Wiley & sons, 1990.
- [59] H. B. Stewart, "Stability of two-phase flow calculation using the two-fluid models," *Journal of Computational Physics*, vol. 33, pp. 359–270, 1979.
- [60] M. Arai, "Characteristics and stability analysis for two-phase flow equation systems with viscous terms," *Nuclear Science and Engineering*, vol. 74, pp. 77–83, 1980.
- [61] D. R. Liles and W. H. Reed, "A semi-implicit method for two-phase fluid dynamics," *Journal of Computational Physics*, vol. 26, p. 77, 1978.
- [62] G. Patnaik, R. H. Guirguis, J. P. Boris, and E. S. Oran, "A barely implicit correction for flux corrected transport," *Journal of Computational Physics*, vol. 71, p. 1, 1987.
- [63] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equation*. Cambridge University Press, 1994.
- [64] A. Harten, "High resolution schemes for hyperbolic conservation law," *Journal of Computational Physics*, vol. 49, pp. 357–393, 1983.
- [65] P. L. Roe, "Characteristic based schemes for the Euler equations," in *Annual Review of Fluid Mechanics*, vol. 18, pp. 337–365, Annual Reviews Inc., 1986.
- [66] AEA Technology, Harwell, Didcot, United Kingdom, *PLAC Pipe Line Analysis Code, User Guide and Manual*, May 1996.
- [67] Computational Fluid Dynamic Services, Harwell Laboratory, United Kingdom, *CFX 4.1 Flow Solver User Guide*, October 1995.
- [68] Fluent Inc., *Fluent User's Guide – Version 5.3*, 1995.
- [69] D. B. Spalding, "Numerical computation of multiphase fluid flow and heat transfer," in *Recent Advances in Numerical Methods in Fluids* (C. Taylor and K. Morgan, eds.), ch. 5, Pineridge Press Limited, 1980.
- [70] S. V. Patankar and D. B. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, vol. 15, pp. 1787–1806, 1972.
- [71] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*. Hemisphere, 1980.

- [72] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics, Volume 2: Specific Techniques for Different Flow Categories*. Springer-Verlag, 1st ed., 1988.
- [73] S. P. Vanka, "Block implicit multigrid solution of Navier-Stokes equations in primitive variables," *Journal of Computational Physics*, vol. 65, no. 1, 1986.
- [74] G. B. Deng, J. Piquet, P. Queutey, and M. Visonneau, "New fully coupled solution of the Navier-Stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 19, no. 2, pp. 605–639, 1994.
- [75] P. R. McHugh and D. A. Knoll, "Fully coupled finite volume solutions of the incompressible Navier-Stokes and energy equations using an inexact Newton method," *International Journal for Numerical Methods in Fluids*, vol. 19, no. 5, pp. 439–455, 1994.
- [76] A. Brandt, "Multigrid techniques: 1984 guide – with applications to fluid dynamics," in *Computational Fluid Dynamics, Lecture Series 1984-04*, Belgium: Von Karman Institute, 1984.
- [77] A. Brandt, "Multi-level adaptive solutions to boundary value problems," *Mathematics of Computations*, vol. 31, pp. 333–390, 1977.
- [78] J. H. Bramble, *Multigrid Methods*. Longman Scientific & Technical, 1993.
- [79] W. L. Briggs, *A Multigrid Tutorial*. Society for Industrial and Applied Mathematics, 1987.
- [80] W. Hackbush, ed., *Multigrid Methods*, vol. 960 of *Lecture Notes in Mathematics*. Berlin: Springer-Verlag, 1982.
- [81] K. Stuben and U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, pp. 1–176. Vol. 960 of Hackbush [80], 1982.
- [82] R. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1962.
- [83] D. Young, *Iterative Solution of Large Linear Systems*. New-York: Academic Press, 1971.
- [84] W. Hackbush, *Multigrid Convergence Theory*, pp. 177–219. Vol. 960 of *Lecture Notes in Mathematics* [80], 1982.
- [85] C. P. Thompson, "Developments in the solution of the incompressible navier-stokes equations: Multigrid algorithms, parallelism and software tools," Tech. Rep. BCS 91/5, IBM Bergen Scientific Centre, 1991.
- [86] M. C. Thompson and J. H. Ferziger, "An adaptive multigrid technique for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 82, pp. 94–121, 1989.

- [87] R. H. Ni, "A multiple grid scheme for solving the Euler equations," *AIAA Paper 81/1025*, 1981.
- [88] A. Jameson, "Solution of the Euler equations for two dimensional transonic flow by a multigrid method," *Applied Mathematics and Computation*, vol. 13, pp. 327–355, 1993.
- [89] W. A. Mulder, "Multigrid relaxation for the Euler equations," *Journal of Computational Physics*, vol. 60, pp. 235–252, 1985.
- [90] W. A. Mulder, "A new multigrid approach to convection problems," *Journal of Computational Physics*, vol. 83, pp. 303–323, 1988.
- [91] P. Lezeau and C. P. Thompson, "MTD technical discussion," Tech. Rep. MTDP5/TR/CRAN/PL960919, AMAC, Cranfield University, 1996.
- [92] D. Bai and A. Brandt, "Local mesh refinement multilevel techniques," *SIAM Journal for Scientific and Statistical Computing*, vol. 8, pp. 109–134, 1987.
- [93] M. J. Berger, "On conservation at grid interfaces," *SIAM Journal on Numerical Analysis*, vol. 24, pp. 967–984, 1987.
- [94] C. Liu, "The finite volume element (FVE) and fast adaptive composite grid (FAC) methods for the incompressible Navier-Stokes equations," in *Proceedings of the fourth Copper Mountain conference on multigrid methods* (J. Mandel, ed.), pp. 319–337, SIAM, 1989.
- [95] R. E. Ewing, R. D. Lazarov, and A. T. Vassilev, "Local refinement techniques for elliptic problems on cell centered grids i: error analysis," *Mathematics of Computations*, vol. 56, pp. 437–461, 1991.
- [96] R. E. Ewing, R. D. Lazarov, and A. T. Vassilev, "Finite difference scheme for parabolic problems on composite grid with refinement in space and time," *SIAM Journal on Numerical Analysis*, vol. 31, pp. 1605–1622, 1994.
- [97] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, pp. 484–512, 1984.
- [98] D. B. Spalding, "A novel finite difference formulation for differential expression involving both first and second derivatives," *International Journal for Numerical Methods in Engineering*, vol. 4, pp. 551–559, 1972.
- [99] B. P. Leonard and J. E. Drummond, "Why you should not use 'hybrid', 'power-law' or related exponential schemes for convective modelling – there are much better alternatives," *International Journal for Numerical Methods in Fluids*, vol. 20, pp. 421–441, 1995.
- [100] C. A. J. Fletcher, *Computational Techniques for Fluid Dynamics, Volume 1: Fundamental and General Techniques*. Springer-Verlag, 1st ed., 1988.

- [101] B. P. Leonard, "A stable and accurate convective modelling procedure based on quadratic upstream interpolation," *Computer Methods in Applied Mechanics and Engineering*, vol. 19, pp. 59–98, 1979.
- [102] W. M. Lai, D. Rubin, and E. Krempl, *Introduction to Continuum Mechanics*. Pergamon Press, 3rd ed., 1993.
- [103] J. M. Ortega, *Iterative Solution of Non-Linear Equations in Several Variables*. New-York: Academic Press, 1970.
- [104] J. Su and J. E. Renaud, "Automatic differentiation in robust optimisation," in *6th AIAA/NASA/ISSMO Symposium on Multi-Disciplinary Analysis and Optimisation – Part II*, 1996.
- [105] A. Carle, L. L. Green, and C. H. Bischof, "Applications of automatic differentiation in CFD," *AIAA Paper 94/2197*, 1994.
- [106] C. H. Bischof, A. Carle, G. Corliss, and A. Griewank, "ADIFOR: Generating derivative codes for fortran programs," *Scientific Programming*, vol. 1, pp. 11–29, 1992.
- [107] C. H. Bischof, "Experiences with the application of the ADIC automatic differentiation tool to the CSCMDO 3-D volume grid generation code," *AIAA Paper 96/0716*, 1996.
- [108] J. D. Pryce and J. K. Reid, "AD01, a fortran90 code for automatic differentiation," tech. rep., Rutherford Appleton Laboratory, 1996. In preparation.
- [109] M. Metcalf and J. K. Reid, *Fortran90 Explained*. Oxford University Press, 1990.
- [110] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: the Art of Scientific Programming*. Cambridge University Press, 2nd ed., 1992.
- [111] N. S. Wilkes. Private communication.
- [112] L. C. McInnes. Private communication.
- [113] Z. Gao. Phd Thesis, Cranfield University, In preparation.

Final Credits

This document was typeset using \LaTeX ,
probably the best piece of software in the world.
The figures were mostly produced using
the GNU PLOT package.