Chapter 4

Solution of the Navier–Stokes Equations

The motion of a fluid can be described by the Navier–Stokes equations, which are the continuity equation and the non-linear transport equations for the conservation of momentum, with additional transport equations for any scalar fields (such as temperature and concentration) that affect the flow. The coupling and solution of these equations is a non-trivial operation, and the coupling methods used in this study are derived and described in this Chapter. The basic scheme used is the SIMPLE scheme of Patankar and Spalding[123, 122] as modified by Rhie and Chow[139] for collocated (ie: nonstaggered) meshes, which has been modified to use a fractional-step method for transient flows. At the end of the Chapter the flow solver is applied to two benchmark problems as a test of the correctness of the implementation.

4.1 The Navier–Stokes Equations

The motion of a Newtonian fluid is described by the Navier–Stokes equations, which are a set of transport equations for the conservation of momentum and the continuity equation enforcing the conservation of mass. For a three dimensional flow the equations for the Cartesian velocity components \( u \), \( v \) and \( w \) can be written as

\[
\begin{align*}
\frac{\partial}{\partial t}(\rho u) + \nabla \cdot (\rho u u) &= \nabla \cdot (\mu \nabla u) + \frac{d\rho}{dx} + \rho g_x + S_x, \\
\frac{\partial}{\partial t}(\rho v) + \nabla \cdot (\rho u v) &= \nabla \cdot (\mu \nabla v) + \frac{d\rho}{dy} + \rho g_y + S_y, \\
\frac{\partial}{\partial t}(\rho w) + \nabla \cdot (\rho u w) &= \nabla \cdot (\mu \nabla w) + \frac{d\rho}{dz} + \rho g_z + S_z,
\end{align*}
\]

where \( \mu \) is the dynamic viscosity, \( \rho \) the pressure, \( g_i \) the acceleration due to gravity in the \( i^{th} \) coordinate direction, and \( S_i \) are any remaining body forces resolved onto the \( i^{th} \) axis, such as Coriolis forces due to a rotating reference frame. The first group on the left hand side of the equations in (4.1) is the transient term for the rate of change of momentum with respect to time, whilst the second term is the rate of change of momentum due to spatial changes in velocity, denoted advection. On the right hand side, the first term is a diffusion term quantifying shear stresses generated by the viscosity of the fluid and any gradients in the velocity field. This is followed by forces due to pressure gradients, buoyancy forces due to gravity, and a source term for any remaining body forces.

The continuity equation describes the interaction of the orthogonal velocity components, enforcing
the conservation of mass, and may be written as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \quad (4.2)$$

Other equations that affect a fluid flow are the energy and species equations, since these can force fluid motion through the creation of density gradients generating buoyancy forces. For a low speed flow where compressibility effects are ignored the energy equation can be written for the specific enthalpy $h$ as

$$\frac{\partial}{\partial t} (\rho h) + \nabla \cdot (\rho \mathbf{u} h) = \nabla \cdot \left( \frac{k}{c_p} \nabla h \right) + S_h, \quad (4.3)$$

where $k$ is the thermal conductivity, $c_p$ the constant pressure specific heat, and $S_h$ the enthalpy source terms. A similar equation can be written for the conservation of a chemical species $E$,

$$\frac{\partial}{\partial t} (\rho m_\alpha) + \nabla \cdot (\rho \mathbf{u} m_\alpha) = \nabla \cdot (\Gamma_\alpha \nabla m_\alpha) + S_\alpha, \quad (4.4)$$

where $m_\alpha$ is the mass fraction of species $\alpha$, $\Gamma_\alpha$ is the diffusivity of the species, and $S_\alpha$ is the rate of generation of the species from chemical reactions or mass transfer across a boundary.

For this thesis only incompressible flow will be considered, and as such the above equations can be simplified for small variations in temperature and concentration. Using the Boussinesq approximation we assume the density $\rho$ to be a constant and absorb the $\rho g$ term into the pressure gradient, with the exception of the variations in density due to temperature and mass fraction which generate buoyancy forces, which are assumed to be linear functions of temperature and concentration, giving

$$(\rho - \rho_0) g_i = \rho g_i \beta (T - T_0) = \rho g_i B_T, \quad (4.5)$$

$$(\rho - \rho_0) \alpha g_i = \rho g_i \beta_\alpha (m_\alpha - m_\alpha) = \rho g_i B_\alpha,$$

where $\beta$ is the coefficient of volumetric expansion for temperature and $\beta_\alpha$ is it’s equivalent for species.

By dividing the Navier–Stokes equations (4.1) by $\rho$, using the above approximations for the buoyancy forces, and using the kinematic viscosity $\nu = \mu/\rho$, the equations in (4.1) become

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) = \nabla \cdot (\nu \nabla \mathbf{u}) + \frac{1}{\rho} \frac{\partial p}{\partial x} + (B_T + B_\alpha) g_x + \frac{1}{\rho} S_x,$$

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{w}) = \nabla \cdot (\nu \nabla \mathbf{w}) + \frac{1}{\rho} \frac{\partial p}{\partial y} + (B_T + B_\alpha) g_y + \frac{1}{\rho} S_y,$$

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{w}) = \nabla \cdot (\nu \nabla \mathbf{w}) + \frac{1}{\rho} \frac{\partial p}{\partial z} + (B_T + B_\alpha) g_z + \frac{1}{\rho} S_z. \quad (4.6)$$

Similarly, the constant density approximation reduces the continuity equation to

$$\nabla \cdot \mathbf{u} = 0. \quad (4.7)$$

The relationship between energy and temperature can often be approximated as linear, with $h = c_p T$. Using this approximation, and dividing through by the constant density, the energy equation (4.3) can be recast as an equation for temperature,

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u} T) = \nabla \cdot (\alpha \nabla T) + S_T, \quad (4.8)$$

where $\alpha = k/\rho c_p$ is the thermal diffusivity, and the source term $S_T = S_h/\rho c_p$. 
4.2 Solution of the Navier–Stokes Equations

The solution of the system of equations described in the previous section is complicated by their interdependence upon each other, with buoyancy forces arising from the temperature and species equations driving the momentum equations, and the advective terms in all equations depending upon the velocity field. To solve the system some method must be used to couple the equations together, and typically this is done iteratively, with an initial approximation of one equation being used in the solution of the others. The method described here for coupling the equations is the SIMPLE scheme of Patankar and Spalding[123, 122], as modified by Rhie and Chow[139] for collocated (ie: nonstaggered) grids. Rhie and Chow’s method for interpolating velocities from cell centres to cell faces is discussed, and a fractional-step scheme that takes advantage of a semi-explicit time marching scheme is described.

4.2.1 The SIMPLE Velocity–Pressure Coupling Scheme

The SIMPLE velocity–pressure coupling scheme was developed by Patankar and Spalding[123, 122] and has since been refined by a number of authors. The scheme is a predictor–corrector method, with an initial estimate for the velocity field from the Navier–Stokes equations being corrected with the continuity equation to force the conservation of mass. The prediction and correction operations are enclosed in an iterative loop which (hopefully) converges to give a solution that satisfies all the equations in the system. The initial scheme by Patankar and Spalding was for a staggered Cartesian mesh, with the velocity values being located at the faces of finite volume cells, and the pressure, temperature and other scalar variables being located at the cell centres. Rhie and Chow[139] extended the method to use collocated grids, where the velocities and the other variables are all located at the cell centres, and this has been further developed by Perić and other authors[129, 72, 43]. Such a grid allows for an easier conversion to non-Cartesian meshes, and is the method used in this study. It will be initially described for Cartesian meshes, with a generalisation to non-Cartesian meshes being discussed in Chapter 5.

The first stage of the solution process is the solution of the discretised versions of the momentum equations (4.6) using the current estimate of the pressure field, and using a cell face mass flux that is interpolated in some manner from the current estimate of the velocity field (this interpolation procedure is discussed in greater depth in Section 4.2.2). The momentum Equations (4.6) are in the same general form as the generic transport equation, the solution of which was described in Chapter 2. For a given mass flux field and pressure field they can be discretised into equations of the form

\[
\begin{align*}
  a_P u_P^* + \sum_{n=nb} a_n u_n^* &= S_x + A_x \frac{dp}{dx}, \\
  a_P v_P^* + \sum_{n=nb} a_n v_n^* &= S_y + A_y \frac{dp}{dy}, \\
  a_P w_P^* + \sum_{n=nb} a_n w_n^* &= S_z + A_z \frac{dp}{dz},
\end{align*}
\]  

(4.9)

where \( u^*, v^*\) and \( w^* \) are the new estimates of velocity in the \( x, y \) and \( z \) axis, and \( nb \) refers to the cells that are neighbours of cell \( P \), ie: \( nb = E, W, N, S, T, B \). The \( A_x, A_y \) and \( A_z \) terms are the areas of faces of the cell normal to the \( x, y \) and \( z \) axis. The pressure gradient can be found by interpolating the pressure field to the cell faces using a linear interpolation, and then approximating the gradient across
the cell with a centred difference as
\[
\frac{dp}{dx} \approx \frac{p_x - p_w}{\Delta x}, \\
\frac{dp}{dy} \approx \frac{p_n - p_s}{\Delta y}, \\
\frac{dp}{dz} \approx \frac{p_t - p_b}{\Delta z},
\]
for a regular mesh where \(\Delta x, \Delta y\) and \(\Delta z\) are the cell dimensions.

After the calculation of the velocity field estimates, the cell face velocities can be interpolated from their values at the cell centres, and the cell face mass fluxes can be calculated. For the eastern face of a cell the velocity normal to the face is \(u_e\), whilst the face has an area \(A_e\). The mass flow across the face is
\[
m_e = \rho A_e u_e. \tag{4.11}
\]
In general this interpolated velocity field will not be mass conserving (ie: it will not have a discrete divergence of zero), and so will not satisfy the discretised version of the continuity equation,
\[
m_e - m_w + m_n - m_s + m_t - m_b = 0, \tag{4.12}
\]
which can be written for the face velocities on a Cartesian meshes as
\[
\rho A_e u_e - \rho A_w u_w + \rho A_n u_n - \rho A_s u_s + \rho A_t u_t - \rho A_b u_b = 0. \tag{4.13}
\]
We therefore wish to calculate a corrected velocity field, \(u^{**}, v^{**}, w^{**}\) and \(p^{**}\) that is mass conserving, together with a corresponding pressure field. We do so by adding a velocity and pressure correction to the original estimation of the velocity and pressure fields
\[
\begin{align*}
  u^{**} &= u^* + u', \\
  v^{**} &= v^* + v', \\
  w^{**} &= w^* + w', \\
  p^{**} &= p + p',
\end{align*}
\]
where a dash ‘ signifies the correction field.

The expressions in Equation (4.14) are substituted into the \(u\) equation in Equation (4.9)
\[
a_P (u_P^* + u_P') + \sum_{n=nb} a_n (u_n^* + u_n') = S_x + A_x \frac{d}{dx} (p + p'), \tag{4.15}
\]
and the sum of the neighbouring velocity terms approximated by
\[
\sum_{n=nb} a_n (u_n^* + u_n') \approx \sum_{n=nb} a_n u_n^*, \tag{4.16}
\]
which should be valid as \(p' \to 0\) and \(u' \to 0\). Subtracting the momentum equation then gives an expression relating the correction pressure and velocity field to each other,
\[
a_P u_P' \approx A_x \frac{dp'}{dx}, \tag{4.17}
\]
or
\[
\begin{align*}
  u_P' &= \frac{A_x}{a_P} \frac{dp'}{dx}, \\
  v_P' &= \frac{A_y}{a_P} \frac{dp'}{dy}, \\
  w_P' &= \frac{A_z}{a_P} \frac{dp'}{dz},
\end{align*}
\]

where similar approximations have been made for the \( y \) and \( z \) components of velocity. By interpolating the expressions in Equation (4.18) to the faces of the cell, the corrected cell face velocities normal to the face are given by

\[
\begin{align*}
  u'_e &= \frac{A_{xe} y'_e - y'_P}{a_{pe}} \\
  v'_n &= \frac{A_{yn} y'_n - y'_P}{a_{pn}} \\
  w'_t &= \frac{A_{zt} y'_t - y'_P}{a_{pt}},
\end{align*}
\]  

(4.19)

with the \( a_P \) terms being approximated at the faces by a linear interpolation

\[
\begin{align*}
  a_{pe} &= \frac{a_{pe} + a_{me}}{2}, \\
  a_{pn} &= \frac{a_{pe} + a_{ne}}{2}, \\
  a_{pt} &= \frac{a_{pe} + a_{te}}{2}.
\end{align*}
\]  

(4.20)

In this interpolation \( a_{pe} \) is the \( a_P \) term in the equation for the cell \( P \), whilst \( a_{me} \) is the \( a_P \) term in the equation for cell \( E \). Substituting the equations in (4.14) into the discretised continuity equation (4.13) yields,

\[
A_e (u'_e + u'_w) - A_w (u'_w + u'_e) + A_n (v'_n + v'_e) - A_s (v'_s + v'_n) + A_l (w'_l + w'_e) - A_b (w'_b + w'_l) = 0,
\]  

(4.21)

which can be rearranged to form

\[
A_e u'_e - A_w u'_w + A_n v'_n - A_s v'_s + A_l w'_l - A_b w'_b = A_w u'_w - A_e u'_e + A_s v'_s - A_n v'_n + A_b w'_b - A_l w'_l.
\]  

(4.22)

Using the expressions for \( u' \) from Equation (4.19) and factorising yields an equation for the pressure correction

\[
b_P p'_P + b_E p'_E + b_W p'_W + b_N p'_N + b_S p'_S + b_T p'_T + b_B p'_B = c,
\]  

(4.23)

where

\[
\begin{align*}
  b_E &= \frac{A_e^2}{a_{pe}}, \\
  b_W &= \frac{A_w^2}{a_{pw}}, \\
  b_N &= \frac{A_n^2}{a_{pn}}, \\
  b_S &= \frac{A_s^2}{a_{ps}}, \\
  b_T &= \frac{A_l^2}{a_{pt}}, \\
  b_B &= \frac{A_b^2}{a_{pb}}, \\
  b_P &= -(b_E + b_W + b_N + b_S + b_T + b_B), \\
  c &= \frac{1}{\rho} (m_w - m_e + m_s - m_n + m_b - m_l).
\end{align*}
\]  

(4.24)

This can be solved for the pressure correction \( p'_P \), which is then used to update the cell centre and cell face velocities using Equations (4.19) and (4.14), the resulting face velocities satisfying the continuity equation. The pressure field is updated using Equation (4.14), and then the process is repeated, with the new velocity and pressure field being used to calculate the \( u^* \) velocities.

The SIMPLE algorithm is summarised in Figure 4.1 for the solution of a thermally driven three dimensional flow. To obtain the velocity, pressure and temperature fields, an estimate of the velocity field
$u^*$, $v^*$ and $w^*$ is calculated using Equation (4.9). The cell face velocities are then interpolated from the $u^*$ velocity field and the cell face mass fluxes calculated $m^*$. The pressure correction equation (4.23) is solved for $p'$, and the velocity, mass flux, and pressure fields are updated using Equation (4.14). The resulting mass conserving velocity field is then used to solve any transport equations for auxiliary scalar fields, such as Temperature and species concentration.

- set initial fields for $u, v, w, p$ and $T$.
- interpolate to find cell face mass fluxes $m$.
- repeat
  - solve Equation (4.9) for $u^*, v^*, w^*$.
  - interpolate to find cell face mass fluxes $m^*$.
  - calculate pressure correction $p'$ from Equation (4.23)
  - update $u, v, w, p$ and $m$ using Equation (4.14).
  - calculate $T$ and any other scalar fields.
- check for convergence. If converged, halt.

Figure 4.1: The SIMPLE velocity–pressure coupling algorithm.

The divergence of the $m^*$ mass flux field (calculated as the source term in Equation (4.23)) is usually used as a convergence criteria. The system at this stage satisfies the momentum equations (they having just been solved) whilst a divergence free $m^*$ mass flux field signifies the solution also satisfies the continuity equation. After the update of the $m$ fields with the pressure correction $p'$, the divergence should be equal to zero, a non-zero divergence signifying an incorrectly solved pressure correction equation. However, the corrections to the velocity fields means that the momentum equations may no longer be satisfied, and so the algorithm repeats.

To aid the convergence of the method, the velocity, pressure and scalar field updates can be under-relaxed using some relaxation parameter. There are two obvious methods of under-relaxation, either by relaxing the update

$$
\begin{align*}
  u &= u^* + \alpha_u u', \\
  v &= v^* + \alpha_u v', \\
  w &= w^* + \alpha_u w', \\
  p &= p^* + \alpha_p p',
\end{align*}
$$

(4.25)

where $\alpha_u$ and $\alpha_p$ are the relaxation parameters for the velocity and pressure fields respectively, or by relaxing the diagonal coefficient of the linear system for each variable, for example the $u$ velocity equation (4.9) being modified to

$$
\frac{\alpha_p}{\alpha_u} u_p^* + \sum_{n=1}^{nb} a_n u_n^* = S_x + A_x \frac{dp}{dx},
$$

(4.26)

The pressure equation relaxation should always be of the form of Equation (4.25), and no relaxation should be used in the update of the face velocities and mass fluxes. Unlike the transport equation, the pressure equation is always diagonally dominant and doesn’t require the under-relaxation of the diagonal. By not relaxing the face velocity updates a mass conserving flux field is ensured, which can ensure the conservation of energy, momentum and other properties. It can also be noted that if a relaxation of the form of Equation (4.26) is used for the momentum equations, then a non-relaxed value of $\alpha_p$ should be used in the pressure correction and velocity interpolation operations.

To model a transient problem the iterative process outlined in Figure (4.1) is carried out at every time step, using the velocity, pressure and scalar fields from the previous time step as the initial guess for the values at the new time step. This can be quite time consuming, and a more efficient time stepping procedure is described below in Section 4.2.3.

For steady state problems the SIMPLE coupling scheme can be considered as a pseudo-transient process, with an implicit calculation of the momentum equations being corrected via an explicit pressure
correction process, each iteration of the scheme corresponding to a pseudo time step. When modelling a transient flow, each time step comprises a number of pseudo-transient time steps to obtain the converged solution for the physically real time step.

The basic SIMPLE scheme was initially described by Patankar and Spalding in 1972[123]. Since then a number of modifications to the coupling method have been made (aside from the conversion to collocated or non-staggered meshes by Rhie and Chow[139] that is described above) – mostly with the aim of improving computational efficiency and speeding the coupling process. Variants rejoicing in the names of SIMPLEC[175, 135], SIMPLEX[135], SIMPLER[122] and PISO[69] all aim to improve the coupling of the momentum and pressure equations, via minor modifications to the SIMPLE algorithm. In this study the basic SIMPLE algorithm was used.

### 4.2.2 Rhie–Chow Velocity Interpolation

The key to the shift from a staggered to a collocated mesh is the interpolation of the velocity field to the cell faces. A naïve linear interpolation of the cell centre velocities can lead to a pressure checkerboarding process, where the pressures on odd and even numbered cells are uncoupled from each other[122].

The Rhie–Chow interpolation method[139] interpolates in a form consistent with the velocity correction equation (4.18) as follows. The equation for the \( u \) axis component of momentum can be written as

\[
up - \frac{Ax}{ap} \frac{dp}{dx} = \frac{Sz}{ap} - \frac{1}{ap} \sum_{n=nb} a_n un. \tag{4.27}
\]

Writing equation (4.27) for the \( P \) and \( E \) cells as

\[
up - \left( \frac{Ax}{ap} \right)_P \left( \frac{dp}{dx} \right)_P = \left( \frac{Sz}{ap} \right)_P - \left( \frac{1}{ap} \sum_{n=nb} a_n un \right)_P,
\]

\[
u_E - \left( \frac{Ax}{ap} \right)_E \left( \frac{dp}{dx} \right)_E = \left( \frac{Sz}{ap} \right)_E - \left( \frac{1}{ap} \sum_{n=nb} a_n un \right)_E,
\]

and assuming a similar equation can be written for the velocity at the east face of the cell

\[
u_e - \left( \frac{Ax}{ap} \right)_e \left( \frac{dp}{dx} \right)_e = \left( \frac{Sz}{ap} \right)_e - \left( \frac{1}{ap} \sum_{n=nb} a_n un \right)_e. \tag{4.29}
\]

This equation is then approximated by a linear interpolation of the equations for the centres of the \( E \) and \( P \) cells given in Equation (4.28). Performing such an interpolation for the left hand side of Equation (4.29) gives

\[
u_e - \left( \frac{Ax}{ap} \right)_e \left( \frac{dp}{dx} \right)_e = \frac{1}{2} \left( up - \left( \frac{Ax}{ap} \right)_P \left( \frac{dp}{dx} \right)_P + u_E - \left( \frac{Ax}{ap} \right)_E \left( \frac{dp}{dx} \right)_E \right), \tag{4.30}
\]

which can be rewritten as an expression for \( u_e \)

\[
u_e = \frac{up + u_E}{2} + \left( \frac{Ax}{ap} \right)_e \left( \frac{dp}{dx} \right)_e - \frac{1}{2} \left( \left( \frac{Ax}{ap} \right)_P \left( \frac{dp}{dx} \right)_P + \left( \frac{Ax}{ap} \right)_E \left( \frac{dp}{dx} \right)_E \right). \tag{4.31}
\]

The pressure gradients can be approximated by centred differences as

\[
\left( \frac{dp}{dx} \right)_e = \frac{pe - pp}{\delta xe},
\]

\[
\left( \frac{dp}{dx} \right)_P = \frac{pe - pw}{\delta xe + \delta xw}, \tag{4.32}
\]

\[
\left( \frac{dp}{dx} \right)_E = \frac{peE - pp}{\delta xe + \delta x_e}.
\]
The approximation of the \((A_{e}/a_P)_e\) term by a linear interpolation
\[
\left(\frac{A_{e}}{a_P}\right)_e = \frac{1}{2} \left( \left(\frac{A_{e}}{a_P}\right)_e + \left(\frac{A_{e}}{a_P}\right)_p \right),
\]
gives a complete interpolation formula for \(u_e\) as
\[
u_e = \overline{u_e} + \left(\frac{A_{e}}{a_P}\right)_e \left(\frac{dp}{dx}\right)_e - \left(\frac{A_{e}}{a_P}\right)_e \left(\frac{d\rho}{dx}\right)_e,
\]
where an overbar \(\overline{()}\) signifies a linear interpolation of the values at the \(P\) and \(E\) cell centres. Similar expressions can be obtained for the \(l\) and \(f\) face velocities, and the face mass fluxes \(F\) can be calculated by multiplying by the density and the area of the relevant cell face.

### 4.2.3 A Fractional-Step Method for Transient Flow

To solve the transient Navier–Stokes equations using the SIMPLE coupling scheme, the equations for the conservation of momentum (previously given in Equation (4.1) )
\[
\frac{\partial}{\partial t}(\rho u) + \nabla.(\rho \mathbf{u}) = \nabla.(\mu \nabla u) + \frac{\partial p}{\partial x} + \rho g_x + S_x,
\]
\[
\frac{\partial}{\partial t}(\rho v) + \nabla.(\rho \mathbf{v}) = \nabla.(\mu \nabla v) + \frac{\partial p}{\partial y} + \rho g_y + S_y,
\]
\[
\frac{\partial}{\partial t}(\rho w) + \nabla.(\rho \mathbf{w}) = \nabla.(\mu \nabla w) + \frac{\partial p}{\partial z} + \rho g_z + S_z,
\]
may be discretised using either a backwards Euler or Crank–Nicolson time stepping scheme. The Crank–Nicolson discretisation may be summarised as
\[
\frac{u^n - u^{n+1}}{\Delta t} + \frac{1}{2} \left( H^{n+1}(u^*) + H^n(u^n) \right) = -G_x(p^{n+1}) + \frac{\mu}{2\rho} (L(u^*) + L(u^n)),
\]
\[
\frac{v^n - v^{n+1}}{\Delta t} + \frac{1}{2} \left( H^{n+1}(v^*) + H^n(v^n) \right) = -G_y(p^{n+1}) + \frac{\mu}{2\rho} (L(v^*) + L(v^n)),
\]
\[
\frac{w^n - w^{n+1}}{\Delta t} + \frac{1}{2} \left( H^{n+1}(w^*) + H^n(w^n) \right) = -G_z(p^{n+1}) + \frac{\mu}{2\rho} (L(w^*) + L(w^n)),
\]
where \(H^m\) is the discrete advection operator calculated using the velocity field at the \(m\)th time step, \(G\) the discrete gradient, and \(L\) the discrete Laplace operator.

The \(u^*, v^*\) and \(w^*\) velocity fields calculated using Equation (4.36) will not generally satisfy continuity, and so a pressure correction \(p'\) is calculated and applied to correct the velocity fields,
\[
u^{n+1} = u^* + \Delta t G_x p',
\]
\[
v^{n+1} = v^* + \Delta t G_y p',
\]
\[
w^{n+1} = w^* + \Delta t G_z p',
\]
the \(u^{n+1}, v^{n+1}\) and \(w^{n+1}\) fields satisfying the continuity equation.

The new value of \(p^{n+1}\) is then used in Equation (4.36), with the advection operator \(H^{n+1}\) being re-evaluated using the new estimation of the \(n+1\) velocity field, \(u^{n+1}, v^{n+1}\) and \(w^{n+1}\). The process is repeated, hopefully iterating to convergence.

A more efficient method for calculating transient flow is to use a fractional-step method that solves the unsteady Navier–Stokes equations in a segregated manner, with the momentum and pressure equations being solved only once per time step. This removal of the iterative coupling at each time step achieves
a large gain in computational efficiency, a desirable feature in transient flows calculations which are generally computationally expensive. The fractional step methods were first proposed by Harlow and Welch[57] and Chorin[24], with the method of Chorin being modified by Kim and Moin[79] for a staggered mesh finite volume discretisation[79]. The scheme described here[4, 5] uses a modification of the SIMPLE coupling scheme described above, with the advection operator \( H \) for the momentum equations being applied using an Adams–Bashforth time discretisation instead of the Crank–Nicolson discretisation used in Equation (4.36). The discrete momentum equations given in (4.36) are modified to

\[
\begin{align*}
\frac{u^e - u^n}{\Delta t} + \frac{1}{2} H^n(u^n) - \frac{1}{2} H^{n-1}(u^{n-1}) &= -G_x(p^n) + \frac{\mu}{2\rho} (L(u^n) + L(u^n)), \\
\frac{v^e - v^n}{\Delta t} + \frac{1}{2} H^n(v^n) - \frac{1}{2} H^{n-1}(v^{n-1}) &= -G_y(p^n) + \frac{\mu}{2\rho} (L(v^n) + L(v^n)), \\
\frac{w^e - w^n}{\Delta t} + \frac{1}{2} H^n(w^n) - \frac{1}{2} H^{n-1}(w^{n-1}) &= -G_z(p^n) + \frac{\mu}{2\rho} (L(w^n) + L(w^n)),
\end{align*}
\]  

(4.38)

which are solved for the initial estimate of the velocity field, \( u^e, v^e \) and \( w^e \). A pressure correction, \( p' \), is then calculated and applied to obtain a mass conserving velocity field,

\[
\begin{align*}
u^{n+1} &= u^n + \Delta t G_x p' \\
v^{n+1} &= v^n + \Delta t G_y p' \\
w^{n+1} &= w^n + \Delta t G_z p' \\
p^{n+1} &= p^n + p'.
\end{align*}
\]  

(4.39)

In Equation (4.38) the advection operators \( H \) are evaluated for the previous two time steps, and no longer depend on the velocity field at time \( n + 1 \), and the pressure force is explicitly extrapolated from time step \( n \). The equation therefore no longer relies upon values at the \( n + 1 \) time step, and the solution to Equation (4.39) is the final estimate of the velocity and pressure fields for the \( n + 1 \) time step.

Unlike the transient calculations made using a Crank–Nicolson or Backwards Euler discretisation and a SIMPLE coupling scheme, there is no iteration necessary to couple the pressure and momentum equations. Instead the momentum equations and the pressure correction equation can be both solved just once per time step, resulting in a considerable gain in computation efficiency. The method has been shown to be second order accurate in time[4, 5], and shows similar error properties to the Crank–Nicolson second order (in time) SIMPLE scheme described in Equations (4.36) and (4.37).

### 4.3 Boundary Conditions

The individual boundary conditions that are used for the Navier–Stokes equations are essentially the same as those for the generic transport equation given in Section 2.3, but with the different velocity components and the pressure field having a mixture of Neumann and Dirichlet conditions at the boundary.

The simplest boundary condition is that for a stationary wall. At a wall the no-slip condition is imposed, with the velocity components being set to zero,

\[
u = v = w = 0, \tag{4.40}\]

Since the velocity is known at the boundary, the velocity correction at the boundary is \( u' = 0 \), and so from Equation (4.19) a Neumann boundary condition is found for the pressure correction equation of

\[
\frac{dp'}{dn} = 0, \tag{4.41}\]
where \( n \) is a direction normal to the boundary. Whilst a zero normal derivative is imposed in the pressure correction equation, the value for the pressure at the external boundary cell is set using a second order extrapolation from the internal nodes\([72, 129, 43]\). Failure to do so results in velocities normal to the boundary in the cells on the interior side of the boundary. A moving wall is implemented in the same manner as a fixed wall, with the velocities at the face being set to the wall velocity.

Symmetry planes are set with a mixture of Dirichlet and Neumann boundaries. For a symmetry plane normal to the \( x \) axis the velocities are set as

\[
\frac{du}{dx} = \frac{dv}{dx} = 0,
\]

whilst the pressure boundary condition is the same as that for a wall. For symmetry planes normal to the \( y \) and \( z \) axis the velocity boundary conditions are

\[
\frac{du}{dy} = v = \frac{dw}{dy} = 0,
\]

and

\[
\frac{du}{dz} = \frac{dv}{dz} = w = 0.
\]

As with solid boundaries, the pressure correction equation has a Neumann zero normal derivative boundary condition set at symmetry planes.

At flow inlets, where fluid enters the domain being studied, the velocities can be specified, whilst the pressure correction has a zero normal derivative boundary condition as with the wall boundary conditions. Outlet boundaries conditions are more empirical in basis, one typical set of boundary conditions being a zero derivative normal to the boundary for the velocities, with the pressure correction equation having a zero second derivative normal to the boundary. Another form uses an initial estimation of the velocities at the outlet by extrapolating from the interior nodes, with these extrapolated velocities being corrected to ensure global conservation of mass.

At a free surface the pressure is constant, with the velocities having similar boundary conditions to a symmetry plane. However the velocity normal to the surface is zero only for a steady surface, and typically has a non-zero value associated with wave motion of the surface. In general the free surface will only be planar for infinite Froude numbers, and some method must otherwise be found to couple the free surface geometry with the velocity and pressure boundary conditions. This topic is discussed further in Chapter 8.

For the fractional step method, the physical boundary conditions described above were applied to the intermediate velocity fields, \( u^*, v^* \) and \( w^* \).

### 4.4 Benchmark Solutions

To test the SIMPLE coupling scheme, and to compare the Finite Volume discretisations derived in Chapter 2, the code was applied to solve two classic benchmark test problems, both for two-dimensional steady internal flow. The first, the “driven cavity” problem, models flow in a square box which is driven by the motion of the lid of the box. The second models natural convection in a square cavity, where the side walls are at two different temperatures and the upper and lower boundaries are adiabatic.

#### 4.4.1 The Driven Cavity Flow

The driven cavity problem has long been used as a test case for Navier–Stokes solvers, thanks to it’s simple geometry and boundary conditions. The two-dimensional square cavity has no slip boundary
conditions on all walls, the flow being driven by the top wall sliding across the cavity whilst the side and bottom walls are static\(^{1}\). A diagram of the basic geometry is given in Figure 4.2.

The non-dimensional parameter describing such a flow is the Reynolds number, defined for the cavity by

\[
\text{Re} = \frac{U \ell}{\nu},
\]

where \(\ell\) is the width of the cavity, \(U\) is the velocity of the top surface, and \(\nu\) is the kinematic viscosity. For low Reynolds numbers, on the order of 100 or less, the viscosity dominates the flow and the first and second order differencing schemes give similar solutions. However, as the Reynolds numbers increases to 1000 or greater the errors in the solutions given by the first-order schemes become apparent.

Early numerical solutions to this problem include those by Kawaguti\(^{75}\), Simuni\(^{156}\), Mills\(^{111}\), Burggraf\(^{19}\), Runchal, Spalding and Wolfsstein\(^{144, 143}\) and Bozeman and Dalton\(^{15}\), with Burggraf providing a summary of previous work, including the recirculating flow theory of Batchelor\(^{10}\) and the analytic corner eddy model of Moffatt\(^{119}\). In more recent times the problem has been used as a standard case for comparing differencing schemes\(^{35, 56, 131, 124, 43}\). High resolution benchmark solutions for the problem are provided by Ghia, Ghia and Shin\(^{50, 49}\) with solutions for Reynolds numbers in the range of \(100\) to \(10000\) using a second order streamfunction–vorticity formulation. The problem has been extended by other authors to non-orthogonal meshes\(^{130, 36}\) and to three-dimensional flow\(^{34, 35, 184}\).

For the current study the problem was solved for a Reynolds number of 1000 on a series of regular meshes, the coarsest mesh being one of \(19^2\) cells, whilst the finest had \(131^2\) cells. For each mesh a calculation was performed using each of the differencing schemes described in Section 2.2 – the four first order methods, the FOU, Hybrid, Exponential and Power law differencing schemes, the three second order methods, the Central, SOU and MSOU differencing schemes, and the two third order methods, the QUICK and ULTRA differencing schemes. Results were compared with the benchmark solutions of Ghia et al\(^{50}\) which were generated using a finite difference vorticity-streamfunction solver using a Central difference approximation. To increase the data available a set of solutions were generated using a finite difference streamfunction-vorticity program (named SLIDE), which is similar to the method used by Ghia et al, but which used either central or FOU differencing.

\(^{1}\)The top boundary slides, and has a no-slip boundary condition imposed upon it in it’s moving reference frame. Such a boundary condition is different from a slip boundary, which has zero shear stress normal to the boundary.
The streamlines for the flow calculated on a $131 \times 131$ mesh using the QUICK differencing scheme are shown in Figure 4.3. The flow consists of a core of fluid undergoing solid body rotation, surrounded by a shear layer. Two small counter-rotating vortices are located in the lower corners. According to the theory of Moffatt[119] there should be an infinite cascade of corner vortices, but the mesh is only fine enough to capture a single sub vortex in the lower right corner.

Figure 4.3: The streamlines for a Re = 1000 flow calculated on a $131 \times 131$ mesh using the QUICK differencing scheme. Contours for streamfunction are at 0.01 intervals for $-0.11 < \psi < 0$, with further contours at $\pm 10^{-3}, \pm 10^{-4}, \pm 10^{-5}, \pm 10^{-6}$ and $\pm 10^{-7}$.

The streamlines calculated on a coarser mesh are shown in Figure 4.4 for calculations using the FOU and ULTRA differencing schemes. The ULTRA scheme is not greatly different from its fine mesh counterpart, but the solution calculated using FOU scheme has a weaker central vortex, and smaller corner vortices at the lower two corners.

Figure 4.4: The streamlines for a Re = 1000 flow calculated on a $51 \times 51$ mesh using the FOU scheme (left) and the ULTRA scheme (right). Contours are at 0.01 intervals for $-0.11 < \psi < 0$, with further contours at $\pm 10^{-3}, \pm 10^{-4}, \pm 10^{-5}, \pm 10^{-6}$ and $\pm 10^{-7}$.

Plots of the $x$ and $y$ components of velocity, the velocity magnitude, and the vorticity, calculated using the QUICK differencing scheme on a $131 \times 131$ mesh, are given in Figure 4.5. The vorticity and velocity magnitude plots reveal the flow as a central core of constant vorticity undergoing solid body rotation, surrounded by a shear layer that buffers the rotating core and the flow adjacent to the cavity boundaries.
CHAPTER 4. SOLUTION OF THE NAVIER–STOKES EQUATIONS

Figure 4.5: Contours of velocity components in the $x$ and $y$ axis ($u$ top left, $v$ top right), and of the velocity magnitude and vorticity ($|u|$ bottom left, $\zeta$ bottom right), for a Re = 1000 flow calculated on a $131 \times 131$ mesh using the QUICK differencing scheme. Velocity contours are at 0.05 intervals between $\pm 1$, and vorticity contours are at intervals of $\pm 100$.

The convergence of the different schemes as the mesh is refined is shown in Figure 4.6. The $x$ axis plots the grid size, whilst the $y$ axis plots the $u$ velocity at a location $\frac{1}{4}$ above the lower boundary on the vertical centreline of the cavity. This is the location given by Ghia et al as having the maximum negative $u$ velocity on the centreline of the cavity, and was found using a cubic interpolation of the velocity fields from the calculated solutions.

The top five lines are from calculations using the first order differencing schemes—the FOU, Hybrid, Exponential and Power law schemes using the SIMPLE solver described in this chapter, and a FOU scheme in the SLIDE finite difference vorticity-streamfunction solver. The power law and Exponential schemes give almost identical solutions, showing that the polynomial function chosen by Patankar is a good approximation of the exponential function. The first order schemes are all grossly in error, and are converging very slowly towards the correct solution. The most accurate of the first order schemes is the solution calculated using the Hybrid scheme, perhaps because regions where the cell Péclet number is less than 2 are using a central difference approximation. For the finest mesh this corresponds to a velocity less than 0.25, which is the case for most of the lower half of the cavity.

The two central difference solutions (calculated using the Finite Volume SIMPLE code and the SLIDE Finite Difference code) both converge at a similar rate, which is of interest considering the different
method being used to model the flow. The finest solution for the SLIDE finite difference code gives an almost identical solution to that from Ghia et al calculating on the same mesh. The second order upwind scheme overshoots the velocity, but the overshoot is reduced with the MSOU scheme, although the value is still greater than that given by the central difference benchmark. For all of the second order methods the error is less than that for the first order schemes, with the exception of the SOU scheme which has an error comparable to the Hybrid scheme on the coarsest mesh.

Finally the third order schemes, using QUICK and ULTRA differencing, converge faster than the second order methods, the QUICK scheme giving a better solution on the coarser mesh. For both the MSOU and ULTRA flux limiter schemes, the addition of the flux limiter damps the solution. For the SOU scheme, where there is an overshoot in the calculated velocity, this hastens the convergence with the exact solution, whilst for the QUICK scheme which monotonically converges with the exact solution, this has the effect of slowing mesh convergence.

From a comparison of the convergence of the velocity, the third order schemes are clearly the fastest converging methods, with the first order methods being the slowest to converge and exhibiting a large error.

The velocity profile along the vertical centreline of the cavity is shown in Figure 4.7, with solutions being shown for each of the differencing schemes for each calculated mesh, with the benchmark solutions of Ghia et al being included for comparison. For reasons of space no solution is given for the Power differencing scheme, but the solutions are nearly identical to those of the Exponential scheme. The large error in the first order solutions is readily apparent, which make a strong contrast with the comparatively rapid convergence of the third order schemes. The first order schemes under-predict the velocity peak in the lower half of the cavity, and smear out the shear layer between the central core of the main vortex and the boundary flow. On the finest mesh the Hybrid scheme is surprisingly accurate when compared with the other first order schemes, but this is thought to be due to the cell Péclet numbers being low enough on the fine mesh for the method to switch to using a second order central difference discretisation. The error in the SOU scheme is of interest, with a region of reverse flow being predicted on the coarser meshes, and with the velocity in the lower half of the cavity being overestimated.

Figure 4.6: The $u$ velocity at a location $\frac{1}{4}$th's above the cavity floor on the centreline of the cavity. For each of the differencing schemes the calculated velocity is plotted as a function of the mesh size $\Delta x$. The benchmark solution of Ghia et al is offset to $\Delta x = 0$ for clarity, although it was calculated on a regular mesh of $131^2$ points, with a mesh size of $\Delta x = 0.0076$. 
Figure 4.7: Profiles of the \( u \) velocity along the vertical centreline, calculated using different differencing schemes for a range of mesh sizes, and compared with the benchmark solutions of Ghia et al. The schemes used are (clockwise from top right), Hybrid, Central, MSOU, ULTRA, QUICK, SOU, Exponential and FOU differencing.
The smearing effect of a first order scheme is shown in Figure 4.8, which shows the distribution of the vorticity along the vertical centreline of the cavity for the finest mesh solutions obtained using the FOU and QUICK differencing schemes (i.e.: on a $131 \times 131$ mesh), together with the solution obtained with the vorticity-streamfunction finite difference solver on a $129 \times 129$ mesh. The finite volume QUICK solution agrees with the vorticity-streamfunction solution, revealing a central region of constant vorticity, with a thin shear layer at the upper boundary, and the presence of two overlapping regions of shear in the lower half of the cavity, the shear layer at the edge of the vortex core merging with the boundary layer at the base of the cavity. The excessive diffusivity of the first order upwind scheme manifests itself with an over-prediction of the width of the shear layer at the upper boundary, and a merging of the two lower shear layers into a single layer. The size of the constant vorticity core flow is reduced, with the vorticity of this region being under-predicted.

Finally, the vertical velocity profiles along the horizontal centreline are shown in Figure 4.9 for calculations made with the FOU and QUICK differencing schemes. As with the velocities along the vertical centreline the first order scheme grossly under-predicts the velocity whilst the third order scheme rapidly converges to the fine mesh solution.

Figure 4.8: The vorticity $\zeta$ distribution along the vertical centreline of the cavity. Solutions are given for calculations made using the FOU and QUICK differencing schemes on a $131 \times 131$ mesh, and are compared with a solution calculated using a finite difference vorticity-streamfunction code.

Figure 4.9: Profiles of the $v$ velocity along the horizontal centreline, calculated using the FOU (left) and QUICK (right) differencing schemes for a range of mesh sizes, and compared with the benchmark solutions of Ghia et al.
4.4.2 The Natural Convection Flow

Another commonly encountered benchmark is one modelling natural convection in a two-dimensional square cavity, with isothermal conducting side walls and adiabatic upper and lower boundaries. The non-dimensional numbers characterising the flow are the Rayleigh and Prandtl numbers,

$$\begin{align*}
Ra &= \frac{g \beta \Delta T \ell^3}{\nu \alpha}, \\
Pr &= \frac{\nu}{\alpha},
\end{align*}$$

where $\nu$ is the kinematic viscosity, $\alpha$ the thermal diffusivity, $\Delta T$ the temperature difference across the cavity, $\beta$ the volumetric thermal expansion coefficient and $g$ the acceleration due to gravity. For low Rayleigh numbers (on the order of $10^3$ to $10^4$) the flow is dominated by viscosity and the first order scheme are adequate, however, for larger Rayleigh numbers ($> 10^6$) the errors in the first order schemes are expected to become apparent.

The earliest analyses of this problem are those of Batchelor[9] who drew upon the little experimental data that was available at that time, and Poots[132] who modelled the flow numerically, using a hand calculated spectral method, with Misses Faithfull and Watkins acting as the computational engine. Two early finite difference models that were run on electronic computers were those of Wilkes and Churchill[180] and de Vahl Davis[29]. Since that date the method has been used as a benchmark for the comparison of natural convection codes[32, 31], a benchmark solution being given in detail by de Vahl Davis in [30, 73], with another solution being provided by Hortmann and Peric[66].

A flow with a Rayleigh number of $10^6$ and a Prandtl number of 0.71 was calculated using the differencing schemes described in Chapter 2, with calculations being made on meshes ranging in size from $19 \times 19$ to $515 \times 515$ cells. The results are compared with the benchmark solution of de Vahl Davis[30], who calculated the flow using a vorticity-streamfunction finite difference method using a central difference approximation on meshes of dimension $21 \times 21, 41 \times 41$ and $81 \times 81$, and used Richardson extrapolation to estimate a grid independent solution.
CHAPTER 4. SOLUTION OF THE NAVIER–STOKES EQUATIONS

Figure 4.11: The streamlines (left) and isotherms (right) for a $Ra = 10^6$ Pr = 0.71 flow calculated on a $131 \times 131$ mesh using the ULTRA differencing scheme. Streamfunction contours are for the range $-22 < \psi < 0$ at 2 intervals, with further contours at $-23$ and $-23.1$. Temperature contours are for the interval $\pm 0.5$ at 0.05 intervals.

Figure 4.12: The streamlines and isotherms for a $Ra = 10^6$ Pr = 0.71 flow calculated on a $35 \times 35$ mesh using the FOU scheme (top) and the ULTRA scheme (bottom). Streamlines are at 2 intervals between $-26 < \psi < 0$. Isotherms are at 0.05 intervals between $\pm 0.5$. 
The streamlines and temperature field for the flow calculated on a $131 \times 131$ mesh using the ULTRA differencing scheme are shown in Figure 4.11, with the $x$ and $y$ components of velocity, the velocity magnitude, and the vorticity being given in Figure 4.14. The flow is rotationally symmetric about the centre of the cavity, with a jet of fluid attached to the walls of the cavity surrounding a comparatively motionless body of fluid at the centre of the cavity. The velocity is at a maximum in the narrow boundary layers adjacent to the left and right walls, where the fluid is heated and cooled by the boundary. These thermally driven boundary layers discharge jets of fluid along the top and bottom walls, which thicken as they travel along the adiabatic walls, shedding their velocity in the process. The vertical boundary layers are driven by the sharp temperature gradients that exist at the left and right walls, and the horizontal jets are comparatively isothermal. The comparatively stationary fluid in the centre of the cavity is stably stratified with a linear temperature gradient existing between the upper and lower jets.

The fine mesh solutions in Figure 4.11 may be compared with the solutions calculated on a $35 \times 35$ mesh using the FOU and ULTRA schemes, given in Figure 4.12. The ULTRA solution is very similar to the converged solution, whilst the errors in the FOU scheme are quite minor. The first order scheme overestimates the strength of the flow across the cavity, underestimates the temperature of the horizontal jets, and doesn’t capture the sharpness of the temperature gradients and streamlines in the top left/bottom right corners of the cavity.

Figure 4.13: The temperature at the top boundary on the vertical centreline of the cavity. For each of the differencing schemes the calculated temperature is plotted as a function of the mesh size $\Delta x$. The calculated temperatures converge to the solution of de Vahl Davis, which was approximated for a mesh of $\Delta x = 0$ using Richardson extrapolation.

The convergence of the temperature at the top boundary on the vertical centreline is shown in Figure 4.13, with the calculated wall temperature being plotted as a function of the grid size. As the mesh is refined the solutions converge to the extrapolated solution of de Vahl Davis. Excluding the coarsest mesh solutions calculated on a $19 \times 19$ grid, the convergence is quite regular and smooth, justifying the use of Richardson extrapolation. It is interesting to note that the SOU scheme is the fastest converging scheme, whilst the FOU scheme underpredicts the temperature, all other schemes overpredicting the temperature. Surprisingly, the first order Power scheme is faster converging than all of the second and third order schemes, with the exception of the SOU method. This is the opposite of the case with the driven cavity benchmark, and was not an expected outcome of the test.

This unexpected accuracy of the Power scheme is thought to arise from the velocity field being largely...
aligned with the mesh along the vertical and horizontal boundaries of the cavity. Since the flow is largely parallel to the mesh axis the error arising from false diffusion in the lower order schemes is minimised.

Figure 4.14: Contours of velocity components in the $x$ and $y$ axis ($u$ top left, $v$ top right), and of the velocity magnitude and vorticity ($|u|$ bottom left, $\zeta$ bottom right), for a $Ra = 10^6$ $Pr = 0.71$ flow calculated on a $131 \times 131$ mesh using the ULTRA differencing scheme. Velocity contours are at 20 intervals between $\pm300$, with extra contours at $\pm5$, $\pm10$ and $\pm30$. Vorticity contours are at $2 \times 10^9$ intervals between $\pm2 \times 10^4$, with extra contours at $10^3$ and $\pm3 \times 10^3$.

Finally the profiles of velocity along the vertical and horizontal centrelines are plotted for the FOU, Power, Central and ULTRA differencing schemes in Figures 4.15 and 4.16. The FOU scheme overpredicts the $u$ velocity on the vertical centreline, but all other schemes quickly converge to the benchmark solution. The same is true of the vertical velocity profiles in the horizontal centreline shown in Figure 4.16.
Chapter 4: Solution of the Navier-Stokes Equations

Figure 4.15: Profiles of the $u$ velocity along the vertical centreline, calculated for a range of mesh sizes and compared with the benchmark solutions of de Vahl Davis. The solutions are calculated using (clockwise from top left), FOU, Power, ULTRA and Central differencing.

Figure 4.16: Profiles of the $v$ velocity along the horizontal centreline, calculated for a range of mesh sizes and compared with the benchmark solutions of de Vahl Davis. The solutions are calculated using (clockwise from top left), FOU, Power, ULTRA and Central differencing.
4.5 Conclusions

The flow of fluid may be modelled with the Navier–Stokes equations, which describe the conservation of mass and momentum in a fluid. The equations are nonlinear and coupled, and some method must be found to solve them for the velocity and pressure fields, giving a solution that conserves mass and momentum.

In this chapter two coupling schemes have been described. It is argued that the first is more efficient for calculating steady flows, whilst the second is more efficient in the solution of transient flow problems. The methods have been implemented in a CFD code, and the steady flow solver has been used to solve two benchmark test problems, both as a test of the coupling scheme, and in addition as a further test of the advection differencing schemes described in Chapter 2.

For both problems the solutions obtained converged to the benchmark as the mesh was refined. This suggests that the discretisation and coupling code is correctly modelling the Navier–Stokes equations and thermal transport equation for steady flows. All differencing schemes seemed to converge to the benchmark solution as the mesh was refined, although the rate of convergence varied depending on the problem and the differencing scheme.

For the Natural Convection benchmark all schemes converged rapidly to the benchmark solution, with the SOU and Power differencing schemes showing the fastest convergence. In contrast, for the Driven Cavity problem the Power, Exponential and FOU schemes were the slowest to converge, the second order schemes converged rapidly, whilst the third order schemes were fastest of all. The difference in the rates of convergence between the two benchmarking problems is thought to be due to the alignment of the flow with the mesh; for the Natural Convection benchmark the flow is mostly parallel to the mesh, and so the false diffusion from the first order schemes is minimised. However, for the Driven Cavity problem much of the flow is at an angle to the mesh axes, and so the false diffusion is more likely to manifest itself. As such the Driven Cavity problem is a much more severe test of the differencing schemes than the Natural Convection benchmark, and it more readily reveals the inadequacies in the first order differencing schemes.