Coupling hybrid CFD models in simulating IC engine flows

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University

Metadata Record: [https://dspace.lboro.ac.uk/2134/13063](https://dspace.lboro.ac.uk/2134/13063)

Publisher: © Xiaodan Huang

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Pilkington Library

Author/Filing Title .................. HUANG
Vol. No. .............. Class Mark ...........

Please note that fines are charged on ALL overdue items.
Coupling Hybrid CFD Models in Simulating IC Engine Flows

by

Xiaodan Huang

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of
Doctor of Philosophy of Loughborough University
4 May, 2000
by Xiaodan Huang
Dedication

To my family...
Acknowledgements

The author would like to thank Professor Jim McGuirk for his crucial supervision during the final stage of this work and guidance on thesis writing, in particular, his great effort on reviewing and correcting the draft.

The author also would like to thank Dr. Andrew Manners for his supervision in the first four years of this work.

Thanks must also go to Professor Tim Gordon who has provided generous support during difficult time of this work.

The author is very grateful for the help received from Nicola Hughes, Penny Pilkington, Scott Baird and John Hing during thesis writing.

The author has also received constant support from brother Dr. Chiang Huang, and cannot say enough of the appreciations.
Abstract

A novel concept which couples 1D and 2D CFD models in a simulation of unsteady IC engine flows was investigated, and such a coupled model was developed.

Two unified solution procedures which are capable of predicting mixed compressible and incompressible flow fields found in an engine were developed and comparatively studied. One is the pressure correction algorithm, the other is the block implicit algorithm. They provided platforms for the implementation of coupled models. Second order spatial and Euler backward time differencing schemes were adopted. The comprehensive comparative studies were performed on a variety of benchmark flows ranging from steady to unsteady, incompressible to compressible. The data documented have shown that the prediction qualities of the two algorithms were comparable in all calculations. The block implicit procedure required more storage memory generally but it converged faster in all cases except the incompressible flow calculations.

General strategies to couple the 1D CFD model with the 2D CFD model in one calculation were proposed. They were successfully incorporated in both of the unified solution procedures. The predictions from these coupled models for a series of unsteady benchmark flows were competitive in quality with those from single 2D CFD models, however, the computing costs involved were comparatively much lower. In these calculations, the coupled models integrated in the block implicit procedure produced faster convergence than those in the pressure correction procedure, but required more computing resource. In addition, the implicit coupling strategy was more efficient compared to the explicit counterpart.

A 1D and 2D coupled model integrated in the pressure correction procedure was applied to simulate a realistic cylinder-valve-pipe flow. The overall prediction quality is satisfactory compared with experimental measurements. Some discrepancies which occurred were largely attributed to numerical representations of valve mechanism and the lack of turbulence models. For this engine application, the coupled model has shown advantages in computing cost or straightforwardness over a conventional uniform 2D model or boundary condition model.
Contents

Dedication i
Acknowledgements ii
Abstract iii
Nomenclature xi

Chapter 1 Introduction 1

1.1 Brief History of CFD Simulation of IC Engines 2
1.2 Influence of IC Engine Flows on Engine Efficiency 2
1.3 Review of CFD Algorithms For Engine Flow Application 4
  1.3.1 Boundary Condition Method 4
  1.3.2 Secondary Calculation Method 6
  1.3.3 Comprehensive CFD Modelling Method 8
1.4 A Proposal for A Hybrid 1D/Multi-D Coupled CFD Method 10
1.5 Multiblock Grid Systems 12
1.6 Block Coupling Methods in Conventional Multiblock Systems 14
1.7 Review of CFD Algorithms For All Flow Speeds 16
1.8 Review of Numerical Time Marching Schemes 18
1.9 Objectives of the Current Research 21
1.10 Layout of Thesis 22
1.11 References 25

Chapter 2 Unified Pressure Correction Algorithm 31

2.1 Preamble 32
2.2 Review of Pressure-Correction Schemes 32
2.3 Review of Differencing Schemes Used in Pressure-Correction Algorithms 35
2.4 Governing Equations 37
2.5 Unified Pressure-Correction Algorithm 40
   2.5.1 Formulation of the Pressure-Correction Equation 42
   2.5.2 Retarded Pressure 46
2.6 Discretisation Scheme 48
   2.6.1 Review of Differencing Schemes for Convective Flux 48
   2.6.2 Second Order MUSCL TVD Method for Convective Flux 49
   2.6.3 Discretisation Scheme for Diffusive Flux 56
2.7 Boundary Conditions 57
   2.7.1 Overview 57
   2.7.2 Boundary Condition Implementation 58
      2.7.2.1 Subsonic Inlet Boundary 59
      2.7.2.2 Subsonic Outlet Boundary 59
      2.7.2.3 Solid Wall Boundary 60
      2.7.2.4 Centre Line Boundary 62
2.8 Rhie & Chow Pressure Smoothing 62
2.9 Linear Matrix Solver 63
2.10 References 64

Chapter 3 Unified Block Implicit Algorithm 67
3.1 Preamble 68
3.2 Review 68
3.3 Non-Dimensional Form of Governing Equations 69
3.4 Discretisation Scheme

3.4.1 Central Difference Scheme and Artificial Dissipation Model

3.4.2 Artificial Dissipation Terms Near the Boundary

3.4.3 Implicit or Explicit Artificial Dissipation

3.5 Iterative Matrix Inversion Method

3.6 Linear Matrix Solver: Gauss-Seidel Point Iteration

3.7 Multigrid Acceleration Technique

3.7.1 Nonlinear Multigrid Cycle: FAS

3.7.2 Restriction and Prolongation Method

3.7.3 Boundary Condition on Coarse Grid

3.7.4 Artificial Dissipation

3.8 Boundary Conditions

3.9 References

Chapter 4 General Strategy of Coupling Methodology

4.1 Preamble

4.2 Multiblock Grid System

4.3 Data Management

4.3.1 Cell Index System

4.3.2 Solution Variable Storage

4.3.3 Block Connectivity Table

4.4 Coupling Method for 2D/2D Coupled Models

4.5 Coupling Method for 1D and 2D Coupled Model

4.5.1 Conservative vs Non-conservative Interface Treatment

4.5.1.1 Non-Conservative Interpolation Method
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.3 Pressure Smoothing</td>
<td>144</td>
</tr>
<tr>
<td>5.3.4 Summary</td>
<td>146</td>
</tr>
<tr>
<td>5.3.5 TDMA Line Solver</td>
<td>147</td>
</tr>
<tr>
<td>5.3.5.1 Modified SLOR Procedure</td>
<td>147</td>
</tr>
<tr>
<td>5.3.5.2 Modified TDMA Algorithm</td>
<td>151</td>
</tr>
<tr>
<td>5.4 Interface Evaluation in Energy Equation Discretisation</td>
<td>160</td>
</tr>
<tr>
<td>5.4.1 Convective Flux</td>
<td>160</td>
</tr>
<tr>
<td>5.4.2 Other Fluxes</td>
<td>162</td>
</tr>
<tr>
<td>Chapter 6 Coupling 1D and 2D Block Implicit Procedure</td>
<td>163</td>
</tr>
<tr>
<td>6.1 Preamble</td>
<td>164</td>
</tr>
<tr>
<td>6.2 General Equation Form</td>
<td>164</td>
</tr>
<tr>
<td>6.3 Interface Treatment of Artificial Dissipation Terms</td>
<td>172</td>
</tr>
<tr>
<td>6.4 Gauss-Seidel Point Iterative Solver</td>
<td>174</td>
</tr>
<tr>
<td>Chapter 7 Results and Discussion</td>
<td>177</td>
</tr>
<tr>
<td>7.1 Preamble</td>
<td>178</td>
</tr>
<tr>
<td>7.2 Steady-State Flow Calculations</td>
<td>179</td>
</tr>
<tr>
<td>7.2.1 Introduction</td>
<td>179</td>
</tr>
<tr>
<td>7.2.2 Compressible Flows Over the Ni Bump</td>
<td>180</td>
</tr>
<tr>
<td>7.2.2.1 Subsonic Flow</td>
<td>180</td>
</tr>
<tr>
<td>7.2.2.2 Transonic Flow</td>
<td>183</td>
</tr>
<tr>
<td>7.2.2.3 CPU Time and Memory Usage</td>
<td>187</td>
</tr>
<tr>
<td>7.2.2.4 Summary</td>
<td>189</td>
</tr>
<tr>
<td>7.2.3 Square Cavity: Incompressible Driven Cavity Flow</td>
<td>190</td>
</tr>
<tr>
<td>7.2.3.1 Reynolds Number 100 Flow</td>
<td>190</td>
</tr>
</tbody>
</table>
Chapter 8 Recommendations For Future Work and Conclusions 283

8.1 Summary and Conclusions 284

8.2 Recommendation for Future Work 285

8.2.1 Improve Numerical Valve Simulation 286

8.2.2 Improve 1D and 2D Coupling 288

8.2.3 Implementation of a Turbulence Model 289

8.2.4 Extend 1D/2D Coupling Method to 1D/2D/3D Coupling 290

8.2.5 Integrate the PISO Method With Pressure-Correction Approach 294

8.2.6 Investigate High Order Time Scheme 294

8.2.7 Investigate Acceleration Method for Low Speed Flows 294

8.3 References 296
Nomenclature

\( x, y \) \hspace{1cm} \text{two dimensional Cartesian co-ordinates}
\( \xi, \eta \) \hspace{1cm} \text{two dimensional Curvilinear co-ordinates}
\( x_i \) \hspace{1cm} \text{three dimensional Cartesian co-ordinate, } i=1,2,3
\( \xi_i \) \hspace{1cm} \text{three dimensional Curvilinear co-ordinate, } i=1,2,3
\( i, j \) \hspace{1cm} \text{unit vectors in } \xi, \eta \text{ co-ordinate directions respectively}
\( u, v \) \hspace{1cm} \text{Cartesian velocity components in } x, y \text{ direction respectively}
\( u', v' \) \hspace{1cm} \text{corrections to } u, v \text{ respectively}
\( \rho u, \rho v \) \hspace{1cm} \text{mass flux components in } x \text{ and } y \text{ co-ordinate direction respectively}
\( (\rho u)', (\rho v)' \) \hspace{1cm} \text{corrections to } \rho u, \rho v \text{ respectively}
\( u_i \) \hspace{1cm} \text{Cartesian velocity components in } x_i \text{ co-ordinate direction}
\( U, V \) \hspace{1cm} \text{(or } U^c, V^c) \hspace{1cm} \text{contravariant velocity components in } \xi, \eta \text{ direction respectively}
\( \rho \) \hspace{1cm} \text{density}
\( \rho' \) \hspace{1cm} \text{correction to density}
\( t \) \hspace{1cm} \text{time}
\( p \) \hspace{1cm} \text{static pressure}
\( p' \) \hspace{1cm} \text{correction to static pressure}
\( \tau_{ij} \) \hspace{1cm} \text{shear stress component in } x_i \text{ direction acting on an } x_j \text{ direction face}
\( E \) \hspace{1cm} \text{specific total energy}
\( H \) \hspace{1cm} \text{specific total enthalpy}
\( e \) \hspace{1cm} \text{specific internal energy}
\( q_i \) \hspace{1cm} \text{heat flux in } x_i \text{ direction}
\( K \) \hspace{1cm} \text{thermal conductivity}
\( T \) \hspace{1cm} \text{temperature}
\( C_v \) \hspace{1cm} \text{specific heat coefficient at constant volume}
\( C_p \) \hspace{1cm} \text{specific heat coefficient at constant pressure}
\( \mu \) \hspace{1cm} \text{dynamic viscosity}
\( \lambda \) \hspace{1cm} \text{second coefficient of viscosity}
\( R \) \hspace{1cm} \text{gas constant}
\( \text{Re} \) \hspace{1cm} \text{Reynolds number}
\( L \) \hspace{1cm} \text{characteristic length}
\( J \) \hspace{1cm} \text{Jacobian of co-ordinate transformation matrix}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Δt</td>
<td>time step size in a time marching numerical scheme</td>
</tr>
<tr>
<td>Δx, Δy</td>
<td>spatial mesh size in x, y direction respectively</td>
</tr>
<tr>
<td>n</td>
<td>time level in a time marching numerical scheme</td>
</tr>
<tr>
<td>k</td>
<td>iteration level in an iterative numerical scheme</td>
</tr>
<tr>
<td>A, B, C, D</td>
<td>coefficient matrices of a discrete algebraic equation system</td>
</tr>
<tr>
<td>S</td>
<td>source term (or matrix) of a discrete algebraic equation system</td>
</tr>
<tr>
<td>I</td>
<td>unit matrix</td>
</tr>
<tr>
<td>M</td>
<td>Mach number</td>
</tr>
<tr>
<td>( \tilde{\lambda}_i )</td>
<td>upwind difference approximation of first derivative ( \frac{\partial}{\partial x_i} )</td>
</tr>
<tr>
<td>( \delta_i^n )</td>
<td>difference approximation of ( n^{th} ) derivative ( \frac{\partial^n}{\partial x_i^n} )</td>
</tr>
<tr>
<td>f</td>
<td>flux at a control volume cell face in a differencing scheme</td>
</tr>
<tr>
<td>( \Psi )</td>
<td>TVD non-linear limiter</td>
</tr>
<tr>
<td>r</td>
<td>ratio of two adjacent variations of cell centre dependent variable</td>
</tr>
<tr>
<td>( \text{Pr} )</td>
<td>Prandtl number</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>ratio of specific heat coefficients, i.e. ( C_p/C_v )</td>
</tr>
<tr>
<td>E, F</td>
<td>flux terms in PDEs</td>
</tr>
<tr>
<td>( \bar{D} )</td>
<td>artificial diffusion flux in Jameson's artificial diffusion model</td>
</tr>
<tr>
<td>( \varepsilon^{(n)} )</td>
<td>coefficient for ( n^{th} ) order Jameson's artificial dissipation term</td>
</tr>
<tr>
<td>( \beta_{i , m} ) ( \text{for } m = 0, \pm 1, \pm 2 )</td>
<td>coefficients in a five-point stencil</td>
</tr>
<tr>
<td>( G )</td>
<td>amplification factor of a differencing scheme</td>
</tr>
<tr>
<td>( \phi_x, \phi_y )</td>
<td>phase angle</td>
</tr>
<tr>
<td>( k_x, k_y )</td>
<td>wave number</td>
</tr>
<tr>
<td>ielm</td>
<td>cell index number in indirect addressing system</td>
</tr>
<tr>
<td>NI, NJ</td>
<td>dimension of grids in ( \xi, \eta ) direction respectively</td>
</tr>
<tr>
<td>ijk</td>
<td>cell index in the structured grid system</td>
</tr>
<tr>
<td>ijkw, ijke, ijks, ijkn</td>
<td>west, east, south and north cell, respectively, of cell ijk</td>
</tr>
<tr>
<td>cf_i</td>
<td>secondary cell face at 1D/2D interface</td>
</tr>
<tr>
<td>( \chi )</td>
<td>specific function representing a discretisation scheme</td>
</tr>
<tr>
<td>( a_{p,w,e,s,n} )</td>
<td>coefficients in the algebraic equation of cell ( p )</td>
</tr>
<tr>
<td>( \phi )</td>
<td>general transported flow variable</td>
</tr>
<tr>
<td>Q (or q)</td>
<td>dependent variable of the equation</td>
</tr>
</tbody>
</table>
$W_h$ flow variable in a MG scheme at a grid level where the cell space is $h$

$P_{2h}$ forcing function in MG scheme during restriction process

$I_{2h}^h$ interpolation operator in MG scheme during prolongation process
Chapter 1

Introduction
Introduction

1.1 Brief History of CFD Simulation of IC Engines

The application of numerical methods to the calculation of Internal Combustion (IC) engine flows has a history of about half a century. Early numerical analyses were based on models developed from the first law of thermodynamics, and are normally known as thermodynamic models or zero dimension models (Heywood 1988). Since the 1960s, assisted partially by developments in computer technology, engine models based on a detailed fluid dynamics approach began to be employed. Such models are examples of the application of Computational Fluid Dynamics (CFD) techniques which have the huge advantage that they can describe geometrical properties of engine systems and associated flow behaviours. By the 1980s, CFD models had earned the status of being useful tools for engine flow simulation (Gosman 1985a, Gosman 1985b). Meanwhile, the thermodynamic model has been reduced largely to a complementary analysis used in conjunction with a CFD model to address thermodynamically related physical processes such as combustion or even as a secondary model to provide the boundary conditions for the CFD model. Nowadays, engine CFD techniques and methodology have become relatively mature and efficient, so the cost of engine prediction has been much reduced. However, there is still a long way to go before a three-dimensional unsteady CFD simulation of the whole engine system under real-time working condition becomes sufficiently accurate and affordable that it represents the default tool used by engine designers.

1.2 Influence of IC Engine Flows on Engine Efficiency

Real gas flows in an IC engine system are generally three-dimensional. However, they could be approximated as two- or one-dimensional in certain sub-components. Frequently encountered examples are axisymmetrical approximations of flows in combustion chambers with symmetrical configuration (Ramos, et al. 1980) and one-dimensional approximation of flows in long and straight manifold pipes (Liu, et al. 1996). Engine flows under real working condition are also unsteady due to the moving parts in the system such as the piston and valve/port. Moreover, flows in various components of an engine are strongly coupled and correlated. The interaction between them is especially strong during transient processes, it influences not only the unsteady flow evolution but also the steady-state flow structure.
Introduction

While the in-cylinder flow structure has a direct influence on engine performance, the manifold flow has only an indirect impact. For example, it has been found that the scavenging efficiency of a two-stroke engine has a close correlation with the flow pattern in the cylinder (Carapanayotis 1988, Nishimoto et al. 1984), and engine performance depends heavily on an efficient scavenging flow. On the other hand, the in-cylinder flow structure is significantly influenced by the manifold flow as shown in the experimental studies by Nishimoto et al. (Nishimoto et al. 1984) and the computational studies of Diwakar (Diwakar 1987), hence the manifold flow is equally important with respect to its influence on engine performance (Ravi et al. 1992, Benson 1979). In addition, engine performance is also influenced by the unsteadiness of the flow (Heywood 1988). One good example is the dynamic pressure waves in the manifold during the engine flow exchange process, which have a great impact on engine output power (Heywood 1988). These waves are generally generated at the valve or port opening that connects the combustion chamber and manifold system. They are subsequently modified upon reflection at either end of the inlet/exhaust ducts, be they junctions with the atmosphere or with an engine cylinder or a crankcase. The motion of these waves can be adjusted ('tuned') during the intake and exhaust processes, so that a favourable impact on engine power output can be achieved (Brunt 1994). For example, tuned pressure waves in the intake flow can push more fresh gases into the cylinder, resulting in an increase of engine volumetric efficiency or breathing capacity. The pressure waves can be tuned in the exhaust manifold of two-stroke engines so that a low pressure is present during the suction period, leading to more burnt gases being pushed out of the cylinder and consequently more fresh air being drawn into the cylinder. Similarly, a high back-pressure could be present at the time when the exhaust port is due to close so that there is no excess fresh air flow out of the cylinder; this also leads to improvement in engine volumetric efficiency and subsequent enhancement of engine power output. Compared to four-stroke engines, the performance of two-stroke engines is more dependent on dynamic effects such as pressure waves. The scavenging process and the charging process are two typical examples in this type of engine which rely heavily on unsteady pressure waves to improve efficiency.
Introduction

Hence, the understanding of such flows, including the unsteady dynamics of the sub-components of an engine system such as manifold ducts is an essential element in using CFD as a design tool in the most fruitful way.

1.3 Review of CFD Algorithms For Engine Flow Application

The scale of engine flow CFD predictions has long been restricted by the capacity and speed of computers as well as the robustness of numerical methodology. Therefore, although the simultaneous transient simulation of all components in the engine system is most appropriate as discussed above, it was impossible to realise in the early days and is a daunting task even at present. Traditionally, CFD flow analysis has been concentrated on the region of most interest only, while other regions or components of an engine system are excluded from the computation. In the 80's, calculations could only be performed on individual components, e.g. the in-cylinder flows, the manifold duct flows (Diwakar 1987), and only a few geometrical features could be retained in the computational representation. In the past decade, computing power has increased drastically and CFD algorithms have become increasingly mature. As a result, the number of components that could be included in the same calculation has increased continuously, as have the details of the geometrical features that could be represented. For example, by the mid 90's, the port (Haworth, et al. 1990) or even the whole intake manifold (Aita, et al. 1990, McLaudress, et al. 1996) could be included in the cylinder flow simulation, and many important geometry details could be retained. Recently, a transient four-stroke engine simulation encompassing combustion chamber, intake port and intake runner has been achieved on a supercomputer (McLandress et al. 1996, Stephenson, et al. 1996). However, such CFD predictions are still exceedingly expensive or impossible on the less powerful computing facilities, and hence a modelling approach which is based on linking engine components in CFD calculations is still commonplace. The following sections describe some methods for achieving this.

1.3.1 Boundary Condition Method

When only part of an engine system is included in the calculation, artificial boundaries or interfaces are created. The physical flow conditions at the locations of these artificial
Introduction

boundaries must be prescribed. They are frequently supplied by experimental measurements (Andrews et al. 1991). However, experiments are not always feasible because of the high cost or the limited ability of available experimental facilities. This is especially problematic when transient calculations are carried out. Therefore, it is also common that the interface boundary conditions are estimated. For example, Adachi et al. (Adachi et al. 1982) and Carpenter et al. (Carpenter et al. 1986) prescribed the port flow velocities following some theoretical or experimental guidelines in their studies of the two-stroke engine scavenging process. Chung (Chung 1995) derived the boundary conditions at the two ends of intake/exhaust ducts from thermodynamic laws and gas flow equations for a pipe flow simulation. Heywood (Heywood 1988) adopted a quasi-steady method which is based on the steady one-dimensional equations for the engine flow to compute the mass flow rate into and out of the four-stroke engine cylinder through the exhaust or inlet valve to provide the inlet boundary conditions to a detailed model of the in-cylinder process. Blair (Blair 1996) derived analytical conditions from the thermodynamic laws to deduce a series of formulae describing the boundary conditions at pipe-cylinder intersections. His method has resulted in a quite complex boundary condition procedure which was employed to provide the boundary conditions for engine pressure wave computations.

Understandably, in this boundary condition method approach, great efforts have been made to approximate accurately the flow conditions on these artificial boundaries. If the artificial boundary coincides with an existing restriction where the physical flow conditions are known, e.g. the closed exhaust valve in a four-stroke engine, little error would be generated from the boundary condition approximation. If the computational boundary is placed in a region where flows are changing rapidly, the accuracy of the computation is difficult to guarantee because of the difficulty in obtaining a plausible flow estimation for this boundary (Haworth 1990). For example, in the two-stroke engine, experimental studies (Bardsley 1989) have shown that details of the port flow varied dramatically as the geometry changed in time, hence it is not easy to specify flow conditions at the port. In addition, both the pipe and port flow strongly affect the scavenging process and should not, in general, be supplied merely as boundary conditions to the in-cylinder flow simulation. Therefore, it is always recommended that the port flow
Introduction

should be included in the two-stroke in-cylinder flow calculation whenever possible. This is why most of the recent work (Epstein et al. 1991) computes the in-cylinder flow together with the pipe/port flows. This would also allow the prediction of the effects of engine port design changes on scavenging efficiency.

In addition to the difficulties in prescribing flow conditions on these artificial boundaries, artificial boundaries can induce another problem, that is the decoupling of flows which are physically continuous and strongly interacting, in particular in transient flows. It is very difficult to represent the transient interactions between flows inside the computational domain and those that have been excluded from the computation by mere boundary condition specification. Thus, two major sources of numerical errors are introduced at the computation boundaries, which may be especially acute for transient flow simulations.

1.3.2 Secondary Calculation Method

As mentioned above, boundary conditions are particularly difficult to specify in a two-stroke engine computation because the scavenging flow evolution in the combustion chamber is strongly coupled with the flows in the intake, exhaust and transfer ports/pipes. Transient experimental measurements are often too expensive to be considered merely as input to a CFD prediction and if expensive measurements are to be made the role of CFD is questionable. On the other hand, there are no analytical formulae that can be used to calculate the boundary conditions accurately. Under such circumstances, other means of supplying accurate boundary conditions must be found. One widely used method is to perform a separate (sometimes called “two-way coupling precursor”) numerical calculation in the regions that are to be excluded from the main computation (Haworth et al. 1990) to obtain the flow conditions on the interface to the main computational flow region. The role of the precursor numerical calculation is to complement the main computation and hence it may be referred to as a secondary calculation. A simple secondary calculation is usually preferred and the main and secondary calculations are coupled at the boundary. The main difference between this method and the boundary condition method described in the previous subsection is that simultaneous computation of all relevant engine components is achieved to some extent, although not in one calculation.
Introduction

When secondary calculations are employed, the engine flow field is divided into multiple calculation regions, and the option to use specific flow models in each region is introduced. It has been common in the past to apply a complex CFD model to the main flow region under question and a thermodynamic based model to the secondary calculation.

In a thermodynamic model, the solution domain is usually chosen to be a region of the engine system which starts and ends at a particular flow restriction, e.g. the cylinder volume or the intake or exhaust manifold (or a section of the manifold). This solution domain is then modelled as an open thermodynamic system which contains a homogeneous gas at a uniform thermodynamic state. The mass of gas in the volume can increase or decrease with time, and the time-dependent gas state is determined from continuity and the first law of thermodynamics for an unsteady open system, that is, the conservation of mass and energy of an open system. When information is known about the mass flow rate into and out of the volume through its open ends, as well as the heat transfer and work transfer across its boundaries and the boundary displacement, the flow properties of the thermodynamic model can be computed. Generally, the mass flow information is obtained from calculations of one-dimensional isentropic compressible flow.

Both Chung (Chung 1995) and Rakopoulos (Rakopoulos 1995) have employed a one-dimensional CFD model interfaced with a thermodynamic model in the simulation of unsteady gas flows in intake and exhaust systems. In their work, one-dimensional models were applied to the manifold pipes while the combustion chamber or the prechamber were approximated by thermodynamic models to provide the necessary boundary conditions. Pearson et al. (Pearson 1990) also modelled the cylinder flow based on the first law of thermodynamics in their CFD studies of intake pipe flows focusing mainly on the volumetric efficiency. The accuracy of their results seems quite acceptable.

The above described thermodynamic modelling approach is often referred to as a ‘filling and emptying’ model. It can characterise time-varying phenomena, but because it uses spatially averaged assumptions, it cannot describe the spatial variation of flow properties, nor can it predict propagating pressure waves. Chen’s studies (Chen 1992) have shown
Introduction

that the filling-and-emptying model filters the effects of wave augmentation and attenuation. It is thus unable to give satisfactory prediction of engine performance for those engines with strong effects of pressure waves. Therefore, when this type of thermodynamic model is used in the secondary calculation, it is not suitable to be applied to unsteady pressure wave calculations for an engine. However, if the thermodynamic model is replaced by some other models, such as one- or two-dimensional CFD models, which are capable of capturing unsteady pulsation flows, the secondary calculation approach can become suitable for unsteady engine flow application. This theme will be expanded in later sections.

On the whole, the employment of a secondary numerical simulation to provide the necessary boundary conditions leads to a more accurate calculation of the flows in the region of major focus. Furthermore it prevents, under some circumstance, the problem of physical flow decoupling which is inevitable if the boundary condition method is utilised.

1.3.3 Comprehensive CFD Modelling Method

Recently, it has become feasible to compute a number of engine components simultaneously within the same CFD calculation (Tu et al. 1991, McLandress et al. 1996, Stephenson, et al. 1996), mostly using a multi-domain or multi-block framework (see below). Such a computation can still be considered as being composed of a main calculation and secondary calculations. However, since the same CFD model is applied in all parts of the domain (both main and secondary calculations), this is better viewed as a comprehensive CFD calculation of the engine flow.

Generally, the multiple domains would be loaded into the main memory simultaneously if computing facilities allow this (McLandress et al. 1996, Stephenson, et al. 1996). Otherwise, they may be loaded one at a time and computed one by one in sequence (Epstein et al. 1991). In the second approach, because only one block is present in the main memory at a time, the other blocks have to be buffered outside of the memory. This involves shuffling of data from the main memory to disk and vice versa, and the repetition of some routines each time a block is re-loaded. Obviously a large amount of overhead
Introduction

time is used in such calculations as a result of these operations, which could reduce drastically the efficiency and economy of the overall computation.

Epstein et al. (Epstein et al. 1991) have used this approach in their studies of a two-stroke engine flow. Three three-dimensional CFD models have been employed in the cylinder/port, exhaust pipe and transfer pipe regions respectively, occupying three independent blocks in the multi-domain system. The flow in each block was computed separately, i.e. the three regions were loaded into the computer individually. As a result, at any stage of the calculation the block interface between two regions was similar to an artificial boundary. The calculations of the flows in the exhaust and transfer pipes would provide the boundary conditions at the interfaces for the calculations of the cylinder/port flows and vice versa. Conditions on the interface were modified whenever the associated blocks were updated.

This approach can of course provide a much more accurate representation of flow conditions at the block interface, compared to the method described in the last subsection. However, the employment of the same order of multidimensional CFD models here is perhaps not always necessary. Much simpler models can be sufficient in some regions of flow, especially if they are only being considered for the purpose of providing an accurate boundary condition required by the main calculation. It would be an unnecessary waste of computing resource to apply a higher order dimensional CFD model to the flow region if a lower order dimensional CFD model were sufficient. There are actually many flow regions in the engine system that can be approximated with some confidence using lower order one- or two-dimensional flow models. For example, the manifold flows during the engine exchange process. By the early 80's, it had already been shown that one-dimensional simulations were capable of predicting the pressure and mass flow rate accurately for many manifold systems (Chapman 1982) and this type of model has since been recognised as a valuable analytical tool. A similar situation exists for two dimensional CFD modelling of combustion chambers which have predominantly symmetrical configurations. In fact, most 80's engine flow calculations were performed using two dimensional axisymmetric representations of the cylinder (Gosman et al. 1980, 1982), and were claimed as still being informative for many flow situations. Even nowadays, a one- or
two-dimensional calculation model can still represent an efficient and useful prediction, even though a three dimensional simulation may be affordable. For example, if the pipe flows in Epstein's work could be accurately and adequately modelled using one-dimensional CFD models, large amounts of memory and computing time could be saved or perhaps more usefully, memory savings in the essentially 1D regions could be transferred to provide better resolution of the 3D regions.

1.4 A Proposal for A Hybrid 1D/Multi-D Coupled CFD Method

According to the above discussion on existing CFD engine simulation methods, it is believed that they could be improved, particularly for application to a predominantly cylinder-valve/port-pipe flow engine simulation. A proposal is outlined in this work which is intended to remove the root cause of the unfavourable aspects in conventional approaches, so that a CFD engine model which is competitive in both accuracy and efficiency as well as cost can be obtained. As mentioned earlier in analysing the work of Epstein et al., if a lower dimensional model was acceptable in some regions of the engine system, the computing resource and time requirement could be reduced drastically compared to the approach adopted by Epstein et al., whilst the same prediction accuracy could be preserved. Therefore, it is proposed to investigate the use of hybrid CFD models which are constructed by combining CFD models of different spatial dimensions, appropriate to the flow processes in particular sub-domains of the engine system. To demonstrate the success of this approach, the coupled calculation of 1D and 2D models in a multi-block computational domain will be investigated since extension of the principle to the coupling of 1D to 3D or 2D to 3D zones should be straightforward.

The saving of computing resource in the proposed approach is obvious. In addition, this approach could enable the simultaneous loading of all blocks into the main memory and consequently the simultaneous computation of the whole flow domain. As a result, a further saving on computing time might be made due to the elimination of block exchange.

The predictive accuracy of this approach is not expected to deteriorate because the flow region at which the one-dimensional model will be applied is essentially one dimensional.
Introduction

in physical reality. As far as engine exchange flow applications are concerned, a two
dimensional model will be used to represent (for example) an axisymmetric in-cylinder
flow as well as the manifold region near the valve. The one dimensional model will be
applied only to the manifold flow region which is sufficiently far away from the valve
region.

It has been explained above that the conventional secondary calculation method is not
suitable for simulating unsteady flows with pulsating pressure waves. The main obstacle is
the use of the thermodynamic model in the secondary calculation to provide the boundary
condition. The new coupled approach proposed here can be considered as improving the
conventional method by substituting a one dimensional CFD model for the
thermodynamic model, so that the unsteady engine flows with moving pressure waves can
be addressed.

In actual fact Chen (Chen 1992) envisaged a few years ago that even when entire three
dimensional simulation might be affordable, one-dimensional CFD models or
thermodynamic models could still serve as complementary calculations to provide the
boundary conditions for the three dimensional analysis in order to shorten computing
time. Chen has compared the computational quality and efficiency of a one-dimensional
CFD model with some other models, including the filling-and-emptying thermodynamic
model and an acoustic model, in computing unsteady flow problems. His studies have
demonstrated the feasibility of the various simple 1D numerical models in predicting
several aspects of engine flows.

The primary focus of the present work is therefore to develop a methodology to address
coupling problems which occur between flow regions where CFD models of different
dimension are being used. If the interface between the one dimensional and two
dimensional regions is treated appropriately, the engine exchange flow simulated by the
new model will be able to achieve similar quality of predictions as that which would be
obtained by a uniform higher order dimensional model.
**Introduction**

### 1.5 Multiblock Grid Systems

Since one dimensional and two dimensional CFD models will be applied to various regions of the flow field in the proposed coupled approach, each of these regions needs to be marked out and distinguished in the computational domain. This leads rather obviously to the use of a multiblock solution technique (Lai et al. 1993). In a multiblock system, the computational domain is divided into a number of continuous non-overlapping subdomains (blocks or zones). Each of the blocks can then be easily made to correspond to one of the identified flow regions in the hybrid approach.

However, the multiblock system required by the present approach will be different from the conventional one. For the proposed hybrid approach, in the blocks where a one dimensional model is implemented, only one dimensional grids need to be generated. In other blocks, two dimensional grids will be needed. The line marking the division between the one dimensional block and the two dimensional block should obviously be within the flow region where the flow is one dimensional, and some skill or judgement (or trial and error) will be needed to search for this location. In the conventional multiblock system, the same dimensional grids are applied in all blocks.

Nevertheless, some similarities can still be identified between the approach proposed here and conventional multiblock systems. The general multiblock method has been around for over a decade now. In the conventional patched multiblock approach (Rai 1986a), the grids in each block are either generated independently or with multi-block dependencies. In the former approach, the grid lines at the common block interface can be discontinuous in general, and even the number of grid lines that intersect the interface can be different between adjacent blocks. In the latter approach, certain properties of the grid lines crossing the interface must be guaranteed, e.g. gradient continuity, or the number of interface intersection lines from both blocks must be the same.

Although, by definition, the grids in the presently proposed multiblock system are not consistent in dimension from block to block, they can still be interpreted as having this property if viewed from a special angle. A one dimensional grid can be seen as a special two dimensional grid in which the number of grid cells in one direction is unity. It can also
Introduction

be seen as a special three dimensional grid where the number of grid cells in two directions is unity. Two dimensional grids become special three dimensional grids in a similar manner. As a result, the lower order dimension grids in the present multiblock system can be converted to any higher order dimension grids, and subsequently a uniform dimension grid (as used in conventional multiblock systems) can be obtained throughout the domain. With this approach, the present grids become similar to conventional multiblock grids which are discontinuous at the block interface boundary.

The similarity established here between the present and conventional multiblock systems will be very significant when attention is turned to the coupling method between the various blocks which is an important part of this research. This is because the coupling methodology developed for conventional multiblock systems can, with the above similarities established, serve as a good guide to the development of the coupling strategy for the present hybrid coupled model system. Special features of the current system will have to be taken into consideration in the strategy developed for block inter-communication.

As well as conventional multiblock systems, a fundamental similarity is also found between the nature of the block interface in the presently proposed multiblock system and local cell interfaces in an embedded grid system (Powell et al. 1992, Berger et al. 1989). In an embedded grid, the locally refined grid cells are generated by splitting some of the existing (coarse mesh) grid cells into a number of smaller cells while other existing cells stay untouched. An internal cell interface is therefore created between unchanged grid cells and neighbouring split cells. There are special treatments applied to these cell interfaces in an embedded grid system. In the present hybrid (1D/2D) coupled approach, the cells in the one dimensional grid block and the two dimensional grid block at the block interface can be viewed as a coarse-level grid and locally refined grid respectively in embedded grids. Because of such similarities, the interface treatment adopted in the embedded grid approach can also provide guidelines to the development of coupling methods for the hybrid approach proposed and developed in this thesis.
Introduction

1.6 Block Coupling Methods in Conventional Multiblock Systems

It is known in the conventional multiblock application that special procedures, namely a block coupling method or an internal boundary treatment, needs to be developed and implemented in the calculation. This is because a continuous flow region has been divided into separate sub-regions in the multiblock system. The flow would be decoupled from one sub-region to its neighbour if a special coupling procedure was not enforced. The task of block coupling procedures is to establish communication between sub-regions so that a continuous flow calculation can be reproduced as it would be in a single block system.

A variety of coupling methods have been developed in the conventional multiblock approach. They may be largely classified into conservative or nonconservative, implicit or explicit groups. All of the coupling schemes satisfy the essential requirements of consistency with the main solution procedure, stability and efficiency.

In the non-conservative coupling method, the transfer of information from block to block is usually carried out strictly by interpolation (Atta et al. 1983), i.e. the flow state at the internal block boundary is evaluated by extrapolating the flow solutions from its adjacent blocks. For example in the work of Steger et al. (Steger et al. 1987), when a patched multiblock system (Rai 1986b) was used, they evaluated the interface conditions by extrapolating the interior solution from the block grids on either side and then averaging these. The averaging was to maintain numerical stability and permit signal propagation in both directions. On the other hand, when an overlapping grid system is used (Wright et al. 1992), the internal zonal conditions of a block were fully determined by interpolating the interior solution of its abutting block. In applications where the numerical calculation is insensitive to the flow conservation at an internal boundary, e.g. in a continuous flow field with no recirculations, shock waves or strong shear across the interface, the non-conservative method has proven to be satisfactory. Under such circumstances, the advantage of this coupling method of being able to achieve any order of interpolation accuracy at interface becomes attractive. However, in flows where large solution gradients or elliptic features such as recirculation exist in the vicinity of block boundaries, disastrous errors could be introduced into the solution by this method.
Introduction

In the conservative coupling method, a special flux-conserving interpolation approximation is imposed in the evaluation of the internal boundary in order to conserve the flow (Wright et al. 1992). It is acknowledged that when the flow involves discontinuities such as shocks, it is important that the numerical scheme is conservative so that the discontinuity can assume the right strength and physical location in the numerical solution. Therefore, it is essential that the numerical treatment at block boundaries should be conservative in order to allow the discontinuity to move freely across these interfaces. Furthermore, a conservative zonal treatment is important in the calculation of incompressible flows using pressure-correction techniques because fictitious mass sources could be generated at the zonal boundaries if the zonal boundary treatment was non-conservative and this would have a global effect on both the convergence rate and the solution accuracy of the scheme. In the conservative coupling approach, in some cases, a different method from that used in the interior is employed to compute the flux through the cell faces on interface (Rai 1986a, Rai 1986b, Volpe 1993). The basic principle used is that the flux through a cell face of one block should equal the sum of the fluxes through each of the segments into which this cell face was divided at the interface. Rai (Rai 1986a) has explained one way to calculate this segment flux sum (or, equivalently, the interpolation of the flux in the abutting zone) based on the assumption of a piecewise constant variation of the flux at the cell face. The resulting interpolation formula would be different if a piecewise linear or any other variation of the face flux were to be assumed.

In either the conservative or the nonconservative coupling method, the interface can be treated explicitly or implicitly. While the explicit interface treatment will not cause any problem for an explicit solution procedure, this may not be the case for an implicit solution algorithm. In an overall implicit solution procedure with an explicit interface treatment, the boundary conditions at the zonal boundaries would first be guessed and the blocks would be solved implicitly within each zone. Then the zonal boundary conditions would be updated, based on the solution obtained from the neighbouring zones. Rosenfeld et al. (Rosenfeld et al. 1994) have used both explicit and implicit zonal interface treatments in their ADI implicit solution procedure. The steady-state converged solution from both interface treatments was the same. However, the temporal accuracy as well as the convergence of the explicit zonal boundary were considerably poorer than the implicit
Introduction

The zonal boundary. This implies that (at least some of) the efficiency of the original implicit scheme was lost when using the explicit coupling, and explicit interface treatment adversely effected the accuracy of the unsteady solutions. In contrast, when the zonal boundary was treated implicitly, the convergence rate to the steady state was accelerated significantly compared to the explicit treatment, although the implicit interface treatment was more complex than the explicit one.

The coupled approach to be described in this work will obviously need an interface coupling method so that the calculations in the multiple blocks employing mixed one dimensional and two dimensional models can be connected. The current development will be guided by the coupling methods developed for both conventional multiblock grids and embedded grids. Since a moving shock frequently appears in the exhaust manifold system in transient engine exchange flows, a conservative coupling approach is more sensible. In addition, the relative performance of implicit and explicit block interface treatment will need to be thoroughly investigated for the present hybrid approach.

1.7 Review of CFD Algorithms For All Flow Speeds

One potential engine application for the present hybrid approach would be to address simultaneously the transient flows in the cylinder-valve-pipe region. In this region, there is a large discrepancy of flow speeds in various components, and compressible and incompressible flows coexist. The bulk flows in the combustion chamber are generally slow, falling into the category of incompressible flows (flow Mach number can be as low as 0.03). However, in the narrow gap at the valve/port region, the flows can sometimes reach sonic speed. In the exhaust or intake pipes a supersonic flow can be present during the initial intake valve opening or in the exhaust blow down process, large gradients in flow variable or shocks can be generated. Moreover, there can be compressible pressure waves, linear or nonlinear, dancing backwards and forwards superimposed on the bulk flows in the combustion chamber and the manifold system. Therefore, a numerical algorithm which is capable of handling both incompressible and compressible flows is needed in such simulations.
**Introduction**

However, traditionally since only one engine component is calculated, the CFD algorithm adopted to date for engine simulation is usually a dedicated compressible or incompressible scheme, which is normally not suitable for all speed flows. For example, if only the flow in the combustion chamber is to be simulated, an essentially incompressible flow CFD model would be adequate. In fact of course, the bulk (large) pressure changes mean that a pressure-density link has to be included and hence the CFD model does incorporate compressible effects. However, the Mach numbers are always low with no shocks, so these difficult aspects of compressibility need not be built into the pressure treatment within the CFD algorithms. On the other hand, a CFD algorithm well-suited to strongly compressible flow including shocks would be necessary for valve/port or manifold flow prediction.

As a matter of fact, CFD algorithms themselves traditionally have been dominated by schemes dedicated (or ‘tuned’) to high or low flow speeds. There are two main reasons contributing to this. One is that only a small flow region could be modelled in the early days due to memory capacity limitations and, within the region of focus, the range of dominant flow speeds is normally quite narrow. The other is that computation benefit can arise by developing a scheme which is suitable to the properties of a reduced set of governing equations applicable to flows where the Mach number is known to remain either high or low. For example, for incompressible flows, the fact that density variation in time and in space is not dependant on pressure allows approximations to be made. The weak coupling between temperature and pressure permits the decoupling of the energy equation from the flow-field (or strictly one-way coupling). The mathematical properties of the original equation system have changed after modification, therefore, the numerical approach to equations appropriate to high and low Mach flows has been different. Conventionally, incompressible CFD schemes are predominantly pressure-based where the primitive variables are the dependent variables (Patankar 1980, Harlow et al. 1965), whereas the compressible schemes are density-based which solve for the conserved flow quantities (MacCormack 1985).

However, there are a large number of practical flows that are of mixed compressible and incompressible nature. The engineering need to develop algorithms suitable for all flow
speeds has always been there but it was not until the early 80's that there has been growing
research activities in the development of unified algorithms solving equations which are
effective for all speeds. There are largely two trends in this development. On the one hand,
the incompressible algorithm research community investigated the extension of pressure-
based schemes to handle compressibility and shock capturing. On the other hand, the
compressible community sought methods to remove the inaccuracy and inefficiency
properties associated with density-based schemes in calculating incompressible flow.

The primary issue involved in the extension of pressure-based algorithms to high Mach
number flow is shock-capturing. A variety of methods have been proposed in the literature
and a review of the mainstream methods will be given in Chapter 2, where the unified
pressure-correction algorithm implemented in this thesis will be reported. The task
involved in extending the density-based algorithm is the improvement of its performance
at computing incompressible flows. A review of this work is presented in Chapter 3.

As explained above, a numerical algorithm for all flow speeds is needed in this work and
this will be referred to from hereon as a "unified" flow algorithm. There are two choices
for such an algorithm, i.e. either from the pressure-based family or from the density-based
family. However, there are few publications which contain comparative studies of the two
approaches. Therefore, one from each of the two families is developed in this work. The
pressure-based one is a unified SIMPLE pressure correction algorithm and the density
based model is a unified block implicit algorithm. A comparative study is carried out. The
performance of the two algorithms in computing flows covering a wide range of speeds,
steady flows and unsteady flows will be compared. Also their relative characteristics when
modified as necessary for the hybrid approach are investigated. The information extracted
from such a study is aimed at providing guidelines for their applications to engine flows.

1.8 Review of Numerical Time Marching Schemes

Since insight gained from transient analyses of gas flows can guide engine design in a
more fruitful way, it is essential for both two-stroke and four-stroke engine simulation to
employ an unsteady CFD model which can reproduce the dynamic effect of pressure
waves, etc.
Introduction

Numerical study of the detailed fluid motion of engine gas dynamics was first accomplished by Benson (Benson 1982) in the late 70's from a one dimensional CFD model using the method of characteristics. This method was especially tailored to the gas exchange process which is dominated by one-dimensional hyperbolic characteristics. However, the application of this method to multidimensional flow is not practical because of the difficulty in identifying the characteristic paths. Blair's wave motion theory (Blair 1991) is another effective one dimensional gas dynamic analysis method proposed around that period. It computes the wave motion from fundamental wave theory and hence was physically more easy to interpret. However, this method also cannot easily be extended to multidimensional situations. In the early 80's, multidimensional CFD calculations of transient engine flows using finite difference and finite volume methods began to be developed and have been the focus of interest ever since. To handle the unsteady nature of the equations, a time marching scheme is needed regardless of whether time-accurate or only steady-state solutions are sought.

Time-marching schemes are largely classified into explicit, semi-implicit and block implicit procedures. Explicit schemes are simple, straightforward and do not require the storage of intermediate results. Yet, these are usually the most time-consuming procedure for obtaining steady-state conditions due to the CFL stability constraint. For time-accurate calculations, the time step has to be chosen small enough to guarantee the accuracy of the transient solutions. One early but relatively efficient explicit scheme still often used today is that of Jameson (Jameson 1981). In this procedure, reasonably fast convergence can be achieved with the use of local time-stepping, residual smoothing or multigrid (see original reference for details). Jameson (Jameson 1991) among others have extended this technique from steady-state calculations to time-accurate calculations.

The semi-implicit time-marching method may be viewed as a sequential solution procedure in which the equations in the system are solved one by one in turn. During the solution of each equation an implicit time procedure is used but only the dependent variable for that equation is solved, which effectively decouples the equations from each other, at least during the first 'pass' through a time step. The advantage of this procedure is that a scalar matrix problem rather than a block matrix problem needs to be solved, and
Introduction

this directly influences the storage requirement as well as the complexity of the matrix inversion method. Moreover, this procedure allows a different number of equations to be accumulated within the algorithm, depending on the physics involved, without the need for reformulating the solution algorithm. Such an approach is usually more efficient than explicit methods for both steady-state and time-accurate calculations, despite the comparatively higher memory requirement. However, some form of stability condition (like a CFL condition) is still imposed in such schemes due to the weak implicit coupling among the equations.

In a fully implicit procedure the whole equation system is solved simultaneously. As a result, a large non-linear block matrix problem has to be solved at each time step. In theory, this procedure is free from stability restriction. In practice, some kind of stability condition will still exist depending on the method chosen to solve the block matrix problem. However, the restriction induced here is usually not as severe as that of the previous two procedures. Therefore, larger time steps can be allowed in this procedure and the asymptotic steady-state solution can be reached with fewer iteration steps. The memory requirement of the implicit procedure will be much larger compared with the explicit and semi-implicit methods and the CPU time of each iteration will also be greater. However, if the overall computing time of this time-marching method is less, as a result of the rapid convergence, the fully implicit procedure could still be attractive.

With regard to the choices made here for algorithms capable of handling all flow speeds (one pressure-based and one density-based algorithm), the unified pressure correction algorithm is a semi-implicit procedure, whereas the unified block implicit algorithm is a fully implicit one. Hence, in the present work, only the two different implicit time-marching methods will be compared. The performance of the two algorithms will be studied and this will also allow an investigation of the relative performance of the two implicit methods. Specifically, the two schemes will be compared against each other with respect to computing time usage, both within each time-step and over the whole calculation, and also with respect to memory requirement. From these data the cost effectiveness of the two schemes can be assessed, subject to the accuracy of the numerical solutions.
**Introduction**

In time accurate calculations, the accuracy of the transient solutions depends not only on the mesh size but also on the time step size. In this work, with full awareness of the effect of time-marching schemes on the quality of the unsteady calculations, but for the sake of simplicity, a low order time accurate scheme, i.e. a first order time scheme, is employed. However, because of the importance of the time step size on the solution accuracy, an in-depth investigation of the time step influence on the transient solution quality will be carried out before engaging in any numerical experiments on unsteady flows. The conclusions obtained from this study will guide the choice of the time step size in all of the unsteady flow calculations performed later.

1.9 Objectives of the Current Research

There are two main objectives in the research work described in this thesis.

Objective one is to study, compare and contrast two unified solution procedures which are capable of simulating both compressible and incompressible multidimensional flows. One is extended from a semi-implicit segregated pressure-correction algorithm, the other is derived from a block implicit density-based algorithm. A variety of studies of the two procedures will be carried out on a number of flow problems, ranging from incompressible to compressible, steady to unsteady, linear to nonlinear pressure wave propagation, to compare their relative performance, computing cost and efficiency. These test case studies should shed light on which one is more suitable for the IC engine flow predictions of interest here. The superior one will then be applied to compute an engine-related flow and validated by experimental measurement.

Objective two is to develop a hybrid coupling methodology which will enable the simultaneous employment of one-dimensional CFD models and two-dimensional CFD models in a single computation. Possible coupling strategies will be proposed and investigated for their capabilities to ensure the same quality of prediction as that of a uniformly 2D calculation. A general coupling methodology is sought in this work which will be verified by implementation in both of the two unified solution algorithms mentioned above. Although it is tested for just 1D/2D coupling, the methodology should be generally valid and capable of easy extension to 3D. The coupling strategy will be
integrated into the solution procedure in such a way that the solution is independent of the relative locations of the one dimensional and the two dimensional models in the computation domain. The developed hybrid coupling strategy will be validated initially in a series of one-dimensional linear and nonlinear transient pressure wave propagation problems. One coupling strategy will be chosen based on these observations and applied to an engine-related test case. This simulation will enable the validation of the developed hybrid (1D/2D) approach against experiments as well as comparing with fully 2D predictions. Furthermore, it will demonstrate the feasibility, promise and potential of the hybrid coupled approach to practical IC engine applications.

1.10 Layout of Thesis

This thesis is structured as follows:

In this first chapter, Chapter 1, an introduction of the background to the present work has been given. The engine flow phenomena that are of interest here have been briefly explained. Contemporary numerical methods for simulating this type of flow have been reviewed and appraised, followed by a proposal for an approach which is to be researched in this thesis. A very brief literature review has also been given to the multiblock method, differencing schemes and unified algorithms for flow at all speeds.

In Chapters 2 and 3 the two unified solution procedures developed and implemented in this research are laid out in sufficient detail for the reader to appreciate. In Chapter 2, the unified semi-implicit pressure correction procedure is described. The formulation of the pressure-correction equation, the discretisation of convective fluxes and the specification of the boundary conditions are the main issues discussed. Some other issues encountered such as retarded pressure and Rhie&Chow pressure smoothing etc. are also described. In Chapter 3, the unified block implicit solution procedure is described. The difficulties associated with conventional compressible methods in simulating incompressible flow and research on resolving these problems in the literature are briefly reviewed. This is followed by discussion of the development of a unified solution procedure for the present class of flow problems. Main emphasis is given to the differencing scheme for the convective fluxes, the linearisation of the nonlinear matrix system and the matrix inversion
Introduction

algorithm. A special description is given at the end of this chapter to the multigrid acceleration technique.

In Chapter 4, attention is turned to the development of a coupling strategy for use in the hybrid approach. An overview of general coupling strategies is first given. The issues involved are the grid system, data management and the flow coupling at the model interface. The kind of grids required, i.e. discontinuously patched multiblock grids, are demonstrated. An unconventional organisation of the data structure is also illustrated as well as a special index system for cell designation. The flow interface coupling method is first discussed for the situation where the grids are two dimensional in both blocks and the grids are continuously patched at the block interface. This is then extended to a general 1D and 2D hybrid coupled approach. Finally, at the end of this chapter, the most important technique which enables efficient flow information exchange between the 1D and the 2D blocks (referred to here as the 'Base Cell Concept'), is introduced. This technique is the foundation of the coupling methods developed.

In Chapter 5, the implementation of the coupling strategy in a unified pressure correction procedure is reported, including an explicit non-conservative coupling method and an implicit conservative coupling method. The interface treatment in the discretisation of all governing equations is demonstrated. In addition, a special implementation of the Rhie&Chow pressure smoothing is illustrated. A very detailed description is given to a novel modification of hybrid implicit line-solvers (e.g. SLOR and TDMA algorithms), for the realisation of implicit interface coupling.

In Chapter 6, the implementation of an implicit coupling strategy in a unified block implicit solution procedure is briefly reported. The presentation of the interface treatment is given in a general form of the governing equation. The evaluation of the artificial dissipation model at the interface is demonstrated.

In Chapter 7, computational results from comprehensive numerical benchmark tests as well as an IC engine-related flow application are presented. The presentation is organised in the following way. Firstly, predicted results of steady-state flow tests verifying the unified solution procedures are documented. The test cases chosen are inviscid
Introduction

compressible flow over a bump and viscous incompressible driven cavity flows. The CPU computing time and memory usage in all calculations are tabulated. Secondly, multigrid calculation results for the bump flows are reported to document the acceleration performance. Thirdly, a series of numerical experiments which validate both unified solution procedures in simulating unsteady flows are presented. A study of the influence of the mesh size and the time marching step on the accuracy of the prediction is first reported. Two benchmark flow problems are employed: linear propagation of a continuous sinusoidal pressure wave and nonlinear propagation of a single strong pulse disturbance. The CPU time and memory usage in these calculations are also recorded. Fourthly, the presentation moves on to the validation test cases for the coupling methodology. The same unsteady benchmark problems as described above are employed in this assessment. The results from two versions of the 1D and 2D coupled model are presented and compared with fully 2D model predictions. The efficiency and resource requirement of the 1D&2D model calculations are evaluated based on CPU time consumption and memory usage. The implicit coupling methods integrated into the pressure-correction and the block implicit procedures are compared for their relative performance, as are the explicit and implicit coupling methods integrated into the pressure correction procedure. Finally, the predictions of a IC engine related flow by the ID&2D coupled model are reported.

In the last chapter, Chapter 8, final conclusions from the research contained in this thesis are summarised and recommendations for future work are given, based on the numerical results obtained.
Introduction

1.11 References


Introduction


Brunt, M. (1994) Ford MSc Advanced Automotive Engineering, Lecture Notes, Aeronautical and Automotive Engineering Department, Loughborough University


26
Introduction


Introduction


Manners, A.P. (1994) “Computational Fluid Dynamics”, Lecture Notes, Aeronautical and Automotive Engineering Department, Loughborough University


Introduction


Steger, J.L., (1991) 'Thoughts on the Chimera Method of Simulation of Three-Dimensional Viscous Flow', Computational Fluid Dynamics and Aeropropulsion, pp. 1-10, Cleveland, OH, April 24-26


Introduction


Chapter 2

Unified Pressure Correction Algorithm
2.1 Preamble

This chapter outlines the unified pressure-correction method used in this thesis. The section below describes the extension of a traditional pressure-based incompressible methodology so that it is able to predict compressible flows.

Research activities to extend incompressible pressure-based algorithms to the compressible regime have been going on since the early 80's. In the present work, one existing suggestion to extend the SIMPLE algorithm is adopted (McGuirk et al. 1990), but formulated to apply to a non-staggered curvilinear grid and with a new discretisation approach for the convective flux. To make this chapter free standing, some background work is reviewed first, then the extended pressure-correction procedure is documented in detail.

2.2 Review of Pressure-Correction Schemes

In low speed flow with no heating sources, density variations are negligible, so the mass conservation equation becomes a constraint equation for the pressure. Although, its role may be viewed as providing the equation for pressure, it actually only provides an indirect governing equation for pressure whereby a suitable pressure field will yield, upon its use in the momentum equations, a velocity field which has no mass imbalance throughout the solution domain.

Pressure-correction algorithms transform the continuity equation into an equation for the correction to a guessed pressure field. In the SIMPLE algorithm (Patankar & Spalding 1972), the pressure field is initially guessed, preliminary velocity fields are generated by solving the momentum equations, then corrections to the guessed pressure are calculated from a pressure-correction equation based on the mass imbalance error associated with the preliminary velocity field. Subsequently the preliminary pressure and velocity fields are corrected. This procedure is iteratively repeated until simultaneous satisfaction of continuity and momentum equations is obtained. Any other flow variables, such as temperature, are updated at the end of each iteration through an energy equation.
Unified Pressure Correction Algorithm

The pioneering work on the SIMPLE procedure is usually attributed to Patankar and Spalding (Patankar et al. 1972, Patankar 1980). Since then, a number of variants have been developed which have improved various aspects of the SIMPLE scheme. Originally, the SIMPLE scheme was derived as applicable only to Cartesian or Polar coordinate systems and using a staggered velocity-pressure grid. Rhie and Chow (Rhie et al. 1983) and Shyy et al. (Shyy et al. 1988) are among the early researchers who generalised this methodology to curvilinear coordinates and a non-staggered co-located grid arrangement. Such co-located grids simplify the prescription of boundary conditions for complex geometry as well as ensuring the extension to a multiblock technique.

First attempts to extend the SIMPLE procedure to include compressibility started in the early 80's (Patankar 1980). The goal was to obtain a pressure-correction algorithm which was comparable in performance to conventional compressible methods in simulating high speed flows (including shock capturing) but also retain the original efficiency of the algorithm in solving low speed flows. The extension involves the employment of compressible Navier-Stokes equations (i.e. including density variation), the derivation of a modified pressure-correction equation, the inclusion of the energy equation and the coupling of pressure with density via a gas law. There have largely been two approaches of this type. One is referred to here as the mass flux linearisation method, the other is characterised as the mass flux dependent variable method. The former was originally proposed by Issa and Lockwood (Issa et al. 1977), while the latter was suggested by McGuirk and Page (McGuirk et al. 1990). The two approaches differed in how to relate the mass flux correction to the pressure correction (a fundamental step in the pressure-correction equation derivation) and how to ensure the hyperbolic characteristic of the pressure-correction equation at supersonic speeds.

In the mass flux linearisation method (also followed by Rhie 1986, Karki, et al. 1989, Van Doormaal, et al. 1987), pressure is assumed to influence (simultaneously) the velocity through the momentum equation and the density through the equation of state, in such a way that the final corrected velocity and density when multiplied together will produce mass fluxes which satisfy the mass conservation equation. The mass flux correction (e.g. \((pu)'\)) in the continuity equation is first written in terms of separate corrections to both
Unified Pressure Correction Algorithm

density ($p'$) and velocity ($u'$) by linearisation, and then indirectly related to the pressure-correction via momentum and gas-law relations. The pressure-correction equation so obtained is a mixed type convection and diffusion equation rather than the pure diffusion type as found in the incompressible scheme. It was demonstrated by Shyy and Chen (Shyy, et al. 1992) that in this equation, the ratio of the convection contribution to the diffusion contribution is proportional to the square of the Mach number. That is to say, at lower Mach numbers, the diffusion terms are dominate, and the equation exhibits elliptic nature. As Mach number increases, convection effects become dominant and the equation displays a hyperbolic nature. Hence, the pressure-correction equation in this approach has the correct properties at all flow speeds. Early practice approximated cell face density by a linear average which is inappropriate for hyperbolic flows, because it does not correctly represent the physical zone of influence if the cell face Mach number is greater than unity. Later, first order upwind interpolation was adopted which guaranteed the hyperbolic property of the discrete equation for supersonic flows. However, with this procedure, the discontinuous solutions such as shock waves were severely smeared. This has prompted research activities in the literature in search for better shock-capturing techniques for a unified pressure-correction algorithm.

In the mass flux dependent variable method, the conserved flow quantities are chosen as the dependent variables for the momentum equations rather than the primitive variables (velocity), these conserved quantities may also be viewed as the mass flux components. Consequently, the correction of the mass flux appearing in the continuity equation can be related directly to the pressure correction purely from the momentum equation without needing any linearisation. Density corrections arising from the unsteady term in the continuity equation are linked to pressure corrections via the gas law. The pressure-correction equation therefore assumes a similar form to that in the incompressible scheme, it is a Poisson type at all flow speeds if the time term in the continuity equation is neglected. Such an elliptic nature is incorrect for supersonic flows. Remedies to this proposed so far have been the retarded pressure or retarded density methods (McGuirk, et al. 1990, Lai, et al. 1992, Lien, et al. 1993), and details of this are available in the quoted references.
Unified Pressure Correction Algorithm

In either of the above methods, in the limit of incompressible flows where density variations in time and space disappear, the algorithms degenerate naturally to the original incompressible pressure-correction algorithm and the algorithm efficiency at low speeds is hence unchanged. The second (mass-flux dependent method) (McGuirk et al. 1990) approach has been selected for assessment of the unified pressure-correction algorithm in this thesis. Because the mass flux can be linked directly to pressure in the mass flux dependent variable method, it has great benefit for flow problems with shocks due to the fact that the mass flux is one of the conserved variables across the shock whereas primitive variables (velocities) are discontinuous; since the engine flow application of interest in this work is a transonic flow, this feature is advantageous.

2.3 Review of Differencing Schemes Used in Pressure-Correction Algorithms

As well as the basic approach taken to the formation of the pressure-correction equation, the overall performance of a CFD algorithm is crucially determined by the decisions made for discretising the convective (non-linear) terms in the transport equations. Various decisions made in the literature on pressure-correction schemes are briefly reviewed here. Incompressible pressure-correction schemes have often used simple differencing methods in the discretisation of the convective flux, e.g. Hybrid differencing (Patankar 1980) or QUICK differencing (Leonard 1979). In the early development of unified pressure-correction schemes, efforts were primarily focused on the appropriate formulation and discretisation of the pressure-correction equation, and traditional differencing methods were adopted. However, the hybrid method smears shocks severely due to its first order nature, whereas the QUICK method causes oscillations around shocks due to its unbounded behaviour. There are a large number of differencing techniques, however, which have been developed in conventional compressible algorithms (see Chapter 3) which lead to good shock capturing. Therefore, in the last decade or so some of these ideas have been investigated within unified pressure-correction algorithms.

Higher order accurate convection schemes in compressible CFD methods are usually characteristic oriented and constructed following either a MUSCL approach (VanLeer 1979) or a non-MUSCL approach (Yee 1987). First order schemes are monotonic, hence
Unified Pressure Correction Algorithm

bounded and oscillation-free. However, they suffer the same excessive smoothing as classic first order upwind methods used in incompressible algorithms. Higher order schemes are not monotonic, and are hence susceptible to the generation of oscillations in the vicinity of large flow gradients. A variety of techniques have been developed to control the non-monotonic property of the higher order schemes. A popular approach is the TVD technique (Hirsch 1988). This technique exerts a constraint (limiter) to ensure that the total variation of the dependent variable always diminishes so that the differencing method is monotone preserving. As a result, no new extrema are allowed to be generated, and hence the source of oscillation is eliminated. A non-linear TVD limiter (Yee 1989, Harten 1983, 1984, 1991) is usually adopted due to its high sensitivity and selectiveness to oscillations which is good for high resolution of gradients (including shocks). The limiter ensures that just sufficient smoothing (dissipation) is added locally to eliminate the over-shoots or under-shoots in the solution.

Kobayashi & Pereira (Kobayashi, et al. 1992) have implemented a higher-order upwind TVD differencing scheme within a unified pressure-correction algorithm. They compared this scheme with three other schemes (central, quadratic upwind and linear upwind) for shock capturing performance. For a supersonic bump flow problem, the TVD scheme produced the best shock resolution. However, in a similar work by Zhou (Zhou, et al. 1995), for flow around an aerofoil (RAE 2822), the linear upwind scheme was reported to give overall better prediction than the TVD MUSCL scheme. In the work of Lai (Lai, et al. 1992), the shock capturing performance of a unified pressure-correction algorithm was compared with that from a conventional compressible algorithm. The same TVD differencing method was employed in both solution procedures. Computational results for two aerofoil flows (NACA0012 and RAE 2822) revealed that the two solution algorithms were comparable. The shock produced by the pressure-correction algorithm was as sharp as that from the compressible algorithm. This work demonstrated that higher order TVD upwind schemes integrated into a pressure-correction formulation can lead to effective shock resolution.

More recently, Kobayashi & Pereira (Kobayashi, et al. 1996) were the first to incorporate a flux-differencing method of the Godonov type (Hirsch 1988) into a unified pressure-
Unified Pressure Correction Algorithm

correction algorithm. However, the role of the pressure-correction procedure was now interpreted in a very different way. They utilized a different set of dependent variables as well as Riemann solvers in both compressible and incompressible flow regimes, so that the scheme was effective and economical across the whole flow speed spectrum. A segregated pressure-correction procedure was employed in their work as a mechanism to linearise the nonlinear equations resulting from the flux differencing discretisation. Because of this, the flux differencing scheme was effectively applied in the same manner as in a density-based scheme, and hence the shock capture capability of flux differencing schemes was retained. Comparing with the coupled solution procedure used in a density-based scheme, the use of the segregated pressure-correction procedure is a good compromise reconciling CPU time and memory resource requirement. However, compared with the previously described unified pressure-correction schemes, the incorporation of a complicated flux differencing discretisation method increased the cost drastically.

Kobayashi & Pereira (1992) have carried out a comparison study of the accuracy of different convectional discretisation schemes used in a pressure-correction procedure. Their studies show that the Minmod method - a TVD method - is the best among all four schemes investigated. Also the TVD method has been proven as a powerful scheme in shock capturing in density-based procedure. Hence, it was decided to employ a TVD-limiter higher order approach within the pressure-correction procedure adopted in the present work to ensure good shock-capturing.

2.4 Governing Equations

The physical conservation laws for mass, momentum and energy per unit volume in the absence of body force and heat sources can be described by the following set of partial differential equations (using cartesian tensor notation for coordinates and variables):

Conservation of mass

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0 \] 2.1
Unified Pressure Correction Algorithm

Conservation of Momentum

\[
\frac{\partial p u_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j) = \frac{\partial p_i}{\partial x_i} + \frac{\partial}{\partial x_j}(\tau_{ij}) \tag{2.2}
\]

Conservation of Total Energy

\[
\frac{\partial p E}{\partial t} + \frac{\partial}{\partial x_i}(\rho H u_i) = \frac{\partial}{\partial x_i}(u_j \tau_{ij}) - \frac{\partial q_i}{\partial x_i} \tag{2.3}
\]

where the specific total energy \( E \) and specific total enthalpy \( H \) are defined by:

\[
E = e + \frac{1}{2} u_i u_i \tag{2.4}
\]

\[
H = E + p/\rho \tag{2.5}
\]

with \( e \) denoting the specific internal energy.

The heat flux according to Fourier's law is given by:

\[
q_i = -k \frac{\partial T}{\partial x_i} \tag{2.6}
\]

where \( k \) denotes the thermal conductivity and \( T \) is the fluid temperature (defined via \( e=C_v T \)).

For a Newtonian fluid, the stress tensor can be written as:
**Unified Pressure Correction Algorithm**

\[
\tau_{ij} = \mu \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij}
\]

where \( \mu \) and \( \lambda \) are the first and second coefficients of viscosity; according to Stokes hypothesis, the bulk coefficient of viscosity should be identically zero, thus \( \lambda \) is related to \( \mu \) via:

\[
\lambda = \frac{2}{3} \mu
\]

and \( \mu \) is related to temperature via Sutherland's Model:

\[
\frac{\mu}{\mu_{ref}} = \left( \frac{T}{T_{ref}} \right)^{3.5} \frac{T_{ref} + C}{T + C}
\]

These equations contain all the physics of any single phase, single species fluid flow and they are, therefore, the governing equations for both compressible and incompressible flows. This equation system represents a set of five PDEs for six unknowns in three dimensions and four PDEs for five unknowns in two dimensions. The unknowns are density \( (\rho) \), velocity components \( (u_i) \), pressure \( (p) \) and total energy \( (E) \). Obviously one complementary equation is needed to close the system, i.e. an equation of state. Since the fluid of interest in this work is air under normal pressure and temperature conditions, the perfect gas law is added to the above equation system

\[
p = \rho RT
\]

where \( R \) is the gas constant.

Theoretically, the solutions to the above equation system for given initial and boundary conditions will provide all the information of a flow evolution. There exists, however, in general no mathematical techniques which can resolve this equation system exactly, and they must be solved by numerical approximation. In addition, although the above equations are valid at all Reynolds values \( (Re=\rho UL/\mu) \), at high values of Re flows become
Unified Pressure Correction Algorithm

turbulent and even numerical solution of the equation set becomes an impossible task for current computational resources. This requires the use of an averaging procedure (Reynolds-time-average) and the inclusion of a turbulence model to close the system of equations. However, in the present thesis, emphasis will be placed on numerical methods and the problems to be addressed may be viewed as independent from the turbulent nature of the flow, hence in the rest of this thesis, only the laminar flow version of the equations will be considered.

2.5 Unified Pressure-Correction Algorithm

The unified pressure-correction procedure is described here for a non-staggered curvilinear grid system, and the energy equation is also solved for transient compressible flow simulation.

In a general two-dimensional curvilinear (non-orthogonal) coordinate system ($\xi$, $\eta$), equations (2.1) to (2.3) may be transformed to (n.b. the relationship between $\xi$, $\eta$ and $x$, $y$ is defined as a one-to-one mapping via a grid generation procedure):

Continuity

$$\frac{1}{J} \cdot \frac{\partial p}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \rho U \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \rho V \right) = 0$$

2.11

u Momentum

$$\frac{1}{J} \cdot \frac{\partial (\rho u)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \rho u U \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \rho u V \right)$$

$$= -\left( \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \xi_x p \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \eta_x p \right) \right) +$$

$$\frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot (\xi_x \tau_{xx} + \xi_y \tau_{xy}) \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot (\eta_x \tau_{xx} + \eta_y \tau_{xy}) \right)$$

2.12

v Momentum


Unified Pressure Correction Algorithm

\[ \frac{1}{J} \cdot \frac{\partial (\rho v)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \rho v U \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \rho v V \right) \]

\[ = - \left( \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \xi_y p \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \eta_x p \right) \right) + \]

\[ \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot (\xi_x \tau_{xy} + \xi_y \tau_{yy}) \right) \]

\[ + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot (\eta_x \tau_{xy} + \eta_y \tau_{yy}) \right) \]

Energy equation

\[ \frac{1}{J} \cdot \frac{\partial (\rho H - p)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \rho H \cdot U \right) + \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \rho H \cdot V \right) \]

\[ = \frac{\partial}{\partial \xi} \left[ \frac{K}{J} \cdot (\xi_x T_x + \xi_y T_y) \right] \]

\[ + \frac{\partial}{\partial \eta} \left[ \frac{K}{J} \cdot (\eta_x T_x + \eta_y T_y) \right] + \]

\[ \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \cdot (\xi_x \cdot (u \tau_{xx} + v \tau_{xy}) + \xi_y (u \tau_{xy} + v \tau_{yy})) \right] + \]

\[ \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \cdot (\eta_x \cdot (u \tau_{xx} + v \tau_{xy}) + \eta_y (u \tau_{xy} + v \tau_{yy})) \right] \]

where \( J \) is the Jacobian of the transformation (related to the value of an infinitesimal control volume). Note that the momentum equations are still written for the cartesian velocity components \( u \) and \( v \). The contravariant velocity components are defined by

\[ U = \xi_x u + \xi_y v \]

\[ V = \eta_x u + \eta_y v \]

and the shear stresses in the curvilinear system become:

\[ \tau_{xx} = \frac{2}{3} \mu \cdot [2(\xi_x u_\xi + \eta_x u_\eta) - (\xi_y v_\xi + \eta_y v_\eta)] \]

\[ \tau_{yy} = \frac{2}{3} \mu \cdot [2(\xi_x u_\xi + \eta_x u_\eta) - (\xi_y v_\eta + \eta_x u_\eta)] \]
Unified Pressure Correction Algorithm

\[
\tau_{xy} = \mu \cdot [(\xi_y u_\xi + \eta_y u_\eta) + (\xi_x v_\xi + \eta_x v_\eta)]
\]

2.19

temperature gradients become:

\[
T_x = \xi_x \frac{\partial T}{\partial \xi} + \eta_x \frac{\partial T}{\partial \eta}
\]

2.20

\[
T_y = \xi_y \frac{\partial T}{\partial \xi} + \eta_y \frac{\partial T}{\partial \eta}
\]

2.21

In addition, the gas law gives:

\[
T = \frac{1}{C_p} \cdot H - \frac{1}{2} \cdot \frac{u^2 + v^2}{C_p}
\]

2.22

2.5.1 Formulation of the Pressure-Correction Equation

The conserved flow quantities in the momentum equations have been chosen as the dependent variables, i.e. the momentum components per unit volume (or mass fluxes) - \(pu\) and \(pv\). The pressure-correction equation is now formulated from the discretised continuity and momentum equations.

Using 1st order Euler implicit scheme the discretised u-momentum equation is as follows:

\[
\frac{1}{\Delta t} \cdot \frac{1}{j} \cdot [(pu)^{n+1} - (pu)^n] + \left[ \frac{\partial}{\partial \xi} \left( \frac{1}{j} \cdot (pu)^{n+1} \cdot U^{n+1} \right) \right] + \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{j} \cdot (pu)^{n+1} \cdot V^{n+1} \right) \right] = - \left[ \frac{\partial}{\partial \xi} \left( \frac{1}{j} \cdot (\xi_x p^{n+1}) \right) \right] - \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{j} \cdot (\eta_x p^{n+1}) \right) \right] + \left[ \frac{\partial}{\partial \xi} \left( \frac{1}{j} \cdot (\xi_y \tau_{xx} + \xi_x \tau_{xy}) \right) \right]
\]

2.23

\[+ \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{j} \cdot (\eta_x \tau_{xx} + \eta_y \tau_{xy}) \right) \right]
\]

where the shear stress is discretised as:
Unified Pressure Correction Algorithm

\[ \tau_{xx} = \frac{2}{3} \mu \left( 2 \left\{ \xi_x \cdot \left[ \frac{\partial}{\partial \xi} \right] ((\rho u)^{n+1}/\rho^{n+1}) + \eta_x \cdot \left[ \frac{\partial}{\partial \eta} \right] ((\rho u)^{n+1}/\rho^{n+1}) \right\} + \left\{ \xi_y \cdot \left[ \frac{\partial}{\partial \xi} \right] ((\rho v)^{n+1}/\rho^{n+1}) + \eta_y \cdot \left[ \frac{\partial}{\partial \eta} \right] ((\rho v)^{n+1}/\rho^{n+1}) \right\} \right) \]

2.24

similarly for \( \tau_{xy} \) and \( \tau_{yy} \)

Note that \( \left[ \frac{\partial}{\partial \xi} \right] \) and \( \left[ \frac{\partial}{\partial \eta} \right] \) are finite difference operators for \( \frac{\partial}{\partial \xi} \) and \( \frac{\partial}{\partial \eta} \) respectively.

During the iterative process at new time level \( n+1 \) in the pressure-correction procedure, which will be discussed in detail later, only \( \rho u \) and \( p \) in the above discretised u-momentum equation are treated implicitly. As a result, this equation for all grid points at iteration level \( k+1 \) within time step \( n \) to \( n+1 \) can be written in the following matrix form:

\[ ([A_p^\alpha u] + [A_{OD}^\alpha u])(\rho u)^{k+1} = -\left[ \left[ \frac{\partial}{\partial \xi} \right] \left( \frac{1}{J} \cdot \xi_x p \right)^{k+1} \right] - \left[ \left[ \frac{\partial}{\partial \eta} \right] \left( \frac{1}{J} \cdot \eta_x p \right)^{k+1} \right] + (S_{pu}) \]

2.25

Similarly, the matrix form of discretised v-momentum equation can be written as:

\[ ([A_p^\alpha v] + [A_{OD}^\alpha v])(\rho v)^{k+1} = \left( -\left[ \left[ \frac{\partial}{\partial \xi} \right] \left( \frac{1}{J} \cdot \xi_y p \right)^{k+1} \right] - \left[ \left[ \frac{\partial}{\partial \eta} \right] \left( \frac{1}{J} \cdot \eta_y p \right)^{k+1} \right] + (S_{pv}) \]

2.26

where superscript \( k+1 \) denotes the current iteration level in the pressure-correction procedure (note that superscript \( n+1 \) as that in equation (2.23) is dropped in the above two equations for simplicity). In addition, \( () \) are column vectors of \( m \) control volumes, \([ ]\) are sparse \((m \times m)\) matrices. \([A]\) is the coefficient matrix for the implicit terms, \([A_p]\) and \([A_{OD}]\) are the matrices for diagonal and off-diagonal entries respectively. \((S)\) is the source column vector for the explicit terms but excluding pressure gradient terms.

Similarly, the discrete continuity equation can be written in the following matrix form:

\[ \frac{1}{J} \cdot \frac{1}{\Delta t} \{ \{ (\rho)^{k+1} - (\rho)^n \} + \left[ \left[ \frac{\partial}{\partial \xi} \right] \left( \frac{1}{J} \cdot \rho U \right)^{k+1} \right] + \left[ \left[ \frac{\partial}{\partial \eta} \right] \left( \frac{1}{J} \cdot \rho V \right)^{k+1} \right] = 0 \]

2.27

and the energy equation:
 Unified Pressure Correction Algorithm

\[
([A^H_p] + [A^H_{OD}]) (H)^{n+1,j+1} = \frac{1}{J} \cdot \frac{1}{\Delta t} [I] \left( p^{n+1,j+1} - p^n \right) + (S_H)
\]

2.28

At the predictor stage, preliminary solutions for the momentum components, \((\bar{\rho} u)\) and \((\bar{\rho} v)\), are obtained by solving momentum equations (2.25) and (2.26) based on a guessed pressure field \((\bar{p})\) (usually \(p^n\) to start with); a preliminary density solution is obtained from the perfect gas law

\[
(\bar{\rho}) = \frac{\bar{p}}{R(T)}
\]

2.29

where \((\bar{T})\), the guessed temperature, uses \(T^n\), the temperature field from the last time-step \(n\), to start with. If these preliminary solutions do not satisfy the continuity equation (2.27), i.e.

\[
\frac{1}{J} \cdot \frac{1}{\Delta t} \int \left\{ (\bar{p}) - (\rho)^n \right\} + \left[ \frac{\partial}{\partial x} \left( \frac{1}{J} \cdot \bar{\rho} \bar{U} \right) \right] + \left[ \frac{\partial}{\partial y} \left( \frac{1}{J} \cdot \bar{\rho} \bar{V} \right) \right] \neq 0
\]

2.30

the solution procedure will proceed to the corrector stage.

At the corrector stage, the preliminary solutions as well as the guessed pressure field are to be corrected so that the mass conservation condition is satisfied, i.e.

\[
\frac{1}{J} \cdot \frac{1}{\Delta t} \int \left\{ (\bar{p} + \rho') - (\rho)^n \right\} + \left[ \frac{\partial}{\partial x} \left( \frac{1}{J} \cdot (\bar{\rho} \bar{U} + (\rho U)') \right) \right] + \left[ \frac{\partial}{\partial y} \left( \frac{1}{J} \cdot (\bar{\rho} \bar{V} + (\rho V)') \right) \right] = 0
\]

2.31

where the variable with a prime denotes the correction to the preliminary solution, and

\[
(\rho U)' = (\rho u)' \cdot \xi_x + (\rho v)' \cdot \xi_y
\]

2.32

\[
(\rho V)' = (\rho u)' \cdot \eta_x + (\rho v)' \cdot \eta_y
\]

2.33
Unified Pressure Correction Algorithm

The relation between the correction to the pressure and the correction to the mass flux can be obtained by comparing the momentum equation governing the preliminary solutions and that governing the corrected solutions. This relation is:

\[
([A^\mu_p] + [A^\mu_{OP}]) (\rho u)' = -\left[ \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \xi \cdot p' \right) \right] - \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \eta \cdot p' \right) \right]
\]

2.34

The second term on the left hand side of (2.34) is considered small compared with the first term (SIMPLE approximation) and hence dropped. This leads to

\[
(\rho u)' = [A^\mu_p]^{-1} \left\{ -\left[ \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \xi \cdot p' \right) \right] - \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \eta \cdot p' \right) \right] \right\}
\]

2.35

which involves only the inversion of a diagonal matrix \([A^\mu_p]\).

Similarly:

\[
(\rho v)' = [A^\mu_y]^{-1} \left\{ -\left[ \frac{\partial}{\partial \xi} \left( \frac{1}{J} \cdot \xi \cdot p' \right) \right] - \left[ \frac{\partial}{\partial \eta} \left( \frac{1}{J} \cdot \eta \cdot p' \right) \right] \right\}
\]

2.36

The correction to the density is related to pressure correction via the perfect gas law

\[
\rho' = \frac{p'}{RT}
\]

2.37

Rearranging equation (2.30) so that only the terms involving correction variables are collected on the left hand side:

\[
\frac{1}{J} \cdot \frac{1}{\Delta t} \{ I \{ \rho' \} \} + \left[ \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \cdot (p' U') \right] \right] + \left[ \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \cdot (p' V') \right] \right] = \]

2.38

\[
\frac{1}{J} \cdot \frac{1}{\Delta t} \{ \bar{\rho} - (\rho)^n \} - \left[ \frac{\partial}{\partial \xi} \left[ \frac{1}{J} \cdot \bar{\rho} U \right] \right] - \left[ \frac{\partial}{\partial \eta} \left[ \frac{1}{J} \cdot \bar{\rho} V \right] \right]
\]

then, upon substitution of equations (2.35), (2.36) and (2.37) into equation (2.38), an equation for pressure-correction is formed:

45
Unified Pressure Correction Algorithm

\[
\frac{1}{\Delta t \cdot R T^n} \Pi(p') + \left[ \frac{\partial}{\partial \xi} \right] (C) \left[ \frac{\partial}{\partial \eta} \right] (p') + \left[ \frac{\partial}{\partial \eta} \right] (D) \left[ \frac{\partial}{\partial \eta} \right] (p') = \text{massrc}
\]

where C and D represent the coefficients, in an orthogonal grid system they are

\[
C = -\left( \frac{1}{J} \right)^2 \left[ \frac{\xi^2}{A_p^{pa}} + \frac{\xi^2}{A_p^{pv}} \right]
\]

and

\[
D = -\left( \frac{1}{J} \right)^2 \left[ \frac{\eta^2}{A_p^{pa}} + \frac{\eta^2}{A_p^{pv}} \right]
\]

'\text{massrc}' which represents the right hand of equation (2.39) is the same as the right hand side of equation (2.38). Equation (2.39) is solved to generate corrections to the guessed pressure. Corrections to the preliminary mass flux and density solutions are subsequently derived from equations (2.35), (2.36) and (2.37). Then other flow variables can be corrected. The energy equation is solved thereafter using the newly corrected variable values and the temperature is updated. By now, the iterative procedure is ready to proceed to the new iteration level (i.e. k+2). The newly obtained pressure field (i.e. at k+1) is used as the guessed value in the new iteration. The whole process of prediction and correction is repeated until a converged solution is obtained for the current time level n+1. The time marching scheme is then moved to the next time level (n+2) and the pressure-correction iterative process is repeated to obtain the solution for the next time level.

2.5.2 Retarded Pressure

There is a fundamental problem with the pressure-correction equation (2.39) as described above and pointed out also by McGuirk & Page (McGuirk and Page 1990, Page 1990), which is that it remains an elliptic equation in supersonic flows. This poor representation of a physically hyperbolic flow may lead to unphysical numerical solutions or instability, hence alterations of this equation are necessary. McGuirk & Page proposed a method which substitutes the real pressure in the momentum and pressure-correction equations by
Unified Pressure Correction Algorithm

a retarded pressure so that the pressure-correction equation changes to a hyperbolic equation in a supersonic flow. This method is briefly explained here.

The definition of the retarded pressure in a two dimensional curvilinear coordinate system is:

\[
\dot{p} = p + \mu_\zeta \frac{\partial p}{\partial \zeta} + \mu_\eta \frac{\partial p}{\partial \eta} \tag{2.40}
\]

where \( \mu_\zeta \) and \( \mu_\eta \) are functions to control the strength of upstream influence on the retarded pressure. When equation (2.40) is transformed to discretised form, it becomes

\[
\hat{p}_{ij} = p_{ij} + \mu_\zeta \vec{\Delta}_\zeta p + \mu_\eta \vec{\Delta}_\eta p \tag{2.41}
\]

where \( \vec{\Delta}_\zeta p \) and \( \vec{\Delta}_\eta p \) are the upwind difference approximations of the first derivatives \( \frac{\partial p}{\partial \zeta} \) and \( \frac{\partial p}{\partial \eta} \) respectively, the upstream pressure is thus introduced into the evaluation of the pressure used in the correction equation. By using the retarded pressure, when the local flow speed becomes supersonic, the upstream pressure is introduced to the numerical equations. This is a more correct representation of the physics of the flow where a pressure disturbance propagates only in a one-way direction when flow Mach number is greater than one.

This approach has been analysed by Page (Page 1990) who demonstrates that the replacement of the real pressure by the retarded pressure is effectively adding a second derivative of pressure to the original equation. Hence, the role of the retarded pressure can also be interpreted as introducing artificial dissipation to smooth out spurious oscillations resulting from the unphysical representation of the hyperbolic flow by an elliptic pressure-correction equation.

The function \( \mu \) triggers the use of the retarded pressure only as local Mach number becomes greater than a reference value \( M_{\text{ref}} \). Moreover, it adjusts the amount of numerical dissipation added through the retarded pressure so that the shock solution is not smeared out. The formulae for this function are:
Unified Pressure Correction Algorithm

\[ \mu_x = \max\{0, k_1[1 - (M_{ref1}/M_{i-1/2})^2], k_2[1 - (M_{ref2}/M_{i-1/2})^2]\} \quad \text{2.42} \]

\[ \mu_n = \max\{0, k_1[1 - (M_{ref1}/M_{j-1/2})^2], k_2[1 - (M_{ref2}/M_{j-1/2})^2]\} \quad \text{2.43} \]

where \( k_1 = 1 \) and \( M_{ref1} = 1, k_2 = 0.7 \) and \( M_{ref2} = 0.7 \). These are similar to the optimised relations given by Page based on numerical experiments and observations.

2.6 Discretisation Scheme

All partial differential equations in this work are discretised within a finite volume framework. As a result, flow variables and gradients on control volume faces are required to be evaluated. This section describes the method used to achieve this in particular for the flow variable itself (convective terms). The gradient (diffusive terms) is always evaluated using a linear approximation between the cell centre values either side of the face.

2.6.1 Review of Differencing Schemes for Convective Flux

The convective terms in the governing equations are influenced by the directional characteristics of the flow, they are therefore determined by the parabolic/hyperbolic nature of the equation system. The physics contained in the convective flux have been studied extensively in the literature in order to guide the choice of an appropriate numerical method, primarily from Euler flows where all diffusive mechanism is absent, leaving the convective flux as the sole mechanism for flow evolution. This flux then transports flow disturbances and flow information in a manner akin to signal propagation, i.e. in certain directions (characteristic directions) and within confined regions (zone of influence). Due to the directional nature of the convective flux, it is essential that discretisation methods take into account the characteristic directions and zone of influence such that the difference approximation will share the same characteristics determined by the flow physics.

Mathematical analysis (Hirsch 1988) has shown that there are four characteristic directions in one dimensional Euler flows, and different flow information is carried along each of them. The flow state anywhere in the flow field is completely determined by the
Unified Pressure Correction Algorithm

information carried on the four characteristics intersecting at that point. The signals on two of the four characteristics propagate at the flow speed and in the same direction as the flow. Signals on the other two characteristics are acoustic signals superimposed on the flow. One propagates in the direction of the flow (right running wave) and the other against (left running wave). The effective direction of the left running acoustic signal is determined by the difference between the flow speed and the speed of sound. In a subsonic flow, the left running acoustic wave moves against the flow, whereas in supersonic flow, this acoustic wave moves in the same direction as the right running acoustic wave. Design of the differencing scheme for the convective flux is primarily based on the directions of these characteristics. An upwind discretisation scheme for example includes only cells in the local zone of dependence to estimate the convective flux.

There are a few classical upwind techniques (Hirsch 1988), the particular one adopted in this work belongs to the technique called flux vector splitting. In the flux splitting technique the convective flux in each co-ordinate direction is analysed separately, i.e. a quasi-one dimensional approach is adopted. Each of these one-dimensional convective fluxes is broken down into two components, containing the flow influence carried by downstream and upstream propagating characteristic waves respectively. Because the direction of the characteristics is determined by the sign of the corresponding eigenvalue in the Jacobian of that flux, all the eigenvalues of the above flux components are either all positive or all negative, and are the same as the original ones. Instead of the flux itself on a cell face the flux components are approximated individually by upwind methods. The physical information characterised by the zone of influence and carried by different signals is then properly reconstructed in the differencing approximation of the convective flux.

2.6.2 Second Order MUSCL TVD Method for Convective Flux

The flux vector splitting technique employed is a second order MUSCL (Monotone Upstream-centred Schemes for Conservation Laws) (Van Leer 1979) scheme because of its proven capability for sharp shock capturing (Hirsch 1988). The MUSCL approach is essentially a method to construct a high order variable variation over a one-dimensional
Unified Pressure Correction Algorithm

control volume. A detailed explanation can be found in the book of Hirsch (1988), here only a brief description is given of the particular MUSCL method employed.

The MUSCL re-construction expresses the variable within a cell in a general form as

\[ \phi(x) = \phi_i + \frac{1}{\Delta x}(x - x_i)\delta_1 \phi + \frac{3k}{2\Delta x^2}(x - x_i)^2 - \frac{\Delta x^2}{12}\delta_2 \phi \quad \text{for} \quad x_{i-1/2} < x < x_{i+1/2} \quad (2.44) \]

where \( \phi_i \) is the average cell value within cell \( i \) as shown in Figure 2.1 (usually the cell centre value); \( \delta_1 \phi, \delta_1^2 \phi \) are the difference approximations of the first and second derivatives within cell \( i \); \( k \) is a parameter; \( \Delta x \) is the cell size, i.e. \( \Delta x = x_{i+1/2} - x_{i-1/2} \).

So the first term on the right hand side of equation (2.44) is the average value within cell \( i \); the second term represents the linear variation over a cell as shown in Figure 2.1; the third term is the truncation error for the linear variation approximation. Different choices of \( k \) produce different second order differencing schemes.

FIGURE 2.1 Variable Variations Over One Dimensional Cells

If \( \delta_1 \phi, \delta_1^2 \phi \) are the central approximations:

\[ \delta_1 \phi = \frac{1}{2} (\phi_{i+1} - \phi_{i-1}) \quad (2.45) \]
Unified Pressure Correction Algorithm

\[ \delta_i^2 \phi = \phi_{i+1} - 2\phi_i + \phi_{i-1} \quad 2.46 \]

the variable value at cell face \( x_{i+1/2} = x_i + \Delta x/2 \) within cell \( i \) is

\[ \phi^L_{i+1/2} = \phi_i + \frac{1}{4}[(1-k)(\phi_i - \phi_{i-1}) + (1+k)(\phi_{i+1} - \phi_i)] \quad 2.47 \]

and at \( x_{i+1/2} = x_{i+1} - \Delta x/2 \) within cell \( i+1 \) is

\[ \phi^R_{i+1/2} = \phi_{i+1} - \frac{1}{4}[(1+k)(\phi_{i+1} - \phi_i) + (1-k)(\phi_{i+2} - \phi_{i+1})] \quad 2.48 \]

The cell face values can be considered as resulting from a combination of forward and backward extrapolations.

In the present work, flows with pressure wave propagation as well as shock flows are applications of interest, therefore, an upwind discretisation scheme is more appropriate. As a result, \( k = -1 \) is chosen which leads to a 2nd order one-sided upwind scheme which corresponds to a linear one-sided extrapolation at the cell face between the averaged values at the two upstream cells as shown in Figure 2.2

![Figure 2.2 Second Order One-Sided Upwind Variable Interpolation](image-url)
Unified Pressure Correction Algorithm

Now:

$$\phi^L_{i+\frac{1}{2}} = \phi_i + \frac{1}{2}(\phi_i - \phi_{i-1})$$  \hspace{1cm} (2.49)

$$\phi^R_{i+\frac{1}{2}} = \phi_{i+1} - \frac{1}{2}(\phi_{i+2} - \phi_{i+1})$$  \hspace{1cm} (2.50)

Superscript L denotes the situation where the characteristics (i.e. acoustic signal) runs from left to right, while R denotes the other way round.

The cell face variable values are then used to evaluate the numerical flux at the face as shown in Figure 2.2. The numerical flux constructed via the flux splitting method is generally expressed as:

$$f^*_{i+\frac{1}{2}} = f^-_{i+\frac{1}{2}} + f^+_{i+\frac{1}{2}}$$  \hspace{1cm} (2.51)

where $f^*$ denotes the convective flux in the governing equation, $f^-$ and $f^+$ are its two constituent components containing its negative and positive eigenvalues respectively. If the averaged values at the upwind cells are used to calculate the flux components, i.e.

$$f^*_{i+\frac{1}{2}} = f^-_{i+\frac{1}{2}}(\phi_{i+1}) + f^+_{i+\frac{1}{2}}(\phi_i)$$  \hspace{1cm} (2.52)

the scheme is only first order accurate. If the flux components are calculated using the linear approximated upwind values given by equation (2.49) or (2.50), i.e.

$$f^*_{i+\frac{1}{2}} = f^-_{i+\frac{1}{2}}(\phi^R_{i+\frac{1}{2}}) + f^+_{i+\frac{1}{2}}(\phi^L_{i+\frac{1}{2}})$$  \hspace{1cm} (2.53)

the second order upwind scheme is obtained.

However, a simplified flux splitting approach is adopted in this work. Among all characteristics only when bulk flow is subsonic, there is one characteristic propagates in
Unified Pressure Correction Algorithm

the opposite direction to the others, which is the acoustic signal. Usually this signal is insignificant (weak) compared to the others and hence this characteristic wave is ignored here. Subsequently, the characteristic waves considered are of the same sign which is determined by the sign of the flow direction. The numerical flux can therefore be split to

\[
f^*_{i+\frac{1}{2}} = f^+_{i+\frac{1}{2}} = f^+_{i+\frac{1}{2}}(\phi^L_{i+\frac{1}{2}}) \quad u_{i+\frac{1}{2}} > 0 \quad 2.54
\]

if the flow is in the positive direction, and

\[
f^*_{i+\frac{1}{2}} = f^-_{i+\frac{1}{2}} = f^-_{i+\frac{1}{2}}(\phi^R_{i+\frac{1}{2}}) \quad u_{i+\frac{1}{2}} < 0 \quad 2.55
\]

otherwise. This method is not a complete characteristic method but an approximation, however, it is a good approximation and the second order of solution accuracy is not affected, because the order of accuracy is determined by the evaluation of \( \phi_{i+1/2} \).

The above flux differenceing scheme is prone to oscillations around large flow gradients. Therefore, some kind of nonlinear Total Variation Diminishing (TVD) limiter must be imposed to prevent the onset of spurious solutions.

The function of the TVD limiter is to monitor the numerical solution and detect conditions under which a second order scheme will violate the TVD condition and generate new extrema. Upon detection, the non-linear limiter will be triggered so that the TVD property can be preserved. The second order upwind method can be expressed as a first order upwind method plus an additional term. Since the first order upwind will itself not generate new extrema, the occurrence of new extrema must originate from the additional term in the expansion expression. This term corresponds to the difference approximation for the slope of the linear interpolation represented by a 2\textsuperscript{nd} order scheme. The TVD scheme basically controls the variation with the use of a limiting function. The detector used by the TVD limiter to reveal conditions favourable to new extrema production is the ratio of two adjacent variations, and the non-linear limiter puts a band on the slope within a specified range which meets the TVD requirement and leads to a smooth solution, free of oscillation.
In this work, the 'minmod' limiter (Liu, et al. 1995) is incorporated into the above differencing scheme because it is simple and can be easily integrated into a pressure-correction procedure. This limiter is defined by a function whose general form is:

\[ \text{minmod}(x, y) = \text{sgn}(x) \cdot \max[0, \min(|x|, \text{sgn}(x \cdot y))] \]

This function selects the argument with the smallest modulus from a series when they all have the same sign, and zero otherwise, i.e.

\[
\text{minmod}(x, y) = \begin{cases} 
  x & \text{if } |x| < |y| \\
  y & \text{if } |x| > |y| \\
  0 & \text{if } xy < 0 
\end{cases}
\]

As already described, a TVD limiter is used to constrain the slope of the linear function. The TVD limited linear interpolation function equivalent to (2.49) and (2.50) is given by:

\[
\phi^L_{i+\frac{1}{2}} = \phi_i + \frac{1}{2} \cdot \Psi^+(r) (\phi_{i+1} - \phi_i) \\
\phi^R_{i+\frac{1}{2}} = \phi_i + \frac{1}{2} \cdot \Psi^-(r) (\phi_{i+2} - \phi_{i+1})
\]

where \( \Psi(r) \) is the TVD non-linear limiter and is a function of the ratio of two adjacent variations:

\[
\Psi^+(r)_{i+\frac{1}{2}} = \Psi^+(r)_{i+\frac{1}{2}} \\
\Psi^-(r)_{i+\frac{1}{2}} = \Psi^-(r)_{i+\frac{1}{2}}
\]

where \( r \) is the ratio of two adjacent gradients.
Unified Pressure Correction Algorithm

\[
\begin{align*}
  r_{i+\frac{1}{2}}^+ &= \frac{\phi_{i+2} - \phi_{i+1}}{\phi_{i+1} - \phi_i} \\
  r_{i+\frac{1}{2}}^- &= \frac{\phi_{i} - \phi_{i-1}}{\phi_{i+1} - \phi_i}
\end{align*}
\]

A TVD scheme requires that the limiting function \( \Psi(r) \) lie within the range shown by the shaded area in Figure 2.3 (Hirsch 1988).

![Figure 2.3 Limiter region for second-order TVD scheme](image)

The minmod limiting function of

\[
\Psi(r) = \minmod(r, 1)
\]

corresponds to the lowest boundary of the considered TVD domain and is used in this work. Therefore, the linear gradients in (2.58) and (2.59) are limited accordingly and the cell face values become:

\[
\begin{align*}
  \phi_{i+\frac{1}{2}}^L &= \phi_i + \frac{1}{2} \cdot \minmod\left( r_{i+\frac{1}{2}}, 1 \right) \cdot (\phi_i - \phi_{i-1}) \\
  &= \phi_i + \frac{1}{2} \cdot \minmod\left( \delta_{i+\frac{1}{2}}, \delta_{i-\frac{1}{2}} \right)
\end{align*}
\]

55
Unified Pressure Correction Algorithm

\[ \phi_{i+\frac{1}{2}}^R = \phi_{i+1} - \frac{1}{2} \cdot \text{minmod}\left(\frac{\phi_{i+2} - \phi_i}{\frac{\Delta}{2}}, 1\right) \cdot \left(\phi_{i+2} - \phi_{i+1}\right) \]

where \( \delta_{i+\frac{1}{2}} = \phi_{i+1} - \phi_i \). Hence, the minmod limiter effectively forces the slope at the cell face to be always the smaller variation of the current interval and the one upstream, avoiding too rapid a change of adjacent slopes. If the slopes of adjacent intervals change sign, the slope of the current interval is set to zero by the limiter; the approximation for the cell face value thus switches back to first order upwind interpolation.

In multidimensional applications, the above difference scheme can be applied without any modification to each direction independently.

2.6.3 Discretisation Scheme for Diffusive Flux

The diffusion term in the equation is well understood as describing the down-gradient flux of the transport variable. Mathematically, diffusive phenomena are described by terms which contribute to the elliptic nature of the partial differential equation.

The discretisation method used for this type of flux has always been a central scheme where the influence of the upstream and downstream cells are equally weighted in the estimation of the flux at the cell boundary. Hence, the diffusive flux in the u-momentum (2.12), v-momentum (2.13) and energy equations (2.14) are linearly interpolated, for example

\[ \left( \frac{\partial}{\partial s} \rho u \right)_{ef} = \frac{1}{\Delta s} \cdot (\rho u_E - \rho u_P) \]

2.7 Boundary Conditions

It is always a critical issue that appropriate flow conditions are specified at chosen computational boundaries. A boundary specification is only valid if it describes true physical conditions at the boundary. The boundary condition has to be imposed in such a
Unified Pressure Correction Algorithm

way that it is compatible with the numerical scheme used in the interior of the flow so that both stability and accuracy of the global scheme will not be adversely affected.

2.7.1 Overview

In characteristic analysis, flow information at a boundary is transported along the characteristics. Some of the characteristics point inwards to the flow field, while others point outwards. The number of fixed boundary conditions required to close an initial boundary partial differential problem is equal to the number of incoming characteristics which propagate from the boundary towards the inside of the flow region. It is also required to supply numerical boundary conditions corresponding to the outgoing characteristics which transport information from the inside of the domain to influence and modify the flow state at the boundary itself. In order to obtain a well-posed numerical problem with a unique solution, it is essential that incoming characteristics are prescribed via a known fluid state throughout the flow evolution and outgoing characteristics are specified via a condition that is derived from the interior solutions.

Either a dummy row of boundary cells with zero thickness or a row of 'halo' cells are used in most CFD codes to enable the numerical implementation of boundary conditions. When boundary cells are used, the boundary condition is imposed directly at the boundary nodes. When halo cells are utilised, the specification of the boundary condition is achieved indirectly by defining the quantities in the halo cells. These quantities are so defined that the interpolation between them and interior cells reproduces the required flow conditions on the boundary. While both approaches have found wide use, the straightforward dummy boundary cell has been adopted in the present work.

Boundary conditions can be implemented explicitly or implicitly in a time marching scheme. If the boundary is treated in an explicit way, the well-known CFL stability condition is reintroduced by the boundary condition implementation. The favourable stability property of an implicit interior scheme could be significantly worsened. On the other hand, if an implicit boundary condition is coupled with an implicit interior scheme, this does not introduce any additional stability constraints and the global stability property can be maintained.
Boundary conditions are generally expressed as a function of characteristic, conservative or primitive variables, or some combinations of these. The characteristic variables are the quantities transported by the characteristic waves. They contain the disturbance information on the boundary and influence directly the flow state in the interior domain, or the flow information from the interior which influence the boundary state, depending on the direction of the characteristic. From a physical point of view, the most rigorous boundary method is the characteristic boundary method (Hirsch 1988). However, in many circumstances the incoming characteristic variables are not available, and other methods must be developed. For example, the known boundary conditions in the form of specified primitive variables can be applied instead. Stagnation or total flow variables such as total pressure and total temperature can also be specified.

2.7.2 Boundary Condition Implementation

In the present work, internal flow problems are of major interest. The types of physical boundaries involved are upstream flow inlet, downstream flow exit, centre line in axisymmetrical flows and solid walls.

As far as possible, variables imposed as physical boundary conditions are those that have been obtained from experiment. For other numerical boundary conditions, a variety of extrapolation conditions are employed.

Because the pressure-correction procedure is a semi-implicit scheme which is subject to an explicit like stability restriction, to some extent, the boundary conditions have been implemented explicitly in the present solution algorithm. The use of explicit boundary conditions has only a minor degrading impact on the stability range of the interior scheme, and its simplicity of implementation becomes a big advantage.

2.7.2.1 Subsonic Inlet Boundary

In 2D, three characteristics at such a boundary are directed into the inside of the computational domain, with only one pointing towards the outside. Three variables are therefore required by the physical boundary conditions to be fixed. Since the upstream stagnation condition is almost always known in the flow problems considered in this work,
Unified Pressure Correction Algorithm

total pressure and total temperature are imposed at a subsonic inlet boundary, which equivalently fixes the total kinetic energy and the total energy flowing into the solution domain. The third invariant variable supplied is the inflow angle. The numerical boundary condition is imposed by extrapolating the outgoing normal Riemann variable from the interior onto the boundary. When pressure and velocity distributions are uniform along an inflow boundary, the Riemann variables normal to the inlet surface remain constant along the characteristics. Hence, the outgoing normal Riemann invariant at the boundary can be obtained from the ones on the same characteristics. A zero order extrapolation is used in the implementation, and the boundary cell and the immediate interior cell are assumed to be of the same characteristics.

In some flow cases, linear extrapolation of the static pressure has been applied instead. Though zero order pressure extrapolation has proven to be more stable, its zero order accuracy may degrade the second order accuracy of the interior scheme, whereas the first order scheme maintains a globally second order accuracy. Therefore, linear extrapolation is adopted unless it leads to instability or spurious oscillations near the inlet.

2.7.2.2 Subsonic Outlet Boundary

Here there is one characteristic propagating inwards to the solution region, hence one variable must be specified. However, the choice of this physical boundary condition can not be arbitrary due to the interrelation between the physical boundary conditions at the inflow and the outflow boundaries. The flow variables fixed at both ends have to be such that they together lead to a valid and unique converged flow state. The frequently specified variable at a subsonic outlet is the static pressure, which is consistent with an inlet fixed total pressure. This pair will determine the mass flow rate in a converged solution, therefore when these two conditions are imposed as such, the mass flow rate at the inflow cannot be prescribed, otherwise conflict will result.

The three outgoing characteristics at the outlet are satisfied by extrapolating two conservative variables $p_u$ and $p_v$, and total enthalpy $H$. 

2.7.2.3 Solid Wall Boundary

At a stationary solid wall the normal velocity vanishes and no flux penetrates the wall. The vanishing normal velocity is imposed as the single physical boundary condition required at this boundary. The three numerical boundary conditions are specified as the wall pressure, the tangential velocity and the wall temperature. In an Euler equation (inviscid) calculation, both the pressure and the tangential velocity condition are assumed to have zero gradients in the normal direction, and the wall temperature is extrapolated. In Navier-Stokes (viscous) calculations, the tangential velocity must also be set to zero at the wall while the other two variable conditions remain the same. In any extrapolation, linear extrapolation is preferred for the second order interior scheme, but zero order extrapolation is used if the first order extrapolation is unstable in the numerical flow tests. In some test cases in this work, instead of extrapolating the temperature, zero gradient total enthalpy (i.e. \( \frac{\partial H}{\partial n} = 0 \)) at the wall is applied as a numerical boundary condition, and the total enthalpy value of the interior cell next to the wall is taken as the one on the wall. Wall temperature is then calculated from the total enthalpy via thermodynamic relations.

However, the above boundary condition for the pressure on the wall, i.e. zero normal gradient, is only valid when the wall is not curved, and the numerical discretisation of this condition can be expressed as

\[
P_{\text{wall}} = P_{\text{neighbour}}
\]

where the subscript ‘neighbour’ denotes the interior cell immediate next to the wall boundary cell. On a curved wall, the normal gradient of the wall pressure is a non-zero quantity and is given by (Hirsch 1988):

\[
-(\hat{n} \cdot \nabla p) + \rho \hat{v} \cdot (\hat{v} \cdot \nabla)\hat{n} = 0
\]

with \( \hat{n} \) equal to the unit vector along the wall normal. Because the computational mesh system is seldom formed normal to the curved wall, that is mesh points aligned along the wall normal, equation (2.69) has to be projected to the curvilinear coordinates, \((\xi, \eta)\) in
two dimensions, with the coordinate line, either a $\zeta = \text{constant}$ or a $\eta = \text{constant}$ being the wall surface. In the curvilinear coordinate system, the normal of the wall $\hat{n}$ can be defined as

$$\hat{n} = \ddot{\zeta} \zeta/|\ddot{\zeta} \zeta|$$ \hspace{1cm} (2.70)

if wall surface is on a $\zeta$ coordinate line, or

$$\hat{n} = \ddot{\eta} \eta/|\ddot{\eta} \eta|$$ \hspace{1cm} (2.71)

if wall surface is on a $\eta$ coordinate line. Therefore, the normal gradient of the wall pressure can be expressed in the general coordinate system as follows: (e.g. if $\hat{n}$ is given by (2.70))

$$\hat{n} \cdot \nabla \bar{p} = \ddot{\zeta} \zeta/|\ddot{\zeta} \zeta| \cdot \nabla \bar{p}$$ \hspace{1cm} (2.72)

$$= \frac{1}{\sqrt{\eta_x^2 + \eta_y^2}} \left[ (\eta_x \hat{i} + \eta_y \hat{j}) \cdot \left( \frac{\partial \hat{p}}{\partial \xi} \hat{i} + \frac{\partial \hat{p}}{\partial \eta} \hat{j} \right) \right]$$

$$= \frac{1}{\sqrt{\eta_x^2 + \eta_y^2}} \left[ \left( \xi_x \eta_x + \xi_y \eta_y \right) \frac{\partial \bar{p}}{\partial \xi} + \left( \eta_x \xi_x + \eta_y \xi_y \right) \frac{\partial \bar{p}}{\partial \eta} \right]$$

For the second term on the left hand side of equation (2.69), a similar transformation as in equation (2.72) can be performed. As a result, equation (2.69) is expressed in the curvilinear coordinate system and it is no longer difficult to discretise.
Unified Pressure Correction Algorithm

2.7.2.4 Centre Line Boundary

Along the centre line of an axisymmetric flow, the flow is parallel to this line. That is to say, the flow velocity component that is normal to the centre line goes to zero at the centre line. As a result there is no mass flux (or any other flux) across a symmetry line. Therefore, on the centre line boundary, the zero normal velocity at the centre line is prescribed as the physical boundary condition. The other three numerical boundary conditions are specified respectively as the zero gradient of the tangential velocity component at the centre line, and the zero gradient of both total enthalpy and pressure. Any other flow quantities required are derived subsequently.

2.8 Rhie & Chow Pressure Smoothing

When a collocated method is used, it is well known that a pressure-velocity decoupling, appearing in the form of odd-even checkerboard oscillations, can develop in the solutions of pressure and velocity fields. The main cause is the use of a central difference for the pressure gradients and linear interpolation of mass flux at cell faces. The most effective method to eliminate the associated oscillation is to couple the change of mass flow rate in a cell directly with the pressure difference across this cell so that the evolution of the velocity fields in the cell closely follow the variation of the pressure field in the same cell. Rhie&Chow smoothing (Rhie et al. 1983) provides a way to realize such a coupling. In a differencing strategy as described above, the mass flux at cell faces is linearly interpolated according to the linear formulation of the scheme. The Rhie&Chow smoothing replaces this approximation by a modified linear interpolation which can be expressed as

$$
p_{u_i + \frac{1}{2}, j} = \frac{1}{2}(p_{u_i, j} + p_{u_{i+1}, j}) + \text{(psmooth)}_{i + \frac{1}{2}, j}
$$

(2.73)

where
**Unified Pressure Correction Algorithm**

\[
\text{(psmooth)}_{i+\frac{1}{2},j} = \frac{1}{2}\left[ \frac{D_{i,j}^{\text{pu}}(p_{i+1,\frac{1}{2},j} - p_{i-1,\frac{1}{2},j})}{A_{i,j}^{\text{pu}}} - \frac{D_{i+1,j}^{\text{pu}}(p_{i+1,\frac{1}{2},j} - p_{i+1,\frac{1}{2},j})}{A_{i+1,j}^{\text{pu}}} \right] + \left( \frac{D_{i,j}^{\text{pu}}}{A_{i,j}^{\text{pu}}} + \frac{D_{i+1,j}^{\text{pu}}}{A_{i+1,j}^{\text{pu}}} \right)(p_{i+1,j} - p_{i,j}) \]  

(2.74)

where the discretised \( \text{pu} \) momentum equation for cell \((i,j)\) is

\[
A_{i,j}^{\text{pu}}(p_{i,j}) = \sum A_{\text{OD}}^{\text{pu}}(\rho_{\text{OD}}) + B_{i,j}^{\text{pu}} + D_{i,j}^{\text{pu}}(p_{i+1,\frac{1}{2},j} - p_{i-1,\frac{1}{2},j}) \]  

(2.75)

### 2.9 Linear Matrix Solver

The second order MUSCL discretisation applied to the transport equations listed in equations (2.2) and (2.3) results in a pentadiagonal matrix problem. The solver adopted in the pressure-correction procedure to invert this matrix for all transport equations and for the pressure-correction equation itself is the simple \( \text{SLOR} \) line solver (Fletch 1988). This solver transforms the pentagonal problem to a tridiagonal matrix problem by choosing one line of unknowns in which the unknowns on the lines other than this line are moved to the explicit source term temporarily. Then the tridiagonal matrix is inverted by the TDMA procedure (Fletcher 1988). This is not the most efficient way to solve the algebraic equations, particularly for the pressure-correction variable where conjugate gradient methods (Shewchuk 1994) have been proposed in the literature. However, in time-dependent problems when the time-step is small to capture time-accurate solutions, this method is often sufficient.
2.10 References


Unified Pressure Correction Algorithm


Unified Pressure Correction Algorithm


Chapter 3

Unified Block Implicit Algorithm
Unified Block Implicit Algorithm

3.1 Preamble

A unified solution procedure based on a conventional density-based algorithm has also been investigated in this thesis, and the details of this procedure are laid out in this chapter.

3.2 Review

Research to extend density-based compressible algorithms to the incompressible regime has been underway for about two decades. In traditional density-based algorithms, the conserved flow variables, i.e. \( \rho, \rho u, \rho v \) and \( \rho e \) in a two dimensional flow are usually used as dependent variables. The direct application of these algorithms to incompressible flows usually suffers two problems: the inaccuracy of solutions and the inefficiency of computation, as has been discussed in Chapter 1, due to round-off errors as density variation shrinks to zero and stiffness of the equation system at low Mach number (large discrepancy between acoustic speed and flow speed).

Techniques proposed in the literature to alleviate the stiffness problem are dominated by the preconditioning technique. A good review has been given by Choi et al. (Choi, et al. 1993) and Turkel et al. (Turkel, et al. 1993). This technique essentially introduces an artificial sound wave whose speed is of the same order as the flow velocity so that good convergence can be obtained at all "Mach numbers". The idea has come from the preconditioning technique in matrix analysis, which scales the matrix eigenvalues to make the condition number close to one. Since the true transient nature of the original equation system is often tampered with by the preconditioning operation, the solution is only meaningful at steady state, unless special solution algorithms such as dual-time stepping or pseudo-time stepping (Buelow et al. 1997, Jameson, et al. 1981) are used.

The invariant density problem can be eliminated by solving for pressure as a main variable instead of density in the solution procedure. Since the pressure varies appreciably in flows at any Mach number, the employment of pressure as the dependent variable is believed to be more appropriate for a unified algorithm and hence adopted in this work. (Note that since the basic algorithm is density-based, then even though it has been re-formulated to have pressure as a main dependent variable, the 'density-based' label is still retained.)
Some density-based methods are implicit solution procedures. In theory, an implicit time marching scheme is free from stability problem. In reality, however, this is difficult to achieve. Nevertheless, the implicit scheme often has better stability properties compared with an explicit time marching scheme, so it is adopted in this work, specifically a block implicit scheme. The block implicit scheme is characterised by simultaneous solution of the equation system over the entire flow field (unlike the pressure-correction method which adopts a segregated solution procedure), the flow variables which determine the flow state in a cell are updated together in the form of a block. In this method, the flows in the whole domain are strongly coupled during the numerical calculation.

### 3.3 Non-Dimensional Form of Governing Equations

As in most density-based methods, the equations in the present method are first non-dimensionalised before developing the solution algorithm. Further, the primitive variables are chosen as solution variables because it will simplify the linearisation of the nonlinear equation system, see below. The variables other than the primitive ones \((u, v, p, T)\) in the Navier-Stokes equations \((2.1) \sim (2.3)\) are re-written in terms of primitive variables, for example

\[
\rho = \frac{p}{RT} \quad \text{(3.1)}
\]

\[
E = C_v T + \frac{1}{2}(u^2 + v^2) \quad \text{(3.2)}
\]

\[
H = C_p T + \frac{1}{2}(u^2 + v^2) \quad \text{(3.3)}
\]

where \(C_v\) and \(C_p\) are constant volume and constant pressure specific heat coefficients respectively, \(R\) is the gas constant, and these are of course linked by

\[
C_v = C_p - R \quad \text{(3.4)}
\]
Unified Block Implicit Algorithm

The governing equations are still cast in conservative form despite the use of primitive variables due to the desire to retain good shock capturing. The reference values - velocity \( \tilde{u}_{\text{ref}} \), density \( \tilde{\rho}_{\text{ref}} \), temperature \( \tilde{T}_{\text{ref}} \) and dynamic viscosity \( \tilde{\mu}_{\text{ref}} \), are chosen to be those of the upstream flow. Length \( \tilde{L}_{\text{ref}} \) is set equal to the characteristic length of the geometry. The Reynolds number, Mach number and Prandtl number are defined as

\[
Re = \frac{\tilde{\rho}_{\text{ref}} \tilde{u}_{\text{ref}} \tilde{L}_{\text{ref}}}{\tilde{\mu}_{\text{ref}}} \tag{3.5}
\]

\[
M_{\infty} = \frac{\tilde{u}_{\text{ref}}}{\sqrt{\gamma \tilde{R} \tilde{T}_{\text{ref}}}} \tag{3.6}
\]

\[
Pr = \frac{\tilde{C}_p \tilde{\mu}_{\text{ref}}}{\tilde{K}} \tag{3.7}
\]

where the quantities with a tilde are dimensional variables. The non-dimensional dynamic viscosity, gas constant and specific heat are defined by:

\[
\mu = \frac{\tilde{\mu}}{\tilde{\mu}_{\text{ref}}} \tag{3.8}
\]

\[
R = \frac{\tilde{R}}{\tilde{u}_{\text{ref}}^2 / \tilde{T}_{\text{ref}}} = \frac{1}{\gamma M_{\infty}^2} \tag{3.9}
\]

\[
C_p = \frac{\tilde{C}_p}{\tilde{u}_{\text{ref}}^2 / \tilde{T}_{\text{ref}}} = \frac{1}{(\gamma - 1) M_{\infty}^2} \tag{3.10}
\]

Other non-dimensional flow variables are defined as
Consequently, the governing equations in two-dimensional curvilinear coordinates become

\[ \frac{\partial}{\partial t} Q(q) + \frac{\partial}{\partial \xi} E(q) + \frac{\partial}{\partial \eta} F(q) = 0 \]

where

\[ q = \begin{bmatrix} p \\ u \\ v \\ T \end{bmatrix} \]

\[ Q = \frac{1}{j} \begin{bmatrix} p \\ \frac{p}{T} \\ \frac{pu}{T} \\ \frac{pv}{T} \\ (C_p - R)p + \frac{1}{2} \frac{pu^2}{T} + \frac{1}{2} \frac{pv^2}{T} \end{bmatrix} \]
Unified Block Implicit Algorithm

\[
E = \frac{1}{J} \cdot \left[ \begin{array}{c}
\frac{p}{T} \cdot U^c \\
\frac{p}{T} \cdot U^c + R p \xi_x - (\xi_x \tau_{xx} + \xi_y \tau_{xy}) \\
\frac{p}{T} \cdot U^c + R p \xi_y - (\xi_x \tau_{xy} + \xi_y \tau_{yy}) \\
\left( C_p p + \frac{1}{2} \frac{p u^2}{T} + \frac{1}{2} \frac{p v^2}{T} \right) \cdot U^c \\
-\left[ (\xi_x u \tau_{xx} + (\xi_x v + \xi_y u) \tau_{xy} + \xi_y v \tau_{yy}) \right] \\
\frac{R C_p u}{Pr Re} \left[ (\xi_x^2 + \xi_y^2) T_\xi + (\xi_x \eta_x + \xi_y \eta_y) T_\eta \right] 
\end{array} \right]
\]

\[
F = \frac{1}{J} \cdot \left[ \begin{array}{c}
\frac{p}{T} \cdot V^c \\
\frac{p}{T} \cdot V^c + R p \eta_x - (\eta_x \tau_{xx} + \eta_y \tau_{xy}) \\
\frac{p}{T} \cdot V^c + R p \eta_y - (\eta_x \tau_{xy} + \eta_y \tau_{yy}) \\
\left( C_p p + \frac{1}{2} \frac{p u^2}{T} + \frac{1}{2} \frac{p v^2}{T} \right) \cdot V^c \\
-\left[ (\eta_x u \tau_{xx} + (\eta_x v + \eta_y u) \tau_{xy} + \eta_y v \tau_{yy}) \right] \\
\frac{R C_p u}{Pr Re} \left[ (\xi_x \eta_x + \xi_y \eta_y) T_\xi + (\eta_x^2 + \eta_y^2) T_\eta \right] 
\end{array} \right]
\]

and

\[
\tau_{xx} = \frac{2 R \mu}{3 Re} \left[ 2(\xi_x u \xi + \eta_x u \eta) - (\xi_y v \xi + \eta_y v \eta) \right] \quad 3.18
\]

\[
\tau_{yy} = \frac{2 R \mu}{3 Re} \left[ 2(\xi_y v \xi + \eta_y v \eta) - (\xi_x u \xi + \eta_x u \eta) \right] \quad 3.19
\]
\[ \tau_{xy} = \frac{R\mu}{Re} \left[ (\xi_y u_x + \eta_y u_\eta) + (\xi_x v_\xi + \eta_x v_\eta) \right] \quad 3.20 \]

\[ U^c = u \cdot \xi_x + v \cdot \xi_y \quad 3.21 \]

\[ V^c = u \cdot \eta_x + v \cdot \eta_y \quad 3.22 \]

E and F contain the contributions from both convective and diffusive fluxes.

### 3.4 Discretisation Scheme

#### 3.4.1 Central Difference Scheme and Artificial Dissipation Model

The convective flux at a cell face is evaluated by linear interpolation of the flux at the cell centres on either side of the face. This is equivalent to a central differencing scheme. It is well known that to ensure good shock capturing, the central differencing scheme needs to be modified by a numerical smoothing mechanism, because the physical viscous terms would not provide sufficient dissipation to eliminate the high frequency spurious errors generated by the nonlinear effect. There are a large number of dissipation models developed in the literature (Hirsch 1988), the one adopted in this work is that of Jameson (Jameson et al. 1981). It has proven to be robust and efficient in an explicit Runge-Kutta solution procedure (Jameson et al. 1981) and a block implicit solution procedure (Caughey 1993). There are two coefficients in this model - artificial viscosities - which need to be determined via trial and error procedures. They remain constant during the calculation. Recently, there have been attempts to improve the adaptability and selectivity of this smoothing model. One idea is to replace the two scalar artificial viscosities with an anisotropic form which varies with the local flow condition, so that the amount of smoothing can be constantly adjusted to an optimum value in both spatial and time domains. For instance, Turkel (Turkel, et al. 1994) utilised the modulus of the flux Jacobian matrices as the artificial viscosity, while Kallinderis (Kallinderis, et al. 1995) determined the coefficients by imposing a limit between the contribution from the numerical dissipation flux and that from the real flux. In the present work, it has been
Unified Block Implicit Algorithm

found that the scalar form of the artificial viscosities can produce satisfactory results for the flow problems of interest, therefore this simple approach has been adopted.

The components of $E$ and $F$ in (3.13) are written as the sum of a physical convective and physical diffusive flux

$$E = \hat{E} + \tilde{E} \quad F = \hat{F} + \tilde{F}$$  \hspace{1cm} 3.23

the first is the convective component, the term with the triangular hat is the diffusive component. Jameson's dissipation model is to add a blend of 1st and 3rd order difference terms to the convective flux at the cell face which is equivalent to adding a 2nd and a 4th derivative numerical diffusion term to the original equation. Thus

$$\hat{E}_{i, j}^{*} + \frac{1}{2} = \hat{E}_{i, j}^{*} + \frac{1}{2} - \tilde{D}_{i, j}^{*} + \frac{1}{2}$$  \hspace{1cm} 3.24

$$\hat{F}_{i, j}^{*} = \hat{F}_{i, j} + \frac{1}{2} - \tilde{D}_{i, j} + \frac{1}{2}$$  \hspace{1cm} 3.25

where $\tilde{D}$ represents the artificial dissipation terms, and the subscript notation is shown in Figure 3.1.
Unified Block Implicit Algorithm

\[ \eta \]

\[ \begin{array}{ccc}
(i-1,j+1) & (i,j+1) & (i+1,j+1) \\
(i-1,j) & (i,j) & (i+1,j) \\
(i-1,j-1) & (i,j-1) & (i+1,j-1) \\
\end{array} \]

\[ \xi \]

FIGURE 3.1 A two dimensional control volume in computational space

As explained already, the convective flux at the cell faces is evaluated by:

\[ \widehat{E}_{i+\frac{1}{2},j} = \frac{1}{2}(\widehat{E}_{i+1,j} + \widehat{E}_{i,j}) \]  
3.26

\[ \widehat{F}_{i,j+\frac{1}{2}} = \frac{1}{2}(\widehat{F}_{i,j+1} + \widehat{F}_{i,j}) \]  
3.27

The artificial dissipation \( \widehat{D}_{i+\frac{1}{2},j} \) is defined as

\[ \widehat{D}_{i+\frac{1}{2},j} = \frac{1}{\Delta t^*} \left[ \varepsilon^{(2)}_{i+\frac{1}{2},j} (\phi_{i+1,j} - \phi_{i,j}) - \varepsilon^{(4)}_{i+\frac{1}{2},j} (\phi_{i+2,j} - 3\phi_{i+1,j} + 3\phi_{i,j} - \phi_{i-1,j}) \right] \]  
3.28

where \( \phi \) is the transported variable governed by the corresponding equation under question, \( \Delta t^* \) is the time that the fastest characteristic wave takes to cross a cell interval.
and \( V_{i+1/2,j} \) is the volume of the control volume that surrounds the cell face, which can be approximated as the average of the volumes of the two cells on either side of this face. The scaling of the smoothing term by \( \Delta t^* \) assures the steady state solution is independent of the time step. The inclusion of the metric Jacobian ensures that the dissipation terms are consistent in dimension with the real diffusive flux derivatives.

The non-linear variable \( \varepsilon^{(2)} \) is defined as

\[
\varepsilon^{(2)}_{i+\frac{1}{2},j} = \max(\theta_{i+1,j}, \theta_{i,j})
\]

where \( \theta_{i,j} \) is adjusted to the flow gradient by the second derivative of the pressure

\[
\theta_{i,j} = k^{(2)} \left| \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{p_{i+1,j} + p_{i,j} + p_{i-1,j}} \right|
\]

in order to detect non-uniform regions in the flow such as near shocks. \( k^{(2)} \) is a scalar coefficient that determines the amount of numerical dissipation added from the second order term. \( \varepsilon^{(2)} \) activates the second order dissipation to suppress strong non-linear numerical oscillations around the flow discontinuity.

The non-linear variable \( \varepsilon^{(4)} \) is defined as:

\[
\varepsilon^{(4)}_{i+\frac{1}{2},j} = \max\left(0, \left( k^{(4)} - \varepsilon^{(2)}_{i+\frac{1}{2},j} \right) \right)
\]

hence it turns on the 4th order dissipation when the second order dissipation is small, i.e. in the smooth flow region. The 4th order dissipation provides background smoothing to damp out the high frequency modes which could stop the residual coming down to the tolerance level. The fourth order smoothing is turned off when \( \varepsilon^{(2)} \) is large, i.e. near shocks, to prevent the excitation of overshoots or undershoots in the vicinity of the shock. \( k^{(4)} \) is a scalar coefficient determining the amount of the fourth order dissipation.
Unified Block Implicit Algorithm

The whole expression in (3.28) is of order $(\Delta x)^3$. Therefore, the artificial dissipation terms have added a 4th order error to the original partial differential equation, two orders lower than the truncation error of the second order differencing scheme. An exception occurs near the shock where the magnitude of the artificial dissipation rises to the order of $(\Delta x)^2$ to stabilise the nonlinear instability. Hence, the global second order accuracy of the solution will not be affected by the addition of this artificial dissipation model.

3.4.2 Artificial Dissipation Terms Near the Boundary

On a boundary, the artificial dissipation model is set to zero. However, it is applied everywhere in the interior of the computational domain.

The central discretisation used in the artificial dissipation terms requires a five-point stencil. While this does not pose any problem at interior cells away from the boundaries, this is not true for cells near a boundary. The stretched stencil implies the utilisation of cells from outside the domain, and there are no such cells defined in the present work. As a result, a one-sided rather than a central differencing operator for the artificial dissipation terms has to be adopted near the boundary. Pulliam (Pulliam 1986) has suggested a procedure to choose and evaluate such a differencing scheme. He found that this dissipation model is dissipative in the entire solution domain if the matrix representation of the dissipation operator is positive definite. Therefore, the coefficients in the one-sided stencil should be determined such that this criterion is satisfied. Pulliam has explained this within a finite difference framework. On a one dimensional grid as shown in Figure 3.2:

![Figure 3.2](image)

**FIGURE 3.2** A one dimensional grid in finite difference system

because the five-point symmetry coefficients of the fourth order dissipation term for the interior points $3 \leq i \leq NJ-2$ are
Unified Block Implicit Algorithm

\[ \beta_{i-2} = 1 \quad \beta_{i-1} = -4 \quad \beta_i = 6 \quad \beta_{i+1} = -4 \quad \beta_{i+2} = 1 \quad 3.32 \]

the one-side coefficients for point \( i = 2 \) should be

\[ \beta_4 = 1 \quad \beta_3 = -4 \quad \beta_2 = 5 \quad \beta_1 = -2 \quad 3.33 \]

for point \( i = NJ - 1 \), they should be

\[ \beta_{NJ-3} = 1 \quad \beta_{NJ-2} = -4 \quad \beta_{NJ-1} = 5 \quad \beta_{NJ} = -2 \quad 3.34 \]

to ensure a positive definite dissipation matrix.

In the current finite volume framework, an example of a one dimensional solution cell is shown in Figure 3.3. The same 5-point stencil to (3.32) can be derived for the interior solution cells \( 3 \leq i \leq NJ-2 \). For the cells near the boundaries \( i = 2 \) or \( i = NJ -1 \), the author has independently developed a one-side differencing method which will produce the same stencil coefficients as those of Pulliam's finite difference relations (3.33) and (3.34)

Assuming \( \Delta \xi = 1 \). For cell \( i = 2 \), the finite volume discretisation of the fourth order derivative proceeds as
Unified Block Implicit Algorithm

\[
\left( \frac{\partial^4 \phi}{\partial \xi^4} \right)_{i=2} = \left( \frac{\partial^3 \phi}{\partial \xi^3} \right)_{i=2} - \left( \frac{\partial^3 \phi}{\partial \xi^3} \right)_{i=1} \\
= \left[ \left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=2} - \left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=3} \right] - \left[ \left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=2} - \left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=1} \right]
\]

Since the linear evaluation of the second order derivative at the cell face \( i = 1 \) will require a point outside the solution domain, this term is assumed zero, which is also consistent with the treatment of the 2\textsuperscript{nd} order dissipation model on the boundary

\[
\left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=1} = 0
\]

Consequently equation (3.35) becomes

\[
\left( \frac{\partial^4 \phi}{\partial \xi^4} \right)_{i=2} = \left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=2} - 2\left( \frac{\partial^2 \phi}{\partial \xi^2} \right)_{i=2}
\]

\[
= [\phi_4 - 2\phi_3 + \phi_2] - 2[\phi_3 - 2\phi_2 + \phi_1] = \phi_4 - 4\phi_3 + 5\phi_2 - 2\phi_1
\]

Therefore, the coefficients in this one-sided stencil are

\[
\beta_4 = 1 \quad \beta_3 = -4 \quad \beta_2 = 5 \quad \beta_1 = -2
\]

which are exactly the same as equation (3.33). When the same procedure is applied to cell \( i = NJ - 1 \), the stencil of (3.34) will be reproduced. Because of the coincidence of the dissipation coefficient matrix in this work with that analysed by Pulliam, the dissipation model in this work is assured to produce satisfactory smoothing up to the boundary.
3.4.3 Implicit or Explicit Artificial Dissipation

From a linear analysis of a nondimensional convective model equation, Pulliam (Pulliam 1986) has concluded that when both second and fourth order dissipation terms are added explicitly to an implicit algorithm, an explicit stability bound is produced which would prevent fast convergence. The additional stability requirement associated with the artificial dissipation model cannot be completely eliminated unless the whole model is inserted implicitly into the implicit solution procedure.

The fully implicit treatment of the fourth derivative term will increase considerably the computational work and storage requirement in a block implicit algorithm. For example, in a two dimensional Approximate Factorisation (AF) algorithm, a block triangle inversion problem will change to a block pentadiagonal inversion problem. However, the non-factorised point iterative inversion algorithm adopted in the present work (see below) will not be affected with respect to computing cost by an implicit treatment of the fourth order dissipation model. Therefore, the use of an implicit dissipation model can be readily implemented. It will be demonstrated later that it is actually essential to use implicit artificial dissipation if larger time steps are to be used in the chosen iterative algorithm. This is due to the poor diagonal dominance of the implicit coefficient matrix resulting from the central differencing scheme. (It is also required for effective realisation of multigrid acceleration, see later)

3.5 Iterative Matrix Inversion Method

A Newton iteration method has been employed to solve the nonlinear equation system.

The semi-discretised form of equation (3.13)

\[
\frac{Q^{n+1} - Q^n}{\Delta t} = -\left( \frac{\partial F^{n+1}}{\partial \xi} + \frac{\partial F^{n+1}}{\partial \eta} \right)
\]

3.39

is equivalent to
Unified Block Implicit Algorithm

\[ \frac{\Delta Q^n}{\Delta t} = -R(Q^{n+1}) \] 3.40

where R(Q) expresses the flux nonequilibrium or the residual of the equation before the steady state solution is converged.

A Newton iterative procedure leads equation (3.39) towards the steady-state solution by generating a sequence of major iterates \( Q^{n+1} = Q^n + \Delta Q^n \), whereby the increment \( \Delta Q^n \) is obtained by solving the nonlinear equation (3.40) using Newton iteration. This iteration procedure is formulated by applying local iteration linearisation to \( R(Q^{n+1}) \) in (3.40) to

\[ R(Q^{n+1}, k+1) \approx R(Q^{n+1}, k) + \left( \frac{\partial R}{\partial Q} \right)^{n+1, k} \Delta Q^{n, k} \] 3.41

where \( k \) is the iteration index, \( \frac{\partial R}{\partial Q} \) gives the Jacobian matrix \( J_R(Q) \) of the residual \( R(Q) \), and consequently, the nonlinear equation (3.40) is transformed to a linear system:

\[ \left( \frac{1}{\Delta t} + J_R^{n+1, k} \right) \Delta Q^{n, k} = -R(Q^{n, k}) \] 3.42

and a linear matrix inversion algorithm is subsequently used to solve this equation system. The correction \( \Delta Q^n \) will be obtained when the Newton iteration converges at time level \( n+1 \). After the solution at current time level \( n+1 \) is updated, i.e. \( Q^{n+1} = Q^n + \Delta Q^n \), a new round of Newton nonlinear iteration is launched for the next time level \( n+2 \).

The classical Newton nonlinear iteration method is of quadratic convergence, but this usually implies considerable computer memory and CPU time requirements for the calculation of the Jacobian. On the other hand, the Jacobian matrices cannot always be evaluated numerically accurately. As a result, the true Newton method is generally not the favoured choice for practical applications; instead, a quasi-Newton method is often used. In the quasi-Newton method, either the Jacobian matrix is approximated or the linear matrix at each Newton iteration is not solved exactly, or both. In both cases, computing time and memory requirements are reduced significantly, although the convergence rate also suffers.
Unified Block Implicit Algorithm

The Jacobian matrix has not been evaluated directly in this work, instead, a linearisation operation based on approximated Taylor Series Expansion is applied to each nonlinear term in the semi-discretised equation. The same approach has been adopted by Chen (1990) and he has demonstrated the equivalence of this approach to the direct approximation of the Jacobian matrix into a difference form.

The Taylor expansion of a nonlinear function \( f(x,y,z) \) can be expressed as

\[
f(x + \Delta x, y + \Delta y, z + \Delta z) = f(x, y, z) + \frac{\partial f}{\partial x} \cdot \Delta x + \frac{\partial f}{\partial y} \cdot \Delta y + \frac{\partial f}{\partial z} \cdot \Delta z + O(\Delta x^2, \Delta y^2, \Delta z^2)
\]

and the approximation can be made:

\[
f(x + \Delta x, y + \Delta y, z + \Delta z) \approx f(x, y, z) + \frac{\partial f}{\partial x} \cdot \Delta x + \frac{\partial f}{\partial y} \cdot \Delta y + \frac{\partial f}{\partial z} \cdot \Delta z
\]

If written in the form for an iterative procedure, equation (3.44) becomes

\[
f(x^{k+1}, y^{k+1}, z^{k+1}) = f(x^k, y^k, z^k) + \left( \frac{\partial f}{\partial x} \right)^k \cdot \Delta x^k + \left( \frac{\partial f}{\partial y} \right)^k \cdot \Delta y^k + \left( \frac{\partial f}{\partial z} \right)^k \cdot \Delta z^k \]

where \( k \) is the iteration index, or

\[
f(x^{k+1}, y^{k+1}, z^{k+1}) = f(x^k, y^k, z^k) + \left( \frac{\partial f}{\partial x} \right)^k \cdot (x^{k+1} - x^k) + \left( \frac{\partial f}{\partial y} \right)^k \cdot (y^{k+1} - y^k) + \left( \frac{\partial f}{\partial z} \right)^k \cdot (z^{k+1} - z^k)
\]

This is the equation that is used here to formulate the linear equation system.

For the equation system (3.13), a finite volume method is first applied to transform this equation to a semi-discretised form. Then, any nonlinear terms are linearised following (3.46). To serve as an illustration, the linearisation of two typical nonlinear terms encountered in this equation system are demonstrated here. One is for the transient term in the continuity equation:
Unified Block Implicit Algorithm

\[
\begin{align*}
(p)^{n+1,k+1} & \approx (\frac{1}{T})^{n+1,k} (p)^{n+1,k+1} - (p)^{n+1,k} - (\frac{1}{T^2})^{n+1,k} (T^{n+1,k+1} - T^{n+1,k}) + (p)^{n+1,k} \\
& = (\frac{1}{T})^{n+1,k} (p)^{n+1,k+1} - (\frac{1}{T^2})^{n+1,k} (T^{n+1,k+1} + (p)^{n+1,k})
\end{align*}
\]

where \( n \) is the time step index and \( k \) is the iteration index in solving nonlinear equation system. Another example is a typical convective term in the momentum equation, similarly after Taylor series expansion operation and combining the same terms:

\[
\begin{align*}
(pu)^{n+1,k+1} & \approx (\frac{pu}{T})^{n+1,k} + (\frac{pu}{T^2} y) (p)^{n+1,k+1} \\
& = (\frac{pu}{T} x) (p)^{n+1,k} + (\frac{pu}{T^2} y) (p)^{n+1,k+1}
\end{align*}
\]

After carrying out such linearisation, the final form of all four linear algebraic equations for a cell can be expressed as:

\[
[A]^p_{i,j} (q)^{n+1,k+1} + \sum_{nb} [A]^b_{i,j} (q)^{n+1,k+1} = (B)^{n+1,k} + (B)^n_{i,j}
\]

where the coefficient \([A]^p\) and \([A]^b\) are 4x4 matrices associated with the current cell and its neighbours respectively, \( (q) \) is the solution variable vector \((p,u,v,T)^T\) as defined in equation (3.14), \( (B) \) is the source vector arising from the last time level or the last nonlinear iteration. The notations follow the ones defined for a difference molecule which is shown in Figure 3.4
The solid lines in Figure 3.4 are grid lines forming control volumes. Dependent variables as well as coefficients in equation (3.49) are stored at the intersections of the dashed lines in a collocated cell centred manner.

The matrix form of equation (3.49) for all control volumes is

\[
[A](q) = (B)
\]

\[
3.50
\]

\[
[A] = \begin{bmatrix}
A_{1,1}^{i,j} & A_{2,1}^{i,j} & \cdots & A_{1,2}^{i,j} \\
A_{i,j-1}^{i,j} & A_{i-1,j}^{i,j} & A_{i,j+1}^{i,j} & \cdots & A_{i,1}^{i,j} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
A_{i,j}^{m,n} & A_{i,j}^{m,n} & \cdots & A_{i,j}^{m,n} & A_{i,j}^{m,n}
\end{bmatrix}
\]

\[
q = \begin{bmatrix}
q_{1,1} \\
q_{2,1} \\
\vdots \\
q_{i,j} \\
q_{i+1,j} \\
\vdots \\
q_{m,n}
\end{bmatrix}
\]

\[
3.51
\]
Equation (3.50) represents a symmetric block matrix problem with either 5 or 9 banded diagonals. A linear block matrix solver is needed to resolve this large but sparse equation system.

3.6 Linear Matrix Solver: Gauss-Seidel Point Iteration

There are a large number of algorithms available in the literature to solve an equation system like (3.50). The direct inversion algorithm, i.e. Gaussian Elimination method, is very costly especially for fine meshes, so alternative algorithms are usually employed. Two large families are the factorisation methods and the non-factored relaxation methods. The factorisation method splits the large stencil matrix into a product of several simultaneous matrices for which efficient direct inversion procedures can be designed. Among these, the lower and upper diagonal banded matrix decomposition, e.g. the Strongly Implicit Procedure (SIP) method of Stone (Stone 1968), the one dimensional tridiagonal matrix decomposition, e.g. the Approximate Factorisation (AF) method of Beam-Warming (Beam, et al. 1978) and the Alternating-Direction Implicit (ADI) method of Briley-McDonald (Briley, et al. 1977), are popular choices. An error, often referred to as factorization error, is inherent in any factorisation algorithm due to the non-replication of the original matrix upon the multiplication of the decomposed matrix factors. This factorization error could affect the convergence properties, particularly as the time step becomes large.

Non-factored algorithms are based on iterative procedures, such as the classical Jacobi, Gauss-Seidel point or line (SLOR) procedures. The relaxation-type schemes would introduce a preferred direction by the choice of sweep direction. Therefore, they could suffer an inefficiency problem when the preferred direction is not aligned with the one where implicitness is most important. Usually such direction in the flow is unknown during the evolution and could change constantly, hence the optimum sweep direction is a difficult choice. For example, the symmetric line Gauss-Seidel procedure (SLGS) (MacCormack 1985) which is based on back-and-forth symmetric sweeps in conjunction with an upwind discretisation is highly efficient for predominantly streamwise supersonic flows, but the asymptotic convergence rate is increasingly degraded as the amount of
upstream influence decreases such as in the presence of large regions of streamwise elliptic flow (Edwards, et al. 1991). Recently, there has been a resurgence of interest in exploring efficient unfactored iterative solvers, such as the Conjugate Gradient method (CG).

In this work, a Point Gauss-Seidel (G-S) iterative method has been employed to solve the linear equation system due to its low computer resource requirement. This is a successive approximation procedure which leads progressively towards the solutions from an initial guess. Instead of solving all the cells simultaneously across the whole solution domain, the G-S point method decouples the cells and only solves for the unknowns on one cell at a time. Any other cells needed are treated as known and approximated by the solutions at the previous G-S iteration. Because it treats each cell individually, this method belongs to the category of point block implicit schemes.

The G-S method solves a scalar matrix problem \([A](q) = (B)\) by first decomposing the problem to

\[
[[D] + [L] + [U]](q) = (B)
\]

where \([D]\), \([L]\) and \([U]\) are the matrices containing the diagonal, lower and upper entries of matrix \([A]\) respectively. The G-S iteration procedure proceeds as

\[
[[D] + [L]](q)^{k+1} = (B) - [U](q)^k
\]

where \(k\) is G-S iteration index, so equation (3.53) can be expressed for equation (3.50) as

\[
(q)^{k+1} - A_{i,j}^{k+1} = (B)_{i,j} - A_{i,j-1}^{k+1} - A_{i,j+1}^{k+1} - A_{i-1,j}^{k+1}
\]
Unified Block Implicit Algorithm

In a block matrix procedure, the scalar elements corresponding to an entry of the matrix are replaced by a block matrix. Any operations on the scalar element are now applied to the block matrix as a whole.

It is noted that in equation (3.54) the only remaining implicit terms are those corresponding to the diagonal coefficient matrix [D]. During the inversion of this diagonal block matrix [D] problem, the cells are updated in sequence one after another. Each time all unknown variables on one cell are solved simultaneously by inverting a (4x4) block matrix. The variables whose coefficients are the entries in either matrix [L] or [U] are always evaluated using their latest solution values. One G-S iteration cycle is completed when all cells are updated. In this work, only a few cycles of G-S iteration are performed rather than solving to convergence.

Because an indirect addressing method is adopted in this work for the cell index system (to be discussed in detail in Chapter 4), the ordering of the G-S sweep in each cycle is chosen to be either from cell number 1 counting up to cell number N (assuming total number of cells is N), or from cell number N counting down in sequence to cell number 1. The choice of which order is randomly decided by the computer for each cycle. A demonstration is given in Figure 3.5.

![Figure 3.5 Ordering of the sweep](image)

where the numbering represents the ordering of the G-S sweep. The big advantage of the G-S solver is the low cost. Its main disadvantage is that it is not guaranteed to be stable...
Unified Block Implicit Algorithm

unless matrix $[A]$ is diagonally dominant. For high Reynolds number flows, a central
spatial difference is unlikely to lead to a diagonal dominant stencil; the burden of
providing the required dominance is thus laid on the control of time step size. This
effectively introduces a stability condition to the implicit scheme. As mentioned in
subsection 3.4.3, the implicit implementation of the mixed 2$^{nd}$ and 4$^{th}$ order artificial
dissipation could improve the diagonal dominance of the coefficient stencil and therefore
could relax the time step constraint.

3.7 Multigrid Acceleration Technique

The multigrid technique is a well-established method to accelerate the convergence rate of
an iterative linear or nonlinear matrix solver. It has proven to be effective for elliptic flow
calculations and remains an attractive approach for hyperbolic flow calculations. Iterative
matrix solvers are in general efficient at smoothing out the high frequency error on a given
grid, but they are often inefficient at removing the low frequency errors. The multigrid
method helps to overcome this problem by transferring the low-frequency error modes
onto a succession of coarser grids where they become high frequency errors and can be
smoothed out by the iterative solver effectively. Also in hyperbolic flow calculations,
larger time steps could be employed on the coarser meshes, as a result, the errors could be
expelled out of the computation domain quickly. The basic ingredients of a multigrid
scheme are: (i) relaxation, (ii) restriction, and (iii) prolongation.

For the multigrid technique to be effective, the driving scheme has to be a good smoother,
i.e. the high frequency smoothing property of the iterative (relaxation) scheme. By saying
a good smoother, this means that the iterative solver should have good damping
characteristics on high frequency errors. Some high frequency errors are always
introduced at the prolongation process of a multigrid cycle during which coarse grid
corrections are interpolated back to a finer grid. Usually, they cannot be rapidly expelled
out of the computation domain le, so ought to be damped locally, otherwise they will be
carried over to the next multigrid cycle and transformed to alias errors because they are
not resolvable. The alias errors are usually low frequency to the coarse grid, hence cannot
be damped effectively on that grid.
Unified Block Implicit Algorithm

The point block Gauss-Seidel iterative solver has been shown to work well with the multigrid technique (Dick 1991, Sidilkover 1994), because it has a good high-frequency error damping property if optimised. For the central differencing method with blended 2nd and 4th order artificial dissipation, Jameson and Yoon (Jameson, et al. 1986) have shown that the artificial dissipation is a good tool to optimise the shape of the amplification factor of an ADI iterative solver, hence the artificial dissipation is also important to the performance of the multigrid method.

In the present work, the coarser grids in multigrid are created by “standard coarsening”, i.e. coarse grids are created from fine grids by deleting every other line in each coordinate direction. Within a finite volume framework, the coarse grid control volume is composed of four fine grid control volumes (see Figure 3.6). As a result, the number of cells in each coordinate direction has to be $2^n$, with $n$ not less than the number of grid levels to be used by the multigrid.

![FIGURE 3.6 Fine and Coarse grids in Multigrid system](image)

3.7.1 Nonlinear Multigrid Cycle: FAS

A nonlinear Full Approximation Scheme (FAS) is adopted for the present multigrid work. The multigrid procedure is integrated within the Newton nonlinear iteration process. There are a large number of documents in the literature about the FAS Multigrid method.
Unified Block Implicit Algorithm

(Brandt 1984, Sivaloganathan 1987), so only a brief description of the V-cycle method as shown in Figure 3.7 is given here to demonstrate the current implementation.

In this multigrid cycle, data transfer is made from each grid to the next coarser grid after one or several Newton iterative calculations of the nonlinear equation system within a time step, the so called pre-smoothing. After reaching the coarsest grid, the corrections are then successively interpolated back from each grid to the next finer grid with one or a couple of intermediate Newton nonlinear iterations, i.e. post-smoothing, to smooth out prolongation errors. If one iteration of pre-smoothing and post-smoothing is used, the total computational effort in one multigrid cycle in comparison with a single grid is

\[ 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \ldots + \frac{1}{8} + \frac{1}{4} + \frac{1}{2} + 1 \leq \frac{4}{3} \times 2 \]

plus the additional work of restriction and prolongation.

In this multigrid method, the mesh is coarsened by removing every other mesh line in the current grid level, i.e. doubling the mesh spacing. Values of flow variables are transferred to the coarser grid by the rule

\[ w_{2h}^{(0)} = \sum S_h w_h / S_{2h} \]

where \( w \) represents a variable, its subscript denotes the value of mesh spacing parameter, \( S \) is the cell area, and the sum is
over the four cells on the fine grid composing each cell on the coarser grid. This rule
ensures the conservation of mass, momentum and energy. A forcing function on a coarser
grid is then defined as

\[ P_{2h} = \sum R_h - R_{2h}^t(w_{2h}^{(0)}) \]  
3.56

The residual used to drive the correction on the coarser grid is given by

\[ \bar{R}_{2h} = R_{2h}^t(w_{2h}) + P_{2h} = \sum R_h \]  
3.57

This implies that effectively it is the residual on the fine grid that drives the solution on the
next-coarser grid, so that the correction to the current solution on the fine grid is obtained
from the solution on the next-coarser grid. This process is continued in a similar manner
on a sequence of coarser grids, corrections having been computed on every grid level.
After the coarsest grid level has been visited, the corrections are prolonged back to
successively finer grids until the finest grid is reached, then the solution is corrected by

\[ w_h^c = w_h + l_{2h}^h(w_{2h}^c - w_{2h}^{(0)}) \]  
3.58

where \( w_{nh}^c \) is the solution on grid \( nh \) after correction and \( l_{2h}^h \) is an interpolation operator
from grid \( 2h \) to grid \( h \). This cycle is repeated until solution convergence.

### 3.7.2 Restriction and Prolongation Method

The restriction of the solution from fine grid to coarse grid is defined as the arithmetic
average of the variable values of the four fine cells which construct the coarse cell, as
shown in Figure 3.6, i.e.

\[ \phi_1 = \frac{1}{4}(\phi_1 + \phi_2 + \phi_3 + \phi_4) \]  
3.59
Unified Block Implicit Algorithm

The residual on the coarse cell I is collected from those on the fine cells 1 - 4 as

\[ R_1 = R_1 + R_2 + R_3 + R_4 \]  

The coarse-to-fine prolongation of the correction is defined by first order bilinear interpolation in computational space where the grid spacing is taken to be uniform:

\[ \phi_4 = \frac{1}{16} \cdot \{9\phi_I + 3\phi_{II} + \phi_{III} + 3\phi_{IV}\} \]

\[ \phi_5 = \frac{1}{16} \cdot \{3\phi_I + \phi_{II} + 3\phi_{III} + 9\phi_{IV}\} \]

\[ \phi_6 = \frac{1}{16} \cdot \{\phi_I + 3\phi_{II} + 9\phi_{III} + 3\phi_{IV}\} \]

\[ \phi_7 = \frac{1}{16} \cdot \{3\phi_I + 9\phi_{II} + 3\phi_{III} + \phi_{IV}\} \]

3.7.3 Boundary Condition on Coarse Grid

Boundary conditions are imposed on the coarser grids in a similar manner as those on the finest grid. Boundary variables on any grid level are dynamically updated as opposed to frozen during the calculation on the coarse grid. Restriction and prolongation operations are also applied to boundary cells, but with appropriate modifications. Boundary cells on the coarse grid can be involved in the interpolation of the internal cell adjacent to the boundary on the finer grid. In Figure 3.8, boundary cell IV of the coarse grid is restricted by

\[ \phi_{IV} = \frac{1}{2}(\phi_I + \phi_2) \]
Unified Block Implicit Algorithm

boundary cell 3 of fine grid is interpolated by

\[ \phi_3 = \frac{1}{4}(3\phi_{III} + \phi_{IV}) \]  

3.66

and internal cell 5 of fine grid is bilinearly interpolated by

\[ \phi_3 = \frac{1}{8}(\phi_1 + 3\phi_{II} + 3\phi_{III} + \phi_{IV}) \]  

3.67

which are all different from the operations on the interior cells as shown in Figure 3.6.

![Fine and coarse grid in Multigrid: boundary cells](image)

**FIGURE 3.8** Fine and coarse grid in Multigrid: boundary cells

3.7.4 Artificial Dissipation

The success of a multigrid method is critically dependent on the shape of the amplification factor of the differencing scheme. As illustrated by Jameson & Yoon (Jameson, et al. 1986), the amplification factor of the current scheme can be studied by considering a model problem

\[ \phi_t + \phi_x + \phi_y + \varepsilon (\Delta x^3 \phi_{xxxx} + \Delta y^3 \phi_{yyyy}) = 0 \]  

3.68
where \( \varepsilon^{(4)} \) is the artificial viscosity. This method is used to analyse the differencing scheme in the present work.

The author has independently performed this analysis for the differencing scheme adopted in this work. With Euler backward time integration and central spatial differencing, the model equation transforms to:

\[
\frac{1}{\Delta t} (\varphi_{i,j} - \varphi_{i,j}^{n+1}) + \frac{1}{2\Delta x} (\varphi_{i+1,j} - \varphi_{i-1,j}) + \frac{1}{2\Delta y} (\varphi_{i,j+1} - \varphi_{i,j-1}) \\
+ \varepsilon^{(4)} \left\{ \frac{1}{\Delta x} (\varphi_{i+2,j} - 4\varphi_{i+1,j} + 6\varphi_{i,j} - 4\varphi_{i-1,j}) + \varphi_{i-2,j} \right\} = 0
\]

3.69

where superscript \( n+1 \) for the time level is dropped for convenience. Using the Von Neumann stability analysis, the error in the solution is written in the form of the discrete Fourier decomposition as

\[
\varepsilon_{i,j}^{n+1} = \sum_{\kappa_x\kappa_y} A^{n+1} e^{i\kappa_x \cdot i\Delta x} e^{i\kappa_y \cdot j\Delta y}
\]

3.70

where \( \kappa_x \) and \( \kappa_y \) are wave numbers in the x and y directions and \( I = \sqrt{-1} \), or

\[
\varepsilon_{i,j}^{n+1} = \sum_{\phi_x\phi_y} A^{n+1} e^{i\phi_x \cdot i\Delta x} e^{i\phi_y \cdot j\Delta y}
\]

3.71

with \( \phi_x \) and \( \phi_y \) phase angles ranging from \(-\pi\) to \(\pi\). Substituting (3.71) into (3.69) and then dividing the equation by \( A^{n+1} e^{i\phi_x \cdot i\Delta x} e^{i\phi_y \cdot j\Delta y} \) gives (study one Fourier component only, because all components in the Fourier decomposition behave the same in terms of error progress):
Unified Block Implicit Algorithm

\[
\frac{A^n}{A^{n+1}} = 1 + I\left(\frac{\Delta t}{\Delta x} \cdot \sin \phi_x + \frac{\Delta t}{\Delta y} \cdot \sin \phi_y\right) \\
+ \varepsilon^{(4)} \cdot \frac{\Delta t}{\Delta x} (2 \cos 2\phi_x - 8 \cos \phi_x + 6) \\
+ \varepsilon^{(4)} \cdot \frac{\Delta t}{\Delta y} (2 \cos 2\phi_y - 8 \cos \phi_y + 6)
\]

The amplification factor \( G \) is defined as

\[
A_{i,j}^{n+1} = G \cdot A_{i,j}^n
\]

so the amplification factor of the present differencing scheme is

\[
\frac{1}{G} = 1 + I(\sigma_x \sin \phi_x + \sigma_y \sin \phi_y) + 16\varepsilon^{(4)}\left[\sigma_x \left(\sin \frac{\phi_x}{2}\right)^4 + \sigma_y \left(\sin \frac{\phi_y}{2}\right)^4\right]
\]

where \( \sigma_x = \frac{\Delta t}{\Delta x} \) and \( \sigma_y = \frac{\Delta t}{\Delta y} \).

Considering \( \sigma_x = 1 \) and \( \sigma_y = 1 \), when \( \varepsilon^{(4)} = 0 \), i.e. without artificial dissipation,

\[
|G| = \frac{1}{\sqrt{1 + (\sigma_x \sin \phi_x + \sigma_y \sin \phi_y)^2}}
\]

the shape of \( |G| \) is shown in Figure 3.9
A near one amplification factor appear at the high-frequency ends where $\phi_x = \pi$ or $\phi_y = \pi$, this indicates that the damping rate is very poor, since $|G| = 1$, therefore the scheme does not have good damping characteristic for the high frequency components. A multigrid is unable to perform well under these smoother conditions.

However, when artificial dissipation is added to the scheme, i.e. $e^{(4)} \neq 0$,

\[
|G| = \frac{1}{\sqrt{1 + 16 e^{(4)} \left[ \sigma_x \left( \sin \frac{\phi_x}{2} \right)^4 + \sigma_y \left( \sin \frac{\phi_y}{2} \right)^4 \right] + (\sigma_x \sin \phi_x + \sigma_y \sin \phi_y)^2}}
\]

When $e^{(4)} = 0.5$, the distribution of $|G|$ in the $(\phi_x, \phi_y)$ plane becomes the one shown below in Figure 3.10.
The high frequency damping ability is improved significantly. The high frequency range as well as the intermediate frequency range are strongly damped, providing a good smoothing performance for the multigrid method.

When less artificial dissipation is introduced, for instance $\epsilon^{(4)}$ is reduced to 0.1, high frequency damping is found to become less heavy, which is illustrated by the shape of $|G|$ shown in Figure 3.11. This indicates a close relation between the amount of artificial dissipation and the damping effect, and the consequent multigrid performance.
Such a relation can be manifested further if another example which uses even smaller artificial dissipation is studied, with regard to the distribution characteristics of the amplification factor in the phase angle plane. The example of $\varepsilon^{(4)} = 0.05$ is shown in Figure 3.12 for the shape of $|G|$. The magnitude of the amplification factor in the high frequency regime is larger than that of either cases above, therefore high frequency components will be damped less rapidly. However, a large region containing medium to high frequencies still lies in the high damping range of the scheme. Low order damping is only confined to low frequency components, hence reasonable multigrid performance may be sustained, but will certainly not be as good as with larger artificial dissipation.
In summary, without the introduction of artificial dissipation, the point block Gauss-Seidel (G-S) iterative procedure does not have a good damping property for high frequency mode errors. When these artificial dissipation terms are implicitly added and optimised, the high frequency damping property of the G-S solver can be greatly improved. These dissipation terms can change the shape of the amplification factor of G-S solver to make it a favourable driving scheme for successful multigrid acceleration.

On the coarser grids, since the smoothing characteristics are more important than accuracy, Jameson (Jameson 1983a, Jameson 1983b) has found it is desirable to use only a fixed-coefficient second-difference form of the dissipation on coarser grids.

### 3.8 Boundary Conditions

The boundary conditions applied in this unified solution procedure are the same as those imposed in the unified pressure-correction procedure described in the previous chapter except that they are implemented implicitly and solved together with the equations for the interior solution cells. Therefore, the boundary conditions are not reiterated here, detailed descriptions can be found in Chapter 2.
3.9 References


Unified Block Implicit Algorithm


Unified Block Implicit Algorithm


Chapter 4

General Strategy of Coupling Methodology
4.1 Preamble

One major objective of this thesis is to study the novel hybrid model concept proposed in Chapter 1 which couples (for example) one dimensional and two dimensional CFD models (1D&2D models) in simulating a flow in which some regions are characterised by strong two dimensional flow while others can be well approximated by one dimensional flow. In this research, this idea has been developed and investigated within the framework of the two unified solution procedures described in Chapters 2 and 3. In this chapter, the general strategy of hybrid model coupling is explained. The specific implementation in the two solution procedures is discussed individually in Chapters 5 and 6.

As mentioned in Chapter 1, when a combination of different dimensional CFD models is applied in one calculation, a hybrid dimensional multiblock grid system is used. The coupled hybrid dimensional CFD model is thus a simultaneous computational procedure which solves the governing flow equations on a solution domain composed of mixed dimensional multiple zones. This procedure is to be derived from an existing solution procedure applicable to single-zone calculations. Modifications must be made to make the original procedure operational in the new multiblock environment, and they should be carried out in such a way that the important properties of the original procedure are not affected, for instance the conservation property. The central issues in the 1D and 2D coupled model development are thus how the solution procedure handles the hybrid dimensional multi-blocks, and in particular, how the solution information is exchanged between the multiple zones during the calculation.

An adequate treatment is needed at the zonal interface for the zonal communication. It should not affect the stability and accuracy of the global algorithm, as well as the convergence performance. Otherwise the benefit from the coupled modelling approach could be undermined. A special data structure is required which allows easy access to the multi-block data and introduces little ‘overhead’.

Because the 1D and 2D coupled model will be integrated into both semi-implicit and implicit unified solution procedures, and it will be applied to transonic flows, some additional issues require consideration. One is that the coupling treatment should


**General Strategy of Coupling Methodology**

Guarantee that, if the calculation converges, it converges to a weak solution of the governing equations. The second is that both explicit and implicit formulations of the interface coupling treatment are required.

In the following, general issues raised during the development of 1D and 2D coupling are addressed. In addition, a general description of the methodology needed to tackle zonal interface coupling is given. These issues are common to the more general hybrid model - e.g. a 1D&2D&3D coupled CFD model, and the coupling methodology developed here has allowed for possible extension to a general hybrid model.

### 4.2 Multiblock Grid System

As explained in Chapter 1, the multiblock hybrid dimensional grid system used for a 1D/2D coupled model belongs to the discontinuous patched grid branch of the conventional multiblock approach. In any multiblock system, the computational domain is decomposed into a finite number of contiguous subdomains (zones or blocks), with two neighbouring zones sharing a common interface. Each block can be visualised topologically in 2D as having four boundary faces. Each of the boundaries, or a portion of each, can denote a real physical boundary or an internal boundary. The distinction between the physical boundary and the internal boundary is that the former is on the computational boundary while the latter lies inside the computational domain; the latter is described by two sets of boundary grid points, each belonging to a different block. Each zone is meshed separately using a structured grid which is topologically Cartesian, and either two-dimensional or one-dimensional.

An example of a 1D and 2D hybrid grid system is illustrated in Figure 4.1. In this work, any two abutting 2D grid blocks are continuously patched, i.e. the grid lines are continuous across their interface, e.g. interface \( m_1-m_1' \). On the other hand, abutting 1D and 2D blocks create a discontinuous interface, e.g. the \( m_2-m_2', m_3-m_3', m_4-m_4' \). As such, the coupling methods for 2D/2D neighbouring blocks and 1D/2D ones cannot be the same, hence they are developed separately. To simplify the presentation of the coupling method, only a multiblock system with two zones is employed throughout the thesis, be it a 2D/2D zone or a 1D/2D zone. However, the coupling method developed is applicable to
General Strategy of Coupling Methodology

any number of multiple zones. In addition the grid is always shown in the computational space of the curvilinear coordinate system unless this is specifically notified.

![Grid diagram](image)

**FIGURE 4.1** An example of hybrid one dimensional and two dimensional grids

4.3 Data Management

In the present multiblock system, additional book-keeping is needed to store properties at each block boundary face. Also a data structure needs to be adequately designed which would allow efficient cell indexing and solution variable storage to be managed in an economical and flexible way.

4.3.1 Cell Index System

In a single block structured grid, grid lines are continuous and coincident with coordinate lines in the curvilinear coordinate system, grid points can thus be denoted by the coordinates. As a result, the curvilinear coordinate index \((i, j)\) is always used as the designation system for the two dimensional grid.

However, in the present grid system, in general grid lines can be discontinuous from one block to another, i.e. no global coordinate system exists, it is hence impossible to have one \((i, j)\) index system for the entire computation domain. One straightforward alternative indexing method would be to use a dedicated \((i, j)\) designation system in each block. However, it is rather inconvenient from a global point of view to use a separate \((i, j)\) index
General Strategy of Coupling Methodology

system for each block. Hence, another alternative method using the indirect addressing concept is adopted. In an indirect addressing index system, a grid cell (an element) is assigned a sequential identification number (ielm), its location in the grid system is indicated by its cell connectivity relations with other grid cells (this is in fact very similar to the approach adopted for unstructured meshes).

The generation of the indirect addressing system is, however, accomplished through the generation of an individual (i, j) index system in each block. During grid generation, a structured grid is generated in each block and temporarily denoted by a local (i, j) index system. Later all of these are re-ordered globally and allocated a new global cell identification number (ielm) - the indirect addressing system is formed. The relation between the global 1D index number ‘ielm’ with the local 2D index $(i, j)_{iblock}$ in block ‘iblock’ for an interior cell is as follows:

$$
\begin{align}
\text{ielm} &= \{(i - 1) + (NI_{iblock} - 1) \cdot (j - 2)\}_{iblock} + \sum_{k=1}^{iblock-1} (NI_k - 1) \cdot (NJ_k - 1)
\end{align}
$$

4.1

where ‘iblock’ is the sequential number of the block (assigned during the block by block grid generation). $NI_k$ and $NJ_k$ are the dimensions of the structured grid in the block whose ID number is $k$.

The advantages of this indirect addressing index system is the optimum use of memory storage. In addition, very simple modification is required to the code when migrating from a single block to a multiblock system. In this index system, the most important element is the cell connectivity table which gives the information of cell location in the computational space. This table records the neighbouring cells of each grid cell, the connectivity table is not globally oriented, there is a dedicated table for each block.
**General Strategy of Coupling Methodology**

A two dimensional array, `elm2elm(NB, ID)`, in the code is used for the cell connectivity table. The second index of this array corresponds to the ID number of the cell, while the first index denotes the relative location of the neighbouring cell to this cell. A west-east-south-north order (1, 2, 3, 4) is used to describe the relative location as shown in Figure 4.2. For example, cell P has index number `ijk`, its neighbours W, E, S and N have ID numbers `ijkw`, `ijke`, `ijks` and `ijkn`. Therefore, the values of the connectivity table for cell P are:

\[
\begin{align*}
\text{elm2elm}(1,ijk) &= ijkw \\
\text{elm2elm}(2,ijk) &= ijke \\
\text{elm2elm}(3,ijk) &= ijks \\
\text{elm2elm}(4,ijk) &= ijkn
\end{align*}
\]

In the connectivity table for a boundary cell, the values for the non-existing neighbours are set to zero.
4.3.2 Solution Variable Storage

The storage of solution variable data is managed using a vector arrangement method by which all variables are allocated as vectors in a single one dimensional array. This long ID array is called the master array and the vector allocation can be either dynamic or non-dynamic. In the present multiblock domain, the solution variable vectors of all blocks are sequentially stored in the 1D master array, block by block. The master array in this system looks like: \( F(u^1(1), u^1(2), ..., u^1(n_1), u^2(n_1+1), ..., u^2(n_1+n_2), ..., v^1(1), v^1(2), ..., v^1(n_1), v^2(n_1+1), ..., v^2(n_1+n_2), ...) \), where the superscript denotes the serial number of the block, and the number in the bracket denotes the local ID number \( "ijk" \) of the cell.

4.3.3 Block Connectivity Table

Due to the presence of internal boundaries, another table is needed to store the cells abutting the interface, referred to as a block connectivity table. During grid generation, each internal interface is assigned an identification number (ipair), a pair of block connectivity tables (a 2D array in the code) are then created to record the interior cells immediately next to the interface. In addition, the number of such cells in the 2D block is also recorded.

For example, in the model shown in Figure 4.2, because there is only one internal boundary created by two 2D/2D patched blocks, the identification number for this sole interface is one (i.e. ipair = 1). The data stored in block connectivity tables are

\[
\begin{align*}
elmbutt_1 (1, 1) &= p_1^1 \\
elmbutt_2 (1, 1) &= p_1^2 \\
elmbutt_1 (5, 1) &= p_5^1 \\
elmbutt_2 (5, 1) &= p_5^2
\end{align*}
\]

where “elmbutt_1” and “elmbutt_2” are the names of the block connectivity tables for the two blocks respectively. Their first index is the sequence number of the abutting cell, while the second index is the interface ID number. The names on the right hand side of the
expressions correspond to the cells indicated in Figure 4.2 and represent the index number of those cells. The superscript of p is the block number and the subscript is the sequence number of the abutting cell.

The access to the cell connectivity table, block connectivity tables etc. during the calculation account for the major extra computing time overhead associated with the current 1D/2D and 2D/2D system.

4.4 Coupling Method for 2D/2D Coupled Models

The decomposition of a solution domain into zones introduces internal boundaries inside the solution domain, therefore each block face can have two types of segments associated with it - physical boundary segments and internal boundary segments. In this work, a set of dummy control volumes with zero thickness are created on any type of boundary. Markers are generated during the grid generation process to indicate the boundary cells and their associated boundary type. There is a special boundary type identifying an internal zonal boundary. There are no 'halo' cells extending a block into its neighbours, the communication between blocks is handled through the shell of dummy cells which contain the pertinent information. The coupling scheme via the zonal interface is always coded in separate routines in the program developed during this work. However, in the explicit or implicit interface treatment discussed later in this chapter, the coding practice is different and is explained below.

In order to enable block communication and to handle information transfer among zones, it is necessary to determine what information flows across the segments on each boundary face. The required information on the physical boundary segment is specified at the outset of the problem via boundary conditions. However, the information on the internal boundary segment has to be obtained from the internal flow solution. The zonal interface is the bridge of communication. It should be so treated that the flow information in all blocks can be effectively transferred and their solutions smoothly patched together. This is important for the accuracy, the convergence and the stability of the global scheme. Consequently, the interface effect of the zonal interface to the calculation should be kept to minimum.
**General Strategy of Coupling Methodology**

Although the focus of this work is on the investigation of a hybrid 1D and 2D coupled model, a 2D/2D continuously patched model is also used. The coupling method developed for such a 2D/2D multiblock can provide valuable guidance to the development of the interface treatment for the 1D/2D hybrid model, and hence is first considered.

An example of the grid system for this model is shown in Figure 4.3. The grid lines in two adjacent blocks, block 1 and block 2, match exactly at the common interface m-m' and are continuous across the interface to the n\(^{th}\) order derivative.

![Figure 4.3](image)

**FIGURE 4.3** A 2D and 2D continuously patched grid

For example, grid line \(g_{1}^{1}\) in block 1 and grid line \(g_{1}^{2}\) in block 2 join continuously at point p on the vertical line m-m'. The segment on interface m-m' should appear to be no different from an ordinary interior cell face. These characteristics are utilised to develop the procedure for transferring information from block to block.

The coupling procedure can be implemented explicitly or implicitly on the interface.

In the explicit approach, the interface is identified as an explicit boundary, and the coupling is achieved by applying a fixed boundary condition to the interface. Any flow quantities on the interface are frozen during the solution of interior cells, and are updated after the internal domain is solved. The process of calculating flow variables on the
**General Strategy of Coupling Methodology**

interface is the process of applying the ‘explicit boundary condition’, and this is where solution information is exchanged among blocks.

In an implicit approach, the interface is treated as if it were a collection of interior cell faces. The zonal common line m-m' is treated in exactly the same way as other vertical interior lines, such as line n-n'. Consequently, the grid lines from two blocks, e.g. gl¹ and gl², are united into one continuous grid line which is in the virtual single block covering block 1 and block 2. When any segment on the interface m-m' is referred to in the discretisation process, the required quantities on this segment are evaluated using the same method as for the interior cell faces. Consider control volumes cv¹ and cv² in Figure 4.4.

![Diagram](image)

**FIGURE 4.4** Control volumes in the 2D patched grid

These are in block1 and block2 respectively and have a shared cell face ‘cf’ which is a segment on interface m-m’. The variable on cf is evaluated by averaging the values at the centres of the cells on both side of cf, similar to the treatment of wf₁ or ef₂. This retains the order of accuracy and conservative property of the base scheme. The interface boundary becomes essentially ‘invisible’ and a single solution domain is recovered. As a result, the multi-domain computation is able to produce identical results to the one-domain computation of the same problem at every iteration. This property ensures that the
**General Strategy of Coupling Methodology**

multi-domain calculation is as efficient and robust as the one-domain calculation if such one-domain can be constructed.

### 4.5 Coupling Method for 1D and 2D Coupled Model

The grid system for a hybrid 1D/2D coupled model is shown in Figure 4.5.

![Figure 4.5 A hybrid 1D&2D coupled grid](image)

The interface of the two blocks m-m’ is the east cell face of control volume $p^1$ in block 1. In block 2, interface m-m’ is split into a number of segments - cf$_i$, etc. each of which is the west cell face of a control volume $p^2$. The cell in the 1D block has multiple neighbours, while a group of cells in 2D block have one neighbour in common. The existence of this multiple cell neighbour feature presents the biggest challenge to the evaluation of flow quantities at the interface.

Since a conservative finite volume discretisation method is adopted in this work, a variety of flow quantities at cell faces require to be evaluated during the formulation of the discretised equations. The estimations of these flow quantities on the interface or interface segments are obtained based on the interior flow solution in block1 and block2, through which block communication and flow coupling are enforced.
General Strategy of Coupling Methodology

4.5.1 Conservative vs Non-conservative Interface Treatment

The coupling treatment at an interface can lead to either a conservative or nonconservative global scheme if the interior base scheme is conservative, as explained in Chapter I. This aspect is further considered here.

Consider a 1D/2D coupled model in the Cartesian system shown in Figure 4.6.

![Figure 4.6 A 1D&2D coupled grid](image)

The conservative governing equations are represented by a general form:

$$\frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$  \hspace{1cm} 4.2

for steady-state flow. Finite volume discretisation of this equation leads to:

$$\Delta y \cdot [E_{ef} - E_{wf}] + \Delta x \cdot [F_{nf} - F_{sf}] = 0$$  \hspace{1cm} 4.3

When the interior numerical scheme is a conservative one, any flux on the cell face in the above equation is evaluated consistently in all cells where this flux is referred and, when the equations for all control volumes in the computation domain are summed, flux
General Strategy of Coupling Methodology

cancellation occurs; as a result, the sum equals the flux across the physical boundaries of
the computation domain plus the flux across the internal boundaries at block interfaces,
i.e.

\[
\text{Sum} = \sum_{j^1=2^1}^{n_j} (E_{ef(n^1,j^1)} - E_{wf(2^1,j^1)}) \Delta y^1 + \sum_{i=2^1}^{n_i} (F_{nf(i^1,n^1)} - F_{sf(i^1,2^1)}) \Delta x^1
\]

\[+ (E_{ef(n^1,2^2)} - E_{wf(2^1,2^2)}) \Delta y^2 + \sum_{i=2^1}^{n_i} (F_{nf(i^1,2^2)} - F_{sf(i^1,2^2)}) \Delta x^2\]

\[= \sum_{j^1=2^1}^{n_j} (-E_{wf(2^1,j^1)}) \Delta y^1 + E_{ef(n^1,2^2)} \Delta y^2 + \sum_{i=2^1}^{n_i} (F_{nf(i^1,n^1)} \Delta x^1
\]

\[+ \sum_{i=2^1}^{n_i} (F_{nf(i^1,2^2)}) \Delta x^2 + \sum_{i=2^1}^{n_i} (-F_{sf(i^1,2^2)}) \Delta x^2\]

\[+ \left[ \Delta y^2 (-E_{wf(2^1,2^2)}) + \sum_{j^1=2^1}^{n_j} (E_{ef(n^1,j^1)}) \Delta y^1 \right]\]

the superscript on $\Delta x$ or $\Delta y$ denotes the block, a uniform mesh is assumed in each block,
the notation in cell index $(i, j)$ has followed the convention shown in Figure 4.6.

The terms in the square bracket of equation (4.4) represent the total flux across the internal
boundary. The other terms represent the flux sum across the real boundaries of the domain.
If the scheme is to be a global conservative scheme, the numerical discretised equation
should retain the property that the change of a conserved quantity within an arbitrary
volume is only determined by the flux of this quantity across the boundary surface
enclosing this volume (in the absence of internal sources). For equation (4.4), that is to
say, there should not be any internal numerical flux accumulation at the internal interfaces.
Therefore, the sum in the square bracket should be zero for a global conservative scheme.

4.5.1.1 Non-Conservative Interpolation Method

When flow conditions on the interface are evaluated via pure interface interpolation
methods in the interface coupling scheme, a non-conservative scheme usually results.
General Strategy of Coupling Methodology

Generally, solution variables on the interface are interpolated or extrapolated directly from the solutions of the interior domain in this practice, the flux or other flow quantities on the interface are subsequently computed. The conservation of some of the flux could probably be achieved by the implementation of special conservation conditions, however, it is normally not possible to restore the conservation of all the flux at the interface.

As an example for illustration, consider the x direction incompressible Euler momentum equation

\[
\frac{\partial u}{\partial t} + \frac{\partial (uu)}{\partial x} + \frac{\partial (vu)}{\partial y} = -\left(\frac{\partial p}{\partial x}\right)
\]

4.5

On the grid shown in Figure 4.5, the equation for control volume \(p_i^2\) after discretisation becomes:

\[
\frac{(u_2^i - u_0^i)}{\Delta t} + \frac{[(uu)_{cf} - (uu)_{cf}]}{\Delta x} + \frac{[(vu)_{nf} - (vu)_{sf}]}{\Delta y} = -\left[\frac{[P_{ef} - P_{cf}]}{\Delta x}\right]
\]

4.6

If solution variable \(u\) on interface segment \(cf_i\) is evaluated by, for example, linear interpolation

\[
u_{ef_i} = \frac{1}{2}(u_2^i + u_{1i})
\]

4.7

or linear extrapolation

\[
u_{ef_i} = 3u_2^i - 2u_{1i}
\]

4.8

the flux \((uu)\) on this segment in equation (4.6) will be subsequently calculated using this interpolated value. The total flux across the interface from the 2D block to the 1D block is the sum of the flux across each of the interface segments, i.e. \(\Sigma(uu)_{ef_i}\).
On the other hand, the discretised equation for control volume \( p^l \) takes the form of

\[
\frac{(u^l_p - u^0_p)}{\Delta t} + \frac{[(u u)_{m-m'} - (uu)_{wf}]}{\Delta x} + \frac{[(vu)_{nf} - (vu)_{sf}]}{\Delta y} = \frac{[p_{m-m'} - p_{wf}]}{\Delta x} \tag{4.9}
\]

If variable linear interpolation scheme is again adopted

\[
u_{m-m'} = \frac{1}{2} \left( u_{p^l} \pm \frac{1}{N} \sum_{i=1}^{N} u_{p^l_i} \right) \tag{4.10}
\]

or

\[
u_{m-m'} = 3u_{p^l} - 2u_{p^l_{w'}} \tag{4.11}
\]

and the flux \((uu)\) on the interface \(m-m'\) in equation (4.9) is computed accordingly. In most cases, flux \((uu)_{m-m'}\) which is the total flux across the interface from 1D block to 2D block would not equal to \(\Sigma (uu)_{cfl}\).

Although a conserved interface cannot be guaranteed in this interface coupling approach, high order of variable continuity can be achieved and consequently high order global numerical scheme can be retained.

### 4.5.1.2 Conservative Interface Method

It has been demonstrated that to achieve conservation at an interface, the interface should be evaluated in such a way that the flux through an interface entering one block equals the total flux through the same interface leaving the abutting block. This will give rise to a conservative scheme. The general approach adopted in this practice is that, for the block on one side of the interface, the variables on the interface are interpolated and the required interface fluxes are derived. For the block on the other side of the interface, the required flux is then calculated by conserving the flux at the interface rather than deriving it from local variables. For the variable value required on the interface, it has in general to be
General Strategy of Coupling Methodology

deduced from the relevant conserved flux. In this interface treatment approach, high order of variable continuity across the interface is often difficult to achieve.

As an example, consider the equation (4.5) again. When a conservative interface treatment is applied, solution variables \( u \) on the interface segment on the 2D block side are interpolated as before, and the \( (uu)_{c1} \) fluxes in equation (4.6) are evaluated by the same method (equation (4.7) and (4.8)). However, the \( (uu)_{m-m'} \) flux on interface \( m-m' \) in 1D block side is computed in a different way. Instead of computing it from the dependent variables (via equation (4.10) or (4.11)), this flux is computed by adding up all the evaluated flux \( (uu)_{c1} \) through the interface segments on the opposite side, i.e. \( (uu)_{m-m'} = \Sigma (uu)_{c1} \). The square bracket in equation (4.4) is then zero, so global conservation of the numerical scheme is achieved.

With the conservative interface, a flow discontinuity, such as a shock, can assume the right strength and location in the simulation and move freely across the zonal interface during the calculation, a property which would not be ensured if the interface fluxes were not conserved. This guarantee is significant to the potential application to IC engine flows where shocks and nonlinear pressure waves constantly propagate from the manifold into the combustion chamber or vice versa. Hence, in the present work, only the conservative interface treatment is considered for transferring information between the multiple blocks.

4.5.2 Explicit vs Implicit Interface Treatment

Zonal interfaces in hybrid coupled models can be treated either explicitly or implicitly. In an implicit treatment, the interface condition is updated at the same time as the interior solution; in an explicit treatment, the interface conditions are frozen during the solution for the interior domain and updated only after the interior solution is obtained. For an explicit based scheme, an explicit interface treatment can be readily applied without having any adverse effect on the properties of the base algorithm. However, for an implicit based scheme, a simple explicit interface treatment will slow down the convergence rate. Although the implicit interface treatment is more complicated, it is worth investigation.
4.5.2.1 Implicit Interface Method

With implicit zonal coupling, the calculation is synchronized so that the interface is updated simultaneously with the interior domain. In the semi-discretised equation, any flow variables on the interface are evaluated from the implicit variables stored in the interior cells. They are then accumulated into the implicit part of the algebraic equations. For the 1D/2D coupled model in Figure 4.5, the discretised equations for the cells abutting the interface in the 2D block, e.g. $p_1^2$, can be formulated in exactly the same way as those for the interior cells, with the assistance of a 'base cell concept' which will be described later. For the cell next to the interface in the 1D block, however, a very different approach is used to derive its algebraic equation. In order to ensure conservation at the interface so as to allow free movement of flow discontinuity in the solution across the interface, any flow quantities on the interface m-m' are evaluated according to the conservation law. With such an approach, a special matrix solver procedure is required to be designed to invert the resulting multi-domain linear equation system.

In general, an implicit interface in a 1D/2D hybrid model involves the following treatment:

(1) flux or variable at the block interface is expressed implicitly using unknown variables in interior cells, they are then incorporated in the semi-discretised equations for the relevant cells, and appended to the implicit part of the algebraic equation.

(2) all of the blocks are solved simultaneously, variables at the interface are updated at the same time with those in the interior domain.

(3) global solution procedure is iterated to convergence.

A fully implicit and conservative (if applicable) interface treatment can make an implicit solution procedure very robust and efficient globally. In this work, this kind of treatment is adopted.

4.5.2.2 Explicit Interface Method

Analogous to the procedure defined above for the implicit method, an explicit interface treatment will involve the following:

---

119
General Strategy of Coupling Methodology

(1) interface variables are evaluated from the existing flow solution, and other flow quantities are subsequently calculated at the interface.

(2) the governing equations are solved within each block using the frozen flow conditions on the interface.

(3) variables and other flow quantities at the interfaces are updated from newly available interior solutions (lagged update).

(4) this procedure is iterated to convergence.

In the present work, this explicit interface treatment has been implemented in the semi-implicit pressure correction procedure to enable a comparison against the implicit interface method. Special flux-conservation approximations have been imposed to conserve these most critical fluxes at the interface.

4.5.3 Base Cell Concept

In all coupling schemes developed in this work, a special concept, termed the ‘Base-Cell-Concept’, was employed for data manipulation. Interpolation for the interface segment lying on the 2D block side is achieved using this technique. In this scheme, the zonal interface in a 1D/2D hybrid dimension grid is put into a standard stencil, as shown in Figure 4.7.

![FIGURE 4.7 Base cell concept: cell stencil at the interface](image_url)
**General Strategy of Coupling Methodology**

Basically, the stencil involves cells from both zones and is constructed from the base cell and secondary cells. The cell in the one-dimensional block immediately next to the interface is defined as the 'base cell' and denoted by $P^1_i$ in Figure 4.7; there is only one base cell in the stencil. The cell in the two-dimensional block on the other side of the interface is defined as the 'secondary cell' and denoted by $P^2_i$. There are many secondary cells in the stencil. The joint interface in this stencil is comprised of two sides, one constitutes the secondary cell's face, e.g. $cf_i, cf_{i-1}, cf_{i+1}$, the other is the base cell's face $m-m'$. The connectivity information between the base cell and the secondary cells as well as their faces are determined and stored before the main solution procedure. In order to evaluate flow variables and flux at secondary cell's face $cf_i$, the base cell $P^1_i$ is split into a **number of fictitious cells** $p'_i, p'_{i-1}, p'_{i+1}$ etc. as shown in Figure 4.7 which continue the secondary cells across the interface. The base cell is a parent cell in this respect and these fictitious cells are its children. The child cell is designed in such a way that each has a common face with one of the secondary cells at the interface. The continuity of grid lines across the interface is thus recovered. This transforms the discontinuous 1D/2D interface into a virtually patched continuous 2D/2D interface dealt with in section 4.5. Hence, once variables at these fictitious cells are obtained, the evaluation of the interface segment in the 2D block can be handled using the same methods as in section 4.4.

The variable values at fictitious cells are derived from an assumed profile at the cross section of the base cell in the 1D block. A variety of profile choices are possible: piecewise constant, piecewise quadratic or piecewise higher order polynomial. The first two have found the widest use. It is known that the profile in an exactly one-dimensional flow will be constant at the interface, however, this will not be the case if two-dimensional effects are present. Therefore the estimation of fictitious cell values may involve the transformation of the constant profile in the one dimensional block to a cross section distribution appropriate to two dimensional flow if the base cell lies in a region of significant two-dimensional flow. However, in the flow applications of this work, the interface between the one-dimensional block and the two dimensional block is always placed in a region where the flow has become predominantly one dimensional, therefore, the piecewise constant profile is a good approximation for the fictitious cell evaluation.
General Strategy of Coupling Methodology

In a conservative coupling method, once the variables and fluxes at the interface segments for secondary cells are obtained, the flux at the base cell's interface is calculated by summing up the fluxes over all secondary cell interface segments to comply with the requirement of equal flux entering and leaving the interface. Consider the coupled model in Figure 4.7. The computation of numerical flux on interface segments $c_{f_i}$ is straightforward. With the creation of fictitious child cells $p'_i$, this is carried out in the same way as for interior cell faces, i.e. via

$$
\hat{F}_{c_{f_i}} = \Phi [\hat{s}, \hat{q}_{p'_i}, \hat{q}_{p'_i+1}, \ldots, \hat{q}_{p'}, \hat{q}_{p'+1}, \ldots] \tag{4.12}
$$

where $\Phi$ is the function defining the flux at an interior cell face, $s$ is the area normal of the interface segment $c_{f_i}$, $q_{p'_i}$ and $q_{p'_i}$ are the cell-centred variable values for the secondary cell in block 2 and the fictitious child cell in block 1 respectively. The computation of numerical flux on the opposite side of the interface $m-m'$, is slightly more complicated.

In an explicit coupling, it is given by

$$
\hat{F}_{m-m'} = \Phi [\hat{Q}_{m-m'}, q_{p'_1}, q_{p'_2}, \ldots] \tag{4.13}
$$

where $Q$ is the area-weighted average value of the variable $q$ of all the secondary cells in block 2, $q_{p'_1}$ is the cell-centred variable value of the base cell in block 1. It is apparent that the flux evaluated in such way may not be consistent on both sides of the interface. On the other hand, in an implicit coupling, the flux on interface $m-m'$ is given, by the conservation condition

$$
\hat{F}_{m-m'} = \sum_{i=1}^{n} \hat{F}_{c_{f_i}} \tag{4.14}
$$

so the consistency of flux across the interface is ensured.
General Strategy of Coupling Methodology

The above is a general form of zonal treatment for any solution algorithm. The higher the order of accuracy of the discretisation scheme, the more cells penetrating into the interior of the domain are used.

The basic procedure to evaluate flux and variables on the zonal interface in a 'base-cell-concept' scheme can therefore be summarised as:

(1) identify base cell and secondary cells

(2) create fictitious child cells based on base cell

(3) obtain variable values at fictitious cells by appropriate interpolation

(4) evaluate variables and fluxes at secondary cell interface following the interface scheme for a 2D continuous patched grid

(5) evaluate the flux at the base cell interface by the summation of the fluxes at the secondary cell’s interface segments.
Chapter 5

Coupling 1D and 2D Pressure-Correction Procedure
5.1 Preamble

General reviews on coupling strategies for a hybrid 1D/2D model have been presented in Chapter 4; now these are integrated into the unified pressure-correction solution procedure described in Chapter 2. Both explicit and implicit interface treatments are implemented due to the semi-implicit property of this base algorithm. The computing efficiency and overall cost of both coupling approaches are the focus of these studies. Due to the use of the 'base cell concept', the evaluation of conditions at the interface segment in a 1D/2D coupled model and a 2D/2D multiblock model is essentially identical. Therefore, only the discussion of the coupling schemes for the 1D/2D coupled model need to be presented.

The form of governing equations adopted in the unified pressure-correction algorithm (equations (2.11) to (2.14) and (2.39)) are used here to demonstrate the implementation of coupling schemes.

5.2 Explicit Coupling Scheme

The implementation of an explicit interface coupling scheme described in section 4.5.2.2 involves mainly explicit interpolations of solution variables onto the interface. Because the base algorithm is a conservative scheme, a conservative interface treatment is preferred, although conservation of all fluxes is difficult in an explicit coupling method.

5.2.1 Interface Evaluation in Momentum Equation Discretisation

The application of finite volume discretisation and Euler backward time integration transforms momentum equations (2.12) or (2.13) to a semi-discretised form as (here \( v_i : (u, v) \) and \( x_i : (x, y) \)): 
Coupling 1D and 2D Pressure-Correction Procedure

\[
\frac{1}{J \Delta t} (\rho v_i)_{(i,j)}^{k+1} + \left[ \frac{1}{J} (\rho v_i U)^{k+1} \right]_{(i+\frac{1}{2},j)} + \left[ \frac{1}{J} (\rho v_i V)^{k+1} \right]_{(i-\frac{1}{2},j)}_{(i,j+\frac{1}{2})} = -\frac{\xi_{x_i}}{J} \left( P_{(i+\frac{1}{2},j)} - P_{(i-\frac{1}{2},j)} \right) + \frac{\eta_{x_i}}{J} \left( P_{(i,j+\frac{1}{2})} - P_{(i,j-\frac{1}{2})} \right) \\
+ \left[ \frac{\xi_{x_i}}{J} \tau_{x_{x_i}} + \frac{\xi_{x_j}}{J} \tau_{x_{x_j}} \right]_{(i+\frac{1}{2},j)}^{i+\frac{1}{2}} + \left[ \frac{\xi_{x_i}}{J} \tau_{x_{x_i}} + \frac{\xi_{x_j}}{J} \tau_{x_{x_j}} \right]_{(i-\frac{1}{2},j)}^{i-\frac{1}{2}} \\
+ \frac{1}{J \Delta t} (\rho v_i)^n_{(i,j)}
\]

for an unsteady calculation, where superscript ‘k+1’ denotes the current iteration step within the current time step n+1, and ‘n’ denotes the previous time level. Note that the ‘n+1’ superscript has been dropped from the equation for simplification. \(\Delta \xi\) and \(\Delta \eta\) also do not appear as they are both assumed unity. The above equation needs to be further discretised to obtain a final algebraic form; the fluxes on the control volume faces need to be approximated using solution variable values at the cell centres.

A typical interface in a 1D/2D hybrid coupled grid is illustrated in Figure 5.1

---

**FIGURE 5.1 A hybrid 1D/2D coupled grid**
Coupling 1D and 2D Pressure-Correction Procedure

m-m’ is the interface between Blocks 1 and 2, it is also the east cell face of control volume P in block 1; with cf the interface segment, it is also the west cell face of control volume P^2.

5.2.1.1 Interface Treatment for Convective Flux

On an ordinary cell face, e.g. ‘ef2’ for cell P^2 and E^2 in block 2, the convective flux \( \rho v_i U \) or \( \rho v_i V \) is approximated by

\[
(\rho v_i U)^{k+1}_{(i+\frac{1}{2},j)} = (U)^{k}_{(i+\frac{1}{2},j)} \cdot (\rho v_i)^{k+1}_{(i+\frac{1}{2},j)}
\]

5.2

i.e. a multiplication of the iteratively lagged contravariant velocity and the implicit dependent variable. A linear interpolation for the convecting quantity gives

\[
U^k_{(i+\frac{1}{2},j)} = \frac{1}{2}(U^k_{(i,j)} + U^k_{(i+1,j)})
\]

5.3

The implicit (convected) quantity is, on the other hand, estimated from the solution variables according to a specific function \( \chi \)

\[
(\rho v_i)^{k+1,n+1}_{(i+\frac{1}{2},j)} = \chi\left\{(\rho v_i)^{k+1,n+1}_{(i,j)}, (\rho v_i)^{k+1,n+1}_{(i+\pm 1,j)}, (\rho v_i)^{k+1,n+1}_{(i,j+\pm 1)}, \text{etc}\right\}
\]

5.4

According to Chapter 2, the function \( \chi \) here represents the second order TVD MUSCL scheme.

Equation (5.4) generates entries into the algebraic equation for point \((i,j)\) which contribute to the implicit part of the final algebraic equation, while equation (5.3) provides the linearised coefficients. Equation (5.3) and (5.4) demonstrate also that the interpolation methods for the convecting and convected components in a flux can be different. Generally, a much simpler interpolation method is used for the convecting component.
**Coupling 1D and 2D Pressure-Correction Procedure**

On a cell face which lies on 1D&2D block interface, this convective flux is evaluated differently. On the interface segment, e.g. 'cf', flux $\rho v_i U$ is calculated by

\[
(\rho v_i U)^{k+1,n+1}_{(cf)} = (\rho v_i U)^{k,n+1}_{(cf)} = U^{k,n+1}_{(cf)} \cdot (\rho v_i)^{k,n+1}_{(cf)}
\]

5.5

i.e. a multiplication of an explicit contravariant velocity and an explicit convected variable.

The ‘base cell concept’ described in section 4.5.3 is employed to create a series of fictitious cells in the one dimensional block, shown by the dashed lines in Figure 5.1, so that there is a fictitious cell, $P_1$ in Figure 5.1, sharing the interface segment ‘cf’ opposite to $P_2$. Consequently linear interpolation can be carried out between cell $P_2$ and fictitious cell $P_1$ as

\[
U^{k,n+1}_{(cf)} = \frac{1}{2}(U^{k,n+1}_{(P)} + U^{k,n+1}_{(P^i)})
\]

5.6

If the fictitious cell value is assumed to be the same as the base cell value, i.e. a piece-wise constant variable approximation, equation (5.6) becomes

\[
U^{k,n+1}_{(cf)} = \frac{1}{2}(U^{k,n+1}_{(P)} + U^{k,n+1}_{(P^i)})
\]

5.7

To evaluate the explicit dependent variable $\rho v_i U^{(cf)}$, simple linear interpolation rather than MUSCL TVD differencing is used

\[
(\rho v_i)^{k,n+1}_{(cf)} = \frac{1}{2}[(\rho v_i)^{k,n+1}_{(P)} + (\rho v_i)^{k,n+1}_{(P^i)}] = \frac{1}{2}[(\rho v_i)^{k,n+1}_{(P)} + (\rho v_i)^{k,n+1}_{(P^i)}]
\]

5.8

Here the fictitious cell is also used to enable the interpolation.

As a matter of fact, $\rho v_i U^{(cf)}$ has to be linear interpolated if equation (5.7) is adopted. This is because if the general coordinate system is coincident with a Cartesian coordinate system, equation (5.7) becomes
Coupling 1D and 2D Pressure-Correction Procedure

\[ (u)^k_{(cf)} = \frac{1}{2}[(u)^k_{(p)} + (u)^k_{(p')}] \] 5.9

which is actually the linear interpolation of the u velocity. For an incompressible flow, the following expression becomes subsequently valid

\[ (\rho u)^k_{(cf)} = \frac{1}{2}[(\rho u)^k_{(p)} + (\rho u)^k_{(p')}] \] 5.10

and this is the linear interpolation of dependent variable. Hence, if the contravariant velocity component U is evaluated by (5.7), the dependent variable pu in the momentum equation can only be linearly averaged in order to guarantee the consistence of the interface velocity variable in all flow situations. Otherwise, a conflict would occur when estimating the u variable on the interface for different terms in equation (5.1).

In a 1D/2D coupled model, the algebraic equation derived from equation (5.1) for an ordinary cell has a general form:

\[ a_p v_{i(i, j)}^{k+1} + a_w v_{i(i-1, j)}^{k+1} + a_e v_{i(i+1, j)}^{k+1} + a_s v_{i(i, j-1)}^{k+1} + a_n v_{i(i, j+1)}^{k+1} = S_{(i, j)} \] 5.11

However, for those cells in the two-dimensional block next to the interface, e.g. P2, because of the explicit computation of the dependent variable on the interface segment, the convective flux there contributes only a source term in the final algebraic equation, the form of the algebraic equation changes to:

\[ a_p v_{i(i, j)}^{k+1} + a_e v_{i(i+1, j)}^{k+1} + a_s v_{i(i, j-1)}^{k+1} + a_n v_{i(i, j+1)}^{k+1} = S_{(i, j)} - S_w \] 5.12

So there is a discontinuity in the general form of the algebraic equation for the control volumes, and the existence of the internal boundary is clearly visible in the solution procedure.
Coupling 1D and 2D Pressure-Correction Procedure

Now consider interface $m-m'$ in the one dimensional block. The method to evaluate the convective flux on this face is now different. Instead of interpolation, the mass flux at this interface is derived from a mass conservation balance

$$\rho U_{(m-m')} = \sum_i [\rho U_{(cf_i)}]$$  \hspace{1cm} 5.13

i.e. the mass flux across the whole interface $m-m'$ equals the summation of this flux across all segments $cf_i$. The dependent variables on interface $m-m'$ have to be computed in such a way that equation (5.13) is satisfied. The method adopted is area-weighted averaging

$$\rho u_{(m-m')}^k = \frac{1}{\text{area}_{y,(m-m')}} \cdot \sum_i [\rho u_{(cf_i)}^k \cdot \text{area}_{y,(cf_i)}]$$  \hspace{1cm} 5.14

$$\rho v_{(m-m')}^k = \frac{1}{\text{area}_{x,(m-m')}} \cdot \sum_i [\rho v_{(cf_i)}^k \cdot \text{area}_{x,(cf_i)}]$$  \hspace{1cm} 5.15

where $\text{area}_x$ and $\text{area}_y$ are projected areas in the $x$ and $y$ directions respectively. For the same consistency reason as stated for equation (5.8), given the approximation method of (5.13), the contravariant velocity flux is evaluated via

$$U_{(m-m')}^k = \sum_i [U_{(cf_i)}^k]$$  \hspace{1cm} 5.16

However, the satisfaction of consistency could render it difficult to conserve all fluxes on the interface. For example, given the estimations (5.13) to (5.15), it is easy to show that momentum flux might not be conserved

$$[\rho U_{i(m-m')}^{k,n+1}] \neq \sum_i [\rho U_{i(cf_i)}^{k,n+1}]$$  \hspace{1cm} 5.17

On the other hand, if the momentum flux on interface $m-m'$ were calculated by imposing a conservation principle, i.e.
Coupling 1D and 2D Pressure-Correction Procedure

\[ [\rho U v_{i(m-m')}^k] = \sum \left[ \rho U v_{i(cf_i)}^{k+1} \right] \quad 5.18 \]

the conservation of mass flux on the interface is not guaranteed.

It is difficult in an incompressible flow calculation to enforce conservation of both mass flux and momentum flux. The mass flux is a linear function of velocity components, whereas the momentum flux is a quadratic function. The conservation of mass flux implies the velocity component should be interpolated from

\[ u_{m-m'}^{k,n+1} = \sum u_{cf_i}^{k,n+1} \quad 5.19 \]

On the other hand, the conservation of momentum flux implies that \( u^k \) should be interpolated from

\[ \left( u_{m-m'}^{k,n+1} \right)^2 = \sum \left( u_{cf_i}^{k,n+1} \right)^2 \quad 5.20 \]

In the present work, the mass flux on the interface is conserved as opposed to the momentum flux. The momentum flux is computed from the velocity variables that are derived from mass flux conservation. This is to ensure no mass accumulation in any control volume in an incompressible flow calculation which is vital for pressure-correction algorithm stability.

Due to the above interface evaluation, for a cell in the one-dimensional block next to the interface, i.e., \( P \), the final algebraic equation form changes to

\[ a_p v_{i(i,j)}^{k+1} + a_w v_{i(i-1,j)}^{k+1} + a_n v_{i(i,j-1)}^{k+1} + a_s v_{i(i,j+1)}^{k+1} = S_{i(i,j)} - S_e \quad 5.21 \]

where \( S_w \) and \( S_e \) are the additional source terms generated by interface treatment. It is clear that the implicit stencil is shorter near the interface, hence the implicitness is weaker.
Coupling 1D and 2D Pressure-Correction Procedure

5.2.1.2 Interface Treatment for Pressure

The pressure gradient term in the momentum equation is treated explicitly for an ordinary cell in a pressure-correction procedure, the pressure value on the cell face being approximated by linear average, i.e.

\[ p_{k+1, n+1}^{(i+\frac{1}{2}, j)} = \frac{1}{2} (p_{i, j}^{(k, n+1)} + p_{i+1, j}^{(k, n+1)}) \quad 5.22 \]

On the interface segment, e.g. 'cf', this is also evaluated using the linear approximation

\[ p_{(cf)}^{k} = \frac{1}{2} (p_{p}^{k} + p_{l}^{k}) \quad 5.23 \]

If a constant pressure profile is assumed in the base cell

\[ p_{(cf)}^{k} = \frac{1}{2} (p_{p}^{k} + p) \quad 5.24 \]

On the interface m-m', pressure is estimated by a different method from the above. It is derived from pressure force conservation as

\[ \text{area}_{(m-m')} \cdot p_{m-m'}^{k} = \sum_{i} \left[ \text{area}_{(cf_{i})} \cdot p_{(cf_{i})}^{k} \right] \quad 5.25 \]

Since pressure values on interface segments cf, are given by equation (5.23), pressure on interface m-m' can be determined as

\[ p_{m-m'}^{k} = \frac{1}{\text{area}_{(m-m')}} \cdot \sum_{i} \left[ \text{area}_{(cf_{i})} \cdot p_{(cf_{i})}^{k} \right] \quad 5.26 \]

Like equation (5.14) and (5.15), equation (5.25) is an area weighted averaging formula.
Pressure also appears in the Rhie & Chow pressure smoothing expression. A conflict similar to that encountered with the velocity component occurs for the pressure if an attempt is made to conserve both pressure force and the pressure smoothing term at the interface. In this work, pressure force conservation is enforced. Pressure smoothing is calculated using the pressure values derived from equation (5.26).

### 5.2.1.3 Interface Treatment for Viscous Flux

For ordinary cells, some of the normal diffusion flux is evaluated implicitly, while other normal and cross derivative viscous fluxes are evaluated explicitly. Linear interpolation is adopted in evaluating all of these on interior cell faces. For example, an implicit normal diffusion is calculated by:

$$
\left[ \frac{\partial v_j}{\partial \xi} \right]_{i+\frac{1}{2}, j}^{k+1, n+1} = v_{i(i+1, j)}^{k+1, n+1} - v_{i(i, j)}^{k+1, n+1}
$$

an explicit normal diffusion is calculated by:

$$
\left[ \frac{\partial v_j}{\partial \xi} \right]_{i+\frac{1}{2}, j}^{k, n+1} = v_{j(j+1, j)}^{k, n+1} - v_{j(i, j)}^{k, n+1}
$$

and an explicit cross diffusion by:

$$
\left[ \frac{\partial v_i}{\partial \eta} \right]_{i+\frac{1}{2}, j}^{k, n+1} = \frac{1}{4} \cdot \left( v_{i(i+1, j+1)}^{k, n+1} + v_{i(i, j+1)}^{k, n+1} - v_{i(i+1, j-1)}^{k, n+1} - v_{i(i, j-1)}^{k, n+1} \right)
$$

On the interface segment 'cf', the implicit normal diffusion component is evaluated by

$$
\left[ \frac{\partial v_i}{\partial \xi} \right]_{(cf)}^{k+1, n+1} = 2 \cdot \left( v_{i(i+1, j)}^{k+1, n+1} - v_{i(cf)}^{k, n+1} \right)
$$
**Coupling 1D and 2D Pressure-Correction Procedure**

where $v_{i(cf)}^k = (\rho v_{i(cf)}^k) / (\rho_{cf}^k)$ and $(\rho v_{i(cf)}^k)$ is given by equation (5.8) and $(\rho_{cf}^k)$ is evaluated similarly. This treatment is to preserve some of the implicitness on this explicit interface, and also to maintain a consistent approximation. The explicit normal diffusion on this interface segment is calculated by:

$$
\left[ \frac{\partial v_j^k}{\partial \zeta} \right]_{(cf)} = 2 \cdot (v_j^{k(P'_i)} - v_{j(cf)}^k) = v_j^{k(P'_i)} - v_{j(P')}^k - v_j^{k(P)}
$$

and the cross diffusion by:

$$
\left[ \frac{\partial v_{i}^k}{\partial n} \right]_{(cf)} = \frac{1}{2} \cdot (v_{i(cf)}^k - v_{i(cf)}^k) = \frac{1}{4} \cdot (v_{i(N^i)}^{k} + v_{i(N^i)}^{k} - v_{i(S^i)}^{k} - v_{i(S^i)}^{k})
$$

Obviously, the methods are the same as those in equation (5.28) and (5.29); this enhances the similarity between the interface segment and the cell face.

On the whole interface $m-m'$, the viscous flux cannot be derived from a conservation law due to the same conflict problem explained for equations (5.7) and (5.8), so it is simply calculated using the velocity variable that is derived from mass flux conservation, i.e.

$$
(v_{i})_{m-m'}^{k,n+1} = (\rho v_{i})_{m-m'}^{k,n+1} / \rho_{m-m'}^{k,n+1}
$$

where

$$
\rho_{m-m'} = \frac{p_{m-m'}}{RT_{m-m'}}
$$

$p_{m-m'}$ is given by equation (5.26), $T_{m-m'}$ is the interface temperature value from the previous iteration solution.

Hence implicit normal diffusion on the interface $m-m'$ is approximated by:
Coupling 1D and 2D Pressure-Correction Procedure

\[
\left[ \frac{\partial v_i}{\partial \xi} \right]^{k+1,n+1}_{(m-m')} = 2 \cdot \left( v_i^{k,n+1}_{m-m'} - v_i^{k+1,n+1}_{i(P)} \right)
\]

and explicit normal diffusion by:

\[
\left[ \frac{\partial v_j}{\partial \xi} \right]^{k,n+1}_{(m-m')} = 2 \cdot \left( v_j^{k,n+1}_{m-m'} - v_j^{k,n+1}_{j(P)} \right)
\]

with cross diffusion from

\[
\left[ \frac{\partial v_i}{\partial \eta} \right]^{k,n+1}_{(m-m')} = \frac{1}{2} \cdot \left( v_i^{k,n+1}_{i(N)} - v_i^{k,n+1}_{i(S)} \right)
\]

where N and S denote the cells on the north and south of control volume P.

This treatment does not guarantee the conservation of viscous flux at the interface, like the convective momentum flux. This is the disadvantage and inherent weak point of explicit coupling. As will be explained later, implicit coupling can remove these flux conservation problems. However, despite these unfavourable properties, the explicit coupling is appealing because of its simplicity and easy implementation.

5.2.2 Interface Evaluation in Pressure-Correction Equation Discretisation

For the pressure-correction equation given by (2.39), a finite volume discretisation leads to the following equation:
Coupling 1D and 2D Pressure-Correction Procedure

\[
\frac{p'}{\Delta t \cdot RT}^{k, n + 1} + \left[ C \cdot \frac{\partial}{\partial \xi} p \right]^{(i + \frac{1}{2}, j)} + \left[ D \cdot \frac{\partial}{\partial \eta} p' \right]^{(i, j + \frac{1}{2})} = 0
\]

\[
\frac{1}{\Delta t} \left\{ p^{k, n + 1} - \left( \rho \right)^n \right\}
\]

where variables without any superscripts are at current iteration step \( k + 1 \) within the current time step \( n + 1 \).

On the whole interface or on an interface segment, the mass flux in the above equation is evaluated explicitly using the same method as that adopted for the momentum equation.

The evaluation of the pressure-correction variable on the interface has to use a different approach. In a pressure-correction procedure, pressure-correction is a special variable, only defined for the current iteration; the pressure-correction solution at the previous iteration has no correlation with the one at the current iteration. Hence an explicit interpolation method cannot be used to estimate this variable. In this work, the approximation method applied at an explicit physical boundary is employed, an internal interface is in some sense similar to a physical boundary in an explicit coupling scheme.

According to Chapter 2, the pressure-correction gradient at the interface is set to zero - a Neumann boundary condition, because flow conditions at the interface are assumed known during the solution procedure. For the example shown in Figure 5.1, this means

\[
\left( \frac{\partial}{\partial \xi} p \right)_{(cf_i)}^{k + 1, n + 1} = 0 \quad \left( \frac{\partial}{\partial \xi} p' \right)_{(m - m')}^{k + 1, n + 1} = 0
\]

Hence, the algebraic equation for the control volume next to the interface takes the following form
Coupling 1D and 2D Pressure-Correction Procedure

\[ a_p p^{k+1} + a_e p^{k+1} + a_s p^{k+1} + a_n p^{k+1} = S_{(p^2)} \]  

No terms associated with the cells in the adjacent block are present in the implicit part of the equation.

One drawback of such an interface treatment is that it cuts off the information transfer of the pressure-correction, and effectively the pressure, between the blocks, because the interface contains no information of this variable from either block. Also the pressure-correction equation is essentially solved in an independent manner for each block rather than coupled block. A problem consequently arises in an incompressible flow simulation that the pressure solutions on each side of the interface could converge to a different constant, i.e. a pressure jump is created in the converged solution at the interface. However, this problem did not occur in any applications and prediction results of the present work.

5.2.3 Pressure Smoothing

Rhie & Chow pressure smoothing is an essential element in the non-staggered pressure-correction scheme to remove the oscillations triggered by pressure-velocity decoupling. This smoothing is in practice introduced during the approximation of cell face mass fluxes. However, this smoothing is not added on the boundary because flow variables there are not solved but derived according to the boundary condition.

In a 1D/2D coupled model, on an explicit block interface, because flow variables are obtained in a similar manner as at an explicit physical boundary, i.e. they are not solved but derived following conditions imposed by the interface coupling scheme, the interface does not present a pressure-velocity decoupling source to the flow solution. As a result, pressure smoothing is not essential at an explicit interface and its neglect has no impact on stability.
**Coupling 1D and 2D Pressure-Correction Procedure**

### 5.2.4 Linear Matrix Solver

As mentioned in Chapter 2, SLOR and TDMA solvers have been used to solve the linear algebraic matrix problem. In a single block structured grid system, each unknown line in the SLOR procedure is coincident with a grid line. One SLOR cycle sweeps over all grid lines. In the multiblock grid system, because of the explicit interface treatment, blocks are effectively separated from each other; SLOR sweeps thus proceed one zone after another. Hence, a line of unknowns is defined within a zone and coincides with a grid line in that zone, rather than across the entire domain, unlike in a single block system. After all lines in a particular zone are updated, the procedure is repeated in another zone. This sequential zonal solution procedure characterises the SLOR solver in the explicit coupling scheme.

Since the zonal interface is identified as an explicit boundary in an explicit coupling scheme, at any given time level, the values of variables on the interface are frozen until the resolution of all interior cells is completed. As a result the explicit treatment of zonal boundaries could affect the convergence rate, and degrade efficiency and stability.

However, the use of an explicit interface does not raise the need to modify the TDMA solver. In addition, because only a grid line within a block is solved at a time, the matrix to be inverted is much smaller. These characteristics are very different when the implicit interface coupling method is adopted.

### 5.2.5 Summary

The explicit conservative interface coupling method can be summarised as:

1. On the interface segment belonging to the two dimensional block, $\rho u$ and $\rho v$ dependent variables are determined with the assistance of fictitious cells created by the 'Base Cell Concept'. The pressure variable is obtained following the same rule.

2. On the interface belonging to the one dimensional block, $\rho u$ and $\rho v$ variables are calculated by an implementation of a mass flux conservation constraint. Pressure is computed by enforcing a pressure force conservation constraint.
Coupling 1D and 2D Pressure-Correction Procedure

(3) Contravariant velocity components on the interface and interface segments are derived using a similar method to that for \( \rho u \) and \( \rho v \). Velocity components are derived from \( \rho u, \rho v \) and density.

(4) Using velocity and pressure, any other fluxes at the interface are evaluated for the 1D and 2D blocks individually.

Although only mass and pressure force conservation are imposed at the 1D/2D interface in the explicit coupling scheme, this has proven to be sufficient to maintain global conservation for the flow problems of interest. However, it may incur slow convergence due to weak coupling.

5.3 Implicit coupling scheme

As described in Chapter 4, the philosophy of implicit coupling is to minimise the inconsistency between the interface treatment and the interior cell face treatment. In this section, the implementation of an implicit coupling interface method in the pressure-correction procedure is discussed. Various issues raised are reported, in particular, a novel modification of the SLOR and TDMA solvers.

5.3.1 Interface Treatment in Momentum Equation Discretisation

Consider again the coupled 1D/2D mesh in Figure 5.1 and the semi-discretised momentum equation (5.1).

5.3.1.1 Evaluation of Convective Flux on Interface

On an interface segment 'cf' in the two dimensional block, the convective flux is approximated by

\[
[U(\rho v_i)]_{(cf)}^{k+1,n+1} = U_{(cf)}^{k,n+1} \cdot [(\rho v_i)]_{(cf)}^{k+1,n+1}
\] 5.41
Coupling 1D and 2D Pressure-Correction Procedure

This is no different from expression (5.2), which is for the ordinary cell face. The lagged term in the above equation, i.e. the contravariant velocity, is calculated from equation (5.7), the implicit dependent variable being discretised via

\[ (\rho v_i)_{(c)}^{k+1} = \chi \left\{ (\rho v_i)_{(p^1)}^{k+1}, (\rho v_i)_{(p^2)}^{k+1}, \text{etc.} \right\} \]

where \( \chi \) represents the discretisation scheme. \( P^1 \) is the fictitious cell in the one dimensional block. The discretisation function \( \chi \) is taken to be the same as the one applied to ordinary cell faces. Therefore, interpolation of the momentum flux on the interface segment is no different to interior cell faces. The substitution of (5.41), (5.42) and (5.7) into the interface convective flux in equation (5.1) provides implicit terms to the discretised algebraic equation which are associated with the cells in the two dimensional block. An implicit term associated with the cell in the one dimensional block is introduced through the fictitious cell. For example, the algebraic equation for cell \( P^2 \) takes the form

\[ a_p v_i^{n+1} + a_w v_i^{n+1} + a_c v_i^{n+1} + a_s v_i^{n+1} + a_n v_i^{n+1} = S_{(P^2)} \]

Obviously, there are no missing neighbours in the implicit terms, unlike the situation when the interface is treated explicitly.

On interface \( m-m' \) in the one dimensional block, the convective momentum flux is calculated from the conservation of this flux

\[ [U(\rho v_i)]_{(m-m')}^{k+1,n+1} = \sum_j [U(\rho v_i)]_{(c)}^{k+1,n+1} \]

This is in contrast to the indirect method in the explicit interface treatment. It will be shown later that the mass flux at the interface can also be evaluated using a conservation law without introducing any inconsistency. This implicit evaluation in (5.44) generates implicit terms associated with the cells on the two sides of the interface \( m-m' \) in the algebraic equation for the base cell in 1D block.
5.3.1.2 Evaluation of Pressure on Interface

In the semi-discretised equation (5.1), pressure force on either interface or interface segment are evaluated in the same way as in the explicit interface coupling scheme expressed by (5.26) and (5.23).

5.3.1.3 Evaluation of Viscous Flux on Interface

On the interface segment 'cf', the normal diffusion flux component which contains the gradient in the same direction as the dependent variable - momentum component, is evaluated by:

\[
\frac{\partial \nu_i}{\partial \xi}^{n+1}_{(cf)} = \nu_{i(P')}^{n+1} - \nu_{i(P')}^{n+1}
\]

which is the same as equation (5.27) for the ordinary cell face. Other normal diffusion flux components as well as cross diffusion flux are computed explicitly using the same methods as those used in the explicit interface treatment.

On interface m-m', unlike in the explicit interface treatment, the viscous flux can be calculated from a conservation law regardless of how mass flux at the interface is estimated. Hence, this flux is given by the sum over interface segments, i.e.

\[
\left[ \frac{\partial \nu_i}{\partial \xi} \right]^{n+1}_{(m-m')} = \sum_i \left[ \left[ \frac{\partial \nu_i}{\partial \xi} \right]^{n+1}_{(cf_i)} \right]
\]

\[
\left[ \frac{\partial \nu_i}{\partial \eta} \right]^{n}_{(m-m')} = \sum_i \left[ \left[ \frac{\partial \nu_i}{\partial \eta} \right]^{n}_{(cf_i)} \right]
\]

\[
\left[ \frac{\partial \nu_i}{\partial \eta} \right]^{n}_{(m-m')} = \sum_i \left[ \left[ \frac{\partial \nu_i}{\partial \eta} \right]^{n}_{(cf_i)} \right]
\]
representing respectively the implicit normal diffusion, the explicit normal diffusion and the cross diffusion flux.

The most important characteristic introduced by the implicit interface coupling is that the algebraic equation for the one dimensional base cell \( P \) in Figure 5.1 has acquired a stencil which is not a standard 5-point structure, but contains multiple implicit entries which link this cell with all of the cells next to the interface in the two-dimensional block. The form of the equation is:

\[
\sum_{j} a_{w} v_{i(j)}^{k+1} + a_{e} v_{i(E)}^{k+1} + a_{s} v_{i(S)}^{k+1} + a_{n} v_{i(N)}^{k+1} = S_{(P)}
\]  

The significance of such a stencil structure is its impact on the SLOR and TDMA matrix solver, because the original SLOR line solver assumes 5-point stencil banded matrix structure. This will be discussed in detail in a later subsection.

5.3.2 Interface Treatment in Pressure-Correction Equation Discretisation

As described in section 5.2.2 in explicit interface coupling, the treatment of pressure-correction on the interface presents a large discrepancy with the treatment on the ordinary cell face. In implicit interface coupling, this shortcoming can be overcome.

5.3.2.1 Evaluation of Pressure-correction Gradient on Interface

The pressure-correction quantity on the interface segment 'cf' in the semi-discretised pressure-correction equation (5.38) is approximated by the same formula given by equation (5.45), i.e.

\[
\left( \frac{\partial}{\partial \xi} p \right)_{(cf)}^{n+1} = (p')_{(P')}^{n+1} - (p')_{(P)}^{n+1} = (p')_{(P')}^{n+1} - (p')_{(P)}^{n+1}
\]  

It contributes implicit terms associated with the base cell in the one dimensional block to the algebraic equation of the cell in the two dimensional block (control volume \( P^2 \)). The general form of the algebraic equation for \( P^2 \) thus becomes
Coupling 1D and 2D Pressure-Correction Procedure

\[ a_p p_{p}^{n+1} + a_p p_{p_i}^{n+1} + a_e p_{e_i}^{n+1} + a_s p_{s_i}^{n+1} + a_n p_{n_i}^{n+1} = f_{(p_i)} \]  

This form is no different from the one for an ordinary interior cell. The variable at the fictitious cell \( p_i \) will be replaced by the variable at base cell \( p \) after a variable profile in the base cell is assumed. Clearly the equation contains implicit terms not only involving the cells in the same block but also the cell in the adjacent block. The coupling of two blocks is hence incorporated in the pressure-correction equation like in the momentum equation.

On the whole interface \( m-m' \) the pressure-correction quantity is obtained again by conservation:

\[ \left[ \frac{\partial}{\partial \xi} p^{n+1}_{(m-m')} \right] = \sum_i \left[ \frac{\partial}{\partial \xi} p^{n+1}_{(cf_i)} \right] \]  

This expression leads to an algebraic equation which contains multiple implicit terms associated with the cells in the two dimensional block, similar to the momentum equation (5.50). Therefore strong coupling of the 1D and 2D models is also integrated into the pressure-correction equation for the base cell in the one dimensional block.

A major benefit of using an implicit zonal interface is the preservation of the pressure coupling in the pressure-correction equation in contrast to an explicit interface. This is primarily due to the recovery of the pressure-correction link between the blocks through the pressure-correction flux conservation at the interface.

5.3.2.2 Evaluation of Mass Flux on Interface

Mass flux is required in the source terms of equation (5.38), and this is approximated by linear averaging for an interface segment ‘cf’

\[ (\rho U)^{k,n+1}_{(cf)} = \frac{1}{2} [(\rho U)^{k,n+1}_{(p_i)} + (\rho U)^{k,n+1}_{(p')} ] \]  

For the whole interface \( m-m' \), it is computed by conserving this flux through interface segments, i.e.
The ability in this implicit interface treatment to apply simultaneously the conservation constraint on both momentum flux and mass flux without causing conflict is because the dependent variable $p\nu_j$ is evaluated implicitly in the momentum flux but explicitly in the mass flux, that is $p\nu_j$ solutions at different iteration levels are used in the estimation of mass and momentum flux. In an iterative scheme, the same variable at different iteration levels can be treated as two independent quantities in the numerical approximations. Different methods can be used to calculate them. Therefore it is possible, when mass flux at the interface is to be conserved, to use a $p\nu_j$ variable at the previous iteration level $k$ as in equation (5.14) and (5.15); on the other hand, when the momentum flux at the interface is to be conserved, the $p\nu_j$ variable at the current iteration step is derived from equation (5.44). The result of this is (in the final converged solution) the conservation of both mass flux and momentum flux on the interface.

### 5.3.3 Pressure Smoothing

An important issue in the interface coupling treatment is to prevent solution oscillations from occurring. As illustrated in Chapter 2 there exists a pressure-velocity decoupling in the present pressure-correction scheme and Rhie&Chow smoothing is added to suppress it. This decoupling oscillation could appear at the interface as well, and the same pressure smoothing needs to be added at the interface.

As on an ordinary cell face, smoothing is added to the velocity flux variable on the interface segment in the two dimensional block. More fictitious cells in addition to the one next to the interface are however required to enable the implementation of Rhie&Chow smoothing.
**Coupling 1D and 2D Pressure-Correction Procedure**

Consider an internal cell face ‘wwf1’ in a 2D/2D patched multiblock system as shown in Figure 5.2

![Diagram of 2D/2D patched multiblock grid]

**FIGURE 5.2 A 2D and 2D patched multiblock grid**

The flux of contravariant velocity at this face is linearly interpolated and pressure smoothing is then added

\[
(U)_{\text{wwf}1}^{k,n+1} = \frac{1}{2} \cdot \left[ (U)_{w_1}^{k,n+1} + (U)_{w_1}^{k,n+1} \right] + \text{psmooth} \tag{5.55}
\]

or

\[
(U)_{\text{wwf}1}^{k,n+1} = \frac{1}{2} \cdot \left[ (U)_{w_1}^{k,n+1} + (U)_{w_1}^{k,n+1} \right] + \left( \sum_j c_j \cdot p_j \right)_{\text{wwf}1} \tag{5.56}
\]

where j denotes the cell involved in the Rhie & Chow pressure smoothing expression, including both upstream and downstream cells, and c_j is a coefficient.

In an analogous manner, on the patching interface m-m’, this flux at an interface segment, for example ‘ef’, can be evaluated by:
**Coupling 1D and 2D Pressure-Correction Procedure**

\[(U)_{cf}^{k,n+1} = \frac{1}{2} \cdot [(U)_{p1}^{k,n+1} + (U)_{p2}^{k,n+1}] + \text{psmooth} \quad 5.57\]

or

\[(U)_{cf}^{k,n+1} = \frac{1}{2} \cdot [(U)_{p1}^{k,n+1} + (U)_{p2}^{k,n+1}] + \left(\sum_j c_j \cdot p_j\right)_{cf} \quad 5.58\]

The Rhie&Chow smoothing expression will contain terms from both cells in block 1 which are on the left hand side of the interface segment 'cf' and cells in block 2 on the right hand side. Subsequently, this smoothing introduces into the algebraic equations some additional terms associated with the cells in both adjacent blocks, and block coupling is further strengthened.

The 1D/2D coupled grid as shown in Figure 5.1 can be readily converted to the 2D/2D patched grid as shown in Figure 5.2 using the fictitious cells in the one dimensional block. Equation (5.57) and (5.58) remain valid for interface segments in the two dimensional block in Figure 5.1. The formula given by (5.58) is used during the conservation evaluation of the mass flux on the interface m-m' in Figure 5.1 so that the pressure smoothing is contained in the interface mass flux.

**5.3.4 Summary**

With the implicit interface treatment on interface segments in the two dimensional block the dependent variable, the contravariant velocity flux, the pressure force and the pressure-correction gradient are all determined by the same scheme as used for the ordinary cell faces via the 'Base Cell Concept'. The velocity variable and the momentum diffusion flux are derived subsequently. The Rhie&Chow pressure smoothing is calculated from the the pressure solutions at the interface derived from pressure force conservation.

On the interface in the one dimensional block, all fluxes appearing in the semi-discretised equations are computed via conservation expressions. Therefore, a true interface conservation has been achieved in the implicit interface treatment.
Coupling 1D and 2D Pressure-Correction Procedure

The most important feature of implicit interface coupling is the special characteristic of the algebraic equation for the base cell in the one dimensional block. Terms linking multiple neighbours in the two dimensional block are present in this equation. Another important point to emphasis is that blocks are coupled in such a way that the interface is almost no different from a cell face. Such strong coupling proves to be of fundamental benefit for good stability and convergence properties of the overall algorithm in a 1D/2D coupled model.

5.3.5 TDMA Line Solver

One of the most important issues in implicit interface coupling is the solver, i.e. the linear matrix inversion algorithm. For a discretisation scheme with a 5-point stencil, a banded pentadiagonal matrix inversion problem occurs. As explained in Chapter 3, the Successive Line Overrelaxation (SLOR) solver is implemented in this work along with the Thomas Tridiagonal Matrix Algorithm (TDMA). In a single block system, SLOR sweeps across the computational domain line by line in I or J direction; an SLOR cycle terminates after all grid lines have been visited. Upon visiting each grid line, only the variables at the cells on this line are treated as unknowns, hence a 2D pentadiagonal matrix problem is reduced temporarily to a local 1D tridiagonal one. The tridiagonal matrix is inverted by the efficient direct inversion algorithm TDMA, consisting of a forward elimination process and a backward substitution process.

In the 1D/2D coupled multiblock system, the banded pentadiagonal structure is not preserved everywhere in the matrix. As a result, standard SLOR and TDMA procedures are no longer suitable. Modifications to SLOR and TDMA must be made in order to accommodate special features in this matrix. The major modification to the standard SLOR procedure is the definition of an appropriate line of unknowns. For the TDMA algorithm, the challenge is the direct inversion of a matrix which contains unknown entries from both 1D and 2D blocks.

5.3.5.1 Modified SLOR Procedure

When the standard SLOR procedure is applied to a single structured grid system, a line of unknowns is identified in the grid system. For example, in Figure 5.3, the cells on grid line
**Coupling 1D and 2D Pressure-Correction Procedure**

\( j=J \), i.e. cells \((i, J)\) where \(i=(1, NI+1)\), form a typical line of unknowns. A local equation system is defined which includes only the equations for the cells on this line, sequenced from cell \((1, J)\) to cell \((NI+1, J)\).

![Diagram of coupling 1D and 2D systems](image)

**FIGURE 5.3** A solution line in SLOR procedure

In a coupled 1D and 2D system, constant I or J lines do not exist in the global grid, but are only available in the individual block. Therefore, a line of unknowns needs to be "collected" in a different way. Two situations can be encountered during SLOR line assembling in 1D/2D coupled systems, and these are discussed separately below.

In one situation, the line of unknowns is collected from the 1D block to the 2D block as shown in Figure 5.4

![Diagram of SLOR line organization](image)

**FIGURE 5.4** The organisation of SLOR lines in 1D&2D grid system
Coupling 1D and 2D Pressure-Correction Procedure

While in the 1D block, i.e. block1, the line is assembled following the same pattern as that in the single block system and the corresponding grid line is termed the ‘master line’. This line runs from cell (1, 2) to cell (NI^1+1, 2) according to the local (i, j) index system of block 1. However, as the master line crosses the interface, it is not obvious how the line should continue because there are multiple grid lines emanating from the other side of the interface in the 2D block. Since strong block coupling is achieved by the implicit interface treatment, it is specifically contained in the neighbouring cell linkage coefficients in the equations for base cell P and secondary cell P^2_i. Such strong coupling and implicitness can only be well preserved during matrix inversion if all secondary cells are treated as unknowns along with the base cell. In order to be able to solve secondary cells simultaneously with the base cell, and in the mean time achieve a banded tridiagonal matrix problem, it is obvious that additional multiple lines rather than one line have to be defined in block 2 to continue the master line; these are termed ‘branch lines’. The constant J lines in the local (i, j) designation system of block 2 construct these branch lines. Each of the branch lines starts from one secondary cell next to the interface, runs along the grid line which has the same J value as that secondary cell and ends at the real boundary. The equations are collected in the same sequence. Only the dependent variables in the cells on the branch line are treated as unknowns. In the end, the equations for the cells on the master line as well as all branch lines are collected into one overall equation system, e.g.:

“Master line → Branch line 1 → Branch line 2 → .... → Branch line NJ^2-1”

where NJ^2 is the number of J lines in block 2. It should be noted that in the equations for the cells next to the interface, base cells or secondary cells, the referred cell which is in the adjacent block is also included as an unknown on this line. This is to ensure that the implicit coupling is retained in the inversion procedure.

In the alternative situation, the line of unknowns is assembled going from the 2D block to the 1D block as shown in Figure 5.5. This line starts from a grid line in the two dimensional block, for example, ‘Branch line 1’.
**Coupling 1D and 2D Pressure-Correction Procedure**

When it reaches the interface it will continue on the exclusive line in block 1, i.e. the ‘Master line’. Because base cell $P$ on the master line is linked to all secondary cells $P^2_i$ during the interface treatment, all secondary cells should be taken as unknowns while assembling the master line in order to preserve the coupling information. This ensures that information communicates among blocks to avoid decoupling problem. However, this will necessitate the assembling of all other lines in block 2, i.e. any other grid lines that are parallel to ‘Branch line 1’. Equations collected for ‘Branch line 1’ are in a sequence from cell $(NJ^2+1, 2)$ to cell $(1, 2)$. The equations for other branch lines in block 2 are ordered in the same way. All the branch lines and the master line are organised in one overall equation system in the sequence

"Branch line 1 $\rightarrow$ Branch line 2 $\rightarrow$ .... $\rightarrow$ Branch line $NJ^2-1$ $\rightarrow$ Master line"

Clearly, when solving 1D/2D coupled model problem, multiple lines instead of one line are collected during the transformation of the pentadiagonal matrix problem to a tridiagonal one. On each of these multiple lines, the cells on that line are treated as unknowns. In addition, the cells which are in the block adjacent to the block containing this line are also included as unknowns. The reduced equation system is characterised by a banded tridiagonal matrix structure with the exception of the equations for the base and secondary cells. As a result, tridiagonal matrix inversion algorithm TDMA needs also to be modified.


5.3.5.2 Modified TDMA Algorithm

The matrix of the equation system generated from the SLOR assembling process is not a truly banded tridiagonal structure in a 1D/2D coupled model. However, the standard TDMA algorithm can be modified to accommodate those non-tridiagonal parts in the matrix, making use of the inherent relations between matrix entries.

Consider firstly the situation in Figure 5.4, it can be represented by a simplified structure as illustrated in Figure 5.6

![Figure 5.6](image-url)

**FIGURE 5.6** Demonstration of the modification of TDMA solver

The SLOR line sweep starts from the real boundary point in the 1D zone, e.g. point 00, and progresses towards the 2D zone. The line first runs across the 1D zone until it arrives at the zonal internal boundary, from there the master line begins to spread out into branch lines. Multiple lines are launched in the 2D zone from the zonal boundary. Each originates from a point next to the interface and ends at a real boundary point. It should be noted that the zonal boundary points are excluded from these lines because they are not interior points. Each of these lines in the 2D zone is organised and operated as a block unit, for example, points 1, 1E and 1EE constitute one block unit, so do points 2, 2E and 2EE, etc.

The reduced matrix problem takes the form of \( \overline{A} \overline{X} = \overline{B} \) with
Coupling 1D and 2D Pressure-Correction Procedure

\[
\bar{\lambda} = \begin{bmatrix}
    0^W & 0^W & -a_0^W \\
a_{0W} & -a_0 & -a_1 & -a_2 & -a_3 \\
-a_0^1 & a_1 & -a_1^E & a_{1E} & a_{1EE} \\
-1E & a_{1E} & -a_{1EE} & 1EE & 1EE \\
a_E & a_{1EE} & -a_1E & -a_1EE & -a_1E \\
-a_0^2 & a_2 & -a_2^E & a_{2E} & a_{2EE} \\
-a_2 & a_{2E} & -a_{2EE} & 2EE & 2EE \\
-2EE & 2EE & 2EE & 2EE & 2EE \\
-a_0^3 & a_3 & -a_3^E & a_{3E} & a_{3EE} \\
-a_3 & a_{3E} & -a_{3EE} & 3EE & 3EE \\
-a_3^E & a_{3EE} & -a_{3EE} & 3EE & 3EE \\
-3EE & 3EE & 3EE & 3EE & 3EE
\end{bmatrix}
\]

\[5.59\]

\[
\bar{X} = \begin{bmatrix}
\phi_{0W} \\
\phi_0 \\
\phi_1 \\
\phi_{1E} \\
\phi_{1EE} \\
\phi_2 \\
\phi_{2E} \\
\phi_{2EE} \\
\phi_3 \\
\phi_{3E} \\
\phi_{3EE}
\end{bmatrix}
\]

\[
\bar{B} = \begin{bmatrix}
S^{0W} + a_{00} \cdot \phi_{00} \\
S^0 \\
S^1 \\
S^{1E} \\
S^{1EE} + a_{11} \cdot \phi_{11} \\
S^2 \\
S^{2E} \\
S^{2EE} + a_{22} \cdot \phi_{22} \\
S^3 \\
S^{3E} \\
S^{3EE} + a_{33} \cdot \phi_{33}
\end{bmatrix}
\]

\[5.60\]

where the superscript denotes the point for which the equation is dedicated, and the subscript denotes the point with which the coefficient is associated. For example, \(a_{3E}^2\) is the coefficient corresponding to point 3E in the equation for point 3.
**Coupling 1D and 2D Pressure-Correction Procedure**

Obviously matrix $\tilde{A}$ is not strictly a banded structure. However, a local quasi-banded structure can be identified corresponding to each block unit. The banded structure in each block unit is only interrupted at the point connecting the 1D/2D interface. Although the original TDMA algorithm cannot be applied directly to the matrix (5.59), the forward elimination and backward substitution procedure of TDMA can be retained in a modified algorithm.

In the forward process, the elimination operation is applied to each block unit in matrix $\tilde{A}$, that is each block unit is treated as an operation base. The entries below the diagonal are eliminated (set to zero) and terms on the diagonal are normalised to unity. As a result, matrix $\tilde{A}$ changes to

\[
\tilde{A} = \begin{bmatrix}
1 & -w(1) & 0 & -w(2) & -w(3) & -w(4) & -w(5) & -w(6) & -w(7) & -w(8) & -w(9) \\
-1 & -w(1) & 1 & -w(2) & -w(3) & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & -w(3) & 1 & -w(4) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(4) & 1 & -w(5) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(5) & 1 & -w(6) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(6) & 1 & -w(7) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(7) & 1 & -w(8) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(8) & 1 & -w(9) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(9) & 1 & -w(10) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(10) & 1 & -w(11) & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & -w(11) & 1 & -w(12) & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

where $w$, $w_1$ and $w_2$ are the work arrays. The source vector $\tilde{B}$ is changed accordingly to

\[
\tilde{B} = \begin{bmatrix}
f(1) \\
f(2) \\
f(3) \\
f(4) \\
f(5) \\
f(6) \\
f(7) \\
f(8) \\
f(9) \\
f(10) \\
f(11)
\end{bmatrix}
\]
**Coupling 1D and 2D Pressure-Correction Procedure**

where \( f \) is the array storing the modified source. The values of these arrays associated with the 1D zone points are

\[
w(1) = \frac{a_0^{0W}}{a_0^{0W}} \quad f(1) = \frac{S^{0W} + a_0^{0W} \cdot \varphi_{00}}{a_0^{0W}}
\]

\[
w_1(1) = \frac{a_1^0}{a_0^0 - a_0^{0W} \cdot w(1)} \quad f(2) = \frac{S^0 + a_0^0 \cdot f(1)}{a_0^0 - a_0^{0W} \cdot w(1)}
\]

Whereas the array for the 2D zone points in one block unit take the values

\[
w(3) = \frac{a_1^{1E}}{a_1^{1}} \quad f(3) = \frac{S^1}{a_1^{1}}
\]

\[
w(4) = \frac{a_{1E}^{1EE}}{a_{1E}^{1E} - a_1^{1} \cdot w(3)} \quad f(4) = \frac{S^{1E} + a_1^{1E} \cdot f(3)}{a_{1E}^{1E} - a_1^{1} \cdot w(3)}
\]

\[
f(5) = \frac{S^{1EE} + a_{11}^{1EE} \cdot \varphi_{11} + a_{1E}^{1EE} \cdot f(4)}{a_{1EE}^{1EE} - a_{1E}^{1EE} \cdot w(4)}
\]

Similar expressions for arrays \( w \) and \( f \) have also been obtained for other block units.

What is stored in arrays \( w_1 \) and \( w_2 \) are the entries away from the banded diagonal. For example, for one block unit

\[
w_1(2) = \frac{a_2^0}{a_0^0 - a_0^{0W} \cdot w(1)} \quad w_1(3) = \frac{a_3^0}{a_0^0 - a_0^{0W} \cdot w(1)}
\]
During backward substitution, the procedure is very different from that in the standard TDMA algorithm. An intermediate elimination is required.

First of all, within each block unit of the 2D zone, the upper half entries are subtracted to zero which gives

\[
\begin{bmatrix}
1 & -w(1) & \cdots & -w(11)
\end{bmatrix}
\begin{bmatrix}
t(1) \\
t(2) \\
t(3) \\
t(4) \\
t(5) \\
t(6) \\
t(7) \\
t(8) \\
t(9) \\
t(10) \\
t(11)
\end{bmatrix}
\]

where \(w_2'\) denotes that the content of the \(w_2\) array has changed, as does \(\tilde{f}\). By now the only non-zero entries in the matrix portion that corresponds to the 2D zone are either on the diagonal or on the coefficient associated with the base cell in the 1D zone.
Coupling 1D and 2D Pressure-Correction Procedure

The matrix row that corresponds to the equation for the base cell in the 1D zone is now considered. Array $w_j$ is eliminated to zero by the unit diagonals in the block units of the 2D zone

$$
\lambda = \begin{bmatrix}
1 & -w(1) \\
0 & \text{sum} & 0 & 0 & 0 \\
-w_2'(3) & 1 & 0 \\
-w_2'(4) & 0 & 1 & 0 \\
-w_2'(5) & 0 & 1 \\
-w_2'(6) & 0 & 1 & 0 \\
-w_2'(7) & 0 & 0 & 1 \\
-w_2'(8) & 0 & 0 \\
-w_2'(9) & 0 & 1 & 0 \\
-w_2'(10) & 0 & 1 & 0 \\
-w_2'(11) & 0 & 1 \\
\end{bmatrix}
$$

where

$$\text{sum} = 1 - w_1(1) \cdot w_2'(3) - w_1(2) \cdot w_2'(6) - w_1(3) \cdot w_2'(9)$$

$$\tilde{f}(2) = f(2) - w_1(1) \cdot \tilde{f}(3) - w_1(2) \cdot \tilde{f}(6) - w_1(3) \cdot \tilde{f}(9)$$

Now the variable value for point 0 can be obtained directly from:

$$\varphi_0 = \frac{\tilde{f}(2)}{\text{sum}}$$

variable values for all other points can be obtained by substitution:

$$\varphi_{0W} = f(1) + w(1) \cdot \varphi_0$$

$$\varphi_1 = \tilde{f}(3) + w_2'(3) \cdot \varphi_0$$

$$\varphi_{1E} = \tilde{f}(4) + w_2'(4) \cdot \varphi_0$$

$$\varphi_{1EE} = \tilde{f}(5) + w_2'(5) \cdot \varphi_0$$  

etc.

Examination of the alternative situation shown in Figure 5.5 indicates that it can be represented by the simplified structure as illustrated in Figure 5.7
**Coupling 1D and 2D Pressure-Correction Procedure**

![Diagram of 2D Pressure-Correction Procedure](image)

**FIGURE 5.7 Demonstration of the modification of TDMA solver**

Similar to the previous situation, multiple lines are assembled in the SLOR procedure. The line assembling starts from a real boundary point in the 2D zone, and runs towards the 1D zone. Within the 2D zone, the line is assembled as usual, but terminates at the interface. Once all the branch lines in the 2D zone have been collected, the master line in the 1D zone is assembled starting from the interface. On each line only the points on that line are treated as unknowns. However, point P which is the base cell in 1D and 2D model is treated as an unknown in all branch lines. Points 1, 2 and 3 are treated as unknowns on the master line. The equation system after SLOR line assembly becomes $AX = B$ where

\[
A = \begin{bmatrix}
1_{\text{WW}} & 1_{\text{WW}} & 0 & 0 & 0 \\
-\alpha_{\text{1W}} & 1_{\text{WW}} & -\alpha_1 & 0 & 0 \\
-\alpha_{\text{1W}} & a_{\text{1W}} & 1_{\text{WW}} & -\alpha_1 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
X = \begin{bmatrix}
-\alpha_{\text{2W}} & -\alpha_2 & 0 & 0 & 0 \\
-\alpha_{\text{2W}} & 2_{\text{WW}} & -\alpha_{\text{2W}} & 0 & 0 \\
-\alpha_{\text{2W}} & a_{2W} & 2_{\text{WW}} & -\alpha_2 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
-\alpha_{\text{3W}} & -\alpha_{\text{3W}} & 0 & 0 & 0 \\
-\alpha_{\text{3W}} & 3_{\text{WW}} & -\alpha_{\text{3W}} & 0 & 0 \\
-\alpha_{\text{3W}} & a_{3W} & 3_{\text{WW}} & -\alpha_3 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
E = \begin{bmatrix}
-\alpha_{\text{3W}} & -\alpha_{\text{3W}} & 0 & 0 & 0 \\
-\alpha_{\text{3W}} & 3_{\text{WW}} & -\alpha_{\text{3W}} & 0 & 0 \\
-\alpha_{\text{3W}} & a_{3W} & 3_{\text{WW}} & -\alpha_3 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Where $\alpha_i$ are the unknowns in the 1D zone.
Coupling 1D and 2D Pressure-Correction Procedure

\[
X = \begin{bmatrix}
\phi_{1w1} & \cdots & 1 \\
\phi_{1w} & \cdots & 2 \\
\phi_1 & \cdots & 3 \\
\phi_{2w1} & \cdots & 4 \\
\phi_{2w} & \cdots & 5 \\
\phi_2 & \cdots & 6 \\
\phi_{3w1} & \cdots & 7 \\
\phi_{3w} & \cdots & 8 \\
\phi_3 & \cdots & 9 \\
\phi_p & \cdots & 10 \\
\phi_p & \cdots & 11 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
S^{1w1} + a_{11} \cdot \phi_{11} \\
S^{1w} \\
S^1 \\
S^{2w1} + a_{22} \cdot \phi_{22} \\
S^{2w} \\
S^2 \\
S^{3w1} + a_{33} \cdot \phi_{33} \\
S^{3w} \\
S^3 \\
S^P \\
S^E + a_{EE} \cdot \phi_{EE} \\
\end{bmatrix}
\]

5.75

Again, a portion in matrix A which comprises the equations on one of the multiple lines is identified as a block unit in this matrix. The forward and backward operations in the modified TDMA algorithm are carried out on block unit basis.

During forward elimination, matrix A is firstly transformed to

\[
A = \begin{bmatrix}
1 -wk(1) & 0 & 1 -wk(2) & 0 & 1 -wk(3) \\
0 & 1 -wk(4) & 0 & 1 & 1 -wk(5) \\
0 & 1 & 0 & 1 & 1 -wk(6) \\
0 & 1 & 0 & 1 & 1 -wk(7) \\
-a_1 & -a_2 & -a_3 & -a_4 & 1 -wk(8) \\
-a_3 & -a_4 & -a_5 & -a_6 & 1 -wk(9) \\
-a_1 & -a_2 & -a_3 & -a_4 & 1 -wk(10) \\
-a_3 & -a_4 & -a_5 & -a_6 & 1 -wk(11) \\
\end{bmatrix}
\]

5.76
where \( wk \) is a work array. Then, the matrix row that is associated with the equation for the base cell \( P \) is manipulated so that the entries corresponding to the points in the 2D zone are set to zero, i.e.

\[
A = \begin{bmatrix}
1 -wk(1) \\
0 & 1 & -wk(2) \\
0 & 1 & -wk(3) \\
1 & -wk(4) \\
0 & 1 & -wk(5) \\
0 & 1 & -wk(6) \\
1 & -wk(7) \\
0 & 1 & -wk(8) \\
0 & 1 & -wk(9) \\
0 & 0 & 0 & \text{sum} & -wk(10) \\
\end{bmatrix}
\]

where

\[
\text{sum} = 1 + \frac{a_1}{a_p} \cdot (-wk(3)) + \frac{a_2}{a_p} \cdot (-wk(6)) + \frac{a_3}{a_p} \cdot (-wk(9))
\]

Subsequently, the block unit corresponding to the master line in the 1D zone is simplified to

\[
A = \begin{bmatrix}
1 -wk(1) \\
0 & 1 & -wk(2) \\
0 & 1 & -wk(3) \\
1 & -wk(4) \\
0 & 1 & -wk(5) \\
0 & 1 & -wk(6) \\
1 & -wk(7) \\
0 & 1 & -wk(8) \\
0 & 1 & -wk(9) \\
0 & 0 & 0 & 1 & \frac{-wk(10)}{\text{sum}} \\
\end{bmatrix}
\]
Coupling 1D and 2D Pressure-Correction Procedure

The backward substitution procedure is now carried out. Starting from the block unit corresponding to the master line in 1D zone, followed by the other block units corresponding to the branch lines in 2D zone.

The forward elimination and backward substitution in the above modified TDMA procedure has achieved information transfer among zones. During the forward process, flow information in the 2D zone is injected into 1D zone. This is performed by using equations from the 2D zone to manipulate the coefficients in the equation for the abutting interface cell in the 1D zone. During the backward process, the information flow is reversed. The flow is updated first in the 1D zone, then this changed flow state is input to the 2D zone to update the flow field there. A strong coupling of the flow in the whole domain is hence preserved in the stage of matrix inversion procedure.

5.4 Interface Evaluation in Energy Equation Discretisation

The energy equation (2.3) has been added to the governing equation system in the unified pressure-correction algorithm. Two different types of flux can be identified in the semi-discretised form of the equation, i.e. the convective flux and the diffusive flux. In the following, the evaluation of these fluxes on the interface during the discretisation of the energy equation is briefly summarised for both explicit and implicit coupling scheme.

5.4.1 Convective Flux

The convective flux in the energy equation is treated in a similar manner to that in the momentum equation. In the explicit interface treatment, this flux on an interface segment $cf$ is evaluated by

$$\left((\rho U)_{(cf)}^{k+1}\right) = \left((\rho U)_{(cf)}^k\right) \cdot \left(H_{(cf)}^k\right)$$

5.79

where

$$\left((\rho U)_{(cf)}^k\right) = \frac{1}{2}[(\rho U)^k_p + (\rho U)^k_{p^2}]$$

5.80
**Coupling 1D and 2D Pressure-Correction Procedure**

and

\[
(H)_{(cf)}^k = \frac{1}{2}((H)^k_{(p)} + (H)^k_{(p')})
\]  

5.81

On the whole interface m-m', this flux is evaluated in the same way as in equation (5.79), rather than applying a conservation law due to a similar conflict problem as explained for the momentum equation. The mass flux is derived from a conservation law, but enthalpy is estimated from an area weighted average, i.e.

\[
H^k_{m-m'} = \frac{1}{\text{area}_{(m-m')}} \sum_j \text{area}_{(cf_j)} \cdot [(H)^k_{(cf_j)}]
\]  

5.82

In the implicit interface treatment, the convective flux on interface segment 'cf' is approximated by

\[
(\rho U H)^{k+1}_{cf} = (\rho U)^k_{cf} \cdot (H)^{k+1}_{cf}
\]  

5.83

Mass flux \((\rho U)^{k}_{cf}\) is evaluated using the expression in (5.80), the dependent variable \(H^{k+1}_{cf}\) is interpolated by the scheme applied to the internal cell face, i.e.

\[
(H)^{k+1}_{(cf)} = \chi \{ (H)^{k+1}_{(p^1)}, (H)^{k+1}_{(p^2)}, \text{etc} \}
\]  

5.84

On the whole interface m-m', this flux is evaluated directly from a conservation expression

\[
[\rho U H]^{k+1}_{(m-m')} = \sum_j [\rho U H]^{k+1}_{(cf_j)}
\]  

5.85
Coupling 1D and 2D Pressure-Correction Procedure

as for the convective momentum flux.

5.4.2 Other Fluxes

The other interface flux, diffusion flux, is treated using a similar method as that for the viscous flux in the momentum equation. In the implicit coupling scheme, this flux on interface segments is evaluated following the method that is used on the interior cell face. On the whole interface, the conservation method, i.e. summing up all the evaluated flux on the interface segments, is used to derive them directly. In the explicit coupling scheme, this flux is calculated explicitly on both the interface segment and the whole interface from the interface variable values that have been already derived during the evaluation of the convective flux $\rho V_j H$. 
Chapter 6

Coupling 1D and 2D Block Implicit Procedure
6.1 Preamble

The general strategy of coupling 1D and 2D CFD models described in Chapter 4 has also been implemented within the block implicit solution procedure described in Chapter 3. The main purpose is to assess the generality of the coupling strategy. Since boundary conditions have been implemented implicitly in the block implicit algorithm, the 1D/2D interface is also treated implicitly as well as conservatively in order to preserve the integrity of the base algorithm. Because the discretisation scheme, the linearisation method and the solution algorithm in the block implicit procedure are different from those in the pressure-correction procedure, the actual implementation of the implicit interface coupling methodology is different in the two procedures.

Since all the governing equations in the block implicit algorithm can be represented in a single general form, and the implementation of an implicit interface coupling scheme is the same in all equations, the discussion in this chapter is restricted to just one example rather than on each individual equation.

6.2 General Equation Form

The general form of the governing equations (3.13) is rewritten here:

\[
\frac{\partial Q(q)}{\partial t} + \frac{\partial E(q)}{\partial \xi} + \frac{\partial F(q)}{\partial \eta} = 0
\]  

(6.1)

Consider a two dimensional control volume P as shown in Figure 6.1

![Figure 6.1 A control volume in finite volume discretisation method](image)

FIGURE 6.1 A control volume in finite volume discretisation method
Implicit finite volume discretization transforms equation (6.1) to

$$\frac{1}{\Delta t} \cdot (Q_p^{n+1} - Q_p^n) + (E_{ef} - E_{w_f})^{n+1} + (F_{nf} - F_{w_f})^{n+1} = 0 \quad 6.2$$

Note that $\Delta \xi, \Delta \eta$ are dropped because they are assumed unity.

Flux functions $E$ and $F$ can be broken down into a combination of flux components. For example, the $\xi$ direction flux $E$ on the east face of control volume $P$ is made up of

$$E_{ef} = (E^c)_{ef} + \sum_j b_j \left( \frac{\partial E_j^{v-n}}{\partial \xi} \right)_{ef} + \sum_j c_j \left( \frac{\partial E_j^{v-c}}{\partial \eta} \right)_{ef} \quad 6.3$$

where $E^c$ denotes the convective flux component, $E^{v-n}$ and $E^{v-c}$ denote respectively normal and cross diffusive viscous flux components, and $b_j$ and $c_j$ are the coefficients associated with various diffusive flux components. Similarly, the $\eta$ direction flux $F$ on the north face of control volume $P$ can be written

$$F_{nf} = (F^c)_{nf} + \sum_j b_j \left( \frac{\partial F_j^{v-n}}{\partial \eta} \right)_{nf} + \sum_j c_j \left( \frac{\partial F_j^{v-c}}{\partial \xi} \right)_{nf} \quad 6.4$$

All cell face quantities in equations (6.3) and (6.4) are evaluated under the assumption that flow variables are interpolated linearly in any direction. For example:

$$\frac{(E^c)_{ef}}{2} = \frac{1}{2} \cdot [(E^c)_p + (E^c)_E] \quad 6.5$$

$$\left( b_j \frac{\partial E_j^{v-n}}{\partial \xi} \right)_{ef} = (b_j)_{ef} \cdot [(E_j^{v-n})_E - (E_j^{v-n})_p] \quad 6.6$$
Coupling 1D and 2D Block Implicit Procedure

\[
\left( c_j \cdot \frac{\partial E^w_c}{\partial n} \right)_{ef} = (c_j)_{ef} \cdot \frac{1}{4} \left[ (E^v_c)_{NE} + (E^v_c)_N - (E^v_c)_{SE} - (E^v_c)_S \right]
\]

and similarly for the coefficients:

\[
(b_j)_{ef} = \frac{1}{2} \cdot [(b_j)_p + (b_j)_E] 
\]

\[
(c_j)_{ef} = \frac{1}{2} \cdot [(c_j)_p + (c_j)_E] 
\]

As a result, the semi-discretised equation (6.2) can be fully discretised as

\[
\frac{1}{\Delta t} \cdot (Q_p - Q^n_p) + \left\{ \frac{1}{2} \left[ (E^c)_p + (E^c)_E \right] - \frac{1}{2} \left[ (E^c)_w + (E^c)_P \right] \right\} 
\]

\[
+ \sum_j \left\{ (b_j)_{ef} \cdot [(E^v-n)_E - (E^v-n)_p] - (b_j)_{wf} \cdot [(E^v-n)_p - (E^v-n)_w] \right\} 
\]

\[
+ \sum_j \left\{ (c_j)_{ef} \cdot \frac{1}{4} \left[ (E^v-c)_{NE} + (E^v-c)_N - (E^v-c)_{SE} - (E^v-c)_S \right] 
\]

\[
- (c_j)_{wf} \cdot \frac{1}{4} \left[ (E^v-c)_{NW} + (E^v-c)_N - (E^v-c)_{SW} - (E^v-c)_S \right] \right\} 
\]

\[
+(F_{nf} - F_{sf}) = 0
\]

where \( F_{nf} \) and \( F_{sf} \) can be similarly expressed as for \( E_{ef} \) and \( E_{wf} \). Superscript ‘n+1’ for the current time level is dropped for simplicity.

According to Chapter 3, any nonlinear terms in equation (6.10) are linearised based on the Taylor expansion series, for example:
Coupling 1D and 2D Block Implicit Procedure

\[
(E^c_p)^{k+1} = (E^c_p)^k + \left[ \left( \frac{\partial E^c}{\partial p} \right)_p (p^{k+1} - p^k) \right]_p + \left[ \left( \frac{\partial E^c}{\partial u} \right)_p (u^{k+1} - u^k) \right]_p \\
+ \left[ \left( \frac{\partial E^c}{\partial v} \right)_p (v^{k+1} - v^k) \right]_p + \left[ \left( \frac{\partial E^c}{\partial T} \right)_p (T^{k+1} - T^k) \right]_p
\]

where superscript 'k' denotes the Newton iteration level.

After linearisation, the discrete algebraic equation becomes

\[
\begin{bmatrix}
 p \\
 u \\
 v \\
 T
\end{bmatrix}_p^{k+1} + \sum_{NB} \begin{bmatrix}
 (a_p, a_u, a_v, a_T)_p \\
 (a_p, a_u, a_v, a_T)_p
\end{bmatrix}_{NB} = \text{src}
\]

where subscript 'NB' denotes the neighbouring cells to cell P, such as cell ‘E’, ‘NW’ etc. in Figure 6.1. The circular bracket indicates a row vector containing the coefficients for each dependent variable. The term on the right hand side is the explicit source term consisting of flow quantities evaluated at the last time level n or the last Newton iteration step k.

A 1D/2D coupled model is shown in Figure 6.2

FIGURE 6.2 The control volumes in the 1D&2D coupled model
**Coupling 1D and 2D Block Implicit Procedure**

From the appearance, due to the difference between the 1D/2D model interface and ordinary cell faces, special methods are needed to evaluate the interface flow quantities in the semi-discretised equation (6.2).

However, with the creation of fictitious cells such as \( P_i \) based on the ‘base cell concept’ method, the interface segment ‘\( c_{fi} \)’ can be evaluated by the same method as that applied at an ordinary cell face. For example:

\[
E_{cf_i} = (E^c)_{cf_i} + \sum_j \left( b_j \cdot \frac{\partial E^v-n}{\partial \xi} \right)_{cf_i} + \sum_j \left( c_j \cdot \frac{\partial E^v-c}{\partial \eta} \right)_{cf_i}
\]

and

\[
(E^c)_{cf_i} = \frac{1}{2} \cdot [(E^c)_{p_i} + (E^c)_{p_i}]
\]

\[
\left( b_j \cdot \frac{\partial E^v-n}{\partial \xi} \right)_{cf_i} = (b_j)_{cf_i} \cdot [(E^v-n)_{p_i} - (E^v-n)_{p_i}]
\]

\[
\left( c_j \cdot \frac{\partial E^v-c}{\partial \eta} \right)_{cf_i} = (c_j)_{cf_i} \cdot \frac{1}{4} [(E^v-c)_{p_{i-1}} + (E^v-c)_{p_{i+1}} - (E^v-c)_{p_{i-1}} - (E^v-c)_{p_{i+1}}]
\]

The fictitious cell variables are replaced by the base cell variables following an assumption for the variable distribution within the base cell. For instance, a piecewise constant profile leads to:

\[
(b_j)_{cf_i} = \frac{1}{2} \cdot [(b_j)_{p_i} + (b_j)_{p_i}]
\]

\[
(c_j)_{cf_i} = \frac{1}{2} \cdot [(c_j)_{p_i} + (c_j)_{p_i}]
\]
Coupling 1D and 2D Block Implicit Procedure

\[(E^c)p_i = (E^c)\bar{p} \quad (E_j^{y-n})p_i = (E_j^{y-n})\bar{p} \quad (E_j^{y-c})p_{i+1} = (E_j^{y-c})p_{i-1} = (E_j^{y-c})\bar{p}\]

6.19

\[(b_j)p_i = (b_j)\bar{p} \quad (c_j)p_i = (c_j)\bar{p}\]

6.20

The substitution of the above expressions for the flux on interface segments will transform equation (6.2) for cell \(P_i\) to the following discrete equation:

\[
\frac{1}{\Delta t} \cdot (Q_{P_i}^n - Q_{P_i}^n) + \left\{ \frac{1}{2} [(E^c)_{P_i} + (E^c)_{E_i}] - \frac{1}{2} [(E^c)_{\bar{P}} + (E^c)_{P_i}^n] \right\} + \sum_j \left\{ (b_j)_{ef} \cdot \left[ (E_j^{y-n})_{E_i} - (E_j^{y-n})_{P_i} - (b_j)_{ef} \cdot \left[ (E_j^{y-n})_{P_i} - (E_j^{y-n})_{\bar{P}} \right] \right\} 

\[
+ \sum_j \left\{ (c_j)_{ef} \cdot \frac{1}{4} [(E_j^{y-c})_{E_i} + (E_j^{y-c})_{P_{i+1}} + (E_j^{y-c})_{P_{i-1}} - (E_j^{y-c})_{P_i}^n - (E_j^{y-c})_{P_i}^n] \right\}

\[-(c_j)_{ef} \cdot \frac{1}{4} [(E_j^{y-c})_{P_{i+1}} + (E_j^{y-c})_{P_{i-1}} - (E_j^{y-c})_{P_i}^n - (E_j^{y-c})_{P_i}^n] \}

6.21

\[+ (F_{ef} - F_{ef}) = 0\]

The linearisation method described by equation (6.11) is then applied to the nonlinear terms in the above equation. The final algebraic equation can be expressed as:

\[
(a_p, a_u, a_v, a_T)_{P_i}^{p_{k+1}} + \sum_{NB(2)} (a_p, a_u, a_v, a_T)_{NB(2)}^{p_{k+1}}

6.22

+ \sum_{NB(1)} (a_p, a_u, a_v, a_T)_{NB(1)}^{p_{k+1}} = src

169
Coupling 1D and 2D Block Implicit Procedure

where NB(1) and NB(2) denote the neighbouring cells in the one dimensional and two dimensional zones respectively. Clearly equation (6.22) is different from that for an ordinary two dimensional cell, i.e. equation (6.12). Both the two dimensional and one dimensional cells are referred to in equation (6.22). As explained in Chapters 4 and 5, the communication and flow coupling between the 1D and 2D blocks is thus ensured.

The flow quantities on the whole interface m-m’ are evaluated according to the conservation principle. That is, the flux on m-m’ is obtained by summing up the corresponding flux across all interface segments in the abutting two dimensional block. For example,

\[ E_{m-m'} = (E^c)_{m-m'} + \sum_j \left( b_j \cdot \frac{\partial E_j^{v-n}}{\partial \xi} \right)_{m-m'} + \sum_j \left( c_j \cdot \frac{\partial E_j^{v-c}}{\partial \eta} \right)_{m-m'} \quad 6.23 \]

and

\[ (E^c)_{m-m'} = \sum_i (E^c)_{cf_i} = \sum_i \frac{1}{2} [(E^c)_p + (E^c)_{p_i}] \quad 6.24 \]

\[ \left( b_j \cdot \frac{\partial E_j^{v-n}}{\partial \xi} \right)_{m-m'} = \sum_i \left( b_i \cdot \frac{\partial E_i^{v-n}}{\partial \xi} \right)_{cf_i} = \sum_i (b_i)_{cf_i} \cdot [(E_i^{v-n})_{p_i} - (E_i^{v-n})_p] \quad 6.25 \]

\[ \left( c_j \cdot \frac{\partial E_j^{v-c}}{\partial \eta} \right)_{m-m'} = \sum_i \left( c_i \cdot \frac{\partial E_i^{v-c}}{\partial \eta} \right)_{cf_i} \quad 6.26 \]

\[ = \sum_i (c_i)_{cf_i} \cdot \frac{1}{4} [(E_i^{v-c})_{p_i^{i+1}} + (E_i^{v-c})_p - (E_i^{v-c})_{p_i^{i+1}} - (E_i^{v-c})_p] \]

which is equivalent to

\[ E_{m-m'} = \sum_i E_{cf_i} \quad 6.27 \]
When the above expressions are substituted into equation (6.2) for cell \(\overline{P}\), the following discrete equation results:

\[
\frac{1}{\Delta t} \cdot (Q_p - Q^n_p) + \left\{ \sum_i \frac{1}{2} [(E^c)^{p_i} + (E^c)^{W_i}] - \frac{1}{2} [(E^c)^{p} + (E^c)^{W}] \right\} + \sum_j \left\{ \sum_i (b_i)_{cf_i} \cdot [(E^v_{-n})^{p_i} - (E^v_{-n})^{p}] - (b_i)_{wf} \cdot [(E^v_{-n})^{p} - (E^v_{-n})^{W}] \right\} 
\]

\[
+ \sum_j \left\{ \sum_i (c_i)_{cf_i} \cdot \frac{1}{4} [(E^v_{-c})^{p_i} + (E^v_{-c})^{p_i+1}] - (E^v_{-c})^{p} - (E^v_{-c})^{p_i} - (E^v_{-c})^{p_i+1} \right\} 
\]

\[
- (c_i)_{wf} \cdot \frac{1}{4} [(E^v_{-c})^N + (E^v_{-c})^{NW} - (E^v_{-c})^S - (E^v_{-c})^{SW}] \right\}
\]

\[
+ (F_{m-m'} - F_{wf}) = 0
\]

Applying the same linearisation as before, a linear algebraic equation is obtained:

\[
(a_p, a_u, a_v, a_T)_{\overline{P}}^{k+1} + \sum_{NB(1)} (a_p, a_u, a_v, a_T)_{NB(1)} + \sum_{NB(2)} (a_p, a_u, a_v, a_T)_{NB(2)} = \text{src}
\]

again NB(1) denotes the neighbours in the one dimensional zone, while NB(2) denotes the neighbours in the two dimensional zone. Clearly, all two dimensional cells which are next to the interface have been introduced and this marks the major difference to the algebraic equation for an ordinary one dimensional cell. As explained in Chapter 5, the multiple linking coefficients associated with cells in the adjacent block are the key to strong coupling between the 1D and 2D models. In the pressure-correction algorithm, only one
dependent variable at a cell is unknown in each equation, whereas here all dependent variables at a cell are unknowns in every equation.

The coupling inherent in both equation (6.22) and equation (6.29) implies that flow information in any zone will influence flow evolution in all adjacent zones in addition to this zone itself. The impact between zones is immediate, instant and global due to the implicit correlation preserved by the implicit coupling method.

6.3 Interface Treatment of Artificial Dissipation Terms

As explained in Chapter 3, Jameson-type mixed 2nd and 4th order artificial dissipation terms have been added to all governing equations in the block implicit algorithm in order to suppress numerical oscillation associated with the central differencing scheme. Since the central discretisation method is also employed in the evaluation of flow quantities on the 1D/2D interface, this artificial dissipation model needs to be introduced as well at the interface during the course of interface flux approximation.

On a cell face, the artificial dissipation flux has been appended to the physical face flux in a similar way to that described by equation (3.24) and (3.25), and the definition is given in equation (3.28).

On the interface segment ‘cf’, with the presence of fictitious cells in the one dimensional block, the introduction of the artificial dissipation flux to the interface segment can be carried out in the same way as that on an internal cell face, i.e.

\[
E_{cf}^{av} = E_{cf} - \overline{D}_{cf}
\]  

6.30

and the artificial dissipation flux is defined by

\[
\overline{D}_{cf} = \frac{J_{cf}^{-1}}{\Delta t_{cf}} \left[ e_{cf}^{(2)} (\phi_{p2} - \phi_{p1}) - e_{cf}^{(4)} (\phi_{E_i} - 3\phi_{p2} + 3\phi_{p1} - \phi_{W_i}) \right]
\]  

6.31
Coupling 1D and 2D Block Implicit Procedure

where $\phi$ is the conserved variable in the corresponding governing equation. If piecewise constant variable distribution is assumed over any base cells, equation (6.31) becomes

$$\overline{D}_{cf_i} = \frac{J_{cf_i}^{-1}}{\Delta t^*_{cf_i}} \left[ \varepsilon_{cf_i}^{(2)}(\phi_{p_i^2} - \phi_p) - \varepsilon_{cf_i}^{(4)}(\phi_{E_i^2} - 3\phi_{p_i^2} + 3\phi_p - \phi_{W_i}) \right]$$ \hspace{1cm} 6.32

The scaling factor $J^{-1}$ is approximated by

$$J_{cf_i}^{-1} \approx \frac{1}{2} \cdot (J_{p}^{-1} + J_{p_i}^{-1})$$ \hspace{1cm} 6.33

The nonlinear factors adjusting the amount of 2$^{\text{nd}}$ and 4$^{\text{th}}$ order dissipation are defined as

$$\varepsilon_{cf_i}^{(2)} = \max(\theta_{p_i^2}, \theta_p)$$ \hspace{1cm} 6.34

$$\varepsilon_{cf_i}^{(4)} = \max\left\{0, (k^{(4)} - \varepsilon_{cf_i}^{(2)})\right\}$$ \hspace{1cm} 6.35

where the oscillation detectors are given by

$$\theta_{p_i^2} = k^{(2)} \left| \frac{p_{E_i^2} - 2p_{p_i^2} + p_{p_i^2}}{p_{E_i^2} + p_{p_i^2} + p_{p_i^2}} \right| = k^{(2)} \left| \frac{p_{E_i^2} - 2p_{p_i^2} + p_p}{p_{E_i^2} + p_{p_i^2} + p_p} \right|$$ \hspace{1cm} 6.36

$$\theta_p = k^{(2)} \left| \frac{p_{E_i^2} - 2p_{p_i^2} + p_{p_i^2}}{p_{E_i^2} + p_{p_i^2} + p_{p_i^2}} \right| = k^{(2)} \left| \frac{p_{E_i^2} - 2p_{p_i^2} + p_p}{p_{E_i^2} + p_p + p_W} \right|$$ \hspace{1cm} 6.37

Obviously, reference to more cells in the one dimensional zone has been introduced in the discretised equation for the two dimensional control volume $P_i^2$ due to the addition of 4$^{\text{th}}$ order artificial dissipation terms. Consequently, the coupling between 1D and 2D model is further strengthened at the fundamental level in the equations.
The whole interface m-m' in Figure 6.2 has been evaluated in a conservative way. When
the artificial dissipation flux is added to the physical flux at the interface segments, this
should also be included in the flux at m-m' to ensure true interface flux conservation.
Therefore the flux on m-m' is evaluated by

\[ E_{m-m'}^{av} = \sum_i (E_{cf_i} - \vec{D}_{cr_i}) = \sum_i E_{cf_i}^{av} \]  

and similarly for \( F_{m-m'} \).

As a result, in the algebraic equation for cell \( \bar{P} \) in Figure 6.2, the multiple linking terms
associated with the cells in the two dimensional zone involve not only the cells
immediately next to the interface but also the cells that are one cell distance away from the
interface, hence the penetration into the two dimensional zone is further enhanced
compared to the situation when artificial dissipation terms are not present. This again
strengthens the coupling of the two flow regions.

6.4 Gauss-Seidel Point Iterative Solver

The form of the discrete algebraic equation for the control volume that is immediately next
to the 1D/2D block interface is different from the general form for the other interior cells.
This change of equation structure has led to the modification of the SLOR and TDMA
algorithms in the pressure-correction procedure as illustrated in Chapter 5. However, the
Gauss-Seidel iterative matrix solver employed by the block implicit procedure can be
applied in its original format without any modification while solving for the hybrid 1D/2D
coupled model. This is primarily due to the point nature of this solver. When visiting a
control volume during the Gauss-Seidel iterative procedure and updating the flow state
there, the flow conditions at any other control volumes are treated as known. In other
words, all implicit terms in the algebraic equation of this control volume except the ones
associated with the current cell are temporarily treated as explicit source terms, and moved
to the right hand side of equation. The number and nature of the cells referred to in the
equation has no affect on how the G-S procedure is carried out. Hence the special equation
**Coupling 1D and 2D Block Implicit Procedure**

form for the cells abutting the interface is not important from the view point of the G-S procedure.

As an example, if equation (6.12) for an ordinary interior cell $P$ is solved in the Gauss-Seidel procedure as

\[
(a_p, a_u, a_v, a_T)_P = \text{src} - \sum_{NB} (a_p, a_u, a_v, a_T)_{NB}^\text{imp} \left[ \begin{array}{c} p \\ u \\ v \\ T \end{array} \right]_{P}^{\text{imp}} - \sum_{NB} (a_p, a_u, a_v, a_T)_{NB}^\text{exp} \left[ \begin{array}{c} p \\ u \\ v \\ T \end{array} \right]_{NB}^{\text{exp}}
\]

where superscript of the matrix denotes the implicitly (imp) or explicitly (exp) treated terms and the block matrix problem represented by the above equation is subsequently inverted by Gaussian elimination to obtain solutions for all dependent variables at this cell, then equation (6.22) for the cells next to the interface in the two dimensional block would be solved as:

\[
(a_p, a_u, a_v, a_T)_P = \text{src} - \sum_{NB(2)} (a_p, a_u, a_v, a_T)_{NB(2)} \left[ \begin{array}{c} p \\ u \\ v \\ T \end{array} \right]_{P}^{\text{imp}} - \sum_{NB(1)} (a_p, a_u, a_v, a_T)_{NB(1)} \left[ \begin{array}{c} p \\ u \\ v \\ T \end{array} \right]_{NB(1)}^{\text{exp}}
\]

and equation (6.29) for the cell next to the interface in the one dimensional block by:
Coupling 1D and 2D Block Implicit Procedure

\[
\begin{pmatrix}
    p \\
    u \\
    v \\
    T
\end{pmatrix}^{\text{imp}}_p = \text{src}
\]

\[
- \sum_{NB(1)} (a_p, a_u, a_v, a_T)_{NB(1)} \begin{pmatrix}
    p \\
    u \\
    v \\
    T
\end{pmatrix}^{\text{exp}}_{NB(1)} - \sum_{NB(2)} (a_p, a_u, a_v, a_T)_{NB(2)} \begin{pmatrix}
    p \\
    u \\
    v \\
    T
\end{pmatrix}^{\text{exp}}_{NB(2)}
\]

The block matrix problem in (6.40) and (6.41) which is similar to that in equation (6.39) is inverted similarly. Hence, the G-S procedure is no different whether it is applied to a normal cell or the special cells next to the interface in a 1D/2D coupled model.
Chapter 7

Results and Discussion
Results and Discussion

7.1 Preamble

It is well known that any algorithm should be thoroughly validated via proper benchmark problems before being applied as a predictive tool. Therefore, in this work, several benchmark problems have been carefully selected to fulfil this requirement. The choice of test cases is based on the relevance of the flow characteristics contained in them to those contained in the IC engine flows of interest here.

The two solution procedures described above, i.e. block-implicit and pressure-correction algorithms, are aimed at engine exchange flow problems where compressible and incompressible flows could coexist within the same computational domain. Two categories of benchmark problems have been employed to examine this capability, one for high speed flows, the other for low speed flows. Comparisons are made for the relative performance of the two algorithms to examine their relative merits in computing flows at all speeds.

The propagation of compressible pressure waves in engine exchange flows is a key flow feature that must be captured accurately. Two benchmark problems have been selected to verify the ability of the two solution procedures to simulate the generation and transient propagation of pressure waves. One is a linear wave propagation problem, the other includes nonlinear effects.

The developed 1D/2D coupled model is intended to be substituted for a conventional uniformly 2D model in calculations of engine exchange flow, therefore the coupling methodology is also verified using pressure wave benchmark problems. The performance of the coupled model is compared to a 2D model in computing these flows to indicate the feasibility of the coupled modelling approach in engine application.

The numerical benchmark experiments conducted in this work are organised in this chapter in the following way. The results of the basic algorithms (for a 2D model) in simulating steady state flows, covering both compressible and incompressible problems, are presented first. The accuracy, convergence history and memory requirements of the numerical simulations are the main items for discussion. The results from computing unsteady benchmark problems are considered next, specifically for linear and nonlinear
Results and Discussion

travelling pressure waves. The accuracy in terms of waveform, wave amplitude as well as wave speed, and the computing efficiency are the main qualities under scrutiny. The calculated results for the same unsteady benchmark problems from a 1D/2D coupled model are illustrated next. Results are compared against not only the benchmark solutions but also the results of uniform 2D calculations. In addition, the coupling methods implemented in the two solution procedures are compared for relative performance.

These benchmark verifications build up confidence for the application of the 1D/2D coupled model to a realistic engine flow problem for which experimental measurements are available. The selected flow is axisymmetric (Kirkpatrick 1994). Traditionally this type of engine flow would be simulated either by the boundary condition approach where cylinder and pipe flows would be isolated or by a unified model approach where both cylinder and connecting pipe system are computed simultaneously using two dimensional grids for both. Blair et al. (1995) have developed the boundary condition method to predict this flow, the major drawback being the large amount of empirical inputs and corrections required (e.g. the input of discharge coefficients, the correction for heat transfer and frictional wall shear effects). If this problem is tackled via uniform dimensional modelling, the restriction to using homogenous dimensional grids would lead to a large number of cells which may not be necessary. With the coupling methodology developed in this work, unnecessarily large cell numbers can be avoided, as well as the empirical inputs and corrections. As a result, the computing time may be shortened and computing efficiency improved.

7.2 Steady-State Flow Calculations

7.2.1 Introduction

Subsonic and transonic inviscid compressible flow over a 10% thick two-dimension Ni bump (Ni 1982) has been chosen to evaluate the capability of the two algorithms in predicting high speed flows. The viscous incompressible flow in a two dimensional square cavity has been employed to test their ability in calculating low speed flows. Both problems are well suited for code development and testing because of their simple geometry, and they have previously been widely used for code verifications (Eidelman et
Results and Discussion

al. 1984, Allmaras 1997). However, they are challenging tests due to the complex flow features contained in them.

7.2.2 Compressible Flows over the Ni Bump

The Ni bump flow (Ni 1982) is the flow in a channel with a circular arc “bump” on its lower wall. The thickness-to-chord ratio of the circular arc bump is 10%, and the vertical width of the channel equals the length of the bump. The upstream and downstream boundaries of the computational domain are normally located at a distance of one bump length from the start and end points of the bump respectively.

Within the flow speed range of interest to this work, flows can be assumed inviscid for this test problem, therefore Euler equations instead of Navier-Stokes equations are solved. On the upper and lower channel walls, slip-wall boundary condition are imposed. At the inflow boundary of the channel, the total pressure and temperature of the upstream flow are fixed, and flow is assumed normal to the inlet cross section. At the outflow boundary, the static pressure of the downstream flow is imposed.

A structured grid with 257*65 cells is used in all calculations; this is of equal or greater density than the previous calculations in the literature (e.g. Eidelman 1984, Ni 1982) and hence good resolutions may be expected. The convergence criterion is that the maximum absolute residual of the discretised equations normalised by the maximum value of the dependent variable is less than 1.0e-5.

7.2.2.1 Subsonic Flow

When the incoming flow is at Mach 0.5, both codes have predicted a symmetric subsonic flow within the channel which agrees well with the benchmark result (Ni 1982, Eidelman 1984).

Figure 7.1 shows the iso-mach line pattern predicted by the pressure-correction algorithm, Figure 7.2 shows the prediction of the block implicit algorithm.
Results and Discussion

The symmetry of the patterns in both figures is obvious. The Mach number solutions are smooth, they are also in good quantitative agreement with the published results, see for example Ni (1982) and Eidelman (1984). All these are indications of good accuracy of the two algorithms, in particular the proper behaviour of the discretisation schemes used for the convection terms.
Results and Discussion

However, the shape of the isomach lines in Figure 7.2 is not perfectly circular in some local regions. This is probably to do with the spatial discretisation scheme employed in the block implicit procedure, i.e. central differencing with 2\textsuperscript{nd} and 4\textsuperscript{th} order artificial dissipation. The artificial dissipation coefficients used in this case were $e^{(2)} = 0.0$ $e^{(4)} = 0.01$ (i.e. switched off 2nd order artificial dissipation terms in the mixed smoothing model).

Figure 7.3 and Figure 7.4 show the Mach number distribution on the lower and upper wall surfaces.

![Mach Number on Lower and Upper Wall](image)

FIGURE 7.3 Surface Mach number for flow over Ni Bump at $M_{\infty} = 0.5$: Pressure-Correction Algorithm
Results and Discussion

Mach Number on Lower and Upper Wall

\[ \text{Mach Number on Lower and Upper Wall (freestream Mach number } = 0.5) \]

![Graph showing Mach number distribution](image)

FIGURE 7.4 Surface Mach number for flow over Ni Bump at \( M_{\infty} = 0.5 \): Block Implicit Algorithm

The symmetry of the patterns in both figures is again obvious. The Mach number solution are again smooth, and in good quantitative agreement with the published results, for example Ni (1982) and Eidelman (1984).

7.2.2.2 Transonic Flow

When the upstream inflow Mach number \( M_{\infty} \) is 0.675, flow solutions no longer remain subsonic throughout the channel; a supersonic region appears which is terminated by a shock. The flow becomes asymmetric over the bump, and the shock is located at around 72\% of the bump chord. The flow behind the shock is rotational and thus the isomach lines downstream of the shock no longer intersect the lower flat wall at right angles as in the previous subsonic bump flow case.

Figures 7.5 and 7.6 present the iso-mach lines predicted by the two solution algorithms, and Figures 7.7 and 7.8 show the Mach number distribution along the upper and lower walls.
Results and Discussion

FIGURE 7.5  Iso-mach lines for flow over Ni Bump at $M_{\infty} = 0.675$: pressure-correction algorithm

FIGURE 7.6  Iso-mach lines for flow over Ni Bump at $M_{\infty} = 0.675$: block implicit algorithm
Results and Discussion

Mach Number on Lower and Upper Wall
(freestream Mach number = 0.675)

FIGURE 7.7 Surface Mach number for flow over Ni Bump at \( M_{\infty} = 0.675 \): pressure-correction algorithm

Mach Number on Lower and Upper Wall
(freestream Mach number = 0.675)

FIGURE 7.8 Surface Mach number for flow over Ni Bump at \( M_{\infty} = 0.675 \): block implicit algorithm

185
Results and Discussion

Both algorithms have captured the shock at about 72% chord which agrees well with the published results. The strength of the shock is predicted at a value of about 1.38 by both algorithms, which is a slightly high compared to the 1.32 value reported in Allmaras’s paper (Allmaras 1997). Both algorithms have captured the shock within 4 cells, this may be considered acceptably sharp.

It is well known that the quality of shock capturing is predominantly influenced by the differencing scheme used for the convection terms and the treatment of derivatives. In the block implicit algorithm, Jameson’s type of mixed 4th and 2nd order artificial dissipation model (Jameson, et al. 1981) is used in conjunction with the central differencing. This dissipation model is a nonlinear smoothing method specially designed to help produce high quality shock capturing. Although the model was originally developed for an explicit compressible algorithm, the results from the present work demonstrate that it is also effective for a implicit algorithm. The optimum artificial viscosity in this case was found to be $\varepsilon^{(4)} = 1/32$ and $\varepsilon^{(2)} = 1.0$. These values are higher than the ones in Jameson’s original scheme, however, they are very close to the values used by Caughey (1988). In the pressure-correction algorithm, a second order TVD MUSCL differencing method has been implemented. The sharp shock solution obtained from this algorithm is largely the result of this differencing method. The shock is captured within 4 cells in the present application. However, in the literature, 3 cells or even 2 cells have been achieved. This probably has something to do with the retarded pressure treatment. As explained in Page’s thesis (1990), although the replacement of real pressure by retarded pressure was motivated by an incorrect physical behaviour of the pressure-correction equation in hyperbolic flow regions, the retarded pressure treatment behaves like a numerical smoothing mechanism. Because of the smearing effect of the retarded pressure, McGuirk and Page (1990) have proposed a formula which is intended to optimise the smoothing at the shock, it appears to work well with their first order differences and a staggered grid system. The same formula has been used in the current work, however it might not be the optimised one for the present non-staggered collocated grid system and TVD MUSCL differencing scheme. The slightly smeared shock in the present prediction may well be caused by the additional contribution to smoothing provided by the retarded pressure.
Results and Discussion

The maximum and minimum values of the Mach number in both solutions are close, especially the upper limit value which represents shock strength. This is an indication of good accuracy of both algorithms in the overall solutions.

It is noticed that the Mach number results of the pressure-correction algorithm oscillate slightly in the cross stream direction in the after-shock region, as shown in Figure 7.5. Since these tiny oscillations are very close to the wall, they possibly originate from the curved wall boundary treatment. While the boundary condition of zero normal pressure gradient on a slip wall has been implemented in a proper way taking into account the curvature of the wall, the boundary conditions of zero normal velocity as well as zero normal tangential velocity gradient at the slip wall have been implemented in a simplified manner which only represents the curvature approximately, and this is believed to give rise to the observed results. However, the same implementation for slip wall boundary condition have been adopted in the block implicit algorithm, and its Mach number results appear to be fine as shown in Figure 7.6. Therefore, the origin of the problem has not been completely identified but it was believed to be sufficiently small not to require closer investigation.

7.2.2.3 CPU Time and Memory Usage

The CPU time and memory on an SGI Octane workstation used by the calculations discussed so far are documented in Tables 7.1 and 7.2. The relative efficiency and memory requirement of the two solution algorithms in computing these flows can thus be compared.

<table>
<thead>
<tr>
<th>TABLE 7.1</th>
<th>CPU Time Usage of Bump Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach number</td>
<td>Mach 0.5</td>
</tr>
<tr>
<td>algorithm</td>
<td>block implicit</td>
</tr>
<tr>
<td>CPU time* per time step</td>
<td>6.3722</td>
</tr>
<tr>
<td>CFL number</td>
<td>10.0</td>
</tr>
<tr>
<td>no of time-steps to converge</td>
<td>2009</td>
</tr>
<tr>
<td>total CPU time*</td>
<td>3.56</td>
</tr>
</tbody>
</table>

* secs
Results and Discussion

The CFL number is defined by \((u+c)\Delta t/\Delta x\), where \(u\) is the inflow velocity, \(c\) is the speed of sound based on inlet conditions, and \(\Delta x\) is the mesh size near the channel inlet boundary. The value of the CFL number in the above table was the maximum allowable one for the block implicit calculations. The maximum CFL number for pressure-correction calculation is 12.

In computing both Mach number problems, the block implicit algorithm converges within a smaller number of iterations compared with the pressure-correction algorithm when the same CFL number is used, which is primarily due to the strong implicitness of the first algorithm. However, the computing time per time step is considerably more, as a result there is no gain in overall CPU time, in fact the algorithm is some 50% slower than the pressure-correction method.

Although in theory a time-step as large as infinity can be used in an implicit algorithm, there is a practical restriction in the present block implicit scheme imposed by the condition of coefficient matrix positivity (which is ensured only by matrix diagonal dominance) due to the use of an iterative linear matrix solver. The definition of diagonal dominance in a block matrix given in text books has sometimes been found impractical (Chen, et al. 1990), nevertheless, the diagonal dominance of each single equation always provides an assurance of numerical stability. Because the central differencing scheme is used in this solution procedure, the time term has to play an important role in ensuring and enhancing matrix diagonal dominance. Therefore, it is inevitable that some kind of restriction on the time step will be imposed.

It is also well known that the Gauss-Seidel point iterative solver used here is much slower to converge compared with other more implicit iterative solvers, e.g. line solver, SIP solver (Stone, 1968) etc., this is why the CPU time per iteration in the block implicit algorithm is long. The slow convergence property of the G-S point solver has been compensated by a multigrid acceleration technique in the present work, but results for this will be presented in a later section.
Results and Discussion

<table>
<thead>
<tr>
<th>TABLE 7.2 Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>block implicit</td>
</tr>
<tr>
<td>no. of cells</td>
</tr>
<tr>
<td>no. of words* per cell</td>
</tr>
<tr>
<td>Mbyte</td>
</tr>
</tbody>
</table>

* single precision 1 word = 32 bits

Each of the solution grids in the block implicit algorithm requires the storage of 236 variables whereas only 36 variables are required in the pressure-correction algorithm. As a result, the block implicit algorithm requires more than six times the memory of the pressure-correction algorithm.

It is not surprising that the block-implicit algorithm uses substantially more memory than the pressure-correction algorithm due to the simultaneous storage of coefficients for all equations, in contrast to the sequential pressure-correction procedure where only the coefficients for the currently solved equation are stored. Note that the memory requirement could have been even greater if the G-S point solver was not used. It is this attribute that has led to the use of the G-S point solver in this work.

7.2.2.4 Summary

Both block-implicit and pressure-correction algorithms have produced good quality predictions for subsonic and transonic bump flows. The shock in the transonic bump flow has been captured accurately and cleanly by both Central and TVD MUSCL differencing schemes implemented respectively in the block implicit and pressure-correction algorithms. However, in block-implicit solutions, the Mach number distribution in some local subsonic regions was not perfectly circular. In the pressure-correction solutions, small oscillations were observed in the after-shock region of the transonic bump flow.

Block-implicit calculations have required longer total CPU time than pressure-correction ones despite the smaller number of iterations for convergence. Therefore, the block-implicit algorithm is less cost effective compared to the pressure-correction algorithm considering its significantly larger memory usage.
Results and Discussion

7.2.3 Square Cavity: Incompressible Driven Cavity Flow

The laminar incompressible flow in a 2D square cavity whose top wall moves with a uniform velocity in its own plane has served many times as a model problem for testing and evaluating numerical techniques for low speed flow problems (in spite of the singularity). Predictions have been run in the present work for Reynolds numbers 100 and 1000, respectively, under isothermal wall conditions and a driven Mach number of 0.05. For these two Reynolds numbers, published results are available; here the results from Ghia et al. (1982) have been used as a benchmark for assessment.

Inside the cavity, the motion of the top wall sets up a complex vortex structure. At Reynolds numbers 100 and 1000, the flow consists of a primary vortex and two secondary vortices. Unlike the Ni Bump inviscid flows, the low Re cavity flows are viscous, therefore Navier-Stokes equations are to be solved.

A uniform mesh consisting of 129*129 cells has been used in all cavity calculations. This mesh arrangement is consistent with the grid used in Ghia's study. It should be noted that solutions are nondimensionalised values. The corresponding dimensional values will be given wherever relevant.

Generally, because stream function values in the secondary vortices are of considerably lower order compared to the ones in the primary vortex, the minor vortices embedded in the bottom corners of cavity are difficult to illustrate in the global stream function contour plot when the contour intervals are chosen based on the primary vortex. Therefore, in all figures for streamline patterns, the streamlines are “magnified” in the regions where secondary vortices occur, i.e. the contour scale used to display the secondary vortices are different from that for the primary vortex although all vortices are plotted on one graph. The scale for the secondary vortices are placed in all figures near the bottom right corner of the cavity, while the one for the primary vortex is placed near the top right corner.

7.2.3.1 Reynolds Number 100 Flow

Figures 7.9 and 7.10 show the stream-function contours for the Reynolds number 100 driven cavity flow from pressure-correction and block implicit solutions, respectively.
Results and Discussion

FIGURE 7.9 Streamline pattern for primary and secondary vortices: pressure-correction algorithm

FIGURE 7.10 Streamline pattern for primary and secondary vortices: block implicit algorithm
Results and Discussion

The patterns as well as the non-dimensional values of the stream-function in both figures are in excellent agreement with the results from Ghia et al. (1982) and Vanka (1986). (Note that the colour levels are slightly different in these two figures due to the different upper and lower limits which indicate the range of the stream function values. However, the good agreement between the two figures is still easy to identify.) The minor secondary vortices in the bottom left and bottom right regions of the cavity have been well resolved, their extent agrees well with that reported in the literature, indicating the good quality of both algorithms in simulating this benchmark flow.

In the block-implicit calculations, it should be noted that the artificial dissipation model added to the central differencing scheme has been switched off, however, the solutions are oscillation free. Such results have also been observed in the studies of Chen et al. (1991) and Shyy (1992). This is the result of small cell Peclet numbers because of the low overall Reynolds number and the use of a dense grid. The cell Peclet number is everywhere less than two for 129*129 grids (Shyy 1992). In the pressure-correction calculations, although the cell Peclet number is no different from that in the block-implicit calculation, however, the central differencing of both the first order pressure derivative in the momentum equation and the first order convection term in the continuity equation allows the existence of a decoupled checkboard pressure solution, and the Rhie&Chow pressure smoothing model has been used in the calculations to re-couple the mass flow directly to the pressure gradient.

Vanka (1986) has recorded the locations of all vortex centres for this cavity flow. Table 7.3 lists the nondimensionalised values of these locations from Vanka's work and the present work.

<table>
<thead>
<tr>
<th>Vortices</th>
<th>Primary vortex</th>
<th>Lower left vortex</th>
<th>Lower right vortex</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, y)</td>
<td>x</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>Vanka</td>
<td>0.6188</td>
<td>0.7375</td>
<td>0.0375</td>
</tr>
<tr>
<td>p-c results</td>
<td>0.6161</td>
<td>0.7355</td>
<td>0.0337</td>
</tr>
<tr>
<td>implicit results</td>
<td>0.6148</td>
<td>0.7426</td>
<td>0.0333</td>
</tr>
</tbody>
</table>

192
Results and Discussion

The primary vortex locations from the present algorithms are in very good agreement with Vanka’s, the difference is only about 0.5%. The locations of the secondary vortices predicted by the two current algorithms are very close, but they are noticeably different to Vanka’s results. However, Vanka has pointed out that the locations of his secondary vortices were different from the results of some other researchers (Ghia et al. 1982, Schreiber et al. 1983). He argued that the difference was inevitable because the values of the stream function in these vortices are so small, they are, in most cases, below the convergence accuracy of the calculation.

Figures 7.11 and 7.12 show the profiles of the cartesian velocity component $u$ along the vertical line through the geometrical centre of the cavity from the predictions of the pressure-correction and block implicit algorithms respectively. The data from Ghia et al. (1982) are used as a benchmark to evaluate the accuracy of the current results.

![U Velocity Profile](image)

FIGURE 7.11 Profile of $u$-velocity along vertical centre line: pressure-correction algorithm
Results and Discussion

The results of the pressure-correction algorithm are literally coincident at every sample point with Ghia's. The block implicit results are not totally coincident with Ghia's but very close.

Figures 7.13 and 7.14 show the predicted vertical velocity profile along the horizontal line through the geometrical centre of the cavity from the two algorithms. Ghia's results are again plotted in these two figures as benchmark data. Both predictions in the present work agree extremely well with the benchmark results.
Results and Discussion

V Velocity Profile
(along horizontal lines through geometric centre)

FIGURE 7.13 Profile of v-velocity along horizontal centre line: pressure-correction algorithm

FIGURE 7.14 Profile of v-velocity along horizontal centre line: block implicit algorithm
Results and Discussion

7.2.3.2 Reynolds Number 1000 Flow

Figures 7.15 and 7.16 show the contours of stream function for the Reynolds number 1000 driven cavity flow from pressure-correction and block implicit solutions.

![Streamline pattern for primary and secondary vortices: pressure-correction algorithm](image)

**FIGURE 7.15** Streamline pattern for primary and secondary vortices: pressure-correction algorithm
Results and Discussion

Non-dimensional Values

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001420</td>
</tr>
<tr>
<td>-0.054978</td>
</tr>
<tr>
<td>-0.111376</td>
</tr>
<tr>
<td>0.000710</td>
</tr>
<tr>
<td>0.000000</td>
</tr>
</tbody>
</table>

FIGURE 7.16 Streamline pattern for primary and secondary vortices: block implicit algorithm

These streamline patterns as well as the nondimensional values of stream function are again in excellent agreement with the results from either Ghia et al. (1982) or Vanka (1986). The minor secondary vortices at the two bottom corners have been captured by both algorithms. Their extent is in good agreement with that reported in the literature. The bottom left secondary vortex in the block implicit prediction is smaller than that in the pressure-correction prediction. The comparison of the two secondary vortices predicted in this work with the ones by Ghia et al. shows that the bottom right vortex has been resolved very well, however, the bottom left one is comparatively weaker. This is suspected to be caused by the different Mach number employed in the current calculation and Ghia’s. To examine it, another calculation has been run (using the pressure-correction algorithm), in which the speed of the top lid is the same as that in Ghia’s study. This time a very similar pattern (to Ghia’s) of bottom left vortex has been obtained as shown below in Figure 7.17. This indicates that the form of this secondary vortex is slightly influenced by the Mach number, but the overall streamline pattern is hardly affected as can be seen by comparing Figure 7.17 with Figure 7.15.
Results and Discussion

Similar to the Re = 100 computation, since the cell Peclet number in the Re = 1000 computation on a 129*129 grid is predominantly less than two (apart from the top right corner according to Shyy (1992)), the artificial dissipation in the block implicit calculation is set to zero.

Table 7.4 lists the nondimensionalised locations of the vortex centres from the current two algorithms and their comparison with the results of Vanka (1986).

<table>
<thead>
<tr>
<th>Vortices</th>
<th>Primary vortex</th>
<th>Lower left vortex</th>
<th>Lower right vortex</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, y) location</td>
<td>x</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>Vanka</td>
<td>0.5438</td>
<td>0.5625</td>
<td>0.075</td>
</tr>
<tr>
<td>p-c results</td>
<td>0.5339</td>
<td>0.5663</td>
<td>0.0865</td>
</tr>
<tr>
<td>implicit results</td>
<td>0.5503</td>
<td>0.5753</td>
<td>0.05039</td>
</tr>
</tbody>
</table>

FIGURE 7.17 Streamline pattern for primary and secondary vortices: Mach number=3.0e-3
**Results and Discussion**

The difference in the primary vortex locations between the pressure-correction result and Vanka’s is about 1.8% for the x location and 0.6% for the y location, the difference between the block-implicit result and Vanka’s is 1.2% for the x location and 2.3% for the y location. Hence the pressure-correction algorithm seems more accurate than the block-implicit algorithm in computing this driven cavity flow. These results indicate that the moving lid induced bulk recirculation in the cavity penetrates further into the pressure-correction prediction than that in the block implicit one.

Figures 7.18 and 7.19 show the \( u \) velocity profiles along the vertical line through the geometrical centre of the cavity from predictions of the pressure-correction and block implicit algorithms respectively. Again the results from Ghia’s study are used as a benchmark to assess these results.

![U Velocity Profile](image)

FIGURE 7.18 Profile of \( u \)-velocity along vertical centre line: pressure-correction algorithm
Results and Discussion

The pressure-correction results agree extremely well with Ghia’s. This demonstrates that the extended unified pressure-correction algorithm has not lost its good performance characteristics for low speed flow calculation. In contrast, the agreement between the block-implicit results and Ghia’s is not as good. The maximum negative velocity value in the implicit results is smaller than that of Ghia’s. This suggests that a weaker primary recirculation has been predicted by the block implicit algorithm and the recirculation has not penetrated as far as that in Ghia’s simulation, which is consistent with the finding of a comparatively higher location of the primary vortex centre than that in Vanka’s result.

Figures 7.20 and 7.21 show the vertical velocity v profiles on the horizontal line through the geometrical centre of the cavity from the two algorithms. They have again been compared with the benchmark results from Ghia et al. (1982)
Results and Discussion

FIGURE 7.20 Profile of $v$-velocity along horizontal centre line: pressure-correction algorithm

FIGURE 7.21 Profile of $v$-velocity along horizontal centre line: block implicit algorithm
Results and Discussion

Here the excellent agreement between the pressure-correction results and the benchmark is observed again. In the block-implicit results, while the results in the right half region agree reasonably with the benchmark data, a large discrepancy is found in the left half region of the cavity, and the maximum positive \( v \) velocity value is lower. The bottom and left region of the cavity are classified as the far field region for the driving cavity flow when the top wall is moving to the right, because they are the furthest regions from the top right corner. As listed in Table 7.5 that a much larger CFL number was employed in the pressure-correction calculation (about 20 times of the one used in the block implicit calculation), therefore the pressure-correction calculation has predicted a cavity flow at a much advanced time compared to the block-implicit prediction. Consequently, the cavity flow results from block-implicit may not be completely developed yet although numerical convergence has been reached.

7.2.3.3 CPU Time and Memory Usage Comparison

As in the bump flow calculations, the CPU time and memory used in the driven cavity flow calculations have been recorded and compared to assess the relative efficiency and cost of the two algorithms in simulating this incompressible Navier-Stokes flow.

In Table 7.5, the CPU time usage is listed for both flow computations. Also included are the CFL number in each calculation, the CPU time per time step and the number of time-steps to a converged solution.

<table>
<thead>
<tr>
<th>Reynolds number</th>
<th>Re 100</th>
<th>Re 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>block implicit</td>
<td>10.0</td>
<td>10.0</td>
</tr>
<tr>
<td>pressure correction</td>
<td>210.0</td>
<td>210.0</td>
</tr>
<tr>
<td>CFL number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU time per time step (sec)</td>
<td>4.7704</td>
<td>4.7704</td>
</tr>
<tr>
<td>no. of time-steps to converge</td>
<td>2779</td>
<td>5443</td>
</tr>
<tr>
<td>total CPU time (hrs)</td>
<td>3.68</td>
<td>7.21</td>
</tr>
</tbody>
</table>
Results and Discussion

The CFL numbers listed are computed as previously described, they are about the maximum values in the calculations, with the values in the pressure-correction calculation being considerably larger. This is because the present unified pressure-correction algorithm degrades naturally into the originally incompressible pressure-correction method in the low speed flow calculations. It is known that the time step in an incompressible pressure-correction algorithm is not restricted by the speed of sound but by the flow speed.

On the other hand, in the block-implicit calculations, although the strong physical diffusion present in these viscous driven cavity flows would relax the burden on the time term in promoting diagonal dominance in the momentum and energy equations, there is no diffusion terms in the continuity equation which is solved coupled with the equation system, therefore, the time term in the continuity equation still bears all of the burden of providing diagonal dominance. As a result, the allowable CFL number in this algorithm has not been increased in this Navier-Stokes flow computation compared with the one used in the Euler bump calculations. In addition, unlike the pressure-correction algorithm, the time step in the block implicit algorithm is also restricted by the speed of sound in the low speed flow computation. As a result, the computing time of the block-implicit computation is much longer than that of the pressure-correction calculation.

<table>
<thead>
<tr>
<th>TABLE 7.6</th>
<th>Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>algorithm</td>
</tr>
<tr>
<td>no. of cells</td>
<td>128*128</td>
</tr>
<tr>
<td>no. of words per cell</td>
<td>236</td>
</tr>
<tr>
<td>Mbyte</td>
<td>14.75</td>
</tr>
</tbody>
</table>

As has been demonstrated above in Table 7.2, the block-implicit algorithm again takes much more memory than the pressure-correction algorithm (see Table 7.6). Consequently, the pressure-correction algorithm is much more cost effective in computing low speed driven cavity flows.
Results and Discussion

It has been largely recognised in the compressible flow research community that density-based algorithms are inefficient for low speed flow computations due to the time step restriction imposed by the speed of sound. A lot of on-going research in the literature is underway to remedy this disadvantage. The techniques proposed so far revolve around either the removal of the speed of sound from the matrix characteristics (Feng et al. 1990) or the modification of the physical speed of sound to a pseudo one which is of the same order of magnitude with that of the flow speed (Turkel 1984), both by some form of preconditioning applied to the equations. These techniques have not been implemented in the present work, but are certainly areas worthy of investigation in the future.

7.2.3.4 Summary

The unified pressure-correction algorithm has shown good performance in simulating the two low speed driven cavity flows. Both accuracy and efficiency are excellent. This is primarily due to the fact that this algorithm degrades into the original incompressible pressure-correction algorithm when computing low speed flow problems.

The block-implicit algorithm has shown reasonably good accuracy in computing this category of flow. However, it suffers from poor convergence performance and high memory requirements.

Some regions of flow inside IC engine cylinders could be at low speed, therefore, the verification of algorithms in computing low speed flows is essential. This validation exercise on driven cavity benchmark flow problems provides an assessment on the capability, the quality and the cost of the current algorithms in computing two-dimensional incompressible flows that can be encountered in engine exchange flows.

7.3 Steady-State Calculations Employing Multigrid

In the discussion of the compressible bump calculations using the block implicit algorithm, it was observed that one reason affecting the efficiency of this algorithm is the slow convergence rate of the Gauss-Seidel (G-S) block point iterative solver. The multigrid method has been developed for about two decades now as a technique to accelerate the convergence rate of an iterative solver. Hence, in the present work, the
multigrid technique was integrated into the block implicit solution procedure and investigated for the convergence acceleration. (Note, multi-grid methods would also speed up the pressure-correction scheme performance, but this has not been investigated here.)

The effectiveness of multigrid acceleration has been tested on the two bump flow problems described in the previous section. The predictions and the convergence performance of the multigrid method in comparison to its single grid counterpart are reported in the following.

7.3.1 Multigrid Tests: Subsonic Ni Bump Flow

The subsonic bump flow problem in subsection 7.2.2.1 was re-calculated by a multigrid enhanced block implicit algorithm, using the same grid, i.e. 256×64. The multigrid cycle employed was the FAS V-cycle on 4-level multiple grids, so the coarsest grid consisted of 16×4 cells. One sweep was performed at each grid level during both restriction and prolongation procedure except at the coarsest grid level where 4 sweeps were carried out. Figures 7.22 and 7.23 present the convergence history of the multi-grid calculation and comparison with the single grid calculation. Table 7.7 summarises the number of iterations and CPU time required by the multigrid and single grid method. The artificial viscosity coefficients used in the multigrid calculations were $\epsilon^{(2)} = 0.0$ $\epsilon^{(4)} = 0.01$.

<table>
<thead>
<tr>
<th>TABLE 7.7 Convergence Performance (256×64)</th>
<th>4-level multigrid</th>
<th>single grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>no. of iterations</td>
<td>178</td>
<td>2009</td>
</tr>
<tr>
<td>total CPU time (sec)</td>
<td>5866.92</td>
<td>17854.11</td>
</tr>
</tbody>
</table>
Results and Discussion

FIGURE 7.22  Convergence history of multigrid and single grid calculations (number of iterations)

FIGURE 7.23  Convergence history of multigrid and single grid calculations (CPU time)
Results and Discussion

(It should be noted that the CPU time here and in the later Table 7.8 and Figures 7.25 and 7.26 has been recorded on the same computer, however, the machine was different from the one used in Table 7.1)

Table 7.7 illustrates that the number of iterations for the multigrid calculation to achieve convergence is significantly less than that of the single grid calculation, about one eleventh. The overall CPU time is about 3 times shorter in the multigrid calculation due to the longer CPU time per multigrid iteration compared to each of the single grid iterations. As illustrated in table 7.1 the total CPU time used by the pressure-correction calculation for this flow problem was about one and half times faster than the block implicit algorithm running on a single grid, therefore the multigrid version block implicit calculation is about twice as fast as the pressure-correction calculation (but note comments on multi-grid speed up of pressure-correction scheme above).

The solutions from the multigrid calculation are the same (different only in the sixth significant digit) with the single grid solutions, as shown in Figure 7.24

![Figure 7.24 Multigrid solutions of subsonic bump flow: iso-Mach lines](image)

FIGURE 7.24 Multigrid solutions of subsonic bump flow: iso-Mach lines
Results and Discussion

7.3.2 Multigrid Tests: Transonic Ni Bump Flow

The same multigrid technique was applied to the transonic bump flow problem described in subsection 7.2.2.2. It was found that the artificial viscosity used in the single grid calculations, i.e. \( \varepsilon^{(4)} = 1/32 \) and \( \varepsilon^{(2)} = 1.0 \), was also a suitable combination for fast multigrid convergence. The convergence history of the single and multigrid computations is presented in Figure 7.25 and Figure 7.26, with a brief summary of iterations and CPU time used given in Table 7.8.

<table>
<thead>
<tr>
<th>TABLE 7.8</th>
<th>Number of iterations and CPU time to converge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4-level multigrid</td>
</tr>
<tr>
<td>no. of iterations</td>
<td>347</td>
</tr>
<tr>
<td>total CPU time (sec)</td>
<td>13015.42</td>
</tr>
</tbody>
</table>

FIGURE 7.25 Convergence history of multigrid and single grid calculations (number of iterations)
Results and Discussion

In Table 7.8, it is clear that the multigrid technique has resulted in a 5.6 times reduction of the number of iterations to reach convergence and consequently a 1.7 times speed-up of the computing time. Because the pressure-correction algorithm was about 1.4 times faster than the single grid block implicit algorithm for this transonic bump flow (Table 7.1), the speed-up achieved by the multigrid is this time sufficient to make the block implicit algorithm comparable with the pressure-correction algorithm in computing speed.

It is noticed that the multigrid acceleration ratio in the current transonic flow calculation is about a half of that in the subsonic flow calculation; this is probably due to the large error generated by the shock during the prolongation and restriction process which requires more iterations to damp out.

The converged solutions from multigrid calculations again agreed well with those from single grid solutions (see Figure 7.27).
Results and Discussion

Single-grid solution of transonic bump flow: iso-pressure lines

![Single-grid solution of transonic bump flow: iso-pressure lines](image)

Multi-grid solution of transonic bump flow: iso-pressure lines

![Multi-grid solution of transonic bump flow: iso-pressure lines](image)

FIGURE 7.27 Multi-grid solution of transonic bump flow: iso-pressure lines

7.3.3 Summary

The multigrid technique integrated into the present block implicit algorithm has demonstrated a satisfactory acceleration of the computation of bump flow problems. As a result, the efficiency of the block implicit algorithm is significantly improved. The success of the current multigrid method is due to the modified high frequency damping property of the Gauss-Seidel iterative solver as well as the large time step employed at the coarse grid level.
Results and Discussion

7.4 Unsteady Flow Calculations

A key flow feature in the exchange flow between an engine cylinder and feed pipe systems is the dynamic propagation of pressure waves. It is therefore necessary to validate the capability of the algorithms being studied here in tracing pressure wave movement accurately. Two unsteady benchmark flow problems containing a linear and a nonlinear propagating pressure wave respectively have been computed. These two tests involve one-dimensional waves only, corresponding to the wave system observed in a pipe system whose diameter is small compared to the wavelength of the waves. The predictions are assessed for accuracy and efficiency. In addition, issues related to the influence of mesh size and time step size on the accuracy of linear wave solution are investigated.

7.4.1 Computation of Linear Pressure Wave Propagation

7.4.1.1 Relation Between Mesh Size and Accuracy

An investigation was carried out on solution grid dependency in unsteady wave simulations in order to obtain guidelines for grid density and time step size to achieve accurate transient solutions. Since both algorithms have used a first order time discretisation scheme and second order spatial discretisation, the guidelines deduced from one algorithm can be used for the other to some extent, although it is not always the case. Only the pressure-correction procedure is employed in this study.

The benchmark flow employed is a one dimensional linear pressure wave propagation in a long channel as shown in Figure 7.28.
Initially, the constant pressure flow field is disturbed by a small half sinusoidal shape linear acoustic wave described by:

\[ u = 0.005 \sin \left( \frac{\pi}{40} (x - 20) \right) \quad (-20 \leq x \leq 20) \]  

\[ u = 0 \quad (x < -20 \text{ or } x > 20) \]  

\[ p = \frac{1}{\gamma} \left( 1 + \frac{\gamma - 1}{2} u \right)^{\frac{2\gamma}{\gamma - 1}} \]  

\[ \rho = \left( 1 + \frac{\gamma - 1}{2} u \right)^{\frac{2}{\gamma - 1}} \quad \gamma = 1.4 \]  

Nondimensional quantities have been adopted using the reference values given in section 7.4.2 below.
**Results and Discussion**

The amplitude of the pressure disturbance corresponds to 0.5% of the ambient level. The computational domain extends from -50 to 350. Due to the weakness of the disturbance, according to linear theory this pressure wave will propagate at the speed of sound, the waveform will remain unchanged during propagation, i.e. no attenuation of amplitude and no distortion of waveform shape.

The frequency spectrum of this half wavelength sinusoid is centralised around the base frequency (0.0125 units), only a very small portion lies in the high frequency range.

A variety of mesh sizes and time step sizes have been employed in the numerical simulations. The comparisons of analytical and numerical solutions at time t = 200 are shown below in a series of plots. Figure 7.29 compares numerical results using different time steps with the spatial grid density chosen to be fine (i.e. equivalent to 80 grid points/wavelength). It is noticeable that more numerical dissipation has been introduced in the large time step case. The smearing at the corners of the waveform indicates that the damping is especially significant on high frequency components. It seems that the increase of time step does not incur a large phase error. This figure shows dt = 0.05 is an acceptable time-step and will be used in the following study of mesh size influence on accuracy.

![Numerical Solution of Linear Problem](image)

**FIGURE 7.29** Numerical Solution of Different Time Step Size
Results and Discussion

Figures 7.30 and 7.31 compare the results from different spatial grid sizes. In this exercise, a sufficiently small time step size (At=0.05) was employed. Mesh density was measured as the number of grid points within one wavelength of the sinusoidal wave. Mesh densities of 80, 60 and 40 were used for the results shown in Figure 7.30, while densities of 40, 20, 10 and 5 were used for those in Figure 7.31

![Numerical Solution of Linear Wave Problem](image)

FIGURE 7.30 Numerical Solution of Different Spatial Grid Density
**Results and Discussion**

**Numerical Solution of Linear Wave Problem**

(effect of no. of grid points on solution, time step=0.05)

![Graph showing numerical solution of linear wave problem](image)

FIGURE 7.31 Numerical Solution of Different Spatial Grid Density

The numerical results shown in Figure 7.30 are almost identical and very close to the analytical solution, however the left corner of the sinusoidal form is increasingly distorted as the number of grid points decreases, which indicates that the phase error of high frequency components is sensitive to the grid size and, for the present second order scheme, only becomes small when grid density is high enough. Figure 7.31 demonstrates that the waveform is completely distorted when the grid density is lower than about 20 grid points per wave length; the phase distortion of the wave peak becomes more significant as mesh density becomes coarser as well as an increasingly large amplitude error. The occurrence of more severe distortion at the left corner of the waveform is possibly caused by the particular use of one-sided upwind discretisation in the MUSCL scheme which introduces asymmetry into the solutions.

Figures 7.32 and 7.33 illustrate the magnitude of the amplitude error (defined as \(\frac{\text{[actual amplitude - real value]}}{\text{real value}} \times 100\%\)) as grid density changes, while Figure 7.34 illustrates the phase error (defined as \(\frac{\text{[actual phase - real value]}}{\text{real value}} \times 100\%\)).
Results and Discussion

Amplitude Error at l=200 vs. Grid Density

![Amplitude Error Graph](image1)

FIGURE 7.32 Amplitude Error of Numerical Solution (root mean square of sample points on the wave)

Amplitude Error at l=200 vs. Grid Density (Error of Peak Point)

![Amplitude Error Graph](image2)

FIGURE 7.33 Amplitude Error of Numerical Solution (peak point of the sine wave)
Although the phase error of peak point shown in Figure 7.34 indicates that the smaller time step resulted in a larger error, the overall rms value of the phase error is smaller in the smaller time step case. In summary, the variation becomes small after the spatial grid density is greater than around 40 points per wavelength. So, for the current second order spatial difference scheme, about 40 grid points per wavelength are needed in order to resolve accurately the bulk sinusoid wave, although higher frequency components (associated with the corners) can still not be adequately resolved. This result is in agreement with Tam's analysis (Tam 1996).

7.4.1.2 A Train of Sinusoidal Waves

This benchmark problem involves an unsteady flow of an inviscid and incompressible fluid through a channel subject to an oscillatory back pressure. The mean flow is of Mach number 0.1. The problem is illustrated in Figure 7.35, it has been used by many researchers to assess transient solution methods (Merkel 1987, Rogers 1989).
Results and Discussion

A constant total pressure and total temperature boundary condition is imposed at the upstream end of the pipe \((x=0)\), the following time varying static pressure is applied at the downstream end \((x=1)\)

\[
p_{\text{back}}(t) = p_0 + p_e \sin(2\pi ft)
\]

7.4

where \(p_0\) is the ambient pressure or datum pressure, and \(p_e\) is the amplitude of the pressure disturbance; \(p_{\text{tot}} - p_0\) is set to give a mean flow Mach number \(M_0 = 0.1\).

In the limit of small pressure disturbance \((p_e << p_0)\), the oscillatory back pressure is propagated upstream in the flow with no attenuation of amplitude, and the propagation speed remains the same all through the channel, i.e. a linear plane wave. In this study, \(p_e\) is set to 0.01\(p_0\) which can be classified as a linear wave. The oscillating frequency \(f\) is chosen such that the length of the channel equals 5 times the wavelength.
Results and Discussion

Snapshots are taken for the flow solutions at a time when the wave front is just about to be reflected from the upstream end of the channel. At that moment, there should be five wave periods appearing in the channel.

Figure 7.36 shows the pressure distribution along an arbitrary horizontal line across the channel as predicted by the pressure-correction algorithm, Figure 7.37 shows the same distribution obtained from the block-implicit algorithm. The computation grid consisted of 230×5 cells, 230 cells in streamwise x direction, 5 in cross stream y direction (although the flow is essentially one-dimensional). This results in 46 cells fitting each wavelength in the streamwise direction. This grid density can guarantee an acceptable resolution of the waves according to the conclusions drawn from section 7.4.1.1. In both graphs, prediction results from the use of two different time step sizes (one five times smaller than the other) are displayed together to demonstrate the influence of this factor. Clearly both phase and amplitude errors are reduced as the time step reduces. It is also apparent that most of the error occurs in the leading wave and is thus generated by the steep corner at the initial wave-front. It should be noted that the results from a 460×5 grid is plotted for the small time step case in pressure-correction calculation, however, the solution becomes grid independent when the number of cells in the stream direction is greater than 230.

FIGURE 7.36 pressure distribution across the channel: pressure-correction algorithm
Results and Discussion

In the block implicit calculations (Figure 7.37), there is no artificial dissipation added, solutions are produced by a pure central differencing scheme. It is clearly shown that the agreement between the predictions and analytical solutions is improved when the time step size is reduced by one fifth, although as noted above this is predominantly the case for the leading wave, where high frequency errors (at the sharp corner of the leading edge) will be improved. Both predictions become coincident with the analytical solution further away from the first wave. The attenuation in the furthermost wave from the source is considerable in the large time step solution but negligible for the smaller time steps. The phase error is very small throughout the channel for both time steps indicating that the phase error is not as sensitive as the amplitude error to the time step size.

In the pressure-correction calculations (Figure 7.36), a smooth solution has been obtained without the addition of Rhie&Chow pressure smoothing. With a streamwise grid of 231 cells, the reduction of the time step improves drastically not only the amplitude accuracy but also the phase accuracy. This indicates that unlike the block-implicit algorithm, the phase error, i.e., the dispersion error, is sensitive to the time step in p-c algorithm.
Results and Discussion

7.4.1.3 CPU Time Usage

The total CPU time in computing the linear wave propagation problem in section 7.4.1.2 has been recorded, and is listed in table 7.9. The time steps used in these calculations are 0.2e-6 in the pressure-correction algorithm and 0.2e-4 in the block implicit algorithm. The difference of time steps in the two calculations is due to the use of non-dimensionalised variable values in the block implicit algorithm. In this wave propagation problem, the dimensional wave frequency in the pressure-correction calculation is 1715 hz whereas the non-dimensional wave frequency in block implicit calculation is 25 units - i.e. about 70 times smaller, therefore the dimensional time step used in the pressure-correction calculation is expected to be about two order smaller than the non-dimensional time step used in the block-implicit calculation if the same resolution of propagating waves is to be achieved in the two calculations. The grid used in both cases is 231x5.

| TABLE 7.9 CPU Time and Memory Usage (grids: 231x5) |
|-----------------|-----------------|-----------------|
|                 | block implicit  | pressure correction |
| CFL number used | 0.025           | 0.017            |
| total CPU time (hrs) | 2.90           | 4.86            |
| memory (Mbyte)    | 0.828           | 0.126            |

The CPU time used by the block-implicit calculation is less than that used by the pressure-correction calculation, mainly due to the smaller number of inner iterations to reach convergence at each time level of the block implicit calculation (typically 1/3 - 1/5 less than that used in pressure-correction calculations). Nevertheless it is observed that both algorithms use less than 10 inner iterations.

The present pressure-correction scheme is an extension of the SIMPLE scheme. It is known that the SIMPLE procedure is not very efficient in computing unsteady flow due to the discarded terms in the formation of the pressure-correction equation. This renders a poor correction for the pressure field at each iteration, which tends to induce a large number of inner iterations before all equations are solved exactly at each time level.
Results and Discussion

Another procedure, termed PISO, has been developed in the incompressible pressure correction methodology to compute specifically unsteady flows, and is significantly more efficient than the SIMPLE algorithm in the time-accurate calculations (Issa 1986). Therefore, it is believed that the efficiency of the present unified pressure-correction procedure could be improved if the PISO method is incorporated.

7.4.1.4 Summary

For this unsteady linear pressure wave propagation problem, the prediction quality of the pressure-correction and the block-implicit algorithms is comparable. The block implicit calculation is faster than the pressure-correction one, however, its memory requirement is again significantly higher. It is noted however that a SIMPLE variant - the PISO method - could be used to reduce the pressure-correction computing time for transient problems without much increase of memory usage, it is possible therefore that with this modification the pressure-correction method would be eventually more favourable for this category of flow problems.

7.4.2 Nonlinear Pressure Wave Propagation

Nonlinear effects on pressure wave propagation, which very often lead to the formation of a shock, are frequent flow features occurring in engine exchange flows, especially during the exhaust blow-down process. It is thus of great interest to this work to investigate the capability of the developed numerical techniques in simulating such flow phenomena, including the coupling methods which will be covered in the next section. A benchmark test problem which reflects characteristics of nonlinear wave evolution has therefore been employed to validate the two algorithms.

This test case is a one dimensional nonlinear propagation of an initial Gaussian waveform disturbance in a two dimensional channel. The initial wave is given by

\[ t = 0 \quad u = 0.5 \exp \left[ -\left( \frac{\ln 2}{3} \right)^2 \right] \] 7.5
the expressions for pressure and density are the same as (7.2) and (7.3). The computational
domain is \(-50 \leq x \leq 350\) in the wave propagation direction and \(0 \leq y \leq 4\) in the cross stream
direction.

This initial disturbance will propagate nonlinearly due to the high amplitude of the wave,
which includes pressure changes of 50\% of the ambient levels. The leading edge of the
wave will steepen and the trailing edge will become shallower. A shock will form at the
wave front around time \(t = 100\) and the waveform assume a triangular shape, the future
propagation being that for a shock. This is again a 1D problem, an approximate analytical
solution for this wave problem has been given by Whitham (1974). This is used to validate
the current numerical predictions. The initial flow condition in the channel and the
analytical solution of Whitham at time \(t = 200\) are illustrated in Figure 7.38

![Analytical Solution of Nonlinear Wave Propagation](image)

**FIGURE 7.38** Initial Condition and Approximate Analytical Solution at Time 200

Nondimensional variables are used in the series of tests, the scales used are
Results and Discussion

\[ \Delta x = \text{reference} - \text{length} \quad 7.6 \]

\[ a_\infty = \text{reference} - \text{velocity} \quad 7.7 \]

\[ \frac{\Delta x}{a_\infty} = \text{reference} - \text{time} \quad 7.8 \]

\[ \rho_\infty = \text{reference} - \text{density} \quad 7.9 \]

\[ \rho_\infty a_\infty^2 = \text{reference} - \text{pressure} \quad 7.10 \]

\[ T_\infty = \text{reference} - \text{temperature} \quad 7.11 \]

\[ \gamma R = \text{reference} - \text{gas constant} \quad 7.12 \]

Subscript \( \infty \) denotes the upstream flow variable. The relation between the nondimensional and the real values of some variables are given below

\[ \tilde{R} = \frac{1}{\gamma} \quad \tilde{C}_p = \frac{1}{\gamma - 1} \quad \tilde{C}_v = \frac{1}{\gamma(\gamma - 1)} \quad 7.13 \]

\[ \tilde{e} = \frac{1}{\gamma - 1} \frac{\tilde{p}}{\rho} \quad \tilde{h} = \frac{\gamma}{\gamma - 1} \frac{\tilde{p}}{\rho} \quad 7.14 \]

where \( e \) and \( h \) are specific energy and specific enthalpy respectively.

### 7.4.2.1 Predictions

Computational solutions from the pressure-correction algorithm are presented in Figures 7.39 and 7.40, and those from the block implicit calculations in Figure 7.41 and 7.42. The computational grids and time step have been chosen following the guidelines obtained in the previous section, 400 grid points are used in the axial direction. The Rhie&Chow pressure smoothing model is used in the pressure-correction algorithm. The mixed 4th and
Results and Discussion

2\textsuperscript{nd} order artificial dissipation model is used in the block-implicit algorithm with $\varepsilon^{(4)} = 0.01$ and $\varepsilon^{(2)} = 0.0625$.

The computed shock position is used as an indicator of numerical phase error, whereas the shock amplitude is a measure of numerical smoothing error. The present results have been compared against the approximate analytical solution. They are also compared with results from Computational Aero-Acoustic (CAA) higher order schemes (4\textsuperscript{th} order in space, 2\textsuperscript{nd} order in time) published during a NASA Workshop in 1995 (1995). One thing should be noted, the grid density and time step size used in the current calculations are the same as those used in the NASA Workshop. Theoretically the lower order scheme would produce a worse accuracy of solution compared to a higher order scheme on the same mesh.

![U Velocity Profile](image)

FIGURE 7.39 Numerical Solutions of Velocity at Time 200 (pressure-correction algorithm)
Results and Discussion

In the density and u velocity solutions from the pressure-correction algorithm, the shock formed during the non-linear propagation process is cleanly captured. It is quite sharp and free of overshoots or undershoots comparing with most of the numerical results illustrated in the NASA Workshop report. The strength of the shock is slightly higher than the analytical solution. The location of the shock is very accurate considering the long distance the wave has already travelled. Both the shock front and the expansion fan are in very good agreement with the analytical solutions. This indicates a good accuracy of the present pressure-correction algorithm in tracing the nonlinear propagation pressure wave.
Results and Discussion

**U Velocity Profile**

![U Velocity Profile](image1)

**FIGURE 7.41** Numerical Solutions of Velocity at Time 200 (block implicit algorithm)

**Density Profile**

![Density Profile](image2)

**FIGURE 7.42** Numerical Solutions of Density at Time 200 (block implicit algorithm)
Results and Discussion

The results from the block implicit prediction have the same excellent accuracy as those from the pressure-correction prediction. The shock has also been captured cleanly and sharply, the expansion fan has been well predicted. The block implicit scheme, like the pressure-correction scheme, has over predicted the shock strength and smeared the wave form near the corners, as would be expected with a 2nd order accurate method in regions where high frequency components make significant contributions.

The prediction quality of the current two algorithms are highly competitive to those reported in the NASA Workshop. Because they are lower order schemes compared to those in the Workshop, they are comparatively simpler and less expensive.

7.4.2.2 CPU Time Usage

The CPU time spent on the computation of this nonlinear pressure wave propagation flow problem is collated in table 7.10

<table>
<thead>
<tr>
<th>TABLE 7.10</th>
<th>CPU Time and Memory Usage (401*4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFL number used</td>
<td>block implicit</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>total CPU time</td>
<td>1.06</td>
</tr>
<tr>
<td>memory (Mbyte)</td>
<td>1.080</td>
</tr>
</tbody>
</table>

The total CPU time of the block implicit calculation is about half of the pressure-correction calculations, however, the memory usage is about 7 times larger. A similar convergence performance at each time level of the two algorithms as that in the linear wave calculations has been observed, that is the number of inner iterations to converge within each time level undertaken by the block implicit algorithm is much less (about one-fifth) compared to the pressure-correction algorithm.

7.4.2.3 Summary

The numerical results for nonlinear wave propagation prediction from both algorithms compare very well to the approximate analytical solution. Both shock location and
Results and Discussion

strength are captured accurately, comparable in quality to those published in a CAA Workshop (1995) which use 4th order spatial difference schemes on the same grid density. However, regions around the shock and expansion fan corners are smeared out, which illustrates the inability to resolve high frequency components contained in the waveform.

The CFL number used in this category of calculations was comparatively larger than the one used in the linear wave calculations. Nevertheless, the accuracy of nonlinear wave calculation is as good as that of linear wave calculation. When such grid and time size are used in linear wave computation, the accuracy would drop considerably. This is because in the strong nonlinear wave flow, flow disturbances are much larger in magnitude compared to the numerical errors, the effect of numerical solution contamination is not so significant.

Comparing the overall computing time, the block implicit calculation is faster than the pressure-correction one, however, the memory requirement of the block implicit algorithm is considerably larger. Therefore the replacement of the PISO method for SIMPLE could make the pressure-correction procedure attractive for this type of flow calculation too.

7.5 Unsteady Flow Calculations Using a Coupled Algorithm

A 1D/2D coupled CFD model has been proposed in this work for the simulation of engine exchange flow where cylinder and pipe system are calculated simultaneously. The one dimensional model is intended to be applied in the straight pipe region where the flow is dominated by one dimensional propagating pressure waves. The interface of the coupled model will be placed in the one dimensional flow region. The propagating waves, whether linear or nonlinear, would therefore pass through this interface during the course of a transient simulation.

As explained in Chapters 4, 5 and 6, special numerical treatments have been imposed on the interface to ensure the continuity of the flow as well as minimise the interface disturbance to the flow, in particular to moving pressure waves. The success of the interface treatment must be measured by comparing the predicted results from a 1D/2D coupled modelling with those from an entirely 2D modelling on the same flow problems.
Results and Discussion

The unsteady benchmark flow problems described in section 7.4 are employed to evaluate the interface coupling schemes, because they contain typical wave flows encountered in an engine pipe system. The feasibility of applying the 1D/2D coupled model in the engine exchange flow simulation is then assessed on the basis of this series of numerical studies.

In all calculations performed, the two dimensional duct containing the propagating waves has been divided into two computational blocks. One is discretised by a one dimensional grid and the other by a two dimensional grid as sketched in Figure 7.43.

![Diagram of 1D and 2D coupled grids](image)

FIGURE 7.43 An example of the 1D and 2D coupled grids used in the unsteady flow calculations

The 1D grid block can be either to the left or to the right of the two dimensional grid block. The effect of these two possible grid arrangements on the predictions will be studied.

7.5.1 Linear Wave Propagation Problem

The description of the linear pressure wave problem has been given in section 7.4.1.2, the 1D/2D interface is located about half way along the channel. The one dimensional grid block consists obviously of only one cell in the cross stream direction, the two dimensional grid block, on the other hand, consists of 4 cells. In the streamwise direction, 130 cells are uniformly placed in the left hand side block with 100 cells in the right hand side block so that the total number of cells in this direction is the same as that used in the calculations presented in section 7.4.1.2. The same time step as that in section 7.4.1.2 is employed.

230
Results and Discussion

7.5.1.1 Predicted Results

Figures 7.44 to 7.47 present the pressure profile along an arbitrary horizontal line through the channel predicted by the pressure-correction algorithm. Also plotted are the analytical solutions as well as the solutions produced by the two dimensional uncoupled model.

![Pressure Wave Profile](image1)

**FIGURE 7.44** Pressure profile along a horizontal line (implicit 1D/2D coupling)

![Pressure Wave Profile](image2)

**FIGURE 7.45** Pressure profile along a horizontal line (implicit 2D/1D coupling)
Results and Discussion

FIGURE 7.46 Pressure profile along a horizontal line (explicit 1D/2D coupling)

FIGURE 7.47 Pressure profile along a horizontal line (explicit 2D/1D coupling)
Results and Discussion

In the calculations shown in Figures 7.44 and 7.45, an implicit interface coupling scheme is used, whereas in the calculations shown in Figures 7.46 and 7.47, an explicit coupling scheme is employed. The one dimensional grid block is on the left of the two dimensional grid block in Figure 7.44 and Figure 7.46, whereas in Figure 7.45 and Figure 7.47, it is the other way round.

The solutions from the coupled model match almost exactly with those from the 2D model in all above the calculations. Such excellent agreement indicates that the interface treatments have been successful in eliminating the intrusive presence of the 1D and 2D interface. The interface has been made to appear no different from an ordinary internal cell face. The predicted results are insensitive to the two arrangements of 1D and 2D blocks in the computation domain. It is believed that this excellent coupling result arises because of the conservation of mass flux, momentum flux and pressure force at the interface as well as the numerical evaluation methods for these flow variables at the interface. Because this wave flow problem is literally one dimensional and inviscid, the interface treatment in both explicit and implicit interface coupling produces the same numerical representation of mass flux etc. as that obtained in the two dimensional model.

The insensitivity of the prediction to the relative location of the 1D and 2D block is the result of an unbiased evaluation of flow quantities on the interface and the employment of multiple direction sweeps in the SLOR line solver.

Figures 7.48 and 7.49 present the same information as above but resulting from the block implicit calculations. Only the implicit interface coupling treatment is implemented with this solution procedure.
Results and Discussion

[Graph showing pressure wave profile with labels: x distance, pressure]

FIGURE 7.48  Pressure profile along a horizontal line (implicit 1D/2D coupling)

[Graph showing pressure wave profile with labels: x distance, pressure]

FIGURE 7.49  Pressure profile along a horizontal line (implicit 2D/1D coupling)
Results and Discussion

The 1D grid block is on the left of the 2D grid block in Figure 7.48, while it is the other way round in Figure 7.49.

Again the difference of solutions between the 1D and 2D coupled model and the 2D model is hard to distinguish. This is expected after the observation of the coupled model results from pressure-correction algorithm, because the same implicit interface treatment has been employed in the two solution procedures, although detail implementation is different due to the different structure of these two procedures.

7.5.1.2 CPU Time and Memory Usage

CPU time usage and memory requirement of the above calculations are recorded in table 7.11 for the pressure-correction calculations.

| TABLE 7.11 CPU Time Comparison (pressure-correction algorithm) |
|-----------------|-----------------|-----------------|
| grids           | coupled grids   | uncoupled grids |
| coupling method | implicit coupling | explicit coupling | no coupling |
| coupling model  | 1D&2D           | 2D&1D           | 1D&2D        | 2D&1D        | 2D           |
| number of cells | 530             | 620             | 530          | 620          | 920          |
| memory (Mbyte)  | 0.0728          | 0.0851          | 0.0728       | 0.0851       | 0.126        |
| total CPU time (hrs) | 3.56       | 3.89           | 3.80         | 4.17         | 4.86         |

The first column under 'coupled grids' corresponds to the coupled model in which the 1D block is on the left of the two dimensional grid block and implicit interface coupling treatment is used, solutions from this model are illustrated in Figure 7.44. The second column corresponds to the coupled model whose solutions are shown in Figure 7.45, the third column corresponds to Figure 7.46, the fourth column corresponds to Figure 7.47. The calculation corresponding to the column under 'uncoupled grids' has been illustrated in Figure 7.36.

Because the number of cells in the coupled model is less than that in the 2D model, the coupled model has a clear advantage in memory over the 2D model. In addition, the computing time is reduced due to the smaller number of solution points to be resolved.
Results and Discussion

The coupled model calculation does involve some overhead time, however, the gain in overall CPU time in the coupled model indicates that this overhead time is very small.

The implicitly coupled model has actually used less CPU time than the explicitly coupled model despite the larger overhead time required. The main reason is that the implicit coupling model converges much faster at each time level. This is likely to be the result of the high integrity of implicitness maintained by the implicit interface treatment; flow information anywhere in the flow field can be communicated to everywhere in the solution domain at any time. In contrast, the explicit interface treatment has weakened the level of implicitness. As a result, more inner iterations are needed for the solution to converge within a time step. Hence, there is a big gain in computing time by employing an implicit coupling technique despite the complexity of its implementation.

Table 7.12 lists equivalent data for the block implicit calculations.

<table>
<thead>
<tr>
<th>grids</th>
<th>coupled grids (implicit coupling)</th>
<th>uncoupled grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>coupling model</td>
<td>1D&amp;2D</td>
<td>2D&amp;1D</td>
</tr>
<tr>
<td>number of cells</td>
<td>530</td>
<td>620</td>
</tr>
<tr>
<td>memory (Mbyte)</td>
<td>0.477</td>
<td>0.558</td>
</tr>
<tr>
<td>total CPU time(hrs)</td>
<td>1.93</td>
<td>2.14</td>
</tr>
</tbody>
</table>

This table indicates clearly that both computing time usage and memory requirement for the 1D and 2D coupled models are less as compared to the 2D model. Therefore the computing cost is reduced when coupled models are employed.

When the coupled models in the pressure-correction algorithm and block implicit algorithms are compared, it is found that the block implicit procedure converges faster. This is due to the same reason as explained in section 7.4.1.2.
Results and Discussion

7.5.1.3 Summary

The 1D/2D coupled models implemented in the two solution algorithms have reproduced the same results as have been produced by the 2D model for the linear wave benchmark flow problem. The agreement between the 1D and 2D coupled model prediction and the 2D model one is excellent. The considerable computing time and memory reduction brought out by the use of a coupled model have achieved the goal of computing cost reduction with no compromise on prediction quality.

In the pressure-correction algorithm, the implicit interface coupling method is more cost effective compared to the explicit coupling method. However, the final prediction quality of these two coupling schemes are comparable.

The implicit coupled model built into the block implicit procedure is faster than the same coupled model in the pressure-correction procedure, however, it is considerably more expensive in memory. The prediction quality the of two schemes is comparable. The coupled model in the pressure-correction procedure would be improved in computing time if the PISO method were integrated and consequently might become superior.

7.5.2 Nonlinear Wave Problem

Tests in the previous subsection have shown the promise of replacing the 2D model by a 1D/2D coupled model in those engine applications where linear plane waves are the key feature in the 1D flow regions. However, in most engine exchange flows, nonlinear effects may be important. Therefore, the coupling approach needs to be validated on this type of transient problem. The benchmark flow problem described in section 7.4.2 is employed here for this purpose.

The grid system in the coupled models in this series of numerical experiments is similar to that used in the previous linear wave tests. Referring to Figure 7.38, the 1D and 2D interface is located at $x = 150$, and the initial pulse is centred around $x = 0$. The number of cells in the stream direction is 200 in both blocks. 4 cells are placed in the cross stream direction in the two dimensional grid block. The time step used in section 7.4.2.1 is adopted here.
7.5.2.1 Predicted Results

Figures 7.50 and 7.51 present the coupled model results using the pressure-correction algorithm. The one-dimensional grid block in this model is on the left of the two-dimensional grid block and the implicit interface coupling method is implemented.

![U Velocity Profile](image)

FIGURE 7.50 Solutions of velocity from pressure correction algorithm: implicit 1D/2D coupling
Results and Discussion

These figures show that the differences between the coupled model and the 2D model are undetectable.

In this unsteady flow problem, a shock passes across the 1D and 2D block interface during the course of the nonlinear wave propagation. Compared to the sinusoidal wave in the previous linear wave propagation problem, the shock front of the nonlinear wave is more prone to oscillation under the influence of any numerical disturbance. The discontinuity of the mesh in the computational domain of the 1D and 2D coupled model could easily introduce some form of numerical disturbance due to the abrupt change of cell size if the key flow information at the interface has not been faithfully represented. The excellent agreement between the results from the coupled model and that from the 2D model indicates that the implicit coupling treatment, where the strong conservation of all fluxes on the interface is imposed, has described flow conditions on the interface accurately when the shock and the expansion wave passed through. Indeed, the shock condition in Euler inviscid flows can be described by the Rankine-Hugoniot condition which can be

FIGURE 7.51 Solutions of density from pressure correction algorithm: implicit 1D/2D coupling
expressed as a flux conservation statement across the shock. These are the exact conditions that have been imposed on the implicit interface. Therefore, the shock condition is prescribed correctly as it crosses the interface.

When 1D and 2D blocks swap their locations in the duct, predictions agree with the results predicted by the 2D model in an exact format as those in Figures 7.50 and 7.51, therefore they are not shown here. This indicates a negligible affect of the change of block arrangement on the prediction solutions, which further confirms the conclusion drawn in the previous subsection that the unbiased evaluation of flow quantities at the interface and symmetric sweeps in SLOR and TDMA solvers have together achieved the independence of solution on the block arrangement in a 1D and 2D coupled model.

In similar calculations using the pressure-correction procedure but where the coupling at the interface was treated explicitly, the shock moved across the explicit interface accurately without being disturbed. Consequently, the results from these calculations are literally no different to those from the implicit coupling treatment and are not shown here.

The above calculations have been also repeated for the block-implicit algorithm as the basic solution procedure, in this case only the implicit coupling strategy is employed. The results are again literally the same as those obtained in the 2D block implicit calculations, and therefore not presented here. Such results further confirm the conclusions drawn from the 1D&2D model in the pressure-correction algorithm. In addition, it shows that the interface coupling strategy can achieve the same performance when it is implemented on two very different algorithms, although the detailed implementation has been greatly tailored according to the structures of the basic algorithms. This demonstrates the generality of the 1D and 2D interface methodology developed in this work with regard to the base solution procedure it is built upon.

7.5.2.2 CPU Time and Memory Usage

Tables 7.13 and 7.14 list the total CPU time and memory usage by the pressure-correction and block-implicit algorithms respectively during the computation of this benchmark problem employing a coupled model. It can be easily observed that the trend shown here is the same as that demonstrated in Tables 7.11 and 7.12.
Results and Discussion

TABLE 7.13 CPU Time and Memory Usage Comparison

<table>
<thead>
<tr>
<th>grids</th>
<th>coupled grids</th>
<th>uncoupled grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>coupling method</td>
<td>implicit coupling</td>
<td>explicit coupling</td>
</tr>
<tr>
<td>coupling model</td>
<td>1D&amp;2D</td>
<td>2D&amp;1D</td>
</tr>
<tr>
<td>number of cells</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>memory (Mbyte)</td>
<td>0.110</td>
<td>0.110</td>
</tr>
<tr>
<td>total CPU time (hrs)</td>
<td>1.619</td>
<td>1.597</td>
</tr>
</tbody>
</table>

TABLE 7.14 CPU Time and Memory Usage Comparison

<table>
<thead>
<tr>
<th>grids</th>
<th>coupled grids (implicit coupling)</th>
<th>uncoupled grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>coupling model</td>
<td>1D&amp;2D</td>
<td>2D&amp;1D</td>
</tr>
<tr>
<td>number of cells</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>memory (Mbyte)</td>
<td>0.720</td>
<td>0.720</td>
</tr>
<tr>
<td>total CPU time (hrs)</td>
<td>0.542</td>
<td>0.551</td>
</tr>
</tbody>
</table>

7.5.2.3 Summary

The coupled model built into either the pressure-correction or the block-implicit algorithm has produced good quality predictions for the nonlinear unsteady benchmark flow problem, the results match very well with the ones from the two dimensional model. Both computing time and memory usage have been reduced in the coupled model. This indicates that the interface coupling methodology developed in this work is general and appropriate for nonlinear wave and shock propagation problems, and has achieved the goal of the coupled model, i.e. reducing cost without sacrificing prediction quality.

The interface treatment has conserved the key fluxes at the interface, therefore, a nonlinear wave and shock can be computed accurately without any distortion as they are moving across the interface. It should be noted that this is the most challenging flow characteristic for the coupled model to handle, because it is extremely sensitive to the numerical treatment of the interface, and can be easily distorted by the numerical errors introduced
Results and Discussion

by the numerical interface evaluation. The good results are strong evidence of the effectiveness of this flux conservation approach in the interface treatment strategy.

The prediction solutions are independent of the relative locations of the block containing the one dimensional grid and the one containing the two dimensional grid. This removes any concerns on grid block arrangement effects on the prediction, so that the blocks can be arranged in the way which is most suitable for the flow condition concerned. This has an important implication to the engine exchange flow application. It assures that the nonlinear waves travelling in either direction in the straight pipe and from either pipe ends can be reproduced faithfully by the same 1D and 2D coupled model.

7.6 Two-Stroke Engine Exchange Flow Prediction

The validation of the coupled modelling approach is now applied to an engine exhaust flow. Because the benchmark tests have shown that the coupled model integrated into the pressure-correction procedure can produce the same quality of prediction as that in the block implicit procedure with significant less requirement of computing resource (and its computing speed would be improved by introducing a solution method such as PISO), the pressure-correction procedure is preferred in the computations to be described.

7.6.1 Experimental Apparatus Set-up

A “Single Pulse” rig has been built at the Queen’s University of Belfast, especially to test engine exchange algorithms. This rig can generate a flow field which replicates that found in the intake or exhaust process in a two-stroke engine. The flow is predominately 2D and axisymmetric. The detailed configuration of the rig and measurements of the transient flow field in the pipe system and the cylinder have been reported by Kirkpatrick (1994), here only a brief description is given.

The test apparatus consists of a single cycle pressure wave generator consisting of a cylinder connecting via a sliding valve with a long pipe. A schematic diagram of the rig configuration is given in Figure 7.52. The internal diameter of the cylinder is 90mm and its volume is 912 cm$^3$. The internal diameter of the pipe is 25mm.
Results and Discussion

The slide plate contains a circular aperture which, upon actuation, uncovers and then recovers an identical aperture in the cylinder and pipe. The diameter of the plate aperture is 25mm, i.e. the same as the pipe diameter. Initially, a different pressure is present in the cylinder and the pipe. When the sliding valve opens and closes, a single cycle pressure wave is generated. This pressure wave mimics closely that found in a two-stroke engine system, either during exhaust flow or in an intake flow. The pipe is sufficiently long (5.901m) so that the whole process of nonlinear wave development including shock formation can be monitored with no interference from wave superposition due to reflections.

The particular flow selected for computation in the present work is an exhaust blow-down flow. The cylinder release condition is at 1.5 bar and 293 K, while the conditions in the pipe are 1 bar and 293 K. The valve opening time schedule is given in the PhD thesis of Kirkpatrick (1994) and shown in Figure 7.53 in terms of the aperture open area connecting cylinder and pipe, the slide completes its travel in approximately 7.5 ms. With such a set-
Results and Discussion

up, the rig produces a blow-down exhaust pulse similar to a typical two-stroke cycle engine running at 3000rpm.

![Port Area vs Time](image)

FIGURE 7.53 Valve operating schedule

The time history of pressure at locations identified as STATION 1 and STATION 2 in the pipe as well as at STATION 0 in the cylinder (see Figure 7.52) were recorded in the experiment (Kirkpatrick et al. 1994). The experimental data are used to validate the present predictions.

7.6.2 Cylindrical Polar Code for Axisymmetric Flow

Before applying the 1D/2D coupled model to compute this engine flow, the cylindrical polar version of the code developed during the present research was validated. The benchmark flow chosen for this purpose was a simple laminar developing flow in a long circular pipe. Initially, flow in the pipe is motionless and there is no pressure gradient. Upon application of a uniform axial velocity profile at the pipe inlet, flow in the pipe starts to move and develop. At a sufficiently large distance from the entrance section, the axial velocity distribution across the sections become independent of the axial coordinate, and the pressure gradient in the streamwise direction becomes a constant.

For the validation test, the radius of the pipe (R) was 25mm and the pipe was of 2500mm length. The uniform axial velocity ($u_{avg}$) applied at the pipe entrance section was 0.04m/s.
**Results and Discussion**

with a fluid density \( \rho \) of 1.2 units and viscosity \( \mu \) of \( 2.0 \times 10^{-5} \) units; as a result, the Reynolds number of the flow was 100. In the fully developed flow region, the distribution of the axial velocity \( u \) is a parabolic profile as given by: (Schlichting 1979)

\[
\begin{align*}
  u_{\text{parabolic}} &= 2\bar{u}_{\text{avg}} \cdot \left[ 1 - \left( \frac{r}{R} \right)^2 \right] \\
  \text{where } r \text{ is the radial distance to the pipe axis and } R \text{ is the pipe radius. The maximum of this parabolic function is} \\
  u_{\text{max}} &= 2\bar{u}_{\text{avg}} \\
  \text{The pressure gradient in this region is} \\
  \frac{dp}{dx} &= - \left( u_{\text{max}} \cdot \frac{4\mu}{R^2} \right) \\
  \text{So from theory, the maximum axial velocity and pressure gradient in the fully developed region of this pipe flow are 0.08 m/s and -0.01024 N/m}^3 \text{ respectively.}
\end{align*}
\]

The results predicted by the present code using a mesh of \( N_x \) (axial) by \( N_r \) (radial) cells are presented in Figure 7.54 to 7.57. In Figure 7.54, the contours of axial velocity are shown. The cross sectional distribution of the axial velocity in the fully developed flow region is shown in Figure 7.55, and the distribution of the centreline velocity along the axis of the pipe is plotted in Figure 7.56. In Figure 7.57, the pressure distribution (gauge pressure, \( P_{\text{ref}} = P_{\text{exit}} \)) along the axis is presented.
Results and Discussion

FIGURE 7.54 Contours of axial velocity: pipe

FIGURE 7.55 Axial velocity distribution at the cross section in the fully developed region: pipe
Results and Discussion

Axial Velocity (u) Profile along the Axis (circular pipe)

FIGURE 7.56 Axial velocity distribution along the axis: pipe

Pressure Profile along the Axis (circular pipe)

FIGURE 7.57 Pressure distribution along the axis: pipe
Results and Discussion

Clearly, the predicted results (i.e. \( u_{\text{max}} = 0.079 \text{m/s} \), pressure gradient = -0.01 units) are in excellent agreement with the analytical solutions, hence validating the present code for axisymmetric problems.

7.6.3 Valve Simulation

Obviously, during the actual area opening phase of the sliding valve the flow domain is no longer perfect axis-symmetrical in this region. To simplify the problem and retain the 2D modelling, a numerical method to simulate the sliding valve has been developed so that the generation of the pressure wave at the valve can be computed as part of the 2D prediction. The simulation of the valve mechanism which uncovers and then covers the aperture on the cylinder is achieved with the assistance of the block interface treatment of the multiblock grid system.

As described in Chapter 4, a set of boundary cells have to be generated on both sides of a block interface in multiblock grid systems, as shown in Figure 7.58.

![Figure 7.58 Schematic diagram of valve simulation](image)

If the valve location coincides with a block interface of two continuous two dimensional blocks, the interface cells can represent the segments of the valve. For example, in Figure 7.58, the valve lies at the interface of block 1 and block 2, the interface cells 'vc1' to 'vc 6' represent various portions of the valve. The status of each valve portion (open or closed) can be denoted by an indicator assigned to each interface cell. During the solution
Results and Discussion

procedure, depending on the indicator value of each interface cell, action is taken accordingly. For the open cells, the interface treatment is applied; whereas for the closed cells, a wall boundary condition just as applied to a real wall is applied.

Although the actual valve is open and closed continuously, in any numerical calculation, it can only be operated in a discrete manner. Furthermore, the opening and closing in a 2D numerical simulation must be different from the actual one. In the experiment, the valve is opened by the sliding aperture as illustrated in Figure 7.59

![Figure 7.59 Valve operating mechanism: experiment](image)

where the overlapping section of the two apertures (non-shaded) represents the valve open area.

In a 2D numerical simulation where an axi-symmetrical description must be retained, depending on whether the open area is considered to grow from the centre of the aperture outwards or from the edge inwards, the valve can only be represented by the following 4 modes as illustrated in Figures 7.60 to 7.63
Results and Discussion

Cylinder aperture

<table>
<thead>
<tr>
<th>Time</th>
<th>Valve Opening</th>
<th>Valve Closing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 7.60 Valve numerical simulation: MODE 1

Cylinder aperture

<table>
<thead>
<tr>
<th>Time</th>
<th>Valve Opening</th>
<th>Valve Closing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 7.61 Valve numerical simulation: MODE 2

Cylinder aperture

<table>
<thead>
<tr>
<th>Time</th>
<th>Valve Opening</th>
<th>Valve Closing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 7.62 Valve numerical simulation: MODE 3
Results and Discussion

![Cylinder aperture](image)

**FIGURE 7.63** Valve numerical simulation: MODE 4

(the shaded section represents the closed portion of the valve aperture). All the modes are characterised by radial movement of the valve open area. At any time instant, the open or closed section of the numerical valve will be either an annular or a circular section instead of a section formed by two arcs. Despite the difference in valve opening (or closing) manner between the experiment and simulation, the variation of valve area in time in the experiment as shown in Figure 7.53 may still be followed by the numerical valve, although in a discrete manner.

The open or closed status of each valve cell is determined by the location of the cell. In one method (CRITERION 1), when over half of the cell edge lies within the valve open region, it will open. In the other method (CRITERION 2), the valve cell will open whenever it is partially or completely inside the valve open regime. A discontinuous change of valve area with time results, because after one cell is opened (or closed), it will be a period of time later before the next cell opens (or closes). The area-time diagram for the numerical valve is, obviously, different depending on which MODE or CRITERION method is used.

**7.6.4 Numerical Prediction Set-up**

The computational domain for the configuration shown in Figure 7.52 and the grid system applied are illustrated in Figures 7.64 to 7.66.
Results and Discussion

where ‘axis’ is the axisymmetric axis. A multiblock grid system is applied. Blocks 1 and 2 cover the cylinder region, while blocks 3 and 4 cover the pipe. The valve simulation is applied at the block interface between blocks 2 and 3. Apart from block 4, a two dimensional grid has been applied in each of the blocks. In block 4, a one dimensional grid is applied, and therefore the block interface between block 3 and block 4 is the only discontinuous zonal boundary in the whole domain with respect to the grid lines. The location of this 1D and 2D interface has been chosen in a location where it is expected that the pipe flow is quasi-one-dimensional. In order to compare the predictions between a 1D/2D coupled model and a complete 2D model for this flow, all calculations have been repeated on a grid system where block 4 is also meshed with a 2D grid. The two grid systems used in the calculations are shown in Figures 7.65 and 7.66.
Results and Discussion

7.6.5 Predicted Results and Discussion

Pressure time variations at two locations in the pipe and one station in the cylinder (as identified in Figure 7.52) have been recorded in the experiment. Because pressure in the cylinder is largely constant, the position of the monitoring station there is not important. Kirkpatrick (1994) has computed this problem using Blair’s wave tracing method and boundary condition method (Blair 1996). The predicted results from the present 1D/2D coupled model are validated using the experimental measurements and compared with those from a complete 2D/2D model as well as Blair’s method. The results of Kirkpatrick using Blair’s methods are also included in all the figures presented below and a discussion of the comparison between his results and the present results is given in section 7.6.5.4 below. From this series of numerical exercises, the capability of the 1D/2D coupled model in predicting a realistic engine exchange process can be assessed.

An investigation has also been carried out on the influence of the different numerical valve treatments, i.e. the choice of operating MODE as well as the CRITERION for numerical valve cell opening. Four numerical valve operating MODEs and two valve cell open CRITERIONs have been explored. For each calculation discussed below, the valve area-time diagram will be given first.
Results and Discussion

7.6.5.1 Predictions Uses Different Valve Operating MODES

Four calculations have been run using the 4 different valve operating MODEs on a coupled 2D model to investigate the influence of numerical valve simulation on the predictions.

The cell open criterion used is CRITERION 2 during the valve opening process and CRITERION 1 during the valve closing process. A total of 10 cells are placed along the valve boundary, and the grid system as shown in Figure 7.66 is adopted. The valve area-time diagram for the 4 calculations are given below in Figures 7.67 to 7.70.

![Numerical Valve Operation Schedule](image)

FIGURE 7.67 Valve Open Area - Time Diagram: MODE 1
Results and Discussion

Numerical Valve Operation Schedule (Mode 2)

FIGURE 7.68 Valve Open Area - Time Diagram: MODE 2

Numerical Valve Operation Schedule (Mode 3)

FIGURE 7.69 Valve Open Area - Time Diagram: MODE 3
Results and Discussion

Numerical Valve Operation Schedule

(Mode 4)

FIGURE 7.70 Valve Open Area - Time Diagram: MODE 4

Clearly, the corresponding area opened at each open cell is decreasing during the valve opening process in MODE 1 and MODE 2, and has the opposite trend in MODE 3 and MODE 4. During the valve closing process, MODEs 2 and 4 display an increasing trend, whereas MODEs 1 and 3 display an opposite trend. The reason for this is that the edge of the valve cell in this two dimensional axisymmetrical model represents the height of an annular rim. For the same annular heights, the area of the outer annulus is larger than the area of the inner one.

Figures 7.71 to 7.74 present the pressure variation in time at STATION 1 obtained from the 4 calculations respectively.
Results and Discussion

Pressure History at Station 1
(2D/2D model, outer-centre, valve grids: 10)

FIGURE 7.71 Pressure Time History at STATION 1: MODE 1

Pressure History at Station 1
(2D/2D model, outer-outer, valve grids: 10)

FIGURE 7.72 Pressure Time History at STATION 1: MODE 2

257
Results and Discussion

Pressure History at Station 1
(2D&2D model, centre-centre, valve grids: 10)

FIGURE 7.73 Pressure Time History at STATION 1: MODE 3

Pressure History at Station 1
(2D&2D model, centre-outer, valve grids: 10)

FIGURE 7.74 Pressure Time History at STATION 1: MODE 4
Results and Discussion

On the whole, the pressure pulse in all four results follows well the overall shape of the pulse measured in the experiment. However, the peak amplitude of the pulse is underpredicted slightly and the pulse has a leading phase error. The leading phase error occurring in the first half of the pulse is the result of the larger valve open area in the simulation than the experiment at any time instant as shown in Figures 7.67 to 7.70. Because more mass flows out of the cylinder into the pipe, the pressure in the pipe rises faster at any instant. The phase error in the second half of the pulse is the response to the leading phase error generated at the valve opening process. However, this phase error is reduced slightly in the valve closing process. The predicted pressure pulses shown in Figure 7.71 to 7.74 are all characterised by a step pattern. This is essentially the effect of discrete valve cell open operation, and this will be confirmed by the dense grid calculation result to be shown later. At each cell opening or closing, a flow phenomenon similar to that observed in shock tube flow (Hirsch 1988) is triggered. The longer the time lapse between the current cell opening to the next cell opening, the longer is the width of the step in the pressure pulse curve. This explains why the first step of the pulse in Figure 7.71 is wider than that in Figure 7.73. The rate of pressure rise or decrease is a proportional function of the pressure ratio between the cylinder and the pipe.

There are large oscillations near the pulse peak in the MODE 3 and MODE 4 results, and also some in the region just after the pulse. In addition, the steady-state pressure level in the MODE 3 and MODE 4 results is noticeably lower than the experiment, whereas in the MODE 1 and MODE 2 this is very close to the measurement. These features indicate that the option to open the numerical valve from the circumference inwards towards the axis is a superior valve simulation method. For the valve closing operation, the method simulating the valve closing from the outer towards the centre seems to have produced slightly better results (comparing the second half pulses in Figures 7.71 and 7.72).

Figures 7.75 to 7.78 show the pressure pulse at STATION 2 predicted by the same 4 calculations.
Results and Discussion

Pressure History at Station 2
(2D & 2D model, outer-centre, valve grids: 10)

FIGURE 7.75 Pressure Time History at STATION 2: MODE 1

Pressure History at Station 2
(2D & 2D model, outer-centre, valve grids: 10)

FIGURE 7.76 Pressure Time History at STATION 2: MODE 2
Results and Discussion

Pressure History at Station 2
(2D&2D model, centre–centre, valve grids: 10)

FIGURE 7.77 Pressure Time History at STATION 2: MODE 3

Pressure History at Station 2
(2D&2D model, centre–outer, valve grids: 10)

FIGURE 7.78 Pressure Time History at STATION 2: MODE 4
Results and Discussion

The sharp rise at the leading edge of the pulse (non-linear effects) and the wave gradient decrease at the trailing edge (compared to STATION 1) have been captured well in all calculations. The shock amplitude in Figures 7.75 and 7.76 agrees well with the experiment, but it is over-predicted in Figure 7.77 and 7.78. There is one strong oscillation at the front of the shock in all results. The shock has been recorded at an earlier time in all 4 calculations indicating a leading phase error, this is believed to be the same phase error observed at STATION 1 which has been transported to STATION 2. The expansion wave position of the pulse has been predicted very well both in shape and in phase by all calculations. The very small phase error on this half of the pulse observed at STATION 1 becomes even smaller at STATION 2. It is also noticed that the ‘step’ pattern that dominates the pressure pulse predicted at STATION 1 has been smoothed out in the nonlinear wave predicted at STATION 2, this is believed to be the combined effect of nonlinear wave propagation and numerical smoothing. Despite the very poor prediction of the pressure pulse peak at STATION 1 by MODE 3 and MODE 4, the peak of the nonlinear shock wave at STATION 2 has been produced comparatively well. The predicted results at STATION 2 further confirm the conclusions drawn from the observation at STATION 1 concerning which valve simulation method is better.

Figures 7.79 to 7.82 present the time history of the pressure predicted in the cylinder.
Results and Discussion

Pressure History in Cylinder
(2D&2D model, outer–centre, valve grids: 10)

FIGURE 7.79 Pressure Time History in Cylinder: MODE 1

Pressure History in Cylinder
(2D&2D model, outer–outer, valve grids: 10)

FIGURE 7.80 Pressure Time History in Cylinder: MODE 2

263
Results and Discussion

Pressure History in Cylinder

![Graph showing pressure history over time](image)

FIGURE 7.81  Pressure Time History in Cylinder: MODE 3

Pressure History in Cylinder

![Graph showing pressure history over time](image)

FIGURE 7.82  Pressure Time History in Cylinder: MODE 4

264
Results and Discussion

The four predictions of the pressure change in the cylinder with time are in good agreement with the experiment. The rate of change agrees extremely well. The cylinder pressure is however lower at any time instant in the prediction as compared to the experiment, which is again due to the larger open area of the valve in the numerical simulation which results in more mass flowing out of the cylinder and therefore lower cylinder pressure. The predicted results oscillate slightly in the steady-state pressure region.

In Figure 7.79, predicted results are plotted for the pressure at two points in the cylinder. The locations of these two points are shown in Figure 7.52 (point 1 - (0, 0.0125), point 2 - (0.07, 0.045)). This indicates that the pressure is uniform over most of the cylinder region.

In summary, the pure 2D multiblock model has simulated very well the engine flow problem. The pressure pulses passing by STATION 1 and STATION 2 in the pipe have been well captured both in amplitude and in phase. The pressure condition change with time in the cylinder has also been predicted accurately. The numerical valve simulation methods MODE 1 and MODE 2 are superior to MODE 3 and MODE 4.

7.6.5.2 Predicted Results from a Dense Grid

The above simulation was re-run for one of the above calculations (Figures 7.71, 7.75 and 7.79) employing a grid of double the density in the direction perpendicular to the pipe axis, i.e. 20 grid cells across the valve as shown in Figure 7.83. This exercise is used to assess the influence of grid density on the overall predictions as well as the valve simulation.

![Figure 7.83 Dense grid system](image-url)


Results and Discussion

The predicted results are shown in Figures 7.84 to 7.86, and the area-time diagram in Figure 7.87.

Pressure History at Station 1

(2D&2D model, outer-centre, valve grids: 20)

Pressure History at Station 2

(2D&2D model, outer-centre, valve grids: 20)

FIGURE 7.84 Pressure Time History at STATION 1

FIGURE 7.85 Pressure Time History at STATION 2
Results and Discussion

Pressure History in Cylinder
(2D&2D model, outer-centre, valve grids: 20)

FIGURE 7.86  Pressure Time History in Cylinder

Numerical Valve Operation Schedule
(Mode 1, dense grid)

FIGURE 7.87  Valve Area - Time Diagram
Results and Discussion

The predicted results from the previous calculation (Figures 7.71, 7.75 and 7.79) are also plotted for comparison.

At STATION 1, the two predictions match each other in terms of pulse waveform, location and amplitude. The only difference is the pattern of the 'steps'. The dense grid result has twice as many steps. These results confirm that the 'step' pattern appearing in the predictions is indeed caused by the discrete valve operation in the numerical simulation.

At STATION 2, the results from the dense grid are almost coincident with the previous calculation results. However, the oscillation at the shock front becomes worse in the dense grid. It is suspected that this oscillation might have something to do with the grid aspect ratio in the pipe region, because the aspect ratio is already high in the previous grid system and doubled in the dense grid system.

In the cylinder, the pressure prediction from the dense grid is very similar to the previous grid, apart from the steady-state solution which oscillates less.

On the whole, the increase of the number of cells resolving the valve seems only to improve the resolution of the 'steps'. It implies that as the number of valve cells goes to infinity, the 'step' in the pulse at STATION 1 would diminish.

7.6.5.3 Predictions from a 1D/2D Coupled Model

The calculation presented in section 7.6.4, specifically MODE 1 and MODE 4 calculations, are now re-run in a 1D/2D coupled model in order to evaluate the capability and the quality of the 1D/2D coupled modelling concept developed in this research for simulating this engine problem. As stressed already, this is the centre of the interest for the present work. The grid system as shown in Figure 7.65 is employed in the 1D/2D model.

The area-time diagram of the MODE 1 calculation is given in Figure 7.67. The fully 2D model predictions are shown in Figures 7.71, 7.75 and 7.79. In the following, the predictions from a 1D/2D coupled model are presented in Figures 7.88 to 7.90, with the results from Figures 7.71, 7.75 and 7.79 plotted as well for comparison.
Results and Discussion

Pressure History at Station 1
(2D&1D model, outer-centre, valve grids: 10)

![Graph showing pressure history at Station 1]

FIGURE 7.88 Pressure Time History at STATION 1: MODE 1 + 1D/2D model

Pressure History at Station 2
(2D&1D model, outer-centre, valve grids: 10)

![Graph showing pressure history at Station 2]

FIGURE 7.89 Pressure Time History at STATION 2: MODE 1 + 1D/2D model

269
Results and Discussion

Pressure History in Cylinder

(2D&1D model, outer-centre, valve grids: 10)

FIGURE 7.90 Pressure Time History in Cylinder: MODE 1 + 1D/2D model

From the above figures, it is clear that the 1D/2D model predictions of the pressure pulse at the two STATIONs in the pipe and the pressure change with time in the cylinder are almost exactly the same with the fully 2D model predictions. The only noticeable difference occurred in the pressure pulse at STATION 1 where the 1D/2D coupled model produced an oscillation at the first step jump and subsequently a slightly bigger oscillation at the shock front at STATION 2.

The area-time diagram of MODE 4 calculation is given in Figure 7.78. Again, the predictions from the 1D/2D coupled model were almost the same with those from the fully 2D model as presented in Figures 7.74, 7.78 and 7.82, so they are not shown again.

7.6.5.4 Predictions from Different Cell Opening Criteria

A variety of valve cell opening methods were investigated for their influence on the predictions. The valve simulation method MODE 1 was employed in this series of
Results and Discussion

exercises. The calculations are performed on both the 1D&2D coupled model and the 2D&2D coupled model so that the 1D&2D model can be further validated by the experiment as well as compared again with the 2D&2D model to confirm further some conclusions that have been already drawn.

One valve-cell opening method has been used in the previous calculations, the valve area variation with time in this approach being illustrated in Figure 7.67. Based on this approach, the second test was to use CRITERION 2 for valve-cell open during the valve closing process. The corresponding time-valve area diagram is shown below in Figure 7.91. It is obvious that the part of the diagram corresponding to the valve opening process is the same as that shown in Figure 7.67, the difference is in the valve closing process.

The predictions from this test are shown in Figures 7.92 to 7.94.
Results and Discussion

Pressure History at Station 1

FIGURE 7.92 Pressure Time History at STATION 1

Pressure History at Station 2

FIGURE 7.93 Pressure Time History at STATION 2
Results and Discussion

Pressure History in Cylinder

Comparing Figure 7.92 with Figure 7.71 (STATION 1), it is observed, as expected, that the first half of the pressure wave is the same in both calculations due to the same valve and valve-cell opening simulation methods. For the second half of the pulse, because the valve closing area is always smaller at any time instant in this test, compared to calculation described in Figure 7.67, the pressure drops more slowly than the experiment at any time, therefore a lagging phase error eventually occurs at the later stages of this half of the pressure wave after the leading phase error created at the valve opening process has been compensated. This demonstrates again that the valve-cell open method does affect the phase accuracy of the pressure wave. Comparing Figure 7.93 with Figure 7.75 for the STATION 2 pulse prediction, again the first half of pulse is exactly the same, including the shock amplitude. However, the second half of pulse in Figure 7.93 has a lagging phase error, unlike that in Figure 7.75. This is the transfer of the phase error from STATION 1. Comparing Figure 7.94 with Figure 7.79 for the cylinder pressure results, apart from the steady-state pressure level being much lower in Figure 7.94, the two results are almost
Results and Discussion

identical. The lower cylinder pressure obtained from the current test is due to the comparatively more overall mass flowing out of the cylinder during the whole valve operating period.

Still based on the valve-cell open approach as shown in Figure 7.67, the third test is to use CRITERION 1 for both valve opening and closing processes. The resulting area-time diagram is given below in Figure 7.95. This probably represents the best overall simulation of the valve-opening process possible in the current approach.

![Numerical Valve Operation Schedule](image)

**FIGURE 7.95** Valve Opening Area - Time Diagram (CRITERION 1 - CRITERION 1)

The predicted results are shown in Figures 7.96 to 7.98.
Results and Discussion

Pressure History at Station 1
(1D&2D model, outer-centre, valve grids: 10, area always small)

![Graph showing pressure history at Station 1.]

FIGURE 7.96 Pressure Time History at STATION 1

Pressure History at Station 2
(1D&2D model, outer-centre, valve grids: 10, area always small)

![Graph showing pressure history at Station 2.]

FIGURE 7.97 Pressure Time History at STATION 2
Results and Discussion

Pressure History in Cylinder

Comparing Figure 7.96 with Figure 7.71 for STATION 1 pulse predictions, the phase error in Figure 7.71 is removed from both first and second halves of the pulse. In addition, the amplitude of the pulse peak is now much closer to the experiment. This valve-cell method has improved the quality of pulse prediction at STATION 1 as might be expected from the area-time diagram of Figure 7.95.

Comparing Figure 7.97 with Figure 7.75 for the STATION 2 pulse prediction, there is now no phase error, unlike the leading error in Figure 7.75. This confirms that the phase error appearing in the pulse at STATION 2 is largely transferred from the error in the pulse at STATION 1. The oscillations at the shock front in Figure 7.97 are also much smaller compared to those in Figure 7.75, but, the shock amplitude is over-predicted in Figure 7.97.
Results and Discussion

Comparing Figure 7.98 with Figure 7.79 for the cylinder pressure change prediction, both predictions follow the pressure change measured from the experiment well, but Figure 7.98 produces the closest value of all situations tested.

<table>
<thead>
<tr>
<th>TABLE 7.15 CPU Time and Memory Usage Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>Hybrid 1D&amp;2D coupled model</td>
</tr>
<tr>
<td>number of cells</td>
</tr>
<tr>
<td>memory (Mbyte)</td>
</tr>
<tr>
<td>total CPU time (secs)</td>
</tr>
</tbody>
</table>

Table 7.15 lists the memory usage by the hybrid 1D&2D coupled model and the uniform 2D model respectively in computing the above transient engine exhaust flow. Also listed is the CPU time to calculate the pressure wave propagating up to STATION 1 (calculation results are shown in Figure 7.96). The advantage in both computing time and resource requirement by the hybrid coupled model is clearly demonstrated, and the capability of the hybrid coupled modelling approach in reducing the cost of engine flow simulations is proven.

In summary, the valve-cell open method which uses CRITERION 1 during both valve opening and closing processes produces the best simulation. The 1D/2D coupled model in the new tests has again produced the same quality of prediction results as the 2D/2D coupled model.

In most of the figures above, predicted results by Kirkpatrick et al. (1994) (note that they are referred to as Blair’s prediction in these figures due to the use of Blair’s simulation method) for the same test apparatus but using Blair’s 1D wave tracing method (Blair 1996) are plotted. Their predictions are in excellent agreement with experiments. However, this is the result of a rather complex boundary condition treatment included for valve opening. Blair et al. (Blair et al. 1995, Kirkpatrick et al. 1994) stressed that this boundary condition is the most important element of any gas dynamic simulation; failure to accurately predict the wave entering the pipe system renders the simulation of its subsequent propagation
Results and Discussion

extremely difficult. The boundary scenario they used for the simulation of the test apparatus is a pipe/volume interface, and two distinct flow regimes, i.e. inflow and outflow (relative to the volume). Three regions near the boundary interface were identified in Blair’s boundary condition modelling, i.e. volume, throat and pipe regions. Assumptions were made on the thermodynamic nature of the flow process from volume to throat and throat to pipe, and vice versa. Fundamental thermodynamic equations were then implicitly written for each flow process at the boundary interface in terms of the unknown parameters and solved simultaneously with the equation system for the internal pipe flow using a matrix implementation of a Newton-Raphson process for multiple polynomials. Apart from the complexity of the boundary treatment, the modelling of the boundary condition also relied on empirical information. For example, a trial and error method was used to determine a suitable assumption for the flow processes at the boundary interface; some empirical parameters, e.g. a discharge coefficient, were employed in writing the equations for these flow processes and the coefficients are adjusted to give best agreement. So this boundary condition modelling approach requires substantial experimental input. In contrast, the hybrid approach in the present work does not involve any boundary condition specification of the valve flow, it is part of the solution, and hence completely independent of experimental inputs. It is believed therefore that the agreement with the experiments of the present predictions indicates very good performance of the method. The agreement is perhaps not as good as the Kirkpatrick predictions (although the peak pressure at STATION 2 is perhaps better predicted). This is, however, particularly true at the leading and trailing edges of the pressure pulse and the pulse predictons are certainly strongly affected in these regions by the decision to convert the valve opening into a 2D approximation. If this were removed by effecting the hybrid approach to a 2D/3D and 2D/3D/2D/1D system, then it is likely that the CFD predictions would be significantly improved. It must also be pointed out that the present predictions assume laminar flow, whereas the flow is almost certainly turbulent. This is, however, only likely to exert a small influence on the predictions as this will probably only affect the near-wall regions which will probably be a weak effect; nevertheless, this will also improve the current predictions.
Results and Discussion

7.6.5.5 Summary

The 1D/2D coupled model has been demonstrated as capable of predicting this engine exchange flow as faithfully as a uniform 2D model, the prediction results from the two models are almost identical. They both agree very well with the experimental measurements. Therefore, the 1D & 2D coupled modelling approach becomes superior due to its low cost and faster turn-around time.

The numerical method to operate the valve opening affects the prediction results noticeably, as does the assumption of which cells constitute the valve. It is found that the combination of CRITERION 1 and MODE 1 or MODE 2 produces the best simulation among the tested options.
7.7 References


Results and Discussion


281


Chapter 8

Recommendations For Future Work and Conclusions
Recommendations for Future Work and Conclusions

8.1 Summary and Conclusions

In this thesis, an introduction was given in Chapter 1 to contemporary methodologies for predicting IC engine cylinder-valve-pipe flows, which prompted a proposal for this work to examine a new simulation approach involving coupling different dimensions of solutions in different zones, in particular to develop a 1D/2D coupled CFD model. The basis of two unified solution procedures - pressure-correction and block implicit algorithms - were reported in Chapters 2 and 3 respectively. A general strategy to enable the proposed 1D and 2D coupled simulation was outlined in Chapter 4, and its implementation into both solution procedures was described in Chapters 5 and 6 respectively. Benchmark validation has been carried out for the two unified solution procedures including the coupling methodology and was reported in Chapter 7, including the prediction of an experimental rig simulating two-stroke-like engine exhaust flow in a cylinder-valve-pipe system.

The first achievement of the work reported here was a comprehensive investigation and comparative study of a unified, semi-implicit pressure-correction procedure and a fully implicit block-implicit procedure. Both procedures are able to compute flows at all speeds, which makes them possible candidates for IC engine flow predictions. The unified pressure-correction procedure reproduces the same prediction performance for low speed flows as the original incompressible algorithm. Its predictive quality for compressible flows was shown to be comparable with conventional compressible algorithms. The unified block-implicit procedure was as good as the original density-based algorithm for the prediction of compressible flow. It was also shown to be able to simulate low speed flow but suffered from an inefficiency problem. The accuracy of the predictions from the two unified solution procedures were comparable. The unified block implicit procedure was however considerably more expensive compared to the unified pressure-correction procedure with regard to memory requirements. In computing steady-state flows, the block-implicit calculation took more computing time, so the pressure-correction algorithm was superior in this category of flow computations. In computing unsteady flows, the block-implicit calculation took only half the computing time of the pressure-correction method. However, it is believed that PISO or similar variants to the SIMPLE method
would improve the pressure-correction calculation with little increase of resource requirement. Therefore, the pressure-correction procedure is believed to be the preferred approach for IC engine flow simulation.

The second achievement was the development of a novel modelling concept - a 1D/2D coupled CFD model for IC engine cylinder-valve-pipe system simulation. This modelling method has a significant advantage over the traditional uniform 2D approach in terms of computing time and computing resource, and hence computing cost and overhead time. In the prediction of either unsteady benchmark flows or a transient IC engine exhaust blow-down flow, the results from the 1D/2D coupled model were in all respects essentially identical to those from a uniformly 2D model. The cylinder and exhaust flow results were in excellent agreement with the experimental measurements. Both the generation and propagation of nonlinear pressure waves were well captured, as well as the formation of a shock. The cylinder, valve and pipe flows were successfully simulated simultaneously. Hence, the 1D/2D coupled model was capable of predicting this type of IC engine flow with good accuracy. The 1D/2D coupled modelling approach for engine flow calculation is superior than the traditional boundary condition approach, because it does not require empirical inputs and corrections, and it has avoided the complex and sometime cumbersome boundary condition specification. In most cases, the boundary condition approach suffers from an accuracy problem in transient predictions. The 1D/2D coupling approach was demonstrated as being sufficiently generic that it can be integrated into both pressure-correction and block-implicit procedures, and produced a similar quality of predictions. The implicit coupled model was superior than the explicit coupled model due to the full conservation at the 1D/2D model interface. The computing time and resource requirement were primarily determined by the properties of the base solution procedure, but significant savings are clearly possible using this approach.

8.2 Recommendation for Future Work

The results presented in the previous chapter have shown that the objective set for this research work has been successfully achieved. However, they also reveal that some other
Recommendations For Future Work and Conclusions

investigations and improvements should be conducted in the future. These are discussed in the following.

8.2.1 Improve Numerical Valve Simulation

The predicted results for the engine exchange flow problem have shown that the numerical simulation of the valve operation has influenced greatly the accuracy of the solutions, the mark of discrete valve movement is always visible in the results. It is believed that as the number of cells at the interface representing the valve increases, the effect of these would become smaller. However, as the size of the interface cells shrink, the allowable time step in the calculation would also be reduced. It is also obvious that it would be beneficial to resolve the 3D flow in and near the valve open area for highest accuracy, leading to the employment of a 3D model on either sides of the valve, and subsequently introducing a 3D/2D interface as well as a 2D/1D interface. These alternatives would result in an increase of the computing time, although savings compared to a fully 3D calculation would still be possible. However, a better approach to simulate the valve numerically should be investigated.

The valve would be approximated more realistically if the valve open area at each discrete time instant in the calculation can be represented exactly. Los Alamos National Laboratory has developed such a method - SNAPPER LOGIC - for its KIVA-3 codes (Amsden et al. 1992) to enable the simulation of moving pistons. This method represents the position of any moving part accurately at each computing time. The current method may be able to use it to improve the valve simulation here. The SNAPPER method would simulate the moving valve in the following way
At time 1 as illustrated in Figure 8.1, the valve location is coincident with grid line 3, and no snapper logic need to be applied. At time 2, the valve has moved to a location below grid line 3 but still closer to line 3 compared to line 2, in this situation, SNAPPER LOGIC is activated to move grid line 3 from its original location to a new valve location, i.e. where the valve edge is currently positioned. This results in an exchange of control volume size between grid lines 4 and 3, and grid lines 3 and 2; a volume average interpolation method
Recommendations For Future Work and Conclusions

is employed to obtain variable values in the modified cells. At time 3, when the valve moves to a location which is closer to grid line 2 compared to the original location of grid line 3, the SNAPPER LOGIC would force grid line 3 to move back to its original location while grid line 2 now is moved up from its original location to the new valve open location. As a result, control volumes at grid lines 2 and 1, 3 and 2 are now changed, appropriate interpolation is subsequently applied to evaluate variable values in these new snapped cells.

SNAPPER LOGIC is a straightforward method to achieve the exact representation of valve open area at any time instant during the computation. However, because SNAPPER LOGIC can temporarily generate abruptly changed local mesh sizes, it needs to be investigated whether this will introduce similar “steps” in the pressure rise or a time step restriction or additional numerical errors.

8.2.2 Improve 1D and 2D Coupling

In the simulation of the single pulse experiment, slip wall boundary conditions were applied to the pipe and a constant velocity profile is always assumed over the pipe cross section in the 1D portions. While these assumptions are valid in the flow conditions considered in this work, this is not always true. For example, when the pipe flow is slow and boundary layers are fairly thick, real wall boundary conditions and a more acceptable velocity profile shape should be employed. Methods need to be developed to match the flow profile at the interface of the 1D and 2D coupled models. If a 1D flat profile is transferred into the 2D model, this ‘false’ profile would then be developed into a new physically realistic shape by the numerical scheme in the 2D model. A flow disruption near the interface would be expected and flow in the 2D zone would undergo a period of transition before a sensible flow profile were established. The main concern here is to what extent this flow disruption would affect the key flow features, such as pressure waves, and the impact of the transition in the 2D zone on the flow field downstream. Obviously, the more rapid the transition, the smaller the extent of influence. Another approach is to replace the flat profile at the interface cross section given by the 1D model by an equivalent approximated curved profile, and take this as the input to the 2D model. Flow
Recommendations For Future Work and Conclusions

disruption at the interface would be reduced, and a better transition behaviour in the 2D model would be expected.

Investigations also need to be made into how to represent at least some features of a non-constant distribution of the flow variable in the 1D model. For 2D fully developed turbulent flow or transition flow, the shape function which describes the axial velocity profile becomes more complex, and in-depth studies need to be conducted in order to find out ways of representing the profile shape. This would probably involve the appearance of some shape factor coefficients in the 1D equation, as has been used in integral profile boundary layer methods for many years. Once a non-constant variable value is approximated for the cross section in the 1D model, parabolic interpolation instead of linear interpolation method should be used in the 'base-cell-concept' approach to obtain the variable value for the fictitious cell created out of the base cell in the one dimensional model.

8.2.3 Implementation of a Turbulence Model

The engine exchange flow in the single-pulse rig simulated in this work would contain strongly turbulent flow in the vicinity of the moving valve. The flow behaviour downstream of the valve, including the propagating pressure waves, is predominately determined by the flow structure in this region. Therefore, the accurate capture of flow physics in the valve flow region is crucial to the accurate prediction of the whole exchange flow process. In the present work, no turbulence model was employed in the calculation. The good agreement between predicted results and experiments is probably because the time required to develop the turbulence is longer than the formation and propagation of the first pressure wave - the first pressure wave propagation is faster than the turbulence developing time scale. But, after some time, the turbulent effect would come into account. It is believed that a turbulence model should be employed for this type of engine flow simulation if good quality of prediction is to be achieved for any operating conditions. So, the implementation of a turbulence model should be considered in the future work.
8.2.4 Extend 1D/2D Coupling Method to 1D/2D/3D Coupling

The objective of this research was to develop a coupling methodology so that hybrid dimensional CFD models, i.e. mixed 1D, 2D and 3D models, can be used simultaneously in one calculation. Although in this work, the implementation and validation of the developed coupling strategy was demonstrated only on a hybrid 1D/2D model, the coupling method is general and can be readily extend to hybrid 1D/3D and 2D/3D models.

One ultimate application of this hybrid modelling approach is engine flow prediction. For best practical impact, any dimensional model should be possible. Therefore, it is important that coupling methods are extended to 3D and implemented for the 3D/1D and 3D/2D hybrid models.

Results in Chapter 7 have demonstrated that the flux conservation approach in coupling 1D/2D zonal interface has achieved a faithful transfer of linear/nonlinear propagating pressure waves across the interface. Therefore, this coupling approach should also be adopted in the extended 1D/2D/3D hybrid model. Similarly, the proven successful "Base Cell Concept" in implementing the coupling strategy should also be employed.

An example of a hybrid 3D/2D model in either a cartesian coordinate system or a cylindrical coordinate system is illustrated in a schematic way in Figures 8.2 and 8.3 respectively:
Recommendations For Future Work and Conclusions

FIGURE 8.2 Schematic demonstration of a 2D/3D hybrid model in cartesian coordinate system

FIGURE 8.3 Schematic demonstration of a 2D/3D hybrid model in cylindrical coordinate system
Recommendaions For Future Work and Conclusions

As explained in Chapter 4, in a 1D/2D hybrid model, the 1D model can be seen as a special 2D model with only one cell in one particular coordinate direction. Similarly, in a 2D/3D coupled model, the 2D model can be viewed as a special 3D model with only one cell in one coordinate direction along which flow variables are constants. The interface of a hybrid model should lie in the region where flow can be approximated as the lower dimensional flow. It is recommended to employ the conservative coupling method in the 2D/3D model because this method will, according to the results from this work, ensure the flux into each 2D cell through the interface equal the sum of the fluxes through the face butting this 2D cell at the interface. In applying the “Base Cell Concept”, the 2D cell becomes the base, fictitious cells will be generated inside the 2D base cell to form a continuous contact with the 3D cells on the other side of interface.

As stressed already, the target practical application area of the 1D/2D/3D coupled hybrid modelling approach is transient engine flow. For an engine system in real life, a typical CFD model is shown in Figure 8.4 (in this case an unstructured mesh is used)
Recommendations For Future Work and Conclusions

There are real possibilities that flows in some parts of this system, such as the long exhaust pipe, can be approximated as two or one dimensional. With freedom in choosing the dimension of the CFD model according to the flow nature in the local flow regime, the overall model can be optimised in terms of size and accuracy. As a result, computing resource requirement can be reduced, which will then increase the ability to perform a simultaneous calculation of the whole engine system. A simultaneous engine flow model will enable the study of transient interaction of flows in various components, e.g. intake manifold, port/valve, chamber, exhaust pipe, etc., and consequently enable the optimisation of each component design to improve the whole engine system performance. All these would constitute a big step forward towards achieving the goal of simulating transient engine under real-time operating condition.
Recommendations For Future Work and Conclusions

8.2.5 Integrate the PISO Method With Pressure-Correction Approach

It has been mentioned in previous chapters that the computing speed of pressure-correction procedures for transient flows could be improved if PISO (or similar) methods were substituted for the SIMPLE method. Therefore the PISO method should be integrated into the codes and investigated in future work.

8.2.6 Investigate High Order Time Scheme

The studies carried out in Section 7.4.1.1 on the sensitivity of solution accuracy to the mesh/time-step size have shown that (see Figures 7.29, 7.32, 7.33 and 7.34), along with the restriction on the mesh size, there is also a restriction on the time-step size to ensure an acceptable accurate transient solution. The accuracy of transient solutions is influenced by the order of accuracy of time discretisation scheme. When a low order time scheme is employed, a relatively smaller time step is usually required to produce the same quality of transient solution as would be obtained when a higher order accurate time scheme is integrated. The size of time step also influences the efficiency of a time marching scheme. Obviously, the larger the allowable time step, the quicker the solution reaches steady-state. Therefore, in the future, higher-order time accurate integration schemes should be considered, so that a large time step may be allowed in the computation of transient flow problems. Crank-Nicolson second order time scheme is a simple and straightforward method, it can be used as the first try to see the difference the higher order time scheme can bring to the time accurate computation.

8.2.7 Investigate Acceleration Method for Low Speed Flows

The results in Chapter 7 have shown that a block-implicit algorithm is not efficient in solving steady state problems compared to the pressure-correction algorithm. This inefficiency occurs in both compressible and incompressible flow calculations, however, it is particularly severe in low Mach number flow calculations. This is because the time step is restricted by the speed of sound and the difference between the flow speed and the speed of sound is very large. The inefficiency problem in computing compressible flows has been partially resolved in this work by the integration of a Multigrid acceleration technique. Preconditioning techniques are however the most popular method in the
Recommendations For Future Work and Conclusions

literature to tackle the low speed flow computing inefficiency problem, so this technique should be investigated in the future for the possible improvement of the block-implicit algorithm in solving incompressible flows.
Recommendations For Future Work and Conclusions

8.3 References
