Transport processes – Part 6b

Ron Zevenhoven
Åbo Akademi University
Thermal and Flow Engineering / Värme- och strömningsteknik
tel. 3223 ; ron.zevenhoven@abo.fi

Numerical methods – intro
Chapter 6

Numerical Heat and Fluid Flow

6.3 Convection and diffusion
6.4 Calculation of the velocity field
6.5 Boundary conditions
6.6 Chart diagram of a numerical code for CFD

6.3 Convection and diffusion

In this section we introduce a new mechanism into the general transport equation: convection. At this stage, we will simply assume that the velocity field that governs the convective term is known. In the sections that follow the exact procedure for calculating the velocity field will be derived.

**Heat:** $\Phi = c_p T$, $\Gamma = \lambda$, **mass:** $\Phi = c$, $\Gamma = D$

**6.3.1 Steady one-dimensional (1D) convection and diffusion**

The convection-diffusion differential equation is

$$\frac{d}{dx} (\rho U \Phi) = \frac{d}{dx} \left( \Gamma \frac{\partial \Phi}{\partial x} \right)$$

**Steady:** $\partial../\partial t = 0$ \hspace{1cm} (6.35)

where $u$ is velocity in the $x$-direction and $\Gamma$ is the diffusion coefficient for variable $\Phi$. For example, the continuity equation can be easily obtained by specifying $\Phi = 1$ and $\Gamma = 0$

$$\frac{d}{dx} (\rho U) = 0 \hspace{0.5cm} \text{or} \hspace{0.5cm} \rho U = \text{const.}$$ \hspace{1cm} (6.36)

By integrating Eq. (6.35) over the control volume shown in Figure 6.4, we have

$$(\rho U \Phi)_e - (\rho U \Phi)_w = \left( \Gamma \frac{\partial \Phi}{\partial x} \right)_e - \left( \Gamma \frac{\partial \Phi}{\partial x} \right)_w$$ \hspace{1cm} (6.37)
6.3.2 The central differencing scheme (CDS)

In the previous section it is demonstrated how to represent the $\Gamma d\Phi/dx$ term using linear interpolation. For the convective term, the linear interpolation will result in

$$\Phi_e = \frac{1}{2} (\Phi_E + \Phi_P), \quad \Phi_w = \frac{1}{2} (\Phi_P + \Phi_W)$$  \hspace{1cm} (6.38)

The factor $1/2$ originates from the simple assumption that a uniform mesh is used. Now, the integrated discretised equation can be written as

$$\frac{1}{2} \left( \rho U \right)_e (\Phi_E + \Phi_P) - \frac{1}{2} \left( \rho U \right)_w (\Phi_P + \Phi_W) = \frac{\Gamma_c (\Phi_E - \Phi_P)}{\delta x_e} - \frac{\Gamma_w (\Phi_P - \Phi_W)}{\delta x_w}$$  \hspace{1cm} (6.39)

Introducing substitutions:

\[ F = \rho U, \quad D = \frac{\Gamma}{\delta x} \]

\[ F/D \sim \text{Péclet, Pe} \]

the discretised equation emerges in a compact form as

$$a_p \Phi_p = a_E \Phi_E + a_w \Phi_W$$  \hspace{1cm} (6.40)

where

$$a_E = D_e - \frac{F_e}{2}, \quad a_w = D_w + \frac{F_w}{2},$$

$$a_p = D_e + \frac{F_e}{2} + D_w - \frac{F_w}{2} = a_E + a_w + (F_e - F_w)$$  \hspace{1cm} (6.41)

Here $F$ represents the intensity of convection while $D$ represents the intensity of diffusion. The diffusive term is always positive, while the convective term can take either positive or negative values - depending on the flow direction in respect to the coordinate direction. In order to satisfy the continuity condition, it follows that $F_e = F_w$ and the central coefficient can be written as $a_p = a_w + a_E$. Note that for estimation of the

- **Figure 6.8** The second order central differencing scheme (CDS).

\[ a_p \Phi_p = a_E \Phi_E + a_w \Phi_W \]  \hspace{1cm} (6.42)
convective terms, a simple linear interpolation is used, known as the central differencing scheme. The correctness of the linear interpolation in evaluating the convective term can be easily validated by assuming few typical flow situations where either diffusion or convection dominates. Let us assume a case where convection dominates over diffusion, e.g. \( D_c = D_w = 1 \) and \( F_e = F_w = 4 \). If values of \( \Phi_E \) and \( \Phi_W \) are given, then we can calculate values in the central point \( P \) from discretised equation, Eq. (6.41). Let assume two sets of values:

- \( \Phi_E = 200, \Phi_W = 100 \), then since \( a_E = -1, a_W = 3 \) \( \rightarrow \Phi_P = 50 \)
- \( \Phi_E = 100, \Phi_W = 200 \), then since \( a_E = -1, a_W = 3 \) \( \rightarrow \Phi_P = 250 \)

It can be concluded that when \( |F| \) exceeds \( 2D \) unrealistic values appear. How to remove these unrealistic values in such situations? The answer is in introducing an upwind differencing scheme as demonstrated below.

6.3.3 The upwind differencing scheme (UDS)

Since the origin of unrealistic (unbounded) values at the cell face of the control volume is in its linear interpolation, we can use the upwind differencing scheme, which simply projects the upwind central values onto cell-faces - depending on the flow direction, Figure 6.9:

\[
\Phi_e = \Phi_p, \text{ if } F_e > 0 \quad \text{and} \quad \Phi_e = \Phi_E, \text{ if } F_e < 0 \quad (6.43)
\]

Instead of checking the flow direction at each cell face, it is more practical to introduce a more general mathematical operator \( \| A, B \| \) to denote the greater of \( A \) and \( B \). Then, the upwind values can be expressed as

\[
F_e \Phi_e = \Phi_P \| F_e, 0 \| - \Phi_E \| -F_e, 0 \| = \max(A, B) \quad (6.44)
\]

and the discretised equation can be written in a simple form

\[
\begin{align*}
 a_e \Phi_P &= a_E \Phi_E + a_W \Phi_W \\
 a_E &= D_c + \| -F_e, 0 \|, a_W = D_w + \| F_w, 0 \| \\
 a_p &= a_E + a_W + (F_e - F_w) 
\end{align*} \quad (6.45)
\]

Figure 6.9 The first-order upwind differencing scheme (UDS).
It is obvious that the upwind scheme will always produce positive discretised convective coefficients and consequently, all cell-face values will be unconditionally bounded.

### 6.3.4 The hybrid (flux-blending) scheme

The flux-blending discretisation scheme simply combines the central-difference with the upwind schemes. A simple blending parameter is introduced ($\gamma$) so we can calculate the values at the cell-face ($f$) as

$$\Phi_f = \gamma \Phi_f^{CDS} + (1 - \gamma) \Phi_f^{UDS}$$  \hspace{1cm} (6.46)

This scheme gives usually very good results and converges fast for intermediate values of the Péclet number, $Pe = \rho U L / \Gamma$. It can be shown that this number defines the ratio between the strengths of convection and diffusion. Because of its simplicity, this scheme is quite popular and can be found in many CFD codes. Instead of using constant values of the blending parameter ($\gamma$) the local $Pe$ number can be used as a controlling parameter such that for $|Pe| > 2$, the central difference scheme will be replaced by its upwind counterpart.

*Figure 6.9 The first-order upwind differencing scheme (UDS)*.

### 6.3.5 The exact solution

If $\Gamma$ is assumed constant (in addition to $\rho$ and $U$), the one-dimensional convection-diffusion equation

$$\frac{d}{dx}(\rho U \Phi) = \frac{d}{dx} \left( \Gamma \frac{\partial \Phi}{\partial x} \right)$$  \hspace{1cm} (6.47)

can be solved exactly. For a domain $0 \leq x \leq L$, with boundary conditions

$$x = 0, \Phi = \Phi_0, \text{ and } x = L, \Phi = \Phi_L$$  \hspace{1cm} (6.48)

the exact solution is

$$\frac{\Phi - \Phi_0}{\Phi_L - \Phi_0} = \frac{\exp \left( \frac{Pe x}{L} \right) - 1}{\exp \left( Pe \right) - 1}$$  \hspace{1cm} (6.49)
\[
\frac{\Phi - \Phi_0}{\Phi_L - \Phi_0} = \frac{\exp\left(\frac{Pe \cdot x}{L}\right) - 1}{\exp(Pe) - 1}
\]  

(6.49)

The nature of the exact solutions for the one-dimensional convection-diffusion problem can be represented by Figure 6.10. It can be seen that when \(Pe = 0\), the pure diffusion is obtained. For \(Pe > 0\) the values of \(\Phi\) are more influenced by the upstream value \(\Phi_0\). For \(Pe < 0\) the opposite is true. It is now easy to see why our preliminary linearisation is based on this exact solution.

\[\text{Figure 6.10 The nature of the exact solution for the one-dimensional convection-diffusion problem.}\]

For \(Pe < 0\) the opposite is true. It is now easy to see why our preliminary linearisation (central-difference scheme, CDS) failed to give a satisfactory formulation at the cell-faces - the \(\Phi \sim x\) profile is far from being linear except for small values of \(|Pe|\). For large \(|Pe|\) the value of \(\Phi\) at \(x = L/2\) (the grid cell interface) is precisely the assumption made in the upwind scheme, but it is used for all values of \(|Pe|\), not just for large values. When \(|Pe|\) is large, \(d\Phi/dx\) is nearly zero at \(x = L/2\), i.e. the diffusion is almost absent. The upwind scheme always calculates the diffusion term from a linear \(\Phi \sim x\) profile and overestimates diffusion at large values of \(|Pe|\). In the next section we derive a discretisation scheme based on this exact solution.

### 6.3.6 The exponential scheme

Let us first introduce a total flux \(J\) as the sum of convective and diffusive fluxes:

\[
J = \rho U \Phi - \Gamma \frac{d\Phi}{dx}
\]

(6.50)

Now, we have

\[
\frac{d}{dx} \left( \rho U \Phi \right) = \frac{d}{dx} \left( \Gamma \frac{d\Phi}{dx} \right) \to \frac{dJ}{dx} = 0
\]

(6.51)
and after integration over the control volume

\[ J_e - J_w = 0 \] (6.52)

In order to calculate the cell-face values we will use the above derived exact solution of the one-dimensional convection-diffusion equation:

\[ J_e = F_e \left( \Phi_p + \frac{\Phi_p - \Phi_E}{\exp(Pe_e) - 1} \right), \quad Pe_c = \frac{(\rho U) c \delta x_c}{\Gamma_e} = \frac{F_e}{D_e} \] (6.53)

After substitutions

\[ F_e \left( \Phi_p + \frac{\Phi_p - \Phi_E}{\exp(Pe_e) - 1} \right) - F_w \left( \Phi_W + \frac{\Phi_W - \Phi_P}{\exp(Pe_w) - 1} \right) = 0 \] (6.54)

which can be again expressed in a compact form

\[ a_p \Phi_p = a_e \Phi_E + a_w \Phi_W \] (6.55)

where

\[ a_e = \frac{F_e}{\exp(F_e/D_e) - 1}, \quad a_w = \frac{F_w}{\exp(F_w/D_w) - 1}, \quad a_p = a_e + a_w + (F_e - F_w) \] (6.56)

Thus, when used for steady 1D problems, the exponential scheme produces the exact solution for any value of the Péclet number and for any number of grid points. The problem is that the exact solution is valid only for one-dimensional problems - this scheme is not exact for two- or three-dimensional situations, non-zero sources, etc.

6.3.7 The generalised formulation of differencing schemes

![Figure 6.11 Prediction of $\Phi_p$ by various schemes for a range of Péclet numbers.](VST rz18)
All the above mentioned differencing schemes (upwind, central, exact) can be represented by the following general convection-diffusion formulation:

\[
\begin{align*}
    a_p \Phi_p &= a_E \Phi_E + a_W \Phi_W, \\
    a_p &= a_E + a_W + (F_e - F_w), \\
    a_E &= D_e A(|Pe_e|) + || -F_e, 0 ||, \\
    a_W &= D_w A(|Pe_w|) + || F_w, 0 ||
\end{align*}
\]  

\( (6.57) \)

| Scheme            | Expression for \( A(|Pe|) \) |
|-------------------|-------------------------------|
| Upwind            | 1                             |
| Central difference| \( 1 - 0.5|Pe| \)            |
| Exponential (exact)| \( |Pe|/(\exp(|Pe|) - 1) \) |
| Hybrid            | \( || 0, 1 - 0.5|Pe| || \) |
| Power law         | \( || 0, (1 - 0.1|Pe|)^5 || \) |

Table 6.1 *The function \( A(|Pe|) \) for different schemes*

\( |..| = \text{absolute value} \)

\[ A(|Pe|) \]

\[ \text{Upwind} \]

\[ \text{Exponential (exact)} \]

\[ \text{Power law} \]

\[ \text{Hybrid} \]

\[ \text{Central difference} \]

\[ |Pe| \]

*Figure 6.12 The function \( A(|Pe|) \) for different schemes.*
The various schemes can now be extracted by different formulations of the function $A(|Pe|)$ given in Table 6.3 as shown in Figure 6.11. To demonstrate the typical behaviour of different differencing schemes we examine the values of $\Phi_p$ predicted for various values of $\Phi_E$ and $\Phi_W$. Without any loss of generality one can assume the values $\Phi_E = 1$ and $\Phi_W = 0$. In addition, on a uniform mesh both distances $\delta x_c$ and $\delta x_w$ are equal and $\Phi_p$ will be a function of the Péclet number ($Pe = \rho u \delta x / \Gamma$). Here we introduce two additional schemes, i.e. hybrid and power-law as a hybrid differencing scheme that combines the upwind and the central-differencing schemes. It is noted that all schemes (except the central-difference) give a physically realistic solution. For the parameters outside $-2 < Pe < 2$ range, the central-difference scheme produces values that are outside the $\Phi_p (0 - 1)$ range imposed by the boundary values. In order to eliminate these unbounded values, it is necessary to refine the numerical mesh until the $|Pe| < 2$ and then the central-difference scheme will produce bounded solutions.

**Diffusion + convection – numerical**

**Peclet number :** $P \equiv \frac{\rho u L}{\Gamma}$ \hspace{1cm} $L = \delta x \Rightarrow P = \frac{F}{D}$

Setting $\Phi_W = 0.9$, $\Phi_E = 0.5$, $D = 1$ and varying $F$:

<table>
<thead>
<tr>
<th>$F$</th>
<th>$D$</th>
<th>$P$</th>
<th>$\Phi_W$</th>
<th>$\Phi_E$</th>
<th>$\Phi_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.9</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.9</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0.9</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>0.9</td>
<td>0.5</td>
<td>1.1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>0.9</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>-2</td>
<td>1</td>
<td>-2</td>
<td>0.9</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>-3</td>
<td>1</td>
<td>-3</td>
<td>0.9</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>-4</td>
<td>1</td>
<td>-4</td>
<td>0.9</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$F$</th>
<th>$D$</th>
<th>$P$</th>
<th>$\Phi_W$</th>
<th>$\Phi_E$</th>
<th>$\Phi_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.9</td>
<td>0.5</td>
<td>0.7</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.9</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.9</td>
<td>0.5</td>
<td>0.9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>0.9</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>4</td>
<td>0.9</td>
<td>0.5</td>
<td>1.1</td>
</tr>
<tr>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>0.9</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>-2</td>
<td>1</td>
<td>-2</td>
<td>0.9</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>-3</td>
<td>1</td>
<td>-3</td>
<td>0.9</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>-4</td>
<td>1</td>
<td>-4</td>
<td>0.9</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

**Central-difference scheme**

**Upwind scheme**

Taken from J. Brännbacka “Introduction to CFD” course material ÅA (v. 2006)
6.3.8 Numerical ("false") diffusion

It is useful to compare the coefficients \(a_E, a_W\) for the central-difference and upwind schemes given in Table 6.2. One can notice that the upwind scheme is equivalent to replacing \(\Gamma\) in the central-difference scheme with \(\Gamma + \rho U \delta x / 2\), which indicates that the upwind scheme tends to increase the true diffusion coefficient \(\Gamma\) by a false diffusion coefficient \(\rho U \delta x / 2\).

In order to give a more physical interpretation of the numerical ("false") diffusion we consider what will happen at the contact between two parallel fluid streams of equal velocity but with different temperatures. If \(\Gamma = 0\), no mixing layer will form and the temperature discontinuity will persist in the streamwise direction. If \(\Gamma \neq 0\), a mixing layer will form and the temperature will gradually change. The mixing layer will grow in the streamwise direction. The most illustrative situation is to specify the real diffusion to zero. If a smeared temperature profile appears, this is the indication of numerical diffusion.

Let us imagine the situation where the flow is aligned with the \(x\)-direction of the underlying grid (Figure 6.13). The left side boundary has a prescribed sharp temperature profile. Since \(\Gamma = 0\) and there is no flow in the \(y\)-direction, the coefficients \(a_N\) and \(a_S\) will be zero. The coefficient \(a_E\) of the downstream neighbour will also be zero and consequently \(a_W = a_W\). It results in \(\Phi_p = \Phi_w\), i.e. the inlet value along the grid line will be kept constant. It means that no diffusion is taking place (Figure 6.13-above). The situation changes significantly when the identical problem is solved on a grid in which the grid lines are inclined at 45° to the flow direction. If the uniform mesh is applied, i.e. \(\Delta x = \Delta y\), the velocities in the \(x\) and \(y\) directions are equal.

\[
\begin{align*}
\text{Pe} &= \infty \\
\Rightarrow \quad x \\
\Phi_p &= \Phi_w \\
\end{align*}
\]

\[
\begin{align*}
\text{Pe} &= \infty \\
\Rightarrow \quad x \\
\Phi_p &= \frac{1}{2}(\Phi_w + \Phi_s)
\end{align*}
\]

Figure 6.13 Appearance of the false diffusion for different grid distributions.
Let us now analyse the identical problem, this time on a numerical mesh that is rotated by 45°. Under the assumption that the mesh is uniform in both coordinate direction, the upstream values \( a_W \) and \( a_S \) are identical, while the downstream values \( a_E \) and \( a_N \) are zero. Then, the value at the central point \( P \) can be evaluated from

\[
\Phi_P = 0.5\Phi_W + 0.5\Phi_S
\]

(6.58)

If we assume that the temperature profiles of incoming jets have values of 100 and 0, respectively, the resulting values of temperatures at the particular mesh locations can be calculated (Figure 6.13—below). If this is a situation without diffusion, then all values above and below the main diagonal should keep their initial values, i.e. 100 above the diagonal and 0 below that line. It is seen, however, that this is not the case and that the temperature does not exhibit sharp profiles as expected. Obviously, despite specifically setting the real diffusion to zero in the discretised temperature equation, a sort of diffusion process takes place. This numerical contamination is called false or numerical diffusion. From the previous analysis it can be concluded that numerical diffusion takes place when the numerical mesh does not coincide with the flow direction. A value of the diffusion coefficient of the numerical contamination can be expressed as

\[
\Gamma_{false} = \frac{\rho U \Delta x \Delta y \sin(2\alpha)}{4(\Delta y \sin^3(\alpha) + \Delta x \cos^3(\alpha))}
\]

(6.59)

Here, \( U \) is the velocity and \( \alpha \) is the angle between the flow direction and the horizontal coordinate. The numerical diffusion will be zero when the flow direction and the grid coordinate direction are parallel, i.e. \( \alpha = 0^\circ \), while it will reach its maximum when \( \alpha = 45^\circ \).

Because the numerical diffusion coefficient \( \Gamma_{false} \) is proportional to the size of the grid cell (\( \Delta x \)), the numerical diffusion can be reduced by mesh refinements and by aligning as much as possible the numerical mesh with the flow direction. A good estimate of the real importance of numerical diffusion can be made by comparing it with the real diffusion. If the numerical diffusion is still not-negligible compared with real diffusion, further grid refinement should be performed.
6.3.9 UDS versus CDS for one-dimensional convection-diffusion

In order to illustrate further the impacts of the upwind differencing scheme, we consider a simple steady one-dimensional convection-diffusion problem:

$$\frac{\partial (\rho c_p U T)}{\partial x} = \frac{\partial}{\partial x} \left( \lambda \frac{\partial T}{\partial x} \right)$$  \hspace{1cm} (6.60)

Assuming a uniform velocity \( (U = \text{const}) \), homogeneous fluid properties, a uniform mesh \((Ax = \text{const.})\) and constant boundary conditions, we can express the simple discretised forms of the convective terms \((C)\) - obtained with central- and upwind differencing schemes as follows:

\[
C_{CDS} = -\rho c_p U \frac{1}{2} \left( \frac{T_{i-1} - T_i}{Ax} + \frac{T_i - T_{i+1}}{Ax} \right) = -\frac{\rho c_p U}{2Ax} (T_{i+1} - T_{i-1}) \tag{6.61}
\]

\[
C_{UDS} = -\frac{\rho c_p U}{Ax} (T_i - T_{i+1}) \tag{6.62}
\]

In order to visualize the structure of these discretised forms, we can write the convection terms in form of matrices (here for a numerical mesh consisting of 5 control volumes):

\[
C_{CDS} = \frac{-\rho c_p U}{Ax} \begin{bmatrix}
0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 1 \\
0 & 0 & 0 & -1 & 1
\end{bmatrix}, \tag{6.63}
\]

\[
C_{UDS} = \frac{-\rho c_p U}{Ax} \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & -1 & 1
\end{bmatrix}, \tag{6.64}
\]

Note that the diagonal terms always represent the current location of the control volume \((i)\) and the upper and lower diagonal terms are its right \((i+1)\) and left \((i-1)\) neighbours, respectively. The diffusion term can be similarly represented. Since the diffusion term is usually always represented by central differences, it will have the following discretised structure:

\[
D_{CDS} = \frac{\lambda}{Ax^2} \left( \frac{T_{i+1} - T_i}{Ax} - \frac{T_i - T_{i-1}}{Ax} \right) = \frac{\lambda}{Ax^2} (T_{i+1} - 2T_i + T_{i-1}) \tag{6.65}
\]

Again, it can be visualised in a matrix form

\[
D_{CDS} = \frac{\lambda}{Ax^2} \begin{bmatrix}
-2 & 1 & 0 & 0 & 0 \\
1 & -2 & 1 & 0 & 0 \\
0 & 1 & -2 & 1 & 0 \\
0 & 0 & 1 & -2 & 1 \\
0 & 0 & 0 & 1 & -2
\end{bmatrix} \tag{6.66}
\]

Next we solve numerically this one-dimensional convection-diffusion equation. The left and right domain boundaries are kept at constant temperatures, \(T_1=10\), \(T_N=-10\). Different mesh resolutions \((N)\) are investigated. Comparative assessments of UDS and CDS differencing schemes for two different values of \(Pe=5,50\) are shown in Figures 6.14 and 6.15.
As seen, the UDS always produces smooth profiles - independently of the number of control volumes used or the value of the Pe number. The CDS produces extreme spikes for $Pe=50$ for coarse mesh resolutions indicating purely numerical oscillations, Figure 6.15. This clearly indicates that a special care should be taken when choosing the proper differencing schemes for convective terms. The UDS will always produce bounded (non-oscillatory) solutions, but convergence towards the exact solutions is very slow. Note that the exact solutions were reached only after a fine numerical mesh is employed, i.e. $N=68$, 180 control volumes for $Pe=5, 50$, respectively. On the other hand, the CDS approaches the exact solutions rapidly - only $N=8, 40$ control volumes are needed to reach the exact solutions for $Pe=5, 50$, respectively. The problem is that for low numerical resolutions, this scheme introduces non-physical oscillatory behaviour.

6.3.10 **Higher order differencing schemes**

In order to take advantage of the numerical stability of the upwind scheme, but to increase its numerical accuracy, one can apply the quadratic interpolation differencing scheme (QUICK). Instead of considering only the immediate neighbours around the central point (P), additional computational nodes are needed (Figure 6.16). A quadratic interpolation through three consecutive grid points is used - depending on the flow direction.

The cell face value (for a uniform grid distribution) can be calculated as

$$
\Phi_f = \Phi_c + \frac{3}{8} \Phi_d - \frac{2}{8} \Phi_c - \frac{1}{8} \Phi_u
$$

(6.67)

where $C, D, U$ are upwind, downstream and upstream values of dependent variable, respectively. For example, for a flow from left to right (both $U_w$ and $U_e>0$) we have

$$
\Phi_w = \frac{6}{8} \Phi_w + \frac{3}{8} \Phi_p - \frac{1}{8} \Phi_{WW}, \quad \Phi_e = \frac{6}{8} \Phi_p + \frac{3}{8} \Phi_e - \frac{1}{8} \Phi_w
$$

(6.68)

For opposite flow direction (both $U_w$ and $U_e<0$) the following interpolation is used:

$$
\Phi_w = \frac{6}{8} \Phi_p + \frac{3}{8} \Phi_w - \frac{1}{8} \Phi_E, \quad \Phi_e = \frac{6}{8} \Phi_E + \frac{3}{8} \Phi_p - \frac{1}{8} \Phi_{EE}
$$

(6.69)
Figure 6.13: See the caption in the previous figure. Here, we have $P_e = 5, 10, 15, 30, 40$, respectively. Note that the UDS scheme reaches the exact solution for $M = 30$. If not, please contact your local address.

Figure 6.14: Comparative assessment of UDS (right) and CDS (left) differencing schemes for the one-dimensional convection-diffusion problem. The temperature boundary conditions at the left and right walls are 10 and -10, respectively. $P_e = 5, M = 1, 4, 8, 16$ from top to bottom, respectively. Note that the UDS scheme reached the exact solution for $M = 64$. The exact solution is given by solid line $T(x) = T_1 + (T_2 - T_1) \left(\exp \left(\frac{x}{L}\right) - 1 \right) \exp \left(\frac{P_e}{L}\right) - 1$. 
In general, higher upwind differencing schemes can be expressed in a unique simple form:

$$
\Phi_f = \Phi_C + \frac{1}{4} \left[ (1+\beta) (\Phi_D - \Phi_C) + (1-\beta) (\Phi_C - \Phi_U) \right]_{\text{upwind correction}} \tag{6.70}
$$

Now by specifying different values of $\beta$ many of the previous differencing schemes can be recovered and new can be created. The following Table shows several possible variants of differencing schemes:

### 6.3.11 Total variation diminishing schemes (TVD)

In the previous section it was demonstrated that the higher order differencing schemes are simply based on higher order polynomial interpolations. These differencing schemes easily produce local extreme peaks that propagate through the entire discretised domain causing oscillatory behaviour of an iterative procedure. To reduce such oscillatory behaviour or even to avoid divergence of the entire iterative procedure, certain limits can be locally imposed in order to prevent these extreme events. There are two criteria that we want to impose on such controlled discretisation schemes. The first one is to prevent extreme values to be initially created, i.e. the value at the central point (C) should be always bounded by its neighbouring values (upstream - U and downstream -D neighbours).

![Figure 6.16 The QUICK differencing scheme.](image)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>$\beta$</th>
<th>Correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Differencing Scheme (CDS)</td>
<td>1</td>
<td>$\frac{1}{2} \Phi_D - \frac{1}{2} \Phi_C$</td>
</tr>
<tr>
<td>Linear Upwind Differencing Scheme (LUDS)</td>
<td>-1</td>
<td>$\frac{1}{2} \Phi_C - \frac{1}{2} \Phi_U$</td>
</tr>
<tr>
<td>Quadratic Upwind Differencing Scheme (QUICK)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{3}{8} \Phi_D - \frac{2}{8} \Phi_C - \frac{1}{8} \Phi_U$</td>
</tr>
<tr>
<td>Cubic Upwind Differencing Scheme (CUI)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6} \Phi_D - \frac{1}{6} \Phi_C - \frac{1}{6} \Phi_U$</td>
</tr>
</tbody>
</table>

Table 6.3 General representation of higher-order differencing schemes.
The second one includes the condition that if the local extreme values have already been created, they should not be allowed to propagate into the rest of the discretised domain. These conditions can be simply interpreted as a limit in total variation (change) of a dependent variable over a specified interval. This can be simply achieved by controlling the monotonicity of distributions. The total variation (TV) of a general variable $\Phi$ is defined as

$$TV(\Phi) = \sum_{i=-\infty}^{\infty} |\Phi_{i+1} - \Phi_i|$$

(6.71)

which can be expressed over a discretised interval $(1, ..., N)$ as

$$TV(\Phi) = |\Phi_2 - \Phi_1| + ... + |\Phi_N - \Phi_{N-1}|$$

(6.72)

In order to impose monotonicity (to prevent oscillatory behaviour) the following condition must be satisfied:

$$TV(\Phi^V) \leq TV(\Phi^{N-1})$$

(6.73)

The character of such a limiter is illustrated in Figure 6.17 where an initial distribution of variable $\Phi$ with overshoots and undershoots (solid line) is corrected by local limiters (dashed lines) in order to provide the required monotonicity of distributions. We will rewrite the general expression for higher order differencing scheme (with addition of a limiter, $\psi(r)$) as

$$\Phi_j = \Phi_c + \frac{1}{4} \left[ (1 + \beta) \psi(r) (\Phi_D - \Phi_C) + (1 - \beta) \psi \left( \frac{1}{r} \right) (\Phi_C - \Phi_U) \right]$$

(6.74)

\[\Phi_1 \Phi_2 \Phi_3 \Phi_4 \Phi_5 \]

\[\Phi  \rightarrow \text{Limiter} \rightarrow \text{Limit} \]

\[\Phi_0 = \Phi_c, \Phi_1 = \Phi_U, \Phi_2 = \Phi_D, \Phi_3 = \Phi_c, \Phi_4 = \Phi_U \]

\[\Phi \rightarrow \text{Limit} \rightarrow \text{Limitor} \]

\[x \quad W \quad P \quad E \]

Figure 6.17 Typical grid-cluster for a 2D convection-diffusion problem.

In order to provide the conditions for the total variation diminishing (TVD), the limiter should be inside the specific area, as shown in Figure 6.18. By introducing nondimensional variables, $\hat{\Phi}$ and $r$ defined by

$$\hat{\Phi} = \frac{\Phi - \Phi_U}{\Phi_D - \Phi_U}, \quad r = \frac{\Phi_C - \Phi_U}{\Phi_D - \Phi_C}$$

(6.75)

the value at the cell-face can be calculated as

$$\hat{\Phi}_f = \hat{\Phi}_C + \frac{1}{4} \hat{\Phi}_C \left[ (1 + \beta) \psi(r) \frac{1}{r} + (1 - \beta) \psi \left( \frac{1}{r} \right) \right]$$

(6.76)

This expression can be further simplified when symmetry properties of the limiter function are imposed; then final form can be written as

$$\psi(r) = r \psi \left( \frac{1}{r} \right), \quad \Phi_f = \Phi_c + \frac{\psi(r)}{2} (1 - \Phi_c)$$

(6.77)
This form can be regarded as a general formulation of the TVD scheme from which a number of specific scheme known in the literature can be extracted depending on the formulation of the limiter. To illustrate the behaviour of the nondimensional variable $\Phi$ as a function of $r$ two cases are explored - with positive and negative values of $r$ as shown in Figure 6.19. These total variation constraints can be written in terms of nondimensional variables as

$$
\text{For } 0 < \Phi_C < 1 \rightarrow \Phi_f \leq 2\Phi_C, \quad \Phi_f \geq \Phi_C, \quad \Phi_f \leq 1
$$

$$
\text{For } \Phi_C \leq 0 \text{ or } \Phi_C \geq 1 \rightarrow \Phi_f = \Phi_C
$$

or in general form as

$$
\psi(r) = \max(0, \min(2r, \beta_1 + \beta_2, r, r + \beta_2, \beta_3)), \quad \beta_1 + \beta_2 = 1
$$

Many different classes of limiters can be applied to ensure that the solution remains inside the stability region (dashed region in Figure 6.18). For example, the most popular bounded versions of the QUICK scheme can be obtained by specifying the following values of parameters:

- SMART (bounded QUICK): $\beta_1 = 0.75, \beta_2 = 0.25, \beta_1 + \beta_2 r = 0, \beta_3 = 4$
- MUSCL (van Leer): $\beta_1 = \beta_2 = 0.5, \beta_1 r + \beta_2 = 0, \beta_3 = 2$
- UMIST (bounded QUICK): $\beta_1 = 0.25, \beta_2 = 0.75, \beta_3 = 2$

The graphical representations of these different schemes in $\psi(r) - r$ diagram are show in Figure 6.20.
Figure 6.19  The nondimensional distribution of $\Phi$ for positive and negative values of $r$.

Figure 6.20  The flux-limiter diagram for different schemes: first-order upwind, UDS - $\psi(r) = 0$; second-order upwind, LUDS - $\psi(r) = 1$; second-order central differencing, CDS - $\psi(r) = r$; second-order bounded QUICK, MUSCL.
6.3.12 Unsteady two-dimensional (2D) convection and diffusion

The general differential equation describing conservation of the variable $\Phi$

$$\frac{\partial}{\partial t} (\rho \Phi) + \frac{\partial}{\partial x} (\rho U \Phi) = \frac{\partial}{\partial x} \left( \Gamma \frac{\partial \Phi}{\partial x} \right) + S$$

(6.80)

can be recast also as

$$\frac{\partial}{\partial t} (\rho \Phi) + \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y} = S$$

(6.81)

where $J_x$ and $J_y$ are the total (convection and diffusion) fluxes defined by

$$J_x = \rho U \Phi - \Gamma \frac{\partial \Phi}{\partial x}, \quad J_y = \rho V \Phi - \Gamma \frac{\partial \Phi}{\partial y}$$

(6.82)

where $u$ and $v$ are the velocity components in the $x$ and $y$ directions, respectively. Integration of the above equation over the control volume shown in Figure 6.21 results in

$$\frac{(\rho_p \Phi_p - \rho_b \Phi_b) \Delta x \Delta y}{\Delta t} + J_x - J_w + J_n - J_s = (S_C + S_p \Phi_p) \Delta x \Delta y$$

(6.83)

where a linearisation of the source term is applied. The $\rho_b$ and $\Phi_b$ are the values from the previous time step. After volume integration, the continuity equation can be written as

$$\frac{(\rho_p - \rho_b) \Delta x \Delta y}{\Delta t} + F_e - F_w + F_n - F_s = 0$$

(6.84)

where $F_e, F_w, F_n, F_s$ are the mass flow rates through the cell faces.

$$F_e = (\rho U)_e \Delta y, \quad F_w = (\rho U)_w \Delta y, \quad F_n = (\rho V)_n \Delta x, \quad F_s = (\rho V)_s \Delta x$$

(6.85)


Multiplying the discretised continuity equation by $\Phi_p$ and then by subtracting it from the discretised conservation equation for $\Phi$

$$\frac{(\Phi_p - \Phi_b) \rho_b \Delta x \Delta y}{\Delta t} + (J_x - F_e \Phi_p) - (J_w - F_w \Phi_p) + \quad$$

$$\quad (J_n - F_n \Phi_p) - (J_s - F_s \Phi_p) = (S_C + S_p \Phi_p) \Delta x \Delta y$$

(6.86)


Figure 6.21 Typical grid-cluster for the 2D convection-diffusion problem.
Replacing

\[ J_e - F_e \Phi_p = a_E (\Phi_p - \Phi_E), J_w - F_w \Phi_p = a_w (\Phi_w - \Phi_s) \]  

(6.87)

where

\[ a_E = D_e A (|P_e|) + || -F_e,0 ||, \quad a_w = D_w A (|P_e|) + || F_w,0 || \]  

(6.88)

(where \( D_e \) and \( D_w \) as well as \( F_e \) and \( F_w \) contain the area \( \Delta y \) of the faces \( e \) and \( w \)) and following the identical steps for the vertical direction, \( J_n - F_n \Phi_p \) and \( J_s - F_s \Phi_p \), the final form of the discretised two-dimensional conservation equation can be written as

\[ a_p \Phi_p = a_E \Phi_E + a_w \Phi_w + a_N \Phi_N + a_s \Phi_s + b \]  

(6.89)

where

\[ a_E = D_e A (|P_e|) + || -F_e,0 ||, \quad a_w = D_w A (|P_e|) + || F_w,0 || \]
\[ a_N = D_n A (|P_n|) + || -F_n,0 ||, \quad a_s = D_s A (|P_s|) + || F_s,0 || \]

\[ \begin{aligned}
\delta E_p^0 &= \frac{\rho_p \Delta x \Delta y}{\Delta M}, & b &= S_c \Delta x \Delta y + \delta E_p^0 \Phi_p^0 \\
\delta a_p &= \delta a_E + \delta a_N + \delta a_s + \delta a_p - S_p \Delta x \Delta y 
\end{aligned} \]  

(6.90)

The mass flow rates through the faces of the control volume are defined as \( F_e = (\rho U) e \Delta y \) and the diffusive counterparts by

\[ D_e = \frac{\Gamma_e \Delta y}{\delta x_e}, D_w = \frac{\Gamma_w \Delta y}{\delta x_w}, D_n = \frac{\Gamma_n \Delta x}{\delta x_n}, D_s = \frac{\Gamma_s \Delta x}{\delta x_s} \]  

(6.91)

where the Péclet numbers are

\[ Pe_e = \frac{F_e}{D_e}, Pe_w = \frac{F_w}{D_w}, Pe_n = \frac{F_n}{D_n}, Pe_s = \frac{F_s}{D_s} \]  

(6.92)

Note the physical meaning of various coefficients of the discretised equation: the neighbour coefficients \( a_E, a_w, a_N, a_s \) represent the convection and diffusion influence at faces of CV in terms of flow rate (\( F \)) and diffusivity (\( D \)).
6.3.13 Unsteady three-dimensional (3D) convection and diffusion

The discretised conservation equation in three dimensions can be easily derived from its two-dimensional form by adding two additional nodes in the z-direction: T and B - representing top and bottom neighbours, respectively.

\[ a_p \Phi_p = a_E \Phi_E + a_W \Phi_W + a_N \Phi_N + a_S \Phi_S + a_T \Phi_T + a_B \Phi_B + b \]  \hspace{1cm} (6.93)

where

\[ a_E = D_E A(|P_E|) + \parallel F_e, 0 \parallel, \quad a_W = D_W A(|P_W|) + \parallel F_w, 0 \parallel, \]
\[ a_N = D_N A(|P_N|) + \parallel F_n, 0 \parallel, \quad a_S = D_S A(|P_S|) + \parallel F_s, 0 \parallel, \]
\[ a_T = D_T A(|P_T|) + \parallel F_t, 0 \parallel, \quad a_B = D_B A(|P_B|) + \parallel F_b, 0 \parallel, \]

\[ \Delta p = \frac{\rho_p \Delta x \Delta y \Delta z}{\Delta t}, \quad b = S_C \Delta x \Delta y \Delta z + \Delta p \Phi_p^0, \]

and the convective and diffusive discretised coefficients are defined as:

\[ F_e = (pU)_e \Delta y \Delta z, D_e = \frac{\Gamma_e \Delta y \Delta z}{\Delta x} \]
\[ F_w = (pU)_w \Delta y \Delta z, D_w = \frac{\Gamma_w \Delta y \Delta z}{\Delta x} \]
\[ F_n = (pV)_n \Delta x \Delta z, D_n = \frac{\Gamma_n \Delta x \Delta z}{\Delta y} \]
\[ F_s = (pV)_s \Delta x \Delta z, D_s = \frac{\Gamma_s \Delta x \Delta z}{\Delta y} \]
\[ F_t = (pW)_t \Delta x \Delta y, D_t = \frac{\Gamma_t \Delta x \Delta y}{\Delta z} \]
\[ F_b = (pW)_b \Delta x \Delta y, D_b = \frac{\Gamma_b \Delta x \Delta y}{\Delta z} \]

6.4 Calculation of the velocity field

In the previous sections we assumed that the velocity field had already been defined. Of course, in real situations the velocity field is usually not known and it should be calculated. As shown in Section 1.2.2, the velocity field is described by equations of conservation of mass and momentum, Eq. (1.17), which for incompressible fluid and in the absence of any body force can be written as

\[ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho U_j \right) = 0 \]  \hspace{1cm} (6.95)

\[ \frac{\partial \rho U_i}{\partial t} + \frac{\partial}{\partial x_j} \left( \rho U_i U_j \right) = - \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j} \left( \mu \frac{\partial U_i}{\partial x_j} \right) + S_{U_i} \]  \hspace{1cm} (6.96)

where \( U_i \) denotes the components of the velocity vector. It is noted that the momentum equation has a form very similar to the previously analysed convective-diffusion equation\(^1\). The only novelty is the presence of the pressure gradient term. The resulting contribution to the discretised equation is the pressure drop over faces of the control volume, i.e. \( p_w - p_e \). To express this contribution in terms of the grid-point pressures, the linear profile for pressure can be assumed resulting in

\[ p_w - p_e = \frac{p_W + p_P}{2} - \frac{p_P + p_E}{2} = \frac{p_W - p_E}{2} \]  \hspace{1cm} (6.97)

\(^1\)In fact, in Section 1.3 it was said that the diffusion-convection equation is just one of the special forms of the general conservation equations when the source term is absent.
In addition, there is another implication that is far more serious. This can be best seen in Figure 6.22 above where the pressure field is expressed in terms of grid-point values. Of course, such a pressure field cannot be regarded as realistic, but since the alternate pressure values are identical, the corresponding pressure difference \( p_w - p_E \) are seen to be zero. The consequence is that such a nonuniform pressure variations will be felt by the momentum equation as a uniform pressure field! The consequences are even more serious for 2D situations as shown in Figure 6.22 below. A similar kind of problem arises

![Pressure Field Diagram](image)

**Figure 6.22**: Check-board pressure fields in 1D and 2D.

When a discretised form of the continuity equation is formed:

\[
\frac{dU}{dx} = 0 \quad \Rightarrow \quad U_x - U_w = 0 \Rightarrow \frac{U_p + U_E}{2} - \frac{U_w + U_E}{2} = 0 \Rightarrow U_E - U_w = 0 \quad (6.98)
\]

So, instead of having the condition of equal velocities at adjacent grid nodes \((w,e)\), the linear pressure interpolation produced conditions of equal velocities at alternate nodes \((W,E)\).

One of the possible solutions for this problem is to eliminate the necessity to linearly interpolate velocity components at the cell faces. This can be achieved by constructing a so-called staggered mesh, Figure 6.23, in which the velocity components are shifted (this is why this mesh is called staggered) to the cell-face locations. By doing this, we are actually introducing three different sets of the geometrical parameters for 3D situations (since three velocity components are located at different locations). The most important advantage of a staggered mesh is in fact that the discretised continuity equation would contain the differences of adjacent velocity components and this will automatically prevent a wavy velocity field. The second advantage is that the pressure difference between the two adjacent grid points becomes a natural driving force for the velocity component located between these grid points. Thus, by introducing the staggered grid arrangements, a decoupling between velocity and pressure fields is prevented. The price is that a more extensive set of the geometrical parameters (specific for each particular velocity components and remaining scalar variables) has to be introduced in the discretisation procedure.
6.4.1 The discretised form of the momentum equation

The discretised form of the momentum equation can be written in essentially the same manner as for the convection-diffusion equation with addition of the pressure term:

\[
\begin{align*}
    a_c U_e &= \sum a_{nb} U_{nb} + b + (p_p - p_T) A_e \\
    a_v V_n &= \sum a_{nb} V_{nb} + b + (p_p - p_T) A_v \\
    a_w W_s &= \sum a_{nb} W_{nb} + b + (p_p - p_T) A_w
\end{align*}
\]

Here the number of neighbour terms will depend on the dimensionality of the problem: 2 for 1D, 4 for 2D and 6 for 3D. The neighbour coefficients \(a_{nb}\) account for the combined convection-diffusion influence at the control-volume faces. The pressure gradient term is treated separately. This term represents the pressure force acting on the control volume and \(A_e\) being the area on which the pressure difference acts. Of course, in two-dimensional problems \(A_e\) will be \(\Delta y \cdot \Delta z\) and for three-dimensional situations it will be \(\Delta x \cdot \Delta y \cdot \Delta z\). It is obvious that the momentum equations can be solved only when the pressure field is given or is estimated\(^2\). When the momentum equation has finally been solved, there is no guarantee that the obtained velocity field will automatically satisfy the continuity equation. It is necessary, therefore, to impose corrections in order to satisfy the continuity equation. Then, such a corrected velocity field will satisfy the continuity equation, but there is no guarantee that these corrected values will satisfy the initial momentum equation. Obviously, an iterative procedure is needed for correcting/updating instantaneous velocity and pressure fields in such a way that they finally lead to fields that, when convergence is obtained, will satisfy both the continuity and the momentum equations.

\(^2\)It is recalled that for a general 3D problem we have three momentum equations - one for each component of the velocity vector, whereas these three equations contain (for incompressible flows) four unknowns: \(U, V, W\) and \(p\). The problem should be closed by using the continuity equation, which, however, does not contain pressure \(p\). Nevertheless, by solving the continuity equation in addition to the momentum equations makes it possible to get unique solution for the four unknown variables.
**The pressure and velocity corrections**

Our goal is to improve the guessed pressure so that the resulting velocity field will progressively get closer to satisfying the continuity equation. Let assume that the correct pressure \( p \) is obtained from

\[
p = p^* + p'
\]  
(6.100)

where \( p^* \) is the initial pressure and \( p' \) is a pressure correction. The momentum equations for corrected velocity components can be written as

\[
a_e U_e' = \sum a_{ab} U_{ab}' + \left( p_p' - p_E' \right) A_e
\]  
(6.101)

By neglecting the term representing the contribution of the neighbouring cells, the discretised form of the corrected momentum equation is

\[
a_e U_e' = \left( p_p' - p_E' \right) A_e \rightarrow U_e' = d_e \left( p_p' - p_E' \right)
\]  
(6.102)

where \( d_e = A_e / a_e \). Now the velocity-correction expression can be written as

\[
U_e = U_e^* + d_e \left( p_p' - p_E' \right)
\]  
(6.103)

and it shows that the velocity from the previous iteration \( u_e^* \) is corrected in response to the pressure corrections to produce \( u_e \). The correction expressions for the velocity components in other directions can similarly be written:

\[
V_e = V_e^* + d_e \left( p_p' - p_N' \right) \quad W_e = W_e^* + d_e \left( p_p' - p_T' \right)
\]  
(6.104)

Before starting with derivation of the remaining pressure-correction equation, it is important to explain the consequences of neglecting the neighbours contributions in the velocity-correction equation. If expressions such as \( a_{ab} U_{ab}' \) are present, they have to be expressed in terms of the pressure corrections and the velocity corrections at the neighbours of \( U_{ab} \). These neighbours will in return bring their neighbours until all grid points in the discretised domain will be included. It is obvious that such a form of the pressure-correction equation will not be suitable for an iterative procedure. Note that the omission of the \( \sum a_{ab} U_{ab}' \) term makes it possible that the \( p' \) equation can be solved by using a general sequential solver (one variable at a time).

### 6.4.2 The pressure-correction equation

The continuity equation in 3D reads:

\[
\frac{\partial p}{\partial t} + \frac{\partial (\rho U)}{\partial x} + \frac{\partial (\rho V)}{\partial y} + \frac{\partial (\rho W)}{\partial z} = 0
\]  
(6.105)

Integrating this equation over a control volume we get

\[
\frac{(p_p - \rho_0 p^*) \Delta x \Delta y \Delta z}{\Delta t} + [(\rho U)_x - (\rho U)^*] \Delta y \Delta z + [(\rho V)_y - (\rho V)^*] \Delta z \Delta x + [(\rho W)_z - (\rho W)^*] \Delta x \Delta y = 0
\]  
(6.106)

Replacing all velocities by the velocity-correction expressions

\[
U_e = U_e^* + d_e \left( p_p' - p_E' \right), \ldots
\]  
(6.107)

the following discretised equation for \( p' \) is obtained:

\[
a_p p_p' = a_E p_E' + a_W p_W' + a_N p_N' + a_S p_S' + a_T p_T' + a_B p_B' + b
\]  
(6.108)
\[ a_F p_F' = a_E p_E' + a_W p_W' + a_N p_N' + a_S p_S' + a_T p_T' + a_B p_B' + b \]  

(6.108)

where

\[ a_E = \rho_c d_c \Delta y \Delta z, a_W = \rho_c d_c \Delta y \Delta z, a_N = \rho_c d_c \Delta z \Delta x, a_S = \rho_c d_c \Delta z \Delta x, \]
\[ a_T = \rho d_t \Delta z \Delta y, a_B = \rho d_t \Delta z \Delta y, a_p = a_E + a_W + a_N + a_S + a_T + a_B \]
\[ b = \frac{(p_F - \rho_2^0) \Delta x \Delta y \Delta z}{\Delta y} + [(\rho U^*)_c - (\rho U^*)_w] \Delta y \Delta z + \]
\[ [(\rho V^*)_n - (\rho V^*)] \Delta z \Delta x + [(\rho W^*)_r - (\rho W^*)_k] \Delta x \Delta y \]

The last term \( b \) in the pressure-correction equation is essentially the left-side of the discretised continuity equation evaluated in terms of \( U^* \) velocities. If this source term is zero \( (b = 0) \), it means that the \( U^* \) velocities do satisfy the continuity equation and no additional pressure corrections are needed. The term \( b \) thus represents a 'mass-source' which the pressure corrections through their associated velocity corrections must eliminate if the convergence criteria to be satisfied.

### 6.4.3 The SIMPLE algorithm

The procedure for calculating the coupling between the velocity and pressure fields can be summarised in the following steps:

- start with an initial pressure field \( p^* \)
- solve the momentum equations to obtain the intermediate velocity components \( U^*, V^*, W^* \)
- solve the pressure-correction equation to obtain the pressure corrections \( p' \)
- update the intermediate pressure and velocity fields with the corrected values \( p = p^* + p', U = U^* + U', V = V^* + V', W = W^* + W' \)
- solve the discretised equations for other variables (temperature, concentration, turbulence quantities, etc.)
- now make the newly corrected pressure \( p \) as a new initial pressure \( p^* \) and repeat the entire procedure until a fully convergent solution is obtained
This pressure-velocity coupling is called the SIMPLE algorithm (Patankar 1980) (Semi-Implicit Method for Pressure-Linked Equations). The word semi-implicit in SIMPLE is used in order to stress the fact that a term in the neighbour contributions ($\sum a_{nb}U'_{nb}$) is neglected. This term represents an indirect or implicit influence of the pressure correction on velocity. It is important to note that the pressure-correction algorithm is just an intermediate algorithm that leads us to the correct pressure field, but it has no direct effect on the final solution. As long as we get a converged solution, all formulations of the $p'$ equation will give the same final solution. The pressure-correction equation is also prone to the divergence unless some sort of under-relaxation is used, such as e.g.

$$p = p' + \alpha p'$$ \hspace{1cm} (6.109)

SEE BOOK PATANKAR 1980, or other literature

### 6.4.4 Pressure-velocity coupling for collocated grid arrangement

In the previous section we derived a discretised form of the momentum and pressure-correction equations using the staggered grid arrangement, Figure 6.23. We will proceed with derivation of the pressure-velocity coupling for a collocated grid arrangement. This arrangement is a preferable option since all dependent variables are defined at the same grid locations and share an identical set of geometrical parameters. In order to prevent the pressure-velocity decoupling as already demonstrated in previous section, a special interpolation is used. Let us summarize the main derivation steps for one-dimensional situations. Starting from

$$a_p U_p = \sum_{nb} a_{nb} U_{nb} + S_{U} = a_E U_E + a_W U_W - \frac{d p}{d x} \Delta V$$ \hspace{1cm} (6.110)

$$a_p p'_p = \sum_{nb} a_{nb} p'_{nb} + b = a_E p'_E + a_W p'_W + (\rho U)_e - (\rho U)_w$$ \hspace{1cm} (6.111)

we can first calculate the intermediate solution for $u_p$. Then, from this value we exclude the pressure-gradient term

$$U^*_p = U_p + \left( \frac{\Delta V \cdot d p}{a_p \cdot d x} \right)_p$$ \hspace{1cm} (6.112)

The cell-face values of such calculated velocity are simply evaluated by linear interpolation:

$$U^*_r = \frac{1}{2} (U^*_E + U^*_p)$$ \hspace{1cm} (6.113)
Transport processes (TRP)

To find the velocity field \( U \) at cell faces, the pressure gradient is added to \( U^* \). But instead of using the interpolated pressure gradients from the nodes, they should be evaluated at cell faces:

\[
U_e = U^*_e - \left( \frac{\Delta V}{a_p} \frac{d p}{d x} \right)_e = U^*_e - \left( \frac{\Delta V}{a_p} \right)_e \frac{p_E - p_P}{\Delta x} \tag{6.114}
\]

This expression can be further reformulated as follows (under the assumption that both \( a_p \) and \( \Delta V \) are constant since the mesh is uniformly distributed):

\[
U_e = \frac{1}{2} (U^*_E + U^*_P) - \frac{\Delta V}{a_p} \frac{p_E - p_P}{\Delta x} = \frac{1}{2} (U_E + U_P) + \frac{\Delta V}{2a_p} \left( \frac{p_{EE} - p_P}{2\Delta x} + \frac{p_E - p_W}{2\Delta x} \right) - \left( \frac{\Delta V}{a_p} \right)_e \frac{p_E - p_P}{\Delta x} = \frac{1}{2} (U_E + U_P) + \frac{\Delta V}{4a_p\Delta x} (p_{EE} - 3p_E + 3p_P - p_W) \tag{6.115}
\]

The resulting expression adds pressure terms to the standard linear interpolation in evaluating the cell-face velocity. As such, this term will prevent velocity-pressure decoupling (check-board pressure fields) since two adjacent sets of discretised nodes are taken into account. This pressure-velocity coupling was introduced by Rhie and Chow (1983) and this was a significant breakthrough in the development of CFD algorithms for complex three-dimensional geometries. The great majority of numerical codes used for fluid flow and heat transfer calculations is based on this interpolation scheme.

6.5 Boundary conditions

To solve the discretised set of equations derived in previous sections, it is necessary to impose the boundary conditions. Through the boundary conditions, the selected discretised domain interacts with the rest of the physical domain. The most common boundary conditions in discretised finite-volume based method are: inlet, outlet, impermeable wall, symmetry, specified pressure and cyclic (periodic) boundaries.

The inlet boundary condition is usually easy to impose if the inflow conditions are known - all values of dependent variables at this boundary should be directly specified (fluid properties, inlet velocity, temperature, heat flux, turbulence, etc.). The velocity component normal to the inlet should also be specified and, based on this value, a total mass inflow can be calculated. Sometimes, in complex flows where a segment of a domain is only considered into which already a developed flow is entering, it could be difficult to estimate the inlet values and a separate set of computations for a sub-domain preceding the main domain of interest need to be performed to provide the inlet information.

The outlet boundary condition are usually specified sufficiently far downstream where one can assume that there are no significant changes in behaviour of dependent variables at that boundary in the flow direction. In order to achieve such conditions, the flow should leave discretised domain over the entire length of the outlet boundary and it should be as much as possible parallel to the horizontal direction. In order to satisfy these conditions, it is often necessary to significantly extend the computational domain downstream. This is especially the case if a recirculation region exists inside the discretised domain. Then
a simple zero-gradient condition can be applied for the outlet boundary, e.g., $\frac{\partial U}{\partial x} = 0$, $U_{NI,J} = U_{NI-1,J}$, where $U_{NI,J}$ is the velocity component in the flow direction at the outlet boundary, and $U_{NI-1,J}$ is the velocity in the first adjacent control volume inside of the discretised domain. The problem is that this type of boundary treatment does not guarantee that the total mass balance in the system considered will be satisfied. To achieve this, both the inflow and outflow mass flux should be calculated and their ratio will serve as the scaling parameter for the calculations of velocity at the outlet boundary, i.e., $U_{NI,J} = U_{NI-1,J} \cdot \frac{M_{\text{inflow}}}{M_{\text{outflow}}}$.

The impermeable wall condition is the most common boundary encountered in confined fluid flow and heat transfer problems. At this boundary, the no-slip boundary condition is applied ($U=V=W=0$), i.e. all velocity components at the wall have zero values (or the wall velocity if the wall moves). In addition, the scalar variables (temperature, concentration, turbulent kinetic energy, etc.) can have different boundary values (predefined value or flux, constant or varying over the boundary). Note that the specific treatment may be necessary when the flow is turbulent: if the model and the computational mesh do not permit the integration through the viscous/conductive wall layer, the so-called wall functions need to be applied (as discussed in detail in the following chapter).

The symmetry boundary condition assumes that there is no flow through the boundary and that all fluxes are zero. This condition simply assumes that the normal velocity is set to zero at the boundary and remaining components are just mirrored from their inner counterparts, i.e., $\Phi_{NI,J} = \Phi_{NI-1,J}$. This boundary condition is often used for steady flow situations where only a part of the complete discretised domain can be simulated. At the symmetry plane all convective fluxes in the flow-normal direction are zero, i.e. there is no flow, nor scalar transport through this boundary.

Special care has to be taken when time-dependent or three-dimensional computations are performed with symmetry boundary conditions, since many natural flow instabilities can be suppressed by such treatment of the boundary. Moreover, often in such circumstances a symmetry of geometrical domain does not imply the symmetry of the final field solutions; if the problem is unsteady it is thus always advisable to perform computations in the complete geometrical domain.

The periodic or cyclic boundary conditions can be used when the flow does not change in one, e.g. in fully developed flows such as pipes or channels. In this case, one can consider only a small segment of the flow of interest as the solution domain and use to values obtained at the outlet boundary to redefine the values at the inlet plane. This approach is very useful when considering very long geometrical domains because the solutions can be obtained by computing only a small part by imposing periodicity conditions. Special care is needed when not all dependent variables can be treated in this way, e.g. the temperature if there is a heat source of wall heat transfer in the solution domain.

An example of application of imposed boundary conditions is shown in Figure 6.24. The outlet (+) and (-) denote a correct and wrong position of the zero-gradient outlet boundary conditions. The symmetry indicates that instead of solving the entire domain, only its half can be calculated.
6.6 Chart diagram of a numerical code for CFD

Finally, by putting all discretisation principles that are derived in this chapter together, a typical chart diagram of a numerical code for solving the time-dependent fluid flow, heat transfer and turbulence based on the SIMPLE algorithm, is presented in Figure 6.25. The first step is to read all geometrical information from the grid generator. The next step involves the initiation of all dependent variable, i.e. specifying their initial value in the entire domain. The part addressing the main calculations consists of separate subroutines for different variables. For example, in the case of a turbulent flow with heat transfer, the subroutines should provide the velocity components \(U, V, W\), pressure \(P\), temperature \(T\) and, depending on the model (see next chapters), the additional turbulence variables, e.g. - turbulent kinetic energy \(k\), its dissipation rate \(\varepsilon\), temperature variance \(\theta^2\) and others. The next step is to assemble and calculate the discretisation coefficients of the discretised transport equations, which should account for the imposed boundary conditions. The linearised system of equations for all variables can now be solved. After each iteration, the convergence criteria should be checked: if these are not satisfied, the entire process should be repeated. In case of the time-dependent simulations, after reaching the convergence criteria for particular time steps, the cycle is repeated for the following time step until the final time is reached. Finally, the fully convergent results are exported for postprocessing.
Figure 6.25 Typical chart diagram of a numerical code for fluid flow ($U - V - W - P$), heat transfer ($T$) and turbulence parameters ($k$, $\mu^2$) based on the SIMPLE algorithm, Kenjereš (1998).

Sources used

(besides course book Hanjalić et al.)

- J. Brännbacka "Introduction to CFD" course material Åbo Akademi University (version 2006)
- S.V. Patankar "Numerical heat transfer and fluid flow" Hemisphere publishing Corp. (1980)