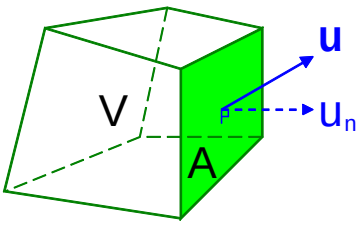

4.1 The Scalar-Transport Equation

This is a generic equation for any transported physical quantity (momentum, energy, ...) For an arbitrary control volume V:

$$\begin{pmatrix}\textbf{RATE OF CHANGE}\\ \textit{inside } V\end{pmatrix} + \begin{pmatrix}\textbf{FLUX}\\ \textit{through boundary of } V\end{pmatrix} = \begin{pmatrix}\textbf{SOURCE}\\ \textit{inside } V\end{pmatrix}$$



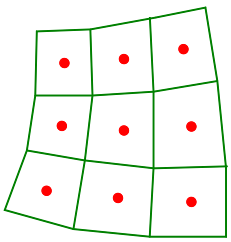
The total flux comprises *advection* (transport by the flow) and *diffusion*. The resulting *scalar-transport* or *advection-diffusion* equation for *concentration* ϕ may be written, for any control volume as:

$$\underset{\textit{rate of change}}{\frac{\mathrm{d}}{\mathrm{d}t}(mass\times\phi)} + \underset{\textit{faces}}{\sum}\Big(\underbrace{mass\ flux\times\phi}_{\textit{advection}} \underbrace{-\Gamma\frac{\partial\phi}{\partial n}A}_{\textit{diffusion}}\Big) = \underset{\textit{source}}{S}$$

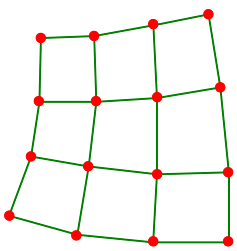
(Non-advective fluxes not described by gradient diffusion can be transferred to the source.)

4.2 Control-Volume Notation

The commonest nodal configurations are *cell-centred* or *cell-vertex*.

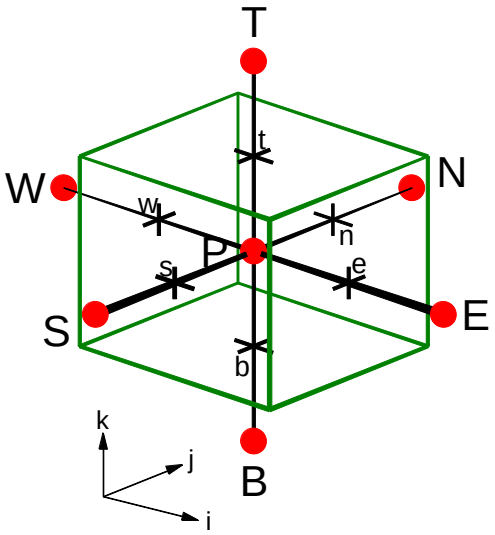


cell-centred



cell-vertex

This course focuses on *structured* meshes using *cell-centred* storage. (Unstructured meshes will be discussed briefly in Section 9, but Versteeg and Malalasekera's book gives a much better description.)



A typical 3-d control volume is shown right. Relative to the cell centre (point *P*) the coordinate directions are commonly denoted *west*, *east*, *south*, *north*, *bottom*, *top* with:

- **lower case** *w*, *e*, *s*, *n*, *b*, *t* used for cell **faces**;
- **upper case** *W*, *E*, *S*, *N*, *B*, *T* for adjacent **nodes**.

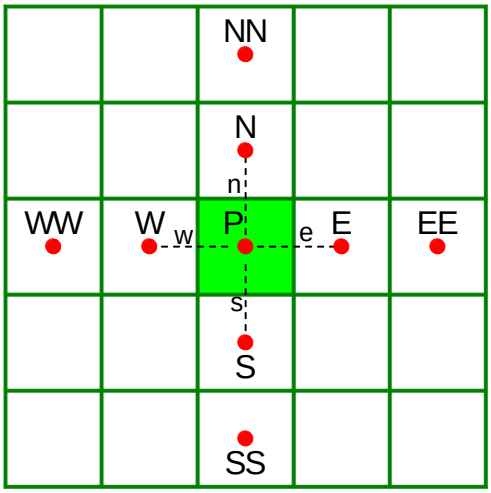
For a Cartesian mesh these would usually correspond to $\pm x$, $\pm y$, $\pm z$ directions respectively.

Cell-face areas will be denoted A_w, A_e, A_s, A_n, A_b, A_t. Cell volumes will be denoted V.

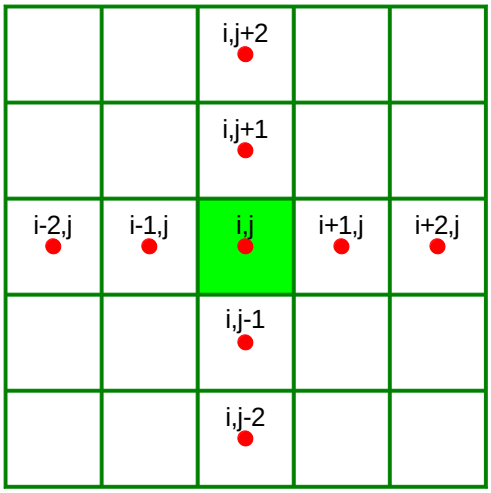
In 2 dimensions one can think of a single layer of cells with unit depth: the cell "volume" would then be equal to the planar area.

When referring to the entire set of control volumes it is common to switch between local geographic and *ijk* notation, so that

$\phi_P \equiv \phi_{ijk}$, $\phi_E \equiv \phi_{i+1\,jk}$, etc.



Local



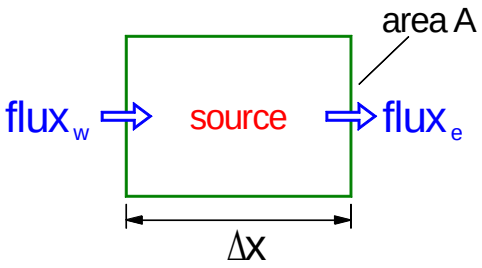
Global

4.3 The Steady-State 1-D Advection-Diffusion Equation

Consider first the steady-state, 1-d advection-diffusion equation. This is worthwhile because:

- it simplifies the analysis;
- it can often be solved by hand;
- subsequent generalisation to 2 and 3 dimensions is straightforward;
- (on structured meshes) discretisation of fluxes is usually carried out coordinate-wise;
- many important theoretical problems are 1-d.

Integral (Control-Volume) Form



Conservation for one control volume gives

$$flux_e - flux_w = source \qquad (1)$$

If ϕ is the amount per unit mass, then the flux is composed of:

advection: $(\rho u A)\phi$

diffusion: $-\Gamma A \dfrac{d\phi}{dx}$

Hence, if s is the source per unit length then the advection-diffusion equation for ϕ is

$$\left(\rho u A\phi - \Gamma A \frac{d\phi}{dx}\right)_e - \left(\rho u A\phi - \Gamma A \frac{d\phi}{dx}\right)_w = s\Delta x \qquad (2)$$

Conservative Differential Form

Dividing by Δx and taking the limit as $\Delta x \to 0$ gives a corresponding differential equation:

$$\frac{d}{dx}\left(\rho u A\phi - \Gamma A \frac{d\phi}{dx}\right) = s \qquad (3)$$

Non-Conservative Differential Form

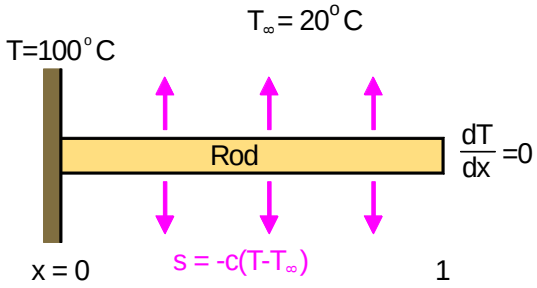
Mass conservation implies that $\rho u A = constant$ and hence (3) can also be written

$$\rho u A \frac{d\phi}{dx} - \frac{d}{dx}\left(\Gamma A \frac{d\phi}{dx}\right) = s \qquad (4)$$

Note:

- This system is quasi-1-d in the sense that the cross-sectional area A may vary. To solve a truly 1-d problem just set $A = 1$. The differential equation is then
$$\frac{d}{dx}\left(\rho u\phi - \Gamma \frac{d\phi}{dx}\right) = s$$
- For the simple cases below, ρ, u, Γ and s are assumed to be known. In the general CFD problem, u is itself the subject of a transport equation and ρ, Γ, S may be functions of the solution. The equation would then have to be solved iteratively.

Classroom Example 1



A thin rod has length 1 m and cross-section 1 cm × 1 cm. The left-hand end is kept at 100 ºC, whilst the right-hand end is insulated. The heat flux across any section of area A is given by

$$-kA\frac{\mathrm{d}T}{\mathrm{d}x}$$

where the conductivity k = 1000 W m^{-1} K^{-1}.

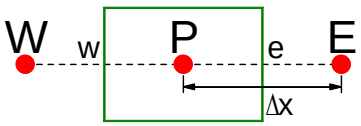
The rod is allowed to cool along its length at a rate proportional to its difference from the ambient temperature (Newton's law of cooling); i.e. the heat loss per unit length is:

$$s = -c(T - T_\infty)$$

where the ambient temperature T_∞ = 20 ºC and the coefficient c = 2.5 W m^{-1} K^{-1}.

(a) Write down and solve the differential equation satisfied by the temperature.
(b) Divide the rod into 5 control sections with nodes at the centre of each section and carry out a finite-volume analysis to find the temperature along the rod.

4.4 Discretising Diffusion



Gradient diffusion is usually discretised by a *central differencing* approximation for the **derivative**:

$$-\Gamma A\left.\frac{\mathrm{d}\phi}{\mathrm{d}x}\right|_e \quad \rightarrow \quad -(\Gamma A)_e\left(\frac{\phi_E - \phi_P}{\Delta x}\right) \quad \rightarrow \quad -D_e(\phi_E - \phi_P) \qquad (5)$$

where

$$D \equiv \frac{\Gamma A}{\Delta x} \qquad (6)$$

is a *diffusive transfer coefficient*.

A similar expression is used for the west face:

$$-\Gamma A\left.\frac{\mathrm{d}\phi}{\mathrm{d}x}\right|_w \quad \rightarrow \quad -D_w(\phi_P - \phi_W)$$

Note:

- In the finite-volume method, fluxes are required at cell **faces**, not nodes.
- This approximation for $(\mathrm{d}\phi/\mathrm{d}x)_e$ is *second-order accurate* in Δx (see later).
- If the diffusivity Γ varies then its cell-face value must be obtained by interpolation.
- Equal weight is applied to the nodes on either side of the cell face, consistent with diffusion acting equally in all directions. Later on, we shall see that this contrasts with advection, which has a directional bias.

4.5 Discretising the Source Term

When the source is proportional to the amount of fluid, the total source strength for the cell is

$$S = s \times V \qquad (\textit{source per unit volume}) \times (\textit{volume})$$

In 1-d problems V is the cell length, Δx, so

$$S = s\Delta x$$

S may depend on the solution ϕ, as in the example above. To form algebraic equations the source term must be linearised:

$$S = b_P + s_P \phi_P \,, \qquad s_P \le 0 \tag{7}$$

If b_P and s_P are functions of the unknown solution ϕ, rather than constants, the equation set has to be solved iteratively. The requirement for s_p to be negative will be explained later.

4.6 Assembling the Algebraic Equations

As seen in the classroom example, when there is no flow (u = 0) the purely diffusive problem discretises as follows.

$$\begin{array}{ccccc} \textit{flux}_e & - & \textit{flux}_w & = & \textit{source} \\ -D_e(\phi_E - \phi_P) & + & D_w(\phi_P - \phi_W) & = & b_P + s_P\phi_p \end{array}$$

Collecting multiples of ϕ_P, ϕ_E and ϕ_W together (noting carefully where there are minus signs):

$$-a_w\phi_W + a_P\phi_P - a_E\phi_E = b_P$$

where $a_W = D_w$, $a_E = D_e$, $a_P = D_w + D_e - s_P$.

This is a special case of the single-cell equation that applies in 2 and 3 dimensions (and where there is advection as well as diffusion):

$$a_P\phi_P - \sum_{\substack{\textit{adjacent} \\ \textit{nodes}}} a_F\phi_F = b_P \tag{8}$$

(8) is a canonical form for the discretised scalar-transport equation.

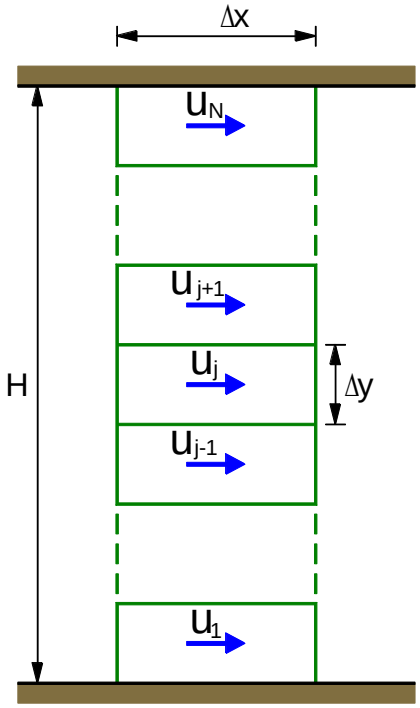
There will be a discretised equation of this form for each variable and for each control volume. For one variable ϕ, if the nodal values are assembled into a vector then the set of algebraic equations takes the form

$$\begin{pmatrix} \ddots & \ddots & & & 0 \\ \ddots & \ddots & \ddots & & \\ & -a_W & a_P & -a_E & \\ & & \ddots & \ddots & \ddots \\ 0 & & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ \vdots \\ \phi_P \\ \vdots \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \vdots \\ b_P \\ \vdots \\ \vdots \end{pmatrix} \qquad \text{or} \qquad \left(\quad\right) \Phi = \mathbf{b}$$

For a 1-d system this is *tri-diagonal*. If the coefficients are constants then it can be solved directly by Gaussian elimination or very efficiently on a computer by the *tri-diagonal matrix algorithm* or TDMA (see the Appendix). If the elements of the matrix are not constants but dependent on the unknown solution ϕ (or other variables) then it must be solved iteratively.

Classroom Example 2 (Computational Hydraulics Exam, June 2009 – extended by part(h))

A 2-d finite-volume calculation is to be undertaken for fully-developed, laminar flow between stationary, plane, parallel walls. A streamwise pressure gradient $\mathrm{d}p/\mathrm{d}x = -G$ is imposed and the fluid viscosity is μ. The depth of the channel, H, is divided into N equally-sized cells of dimension $\Delta x \times \Delta y \times 1$ as shown, with the velocity u stored at cell centres.



(a) What are the boundary conditions on velocity?

(b) What is the net pressure force on a single cell?

(c) Using a finite-difference approximation for velocity gradient, find expressions for the viscous forces on upper and lower faces of the j^{th} cell in terms of the nodal velocities $\{u_j\}$. (You will need to deal separately with interior cells and the boundary cells $j = 1$ and $j = N$.)

(d) By balancing pressure and viscous forces set up the finite-volume equations for the velocity field.

(e) Solve for the nodal velocities in the case $N = 6$, leaving your answers as multiples of $U_0 = GH^2/\mu$. (You are advised to note the symmetry of the problem.)

(f) Using your numerical solution, find the volume flow rate (per unit span), Q, in terms of U_0 and H.

(g) Find the wall shear stress τ_w.

(h) Compare your answers to (e), (f), (g) with the exact solution for plane Poiseuille flow.

4.7 Discretising Advection (Part 1)

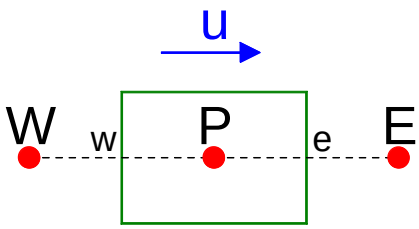
In most engineering flows advective fluxes far exceed diffusive fluxes because the Reynolds number (ratio of inertial forces [mass × acceleration] to viscous forces) is very large.

The 1-d steady-state advection-diffusion equation is

$$flux_e - flux_w = source$$

where, with mass flux C (= $\rho u A$):

$$flux = C\phi - \Gamma A \frac{d\phi}{dx}$$

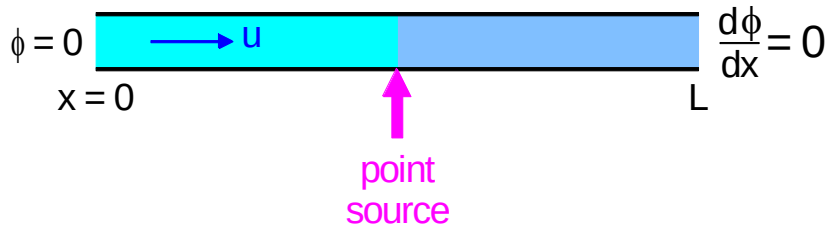


Discretising the diffusion and source terms as before, the equation becomes

$$\underset{advection}{[C_e\phi_e - C_w\phi_w]} + \underset{diffusion}{[-D_e(\phi_E - \phi_P) + D_w(\phi_P - \phi_W)]} = \underset{source}{b_P + s_P\phi_P} \qquad (9)$$

ϕ_e and ϕ_w have yet to be approximated. The problem is how to approximate these **face** values in terms of the values at adjacent **nodes**. A method of specifying these face values in order to calculate advective fluxes is called an *advection scheme* or *advection-differencing scheme*.

Classroom Example 3.

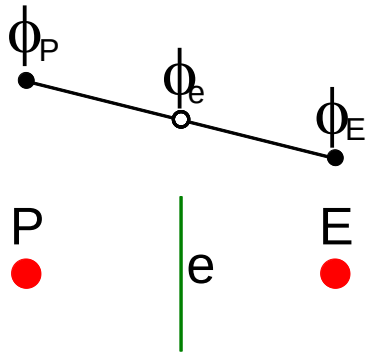


A pipe of cross-section $A = 0.01$ m^2 and length $L = 1$ m carries water (density $\rho = 1000$ kg m^{-3}) at velocity $u = 0.1$ m s^{-1}.

A faulty valve introduces a reactive chemical into the pipe half-way along its length at a rate of 0.01 kg s^{-1}. The diffusivity of the chemical in water is $\Gamma = 0.1$ kg m^{-1} s^{-1}. The chemical is subsequently broken down at a rate proportional to its concentration ϕ (mass of chemical per unit mass of water), this rate amounting to $-\gamma\phi$ per metre, where $\gamma = 0.5$ kg s^{-1} m^{-1}.

Assuming that the downstream boundary condition is $d\phi/dx = 0$, set up a finite-volume calculation with 7 cells to estimate the concentration along the pipe using:
(a) *central*
(b) *upwind*
differencing schemes for advection.

4.7.1 Central Differencing



In *central differencing* for advection the cell-face value is approximated by the average of the nodal values on either side:

$$\phi_e = \tfrac{1}{2}(\phi_P + \phi_E)$$

This is second-order accurate in Δx (see later).

Substituting into (9) (with a similar expression for ϕ_w) gives

$$\tfrac{1}{2}C_e(\phi_P + \phi_E) - \tfrac{1}{2}C_w(\phi_W + \phi_P) - D_e(\phi_E - \phi_P) + D_w(\phi_P - \phi_W) = b_P + s_P\phi_P$$

or, collecting terms,

$$-a_W\phi_W + a_P\phi_P - a_E\phi_E = b_P$$

where:

$$\begin{aligned} a_W &= \tfrac{1}{2}C_w + D_w \\ a_E &= -\tfrac{1}{2}C_e + D_e \\ a_P &= a_E + a_W - s_P + (C_e - C_w) \end{aligned} \qquad (10)$$

(By mass conservation, $C_e - C_w = 0$, so that the expression for a_P can be simplified.)

The graphs below show the solution of an advection-diffusion problem with no sources, constant diffusivity and ϕ fixed as 0 and 1 at upstream and downstream ends for cases

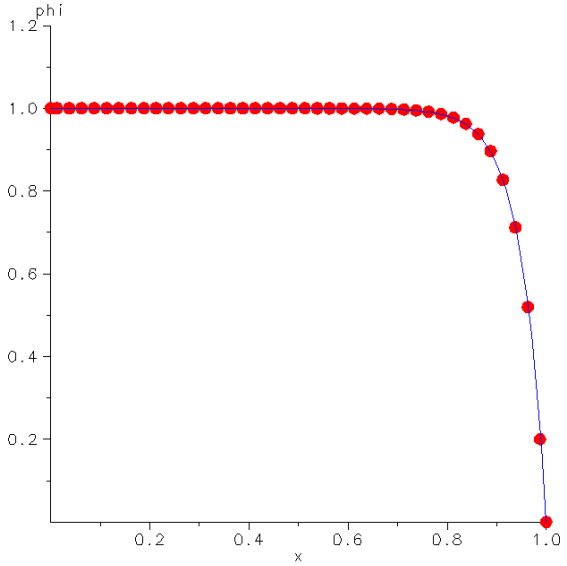
Pe = 1/2 (advection « diffusion) (equation: $-\tfrac{5}{4}\phi_W + 2\phi_P - \tfrac{3}{4}\phi_E = 0$)

Pe = 4 (advection » diffusion) (equation: $-3\phi_W + 2\phi_P + \underline{\phi_E} = 0$)

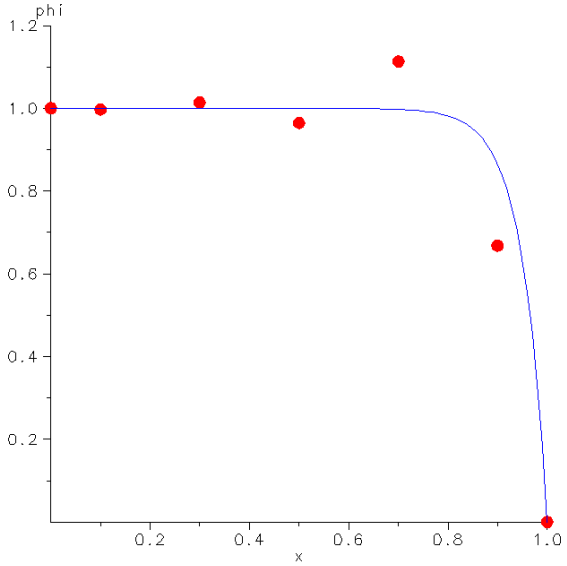
where the *Peclet number* Pe is defined by

$$\text{Pe} = \frac{C}{D}\left(\text{i.e. } \frac{\textit{advection}}{\textit{diffusion}}\right) = \frac{\rho u \Delta x}{\Gamma} \qquad (11)$$

Pe = 1/2



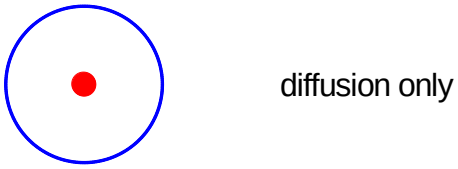
Pe = 4



In the first case the solution is good (consistent with second-order accuracy).

In the second case there are pronounced "wiggles" in what should be a perfectly smooth solution. What is wrong?

Mathematically, when the cell Peclet number Pe is bigger than 2, the a_E **coefficient becomes negative**, meaning that, for example, an **increase** in ϕ_E would lead to a **decrease** in ϕ_P. This is impossible for a quantity that is simply advected and diffused.



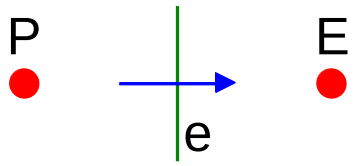
Physically, the advection process is **directional**; it transports properties only in the direction of the flow. However, the central-differencing formula assigns **equal weight** to both upwind and downwind nodes.

4.7.2 Upwind Differencing

In *upwind differencing* ϕ_{face} is taken as the value of ϕ at whichever is the upwind node; i.e. in one dimension:

$$\phi_e = \begin{cases} \phi_P & (\text{if } u > 0) \\ \phi_E & (\text{if } u < 0) \end{cases}$$

This is only first-order accurate in Δx (see later) but acknowledges the directional nature of advection. The alternatives can be summarised as



$$\phi_{face} = \phi_U$$

where subscript U denotes whichever (P or E) is the **upwind** node for that face.

With this scheme, substituting into (9) gives

$$-a_W\phi_W + a_P\phi_P - a_E\phi_E = b_P$$

where:

$$\begin{aligned} a_E &= \max(-C_e, 0) + D_e \\ a_W &= \max(C_w, 0) + D_w \\ a_P &= a_E + a_W - s_P \end{aligned} \qquad (12)$$

If the "max" bit confuses you, consider separately the two cases where the mass flux is positive (flow from left to right) or negative (flow from right to left).

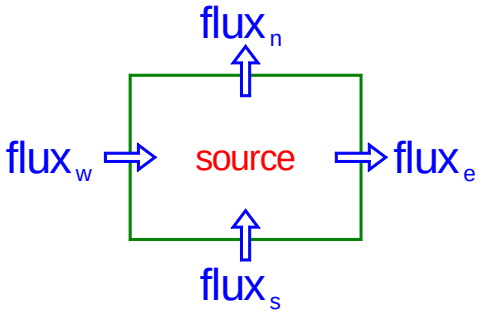
When applied to the same advection-diffusion problem as the central-differencing advection scheme it is found that:

- when Pe = ½ the upwind-differencing scheme is not as accurate as central differencing; this is to be expected from its order of accuracy (see later);
- when Pe = 4 the upwind-differencing solution is not particularly accurate, but the "wiggles" disappear.

In all cases, however, both a_W and a_E are unconditionally positive.

So there is a pay-off – *accuracy* versus *boundedness* (absence of wiggles). After a brief examination of 2-d or 3-d discretisation, the following sections examine the desirable properties of discretisation schemes, the constraints that these properties impose and more advanced advection schemes that are both accurate and bounded.

4.8 Extension to 2 and 3 Dimensions



For a multi-dimensional flow the net flux out of a cell can be obtained by summing the **outward** fluxes through all faces.

For quadrilateral elements (in 2-d) or hexahedral elements (in 3-d) the net flux out of a cell is simply the sum of the net fluxes through opposing sides and the general conservation equation may be written:

$$(flux_e - flux_w) + (flux_n - flux_s) + (flux_t - flux_b) = source \qquad (13)$$

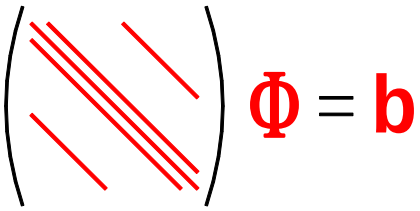
where *flux* here refers to a "forward" flux; i.e. in the direction of the increasing coordinate.

The discretised equations are still assembled in the same matrix form

$$a_P \phi_P - \sum_F a_F \phi_F = b_P \qquad (14)$$

with the summation extended to include nodes in the other directions.

Combining the individual equations (14) from all control volumes gives a matrix equation for the set of nodal values. In two dimensions this gives the banded matrix system shown. Further bands appear in three dimensions.



Thus, equation (14) has the same form in 1-, 2- or 3-d problems, but in multiple dimensions the summation is extended to the other (*S*, *N*, *B*, *T*) nodes and there is a corresponding increase in the number of non-zero diagonals in the assembled matrix equation. Although the matrix equation is almost as easy to write down as in 1-d, it is now much harder to solve (see Section 4.13).

Equation (14) also describes the discretisation for a single control volume in an *unstructured* mesh. However, since the nodes do not have a simple *ijk* indexing, the resulting matrix and solution method for unstructured meshes are much more complicated.

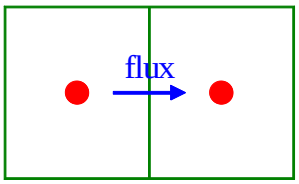
4.9 Discretisation Properties

(i) *Consistency*

An approximation is *consistent* if the discretised equations are equivalent to the continuum equations in the limit as the grid size tends to zero. E.g., by the definition of a derivative,

$\frac{\phi_E - \phi_P}{\Delta x}$ is a consistent approximation for $\frac{\partial \phi}{\partial x}$.

(ii) *Conservativeness*



A scheme is *conservative* if fluxes are associated with faces, not cells, so that what goes **out** of one cell goes **into** the adjacent cell.

This is automatically built into the finite-volume method. Computationally, it is most efficient if fluxes are first calculated facewise and then distributed to the relevant cells. Otherwise each flux would be calculated twice – once for each cell on either side of a face.

(iii) *Transportiveness*

An advection scheme is *transportive* if it is **upstream-biased**. In practice this means higher weighting to nodes on the upstream side of a face.

(iv) *Boundedness*

A flux-differencing scheme is *bounded* if, in an advection-diffusion problem without sources:

- the value of ϕ at a node always lies between the maximum and minimum values at surrounding nodes;
- ϕ = constant is a possible solution.

This imposes conditions on the matrix coefficients a_P, a_W, a_E, ... If there are no sources then

$$a_P \phi_P - \sum a_F \phi_F = 0 \qquad (15)$$

Suppose that ϕ is only non-zero at one adjacent node, F. Then

$$a_P \phi_P - a_F \phi_F = 0 \qquad \text{or} \qquad \phi_P = \frac{a_F}{a_P} \phi_F$$

Since ϕ_P must lie between 0 and ϕ_F, this requires that

$$0 \le \frac{a_F}{a_P} \le 1$$

Hence, a_F and a_P must have the same sign (invariably positive in practice). Thus, we require:

$$a_F \ge 0 \text{ for all } F \qquad (\text{“}\textit{positive coefficients}\text{”}) \qquad (16)$$

(Contravening of the positivity condition leads central-differencing to produce “wiggles”.)

If equatin (15) is also to admit the solution ϕ = constant then this can be divided out to yield

$$a_P = \sum a_F \qquad (\text{“}\textit{sum of neighbouring coefficients}\text{”}) \qquad (17)$$

(v) *Stability*

A **solution method** (not an advection scheme) is *stable* if small errors do not grow in the course of the calculation. This determines whether it is possible to obtain a converged solution: it says nothing about its accuracy or whether it is bounded. Stability is strongly influenced by how the **source** term is discretised.

Any source term should be linearised as $b_P + s_P\phi_P$; the complete equation for one cell is then

$$a_P\phi_P - \sum a_F\phi_F = b_P + s_P\phi_P$$

If the solution-dependent part of the source ($s_P\phi_P$) is transferred to the LHS then the diagonal coefficient is modified to read

$$a_P = \sum a_F - s_P$$

Numerical stability requires *negative feedback*; otherwise, an increase in ϕ would lead to an increase in the source, which would lead to a further increase in ϕ and so on. Thus:

$s_P \le 0$ ("*negative-slope linearisation of the source term*")

If this condition and the positivity of the a_F is maintained then

$a_P \ge \sum |a_F|$ ("*diagonal dominance*")

The last condition is, in fact, a necessary requirement of many matrix solution algorithms.

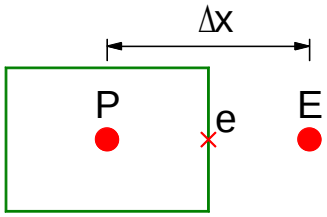
To summarise, boundedness and stability place the following constraints on the discretisation of flux and source terms:

positive coefficients:	$a_F \ge 0$ for all F
negative-slope linearisation of the source term:	$source = b_P + s_P\phi_P, \quad s_P \le 0$
sum of neighbouring coefficients:	$a_P = \sum_F a_F - s_P$

(vi) *Order*

Order is a measure of accuracy. It defines how fast the error in a numerical approximation diminishes as the grid spacing gets smaller.

If, on a uniform grid of spacing Δx, the truncation error is proportional to Δx^n as $\Delta x \to 0$ then that scheme is said to be *of order n*.



Order can be established formally by a Taylor-series expansion about a **cell face**. e.g. for the nodes either side of the east face:

$$\phi_E = \phi_e + \left(\frac{d\phi}{dx}\right)_e \left(\frac{\Delta x}{2}\right) + \frac{1}{2!}\left(\frac{d^2\phi}{dx^2}\right)_e \left(\frac{\Delta x}{2}\right)^2 + \frac{1}{3!}\left(\frac{d^3\phi}{dx^3}\right)_e \left(\frac{\Delta x}{2}\right)^3 + \cdots \quad (18)(a)$$

$$\phi_P = \phi_e - \left(\frac{d\phi}{dx}\right)_e \left(\frac{\Delta x}{2}\right) + \frac{1}{2!}\left(\frac{d^2\phi}{dx^2}\right)_e \left(\frac{\Delta x}{2}\right)^2 - \frac{1}{3!}\left(\frac{d^3\phi}{dx^3}\right)_e \left(\frac{\Delta x}{2}\right)^3 + \cdots \quad (18)(b)$$

Subtracting (18)(a) – (b) gives:

$$\phi_E - \phi_P = 0 \quad + \quad \left(\frac{\mathrm{d}\phi}{\mathrm{d}x}\right)_e \Delta x \quad + \quad 0 \quad + \quad \frac{1}{3}\left(\frac{\mathrm{d}^3\phi}{\mathrm{d}x^3}\right)_e \left(\frac{\Delta x}{2}\right)^3 \quad + \quad \cdots$$

whence, dividing by Δx,

$$\frac{\phi_E - \phi_P}{\Delta x} = \left(\frac{\mathrm{d}\phi}{\mathrm{d}x}\right)_e + O(\Delta x^2)$$

As the leading error term is $O(\Delta x^2)$, $\dfrac{\phi_E - \phi_P}{\Delta x}$ is a second-order approximation for $\left(\dfrac{\mathrm{d}\phi}{\mathrm{d}x}\right)_e$.

Alternatively, adding (18)(a) + (b) gives:

$$\phi_P + \phi_E = 2\phi_e \quad + \quad 0 \quad + \quad \left(\frac{\mathrm{d}^2\phi}{\mathrm{d}x^2}\right)_e \left(\frac{\Delta x}{2}\right)^2 \quad + \quad \cdots$$

whence:

$$\tfrac{1}{2}(\phi_P + \phi_E) = \phi_e + O(\Delta x^2)$$ (read $O(\Delta x^2)$ as "of the *order* of Δx^2").

As the error term is $O(\Delta x^2)$, the central-differencing formula $\tfrac{1}{2}(\phi_P + \phi_E)$ is a second-order approximation for ϕ_e. On the other hand, the upwind-differencing approximations ϕ_P or ϕ_E (depending on the direction of the flow) are first-order accurate.

Higher accuracy can be obtained by using more nodes.

Schemes of low-order accuracy, e.g. upwind, lead to substantial numerical diffusion in 2-d and 3-d calculations when the velocity vector is not aligned with the grid lines.

Notes.

(1) *Order* is an *asymptotic* concept; i.e. it refers to behaviour as $\Delta x \to 0$. In this limit only the first non-zero truncation term in the Taylor series is important. However, the full expansion includes terms of higher order which may be non-negligible for finite Δx.

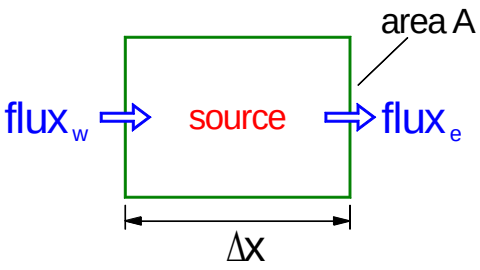
(2) *Order* refers to the *theoretical truncation error* in the approximation, not the computer's *round-off error* (the accuracy with which it can store floating-point numbers).

(3) The more accurate a scheme then, in principle, the greater the reduction in numerical error as the grid is made finer, or, conversely, the less nodes required for a given accuracy. However, high-order schemes tend to require more computational resources and often have boundedness or stability problems. Also, the law of diminishing returns applies once the truncation error becomes of similar size to the computer's floating-point round-off error.

4.10 Discretising Advection (Part 2)

With an understanding of the desirable properties of a flux-differencing scheme it is now possible to examine more advanced schemes.

4.10.1 Exponential Scheme (Patankar, 1980)



The 1-d advection diffusion equation is

$$flux_e - flux_w \equiv source$$

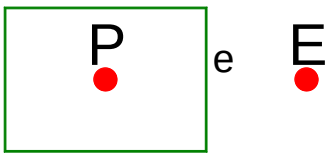
or, dividing by cell length Δx and proceeding to the limit:

$$\frac{\mathrm{d}}{\mathrm{d}x}(\rho u A\phi - \Gamma A\frac{\mathrm{d}\phi}{\mathrm{d}x}) = s \qquad (19)$$

If there are no sources ($s = 0$) then the total flux must be constant:

$$flux = \rho u A\phi - \Gamma A\frac{\mathrm{d}\phi}{\mathrm{d}x} \quad = constant$$

If ρ, u, A and Γ are constant this equation can be solved exactly, with boundary conditions $\phi = \phi_P$ and $\phi = \phi_E$ at adjacent nodes, to give (see the Examples):



$$flux_e = C\left(\frac{\mathrm{e}^{Pe}\phi_P - \phi_E}{\mathrm{e}^{Pe} - 1}\right)$$

where

$C = \rho u A$ = mass flux

$D = \dfrac{\Gamma A}{\Delta x}$ = diffusive transfer coefficient

$\mathrm{Pe} = \dfrac{C}{D} = \dfrac{\rho u \Delta x}{\Gamma}$ = Peclet number

.

With a similar expression for the west face, one obtains

$$flux_e - flux_w = a_P\phi_P - \sum a_F\phi_F$$

where:

$$a_W = \frac{C\mathrm{e}^{\mathrm{Pe}}}{\mathrm{e}^{\mathrm{Pe}} - 1}, \qquad a_E = \frac{C}{\mathrm{e}^{\mathrm{Pe}} - 1}, \qquad a_P = a_E + a_W \qquad (20)$$

Assessment

- Conservative by construction.
- Transportive, because there is a larger weighting on the upwind node.
- Bounded: all a_F are positive and a_P is the sum of the neighbouring coefficients.

To see the last two of these you will have to consider separately the cases $C > 0$ (for which $\mathrm{e}^{\mathrm{Pe}} > 1$) and $C < 0$ (for which $\mathrm{e}^{\mathrm{Pe}} < 1$).

This scheme, by construction, gives the **exact** solution for **zero sources** and **constant velocity and diffusivity** ... but this is something we could have found analytically anyway. The scheme has never really found favour because:

- the scheme is not exact when u or Γ vary, or if there are sources, or in 2-d or 3-d flow;
- exponentials are extremely expensive to compute.

4.10.2 Hybrid Scheme (Spalding (1972)

This is an approximation to the exponential scheme which amounts to:

- central differencing if $|\text{Pe}| \le 2$;
- upwind differencing (with zero diffusion!) if $|\text{Pe}| > 2$.

$$flux_e - flux_w = a_P\phi_P - \sum a_F\phi_F$$

where (if $u > 0$ and the mass flux C and diffusive transport coefficient D are constant):

$$\begin{cases} a_W = \frac{1}{2}C + D, & a_E = -\frac{1}{2}C + D & \text{if } \text{Pe} \equiv C/D \le 2 \\ a_W = C, & a_E = 0 & \text{if } \text{Pe} > 2 \end{cases} \qquad (21)$$

$$a_P = a_E + a_W$$

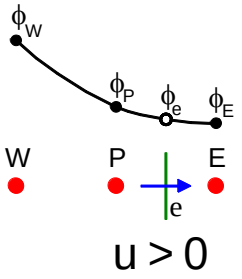
Assessment
The scheme is conservative, transportive and bounded.

The hybrid scheme remained popular in commercial codes for a long time because it was stable and robust. However, most flows of interest operate in the high-advection/low-diffusion regime, where this scheme amounts to first-order upwinding with no diffusion. Modern CFD practitioners seek much higher accuracy.

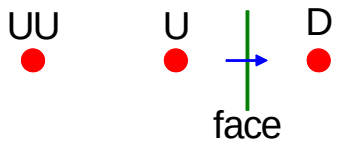
Patankar also developed a *power-law* approximation to the exponential scheme, to overcome the heavy-handed switch-off of diffusion at $\text{Pe} = 2$. However, "powers" are just as computationally expensive as exponentials.

4.10.3 QUICK (QUadratic Interpolation for Convective Kinematics – Leonard, 1979)

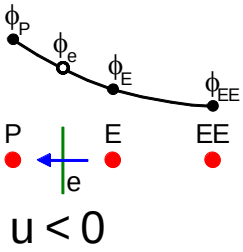
Fits a quadratic polynomial through **3 nodes** to get 3rd-order accuracy.



For each cell face, QUICK uses the nodes either side of the cell face, plus a further upwind node depending on the direction of the flow as shown right.



To emphasise the conservation property which associates fluxes with cell *faces*, not nodes, we shall, in future, for all such *three-point* schemes use the notation ϕ_D, ϕ_U and ϕ_{UU} for the Downwind, Upwind and Upwind-Upwind nodes at any particular face.



By fitting a quadratic polynomial to these nodes (see the Examples) the QUICK scheme gives:

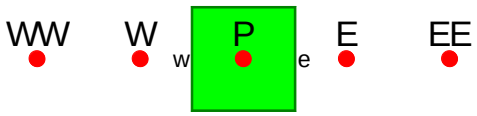
$$\phi_{face} = -\tfrac{1}{8}\phi_{UU} + \tfrac{3}{4}\phi_U + \tfrac{3}{8}\phi_D \qquad (22)$$

For example, if $u > 0$ on the east face then:

$$\phi_e = -\tfrac{1}{8}\phi_W + \tfrac{3}{4}\phi_P + \tfrac{3}{8}\phi_E$$

whereas, if $u < 0$:

$$\phi_e = -\tfrac{1}{8}\phi_{EE} + \tfrac{3}{4}\phi_E + \tfrac{3}{8}\phi_P$$



Assessment

- 3rd-order accurate.
- Conservative by construction.
- Transportive (upwind bias in the selection of the third node and relative weightings).
- Not bounded; (for example, if $u > 0$ then a_E is negative – see the Examples).

Despite boundedness not being guaranteed (which can be a major problem in turbulent flows, where certain turbulence variables are required to be positive – see Section 8) the high-order accuracy of the QUICK scheme make it popular and widely-used.

4.10.4 Flux-Limited Schemes

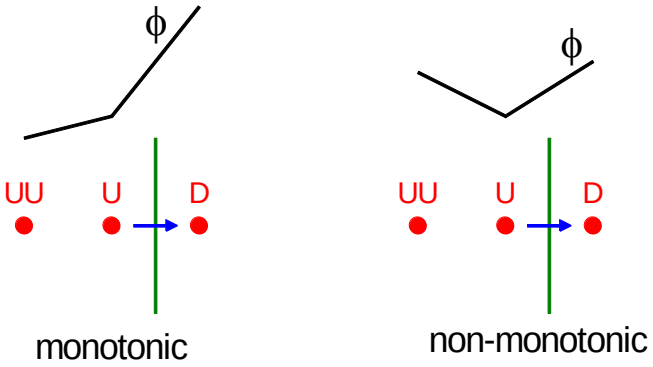
Hitherto we have only seen schemes where the matrix coefficients a_F are constants (i.e. independent of the solution ϕ). The only unconditionally-bounded scheme of this type is first-order upwind differencing. Schemes such as QUICK, which fit a polynomial through several points, are prone to generate cell-face values which lie outside the interpolating values ϕ_D, ϕ_U, ϕ_{UU}. To prevent this, modern schemes employ *solution-dependent limiters*, which enforce boundedness while trying to retain high-order accuracy wherever possible.

For three-point schemes, ϕ is said to be:

monotonic increasing if $\phi_{UU} < \phi_U < \phi_D$,

monotonic decreasing if $\phi_{UU} > \phi_U > \phi_D$.

A necessary condition for boundedness is that the schemes must default to first-order upwinding (i.e. $\phi_{face} = \phi_U$) if ϕ is not locally monotonic (either increasing or decreasing).

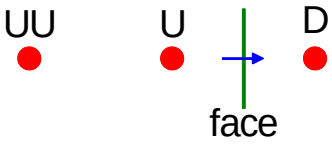


Monotonic variation in ϕ may be gauged by whether the changes in ϕ between successive pairs of nodes have the same sign; i.e.

$$\textit{monotonic} \quad \Leftrightarrow \quad (\phi_D - \phi_U)(\phi_U - \phi_{UU}) > 0$$

Such schemes can then be written (in the notation of Versteeg and Malalasakera, if not in the manner in which they are programmed!) as the sum of upstream value (ϕ_U) and a solution-dependent fraction of the difference between downstream and upstream nodal values:

$$\phi_{face} = \begin{cases} \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U) & \text{if monotone} \\ \phi_U & \text{otherwise} \end{cases}$$



where r is the ratio of successive differences (> 0 where monotonic):

$$r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U}$$

The limiter $\psi(r)$ is given below for various common schemes.

Scheme	$\psi(r)$ (for $r > 0$)	
UMIST	$\min\{2,\ 2r,\ \frac{1}{4}(1+3r),\ \frac{1}{4}(3+r)\}$	Upstream Monotonic Interpolation for Scalar Transport (Lien and Leschziner, 1993). A limited variant of QUICK, this is 3rd-order accurate where monotone.
Harmonic	$\dfrac{2r}{1+r}$	Van Leer (1974). Second-order accurate where monotone.
Min-mod	$\min(r,1)$	Roe (1985)
Van Albada et al.	$\dfrac{r+r^2}{1+r^2}$	Van Albada et al., (1982)

In all cases $\psi(r)$ is taken as 0 if non-monotonic ($r < 0$).

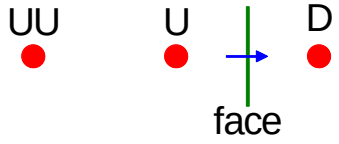
The choice of these examples is (obviously!) parochial. Many other equally-good schemes exist: see Versteeg and Malalasekera (2007). The important points about these schemes are that they are: (a) bounded; and (b) non-linear (i.e. the resulting matrix elements are themselves functions of the solution, ϕ). The last property means that an **iterative** numerical solution is inevitable.

Classroom Example 4.

The Van Leer harmonic advection scheme is given by

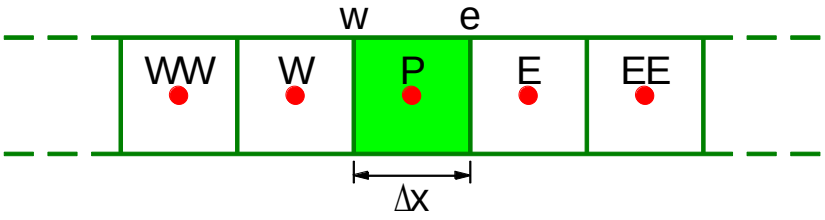
$$\phi_{face} = \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U)$$

$$\psi(r) = \begin{cases} \dfrac{2r}{1+r} & \text{if } r > 0 \\ 0 & \text{otherwise} \end{cases} \qquad r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U}$$



where nodes *UU*, *U* and *D* are determined by flow direction as shown.

A local arrangement of nodes and faces in a 1-d finite-volume mesh with standard geographical notation and uniform mesh spacing Δx is shown below.



The values of a scalar ϕ at these nodes are:

$\phi_{WW} = 2$, $\phi_W = 4$, $\phi_P = 6$, $\phi_E = 7$, $\phi_{EE} = 6$.

Using the Van Leer harmonic advection scheme, calculate the value of ϕ on the west (*w*) and east (*e*) cell faces if the velocity throughout the domain is:

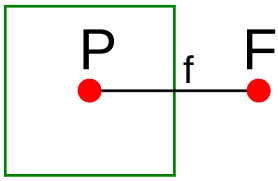
(a) positive;

(b) negative.

4.11 Implementation of Advanced Advection Schemes

The general steady-state scalar-transport equation is

$$\sum_{faces}\Big(\underbrace{\textit{mass flux}\times\phi}_{\textit{advection}}\;\underbrace{-\Gamma\frac{\partial\phi}{\partial n}A}_{\textit{diffusion}}\Big)=\underbrace{S}_{\textit{source}} \tag{23}$$



If C_f and D_f are the outward mass flux and diffusive transport coefficient on cell face f, then, with the standard discretisation for diffusion and sources, (23) becomes

$$\sum[C_f\phi_f+D_f(\phi_P-\phi_F)]=b_P+s_P\phi_P$$

where F denotes an adjacent node, P is the index of the cell-centre node and the sum is over faces of that cell. Since $\Sigma C_f = 0$ by mass conservation, it is convenient to first subtract $(\Sigma C_f)\phi_P$ (which is 0) from the LHS:

$$\sum[C_f(\phi_f-\phi_P)+D_f(\phi_P-\phi_F)]=b_P+s_P\phi_P \tag{24}$$

An advection scheme (Upwind, Central, QUICK, …) is needed to specify the cell-face value ϕ_f. Because many matrix-solution algorithms require positive coefficients and diagonal dominance, it is common practice to separate ϕ_f into the *Upwind* part plus a correction; i.e.

$$\phi_f=\phi_U+(\phi_f-\phi_U)$$

Then, for the advective flux:

$$\begin{aligned}C_f(\phi_f-\phi_P) &= \underbrace{C_f(\phi_U-\phi_P)}_{upwind} &&+\underbrace{C_f(\phi_f-\phi_U)}_{correction}\\ &= \max(-C_f,0)(\phi_P-\phi_F) &&+C_f(\phi_f-\phi_U)\end{aligned} \tag{25}$$

The first part of (25) always gives rise to a positive matrix coefficient:

$$a_F=\max(-C_f,0)+D_f$$

The latter part is transferred to the RHS of the equation as a *deferred correction*; ("deferred" because it is treated explicitly and won't be updated until the next iteration):

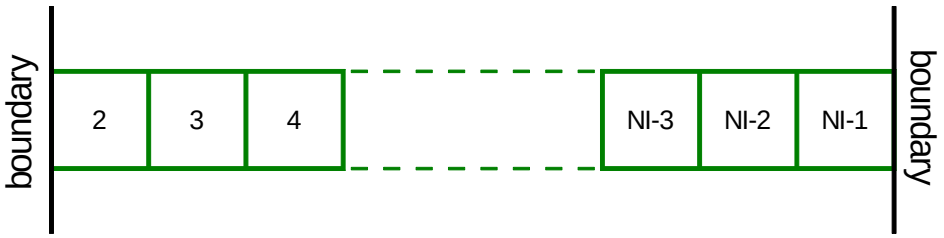
$$\sum_F a_F(\phi_P-\phi_F)=b_P+s_P\phi_P\underbrace{-\sum_{faces}C_f(\phi_f-\phi_U)}_{deferred\ correction}$$

4.12 Implementation of Boundary Conditions

The most common types of boundary condition are:

- value ϕ specified (*Dirichlet* boundary condition);
  e.g. ϕ = constant (e.g. velocity at a wall, or temperature fixed at some surface);
- gradient $\partial\phi/\partial n$ specified (*Neumann* boundary condition).
  e.g. $\partial\phi/\partial n = 0$ on a symmetry plane, or at an outflow boundary.

In the finite-volume method, both types of boundary condition can be implemented by **transferring the boundary flux to the source term**.

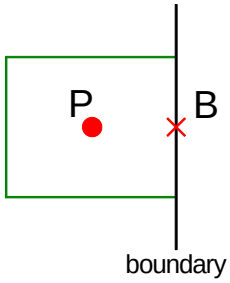


For a cell abutting a boundary:

$$\sum_{\substack{not\\ boundary}} flux + flux_{boundary} = source$$

$$\Downarrow$$

$$a_P\phi_P - \sum_{\substack{not\\ boundary}} a_F\phi_F = b_P - flux_{boundary}$$



There are two modifications:

- the a_F coefficient in the direction of the boundary is set to zero;
- the outward boundary flux is subtracted from the source terms.

If *flux*(ϕ) is specified on the boundary, then this is immediate. If ϕ itself is fixed on the boundary node B then, with Δx the width of the cell and hence ½Δx the distance between boundary and internal nodes,

$$flux(\phi) = -\Gamma A \left.\frac{\partial\phi}{\partial x}\right|_{boundary} \quad \rightarrow \quad -\Gamma A\left(\frac{\phi_B - \phi_P}{\frac{1}{2}\Delta x}\right) = -2D(\phi_B - \phi_P)$$

To subtract this flux from the source term requires a simple change of coefficients:

$$b_P \rightarrow b_P + 2D\phi_B\,, \qquad s_P \rightarrow s_P - 2D \tag{26}$$

You should revisit the classroom examples of earlier sections to see this in action. (Note that the above strictly uses a first-order approximation for the gradient of ϕ at the boundary. A second-order approximation using so-called "ghost nodes" is possible, but the extra complexity is not merited where diffusion is much smaller than advection.)

4.13 Solution of the Algebraic Equations

The discretisation of a single scalar-transport equation over a single control volume produces an algebraic equation of the form

$$a_P \phi_P - \sum a_F \phi_F = b_P$$

where the summation is over adjacent nodes. Combining the equations for all control volumes produces a set of simultaneous equations, i.e. a matrix equation:

$$\mathbf{A\Phi = b}$$

where $\mathbf{\Phi}$ is the vector of nodal values. Matrix **A** is *sparse* (i.e. has only a few non-zero elements). Many algebraic methods can be used to tackle this problem; some of the simpler ones are mentioned below.

4.13.1 Matrix Solution Algorithms

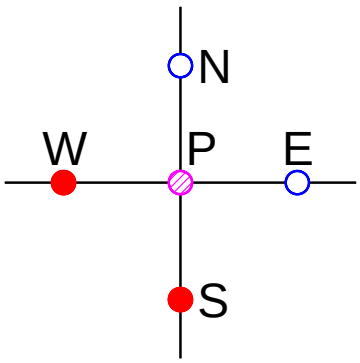
Gaussian Elimination

This is a direct (i.e. non-iterative) method. It consists of a sequence of row operations to obtain zeros below the main diagonal (upper-triangular matrix), followed by back-substitution.

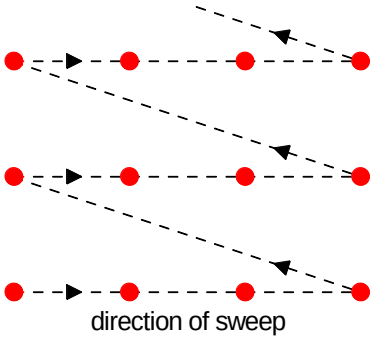
In general, it is inefficient because it tends to fill in sparse matrices (tridiagonal systems are an important exception), whilst for fluid-flow problems the matrix elements vary with the solution, so that an iterative solution is necessary anyway.

Gaussian elimination is OK for small hand calculations, but not recommended for large systems of equations.

Gauss-Seidel



Rearrange the equation for each control volume as an iterative update for each node in terms of the surrounding nodal values. Then repeatedly cycle through the entire set of equations until convergence is achieved.



For example, in 2-d:

$$-a_S \phi_S - a_W \phi_W + a_P \phi_P - a_E \phi_E - a_N \phi_N = b_P$$

is replaced by the iterative set (asterisk * denoting the "most recent" value):

$$\phi_P = \frac{1}{a_P}(b_P + a_S \phi_S^* + a_W \phi_W^* + a_E \phi_E^* + a_N \phi_N^*)$$

Gauss-Seidel is simple to code and is often used for unstructured grids. However, it tends to converge slowly for large matrices and may require substantial under-relaxation (see below).

Classroom Example 5.

(a) Show how Gauss-Seidel can be used to solve the following matrix equation iteratively, and conduct 3 Gauss-Seidel sweeps.

$$\begin{pmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 6 \\ 13 \end{pmatrix}$$

Write a computer program (using any programming language or application of your choice) to solve this problem iteratively.

(b) Now try to do the same for the matrix equation (which actually has the same solution):

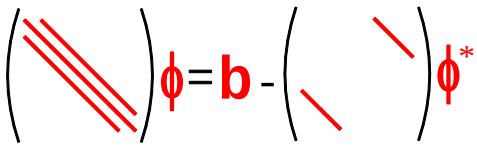
$$\begin{pmatrix} 1 & -4 & 0 & 0 \\ -4 & 1 & -4 & 0 \\ 0 & -4 & 1 & -4 \\ 0 & 0 & -4 & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \begin{pmatrix} -7 \\ -14 \\ -21 \\ -8 \end{pmatrix}$$

Why does Gauss-Seidel not converge in this case?

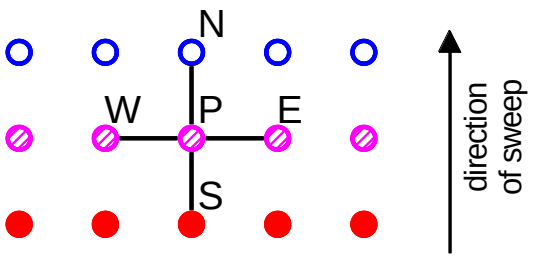
Line-Iterative Procedures ("Line Gauss-Seidel")

Along any one coordinate line, the system is tri-diagonal; e.g. in the *i*-direction:

$$-a_W \phi_W + a_P \phi_P - a_E \phi_E = b_P - \sum_{\substack{not \\ W,E}} a_F \phi_F^*$$



Thus, using the tri-diagonal matrix algorithm **a whole line can be updated at one go** and information can propagate right across the domain in one iteration, rather than (as in Gauss-Seidel) one node at a time. A typical single iteration would consist of applying the update for each successive *i* line, then for each successive *j* line, then for each successive *k* line.



This is probably the most popular method for block-structured grids, and is the basis of most of our in-house research codes. Note, however, that it doesn't work for unstructured grids.

4.13.2 Convergence Criteria

For any individual cell the *residual* is the error (LHS minus RHS) in its discretised equation:

$$res = a_P \phi_P - \sum_F a_F \phi_F - b_P$$

Iteration is stopped when the *total residual error* becomes less than some small, pre-defined tolerance. The total residual error is a suitably-weighted sum over the errors for all cells; e.g.

sum of absolute residuals: $\sum_{cells} |res|$

root-mean-square (rms) error: $\sqrt{\frac{1}{N}\sum_{cells}(res)^2}$

The tolerance is a matter of judgement. To avoid dependence on units it is often set to a small fraction (e.g. 10^{-4}) of the error at the first iteration. However, this has its own deficiencies because it obviously depends on how close the initial values are to the final solution.

4.13.3 Under-Relaxation

If iterative algebraic methods are applied to non-linear, coupled equations then large changes in variables over an iteration may cause instability. To overcome this, *under-relaxation* applies only a fraction of the projected **change** at each iteration. Since incremental changes vanish as the solution is approached, under-relaxation makes no difference to the final result.

The discretised scalar-transport equation for one cell:

$$a_P \phi_P - \sum a_F \phi_F = b_P$$

can be rearranged as

$$\phi_P = \frac{\sum a_F \phi_F + b_P}{a_P}$$

This can be written as the sum of the previous iteration value plus the **change** in ϕ:

$$\phi_P = \phi_P^{prev} + \left(\frac{\sum a_F \phi_F + b_P}{a_P} - \phi_P^{prev} \right)$$

If, instead, only a fraction α of the projected change in ϕ is applied then

$$\phi_P = \phi_P^{prev} + \alpha \left(\frac{\sum a_F \phi_F + b_P}{a_P} - \phi_P^{prev} \right)$$

α is called an *under-relaxation factor*.

The equation can then be rearranged back as

$$a_P \phi_P - \sum \alpha a_F \phi_F = \alpha b_P + (1-\alpha) a_P \phi_P^{prev}$$

Hence, under-relaxation is easily implemented by a simple change of coefficients:

$$\begin{aligned} a_F &\rightarrow \alpha a_F \\ b_P &\rightarrow \alpha b_P + (1-\alpha) a_P \phi_P^{prev} \end{aligned} \qquad (27)$$

This makes the equations more diagonally dominant (a_F smaller), improving stability.

Note that, whilst some under-relaxation is necessary to prevent divergence, too much under-relaxation will slow down convergence.

Summary

- The generic scalar-transport equation for a particular control volume has the form
  rate of change + *net outward flux* = *source*

- *flux* ≡ rate of transport through a surface and consists of:
  advection: transport with the flow (other authors prefer *convection*);
  diffusion: net transport by random molecular or turbulent fluctuations.

- Discretisation of the (steady) scalar-transport equation yields an equation of form
  $$a_P\phi_P - \sum_F a_F\phi_F = b_P$$
  for each control volume, where the summation is over adjacent nodes.

- The collection of these simultaneous equations on a structured mesh yields a matrix equation with *limited bandwidth* (i.e. few non-zero diagonals), typically solved by iterative methods such as Gauss-Seidel or line-Gauss-Seidel.

- Source terms are linearised as
  $$b_P + s_P\phi_P \,, \quad s_P \le 0 \,.$$

- Diffusive fluxes are usually discretised by central differencing; e.g.
  $$-\Gamma\left.\frac{\partial\phi}{\partial x}\right|_e A \;\rightarrow\; -\frac{\Gamma A}{\Delta x}(\phi_E - \phi_P)$$

- *Advection schemes* are means of approximating ϕ on cell faces in order to compute advective fluxes. They include upwind, central, exponential, hybrid, QUICK, and various flux-limited schemes.

- General desirable properties for a numerical scheme:
  consistency
  conservativeness
  boundedness
  stability
  transportiveness
  accuracy / *high order*

- Boundedness and stability impose certain constraints on the discretisation:
  $a_F \ge 0$ for all F (“positive coefficients”)
  $s_P \le 0$ (“negative feedback in the source term”)
  $a_P = \sum_F a_F - s_P$ (“sum of the neighbouring coefficients”)

- To ensure positive coefficients (and implement non-linear schemes), advective fluxes are often decomposed into
  “*Upwind*” + “*deferred correction*”
  with the latter being transferred to the source term and treated *explicitly* (i.e. fixed for this iteration).

- Boundary conditions can be implemented by transferring boundary fluxes to the source terms.

- Under-relaxation is usually required to solve coupled and/or non-linear equations.

References

Leonard, B.P., 1979, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Meth. Appl. Mech. Engng, 19, 59-98.

Lien, F.S. and Leschziner, M.A., 1993, Upstream monotonic interpolation for scalar transport with application to complex turbulent flows, Int. J. Numer. Methods Fluids, 19, 293-312.

Patankar, S.V., 1980, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill.

Roe, P.L., 1985, Some contributions to the modelling of discontinuous flows, Lectures in Applied Mechanics, 22, Springer-Verlag, 163-193.

Spalding, D.B., 1972, A novel finite-difference formulation for different expressions involving both first and second derivatives, Int. J. Numer. Meth. Engng, 4, 551-559.

Sweby, P.K., High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal., 21, 995-1011.

Van Albada, G.D., Van Leer, B. and Roberts, W.W., 1982, A comparative study of computational methods in cosmic gas dynamics, Astron. Astrophys., 108, 76-84.

Van Leer, B., 1974, Towards the ultimate conservative difference scheme II: monotonicity and conservation combined in a second-order scheme, J. Comput. Phys., 14, 361-370.

Versteeg, H.K. and Malalasekera, W., 2007, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, 2nd Edition, Pearson.

Appendix: Tri-Diagonal Matrix Algorithm

System of equations (note the signs!):

$$-a_i\phi_{i-1} + b_i\phi_i - c_i\phi_{i+1} = d_i \ , \quad i = 1, \ldots, N$$

Either ϕ_0 and ϕ_{N+1} are given, or $a_1 = c_N = 0$.

$$\begin{pmatrix} b_1 & -c_1 & 0 & \ddots & 0 \\ \ddots & \ddots & \ddots & 0 & \ddots \\ 0 & -a_i & b_i & -c_i & 0 \\ \ddots & 0 & \ddots & \ddots & \ddots \\ 0 & \ddots & 0 & -a_N & b_N \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_i \\ \vdots \\ \phi_N \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_i \\ \vdots \\ d_N \end{pmatrix}$$

Forward pass:

$$P_0 = 0, \qquad Q_0 = \phi_0$$

$$P_i = \frac{c_i}{b_i - a_i P_{i-1}}, \qquad Q_i = \frac{d_i + a_i Q_{i-1}}{b_i - a_i P_{i-1}}, \qquad i = 1, \ldots, N$$

Backward pass:

$$\phi_i = P_i\phi_{i+1} + Q_i, \qquad i = N, \ldots, 1$$

The method is guaranteed to converge if the coefficients are non-negative and diagonally dominant:

$a_i \geq 0, \quad c_i \geq 0, \quad a_i + c_i \leq b_i$ for all i

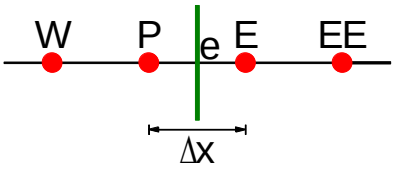
except in the degenerate case where zero a_i or c_i allow successive rows to be multiples of each other and the matrix is consequently singular (rank < N).

Exercise: Code your own tridiagonal solver as a subroutine and test it on one of the classroom examples.

Examples

Q1.

Consider the uniform, one-dimensional arrangement of nodes shown right. Face *e* lies half way between *P* and *E* nodes.



(a) Show that the central-differencing schemes

$$\phi_e \approx \tfrac{1}{2}(\phi_P + \phi_E)$$

$$\left.\frac{\mathrm{d}\phi}{\mathrm{d}x}\right|_e \approx \frac{\phi_E - \phi_P}{\Delta x}$$

are second-order accurate approximations for ϕ_e and $(\mathrm{d}\phi/\mathrm{d}x)_e$ respectively.

(b) Making use of the *W* and *EE* nodes also, find symmetric fourth-order-accurate approximations for ϕ_e and $(\mathrm{d}\phi/\mathrm{d}x)_e$.

Q2.

(a) By fitting a quadratic function $\phi(x)$ to values ϕ_W, ϕ_P and ϕ_E at $x = -\tfrac{3}{2}\Delta x$, $x = -\tfrac{1}{2}\Delta x$ and $x = \tfrac{1}{2}\Delta x$, and taking its value at $x = 0$, deduce the formula for the QUICK advection scheme (in the case of positive *x*-velocity):

$$\phi_e = -\tfrac{1}{8}\phi_W + \tfrac{3}{4}\phi_P + \tfrac{3}{8}\phi_E .$$

(b) By expanding ϕ_W, ϕ_P and ϕ_E as Taylor series in terms of ϕ and its derivatives at cell face *e*, show that requiring a constant-coefficient combination of these to be third-order accurate leads to the same expression as in part (a).

Q3.

If the continuity (mass-conservation) equation were to be regarded as a special case of the general scalar-transport equation, what would be the expressions for ϕ, Γ and *S*?

Q4. (From Patankar, 1980)

The source term for a dependent variable ϕ is given by $2 - 3|\phi_P|\phi_P$. If this term is to be linearised as $b_P + s_P\phi_P$, comment on the following practices (ϕ_P^* denotes the value from the previous iteration):

(a) $b_P = 2 - 3|\phi_P^*|\phi_P^*$, $s_P = 0$

(b) $b_P = 2 - 3\phi_P^{*2}$, $s_P = 0$

(c) $b_P = 2 - 4|\phi_P^*|\phi_P^*$, $s_P = |\phi_P^*|$

(d) $b_P = 2$, $s_P = -3|\phi_P^*|$

Q5.

Consider the advection-diffusion equation with no sources (and unit cross-sectional area):

$$\frac{\mathrm{d}}{\mathrm{d}x}(\rho u\phi - \Gamma\frac{\mathrm{d}\phi}{\mathrm{d}x}) = 0$$

where ρ, u, Γ are constants. If $\phi = \phi_P$ at $x = 0$ and $\phi = \phi_E$ at $x = \Delta x$, find the value of the flux

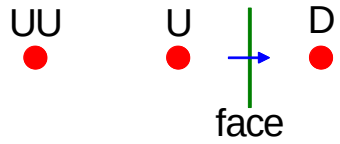
$$\rho u\phi - \Gamma\frac{\mathrm{d}\phi}{\mathrm{d}x}$$

at any point between $x = 0$ and Δx. (This is the basis of the exponential differencing scheme).

Q6.

The QUICK scheme fits a quadratic function to three nodal values to estimate the value of a scalar at a cell face, according to

$$\phi_{face} = -\tfrac{1}{8}\phi_{UU} + \tfrac{3}{4}\phi_U + \tfrac{3}{8}\phi_D$$



(a) For a two-dimensional problem, write down expressions for

$$\phi_e,\ \phi_w,\ \phi_n,\ \phi_s$$

in terms of the values at neighbouring nodes, assuming that velocity components u and v are known, constant and positive.

(b) Neglecting diffusion, and assuming a uniform source s per unit volume, derive an algebraic discretisation of the conservation equation

$$\sum(outward\ \ flux) = source$$

in the form

$$a_P\phi_P - \sum a_F\phi_F = b_P$$

where the sum is over local nodes. (Assume the cell to be Cartesian, with unit depth in the z direction, face areas A_e, A_w, A_n, A_s and volume V).

(c) How do the expressions for ϕ_w and ϕ_e change if $u < 0$.

(d) Which of the following properties does the QUICK scheme satisfy:

transportiveness;

boundedness?

(e) Split the QUICK expression for ϕ_{face} in the form

"Upwind differencing" + "deferred correction"

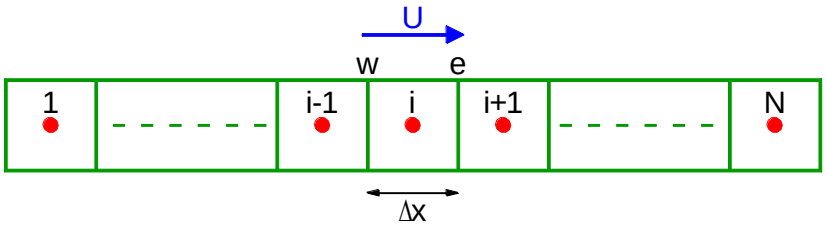
Why is this decomposition used?

Q7. (Exam 2013)

The steady-state advection and diffusion of a scalar with concentration ϕ along a 1-d pipe may be described by the equation

$$\frac{\mathrm{d}}{\mathrm{d}x}(\rho u\phi - \Gamma\frac{\mathrm{d}\phi}{\mathrm{d}x}) = s \qquad \textbf{(*)}$$

where ρ is density, u is velocity, Γ is diffusivity and s is source density. This is to be solved numerically by a finite-volume method on the uniform mesh of N cells shown below.



(a) By integrating equation (*) over a general one-dimensional cell centred on node i, write the 1-d advection-diffusion equation in finite-volume form.

(b) Write down numerical approximations for ϕ and $\mathrm{d}\phi/\mathrm{d}x$ on the 'east' and 'west' faces of internal cell i, using upwind differencing for advection and centred differencing for diffusion. (Assume that flow is from left to right.)

(c) Define the *order* of a numerical scheme and state, without proof, the order of your numerical schemes for advection and diffusion in part (b).

(d) Use your approximations from part (b), together with appropriate boundary treatment, to solve equation (*) numerically on a 4-cell mesh in the case

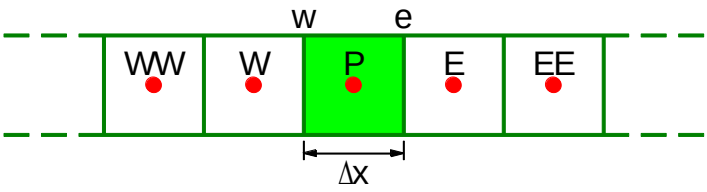
$\rho u = 1, \quad \Gamma = 0.5, \quad s = 2$ (in appropriate units),

with end boundary conditions

$\phi = 0$ at $x = 0$, $\quad \mathrm{d}\phi/\mathrm{d}x = 0$ at $x = 2$.

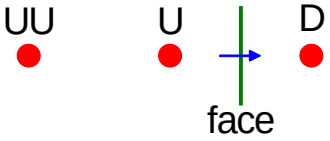
(e) Solve equation (*) exactly in this case and compare with your numerical solution at the nodal points.

Q8. (Exam 2011 – part)
A local arrangement of nodes and faces in a 1-d finite-volume mesh with standard notation and uniform mesh spacing Δx is shown right.



A general expression for ϕ on a cell face with a 3-point advection scheme is

$$\phi_{face} = \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U), \qquad \text{where} \qquad r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U}$$



where subscripts *D*, *U* and *UU* denote "downstream", "upstream" and "double-upstream" nodes, respectively.

(a) Write down the (constant) values of ψ for first-order upwind and central advection schemes.

(b) The Min-Mod flux limiter is

$$\psi(r) = \max(0, \min(1, r))$$

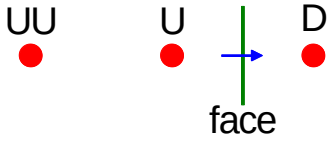
In a particular case, ϕ takes the following values at local nodes:

$$\phi_{WW} = 3, \quad \phi_W = 3, \quad \phi_P = 4, \quad \phi_E = 7, \quad \phi_{EE} = 5,$$

Find the values of ϕ on face *e* if the velocity on that face is (i) positive; (ii) negative.

Q9.
The general three-point scheme for the cell-face value of a transported scalar ϕ is

$$\phi_{face} = \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U), \qquad r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U}$$



where upstream (*U*) and downstream (*D*) nodes are defined by flow direction.

(a) The *linear upwind differencing* (LUD) and QUICK schemes are:

LUD: $\phi_{face} = \tfrac{3}{2}\phi_U - \tfrac{1}{2}\phi_{UU}$

QUICK: $\phi_{face} = -\tfrac{1}{8}\phi_{UU} + \tfrac{3}{4}\phi_U + \tfrac{3}{8}\phi_D$

Identify the functional form of $\psi(r)$ for each of these schemes.

(b) Sweby's conditions for *total-variation-diminishing* (TVD) schemes are that

$\psi = 0$ if $r \le 0$

$\psi \le \min(2r, 2)$ if $r \ge 0$

Show that LUD and QUICK both contravene these conditions for some $r \ge 0$.

(c) The Van Albada scheme has

$$\psi(r) = \begin{cases} 0, & \text{if } r \le 0 \\ \dfrac{r + r^2}{1 + r^2}, & \text{if } r \ge 0 \end{cases}$$

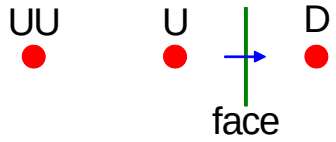
Show that this satisfies Sweby's criteria and also the symmetry property

$$\frac{\psi(r)}{r} = \psi\left(\frac{1}{r}\right) \qquad \text{for } r > 0$$

Q10. (Exam 2012)

The QUICK advection scheme may be written

$$\phi_{face} = -\frac{1}{8}\phi_{UU} + \frac{3}{4}\phi_U + \frac{3}{8}\phi_D ,$$

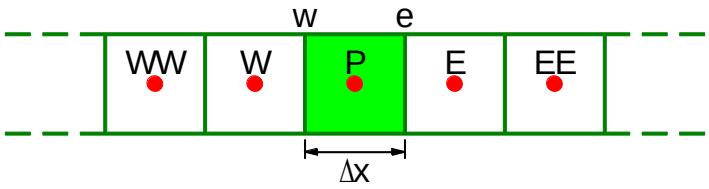


where downwind (D) and upwind (U, UU) nodes relative to a cell face and direction of flow are defined right.

(a) Define the terms *transportive* and *bounded* when applied to an advection scheme and state, without proof, whether QUICK is transportive and/or bounded.

(b) Prove that QUICK is 3rd-order accurate (on a uniform mesh).

(c) The values of ϕ at successive nodes of a cell-centred structured mesh are shown in the figure below. The flow is everywhere from left to right. Using the QUICK scheme find the values of ϕ on cell faces marked *w* and *e*.



$\phi_{WW} = 1, \quad \phi_W = 2, \quad \phi_P = 5, \quad \phi_E = 3, \quad \phi_{EE} = 2$

(d) The UMIST scheme is a variant of the QUICK scheme defined by

$$\phi_{face} = \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U) ,$$

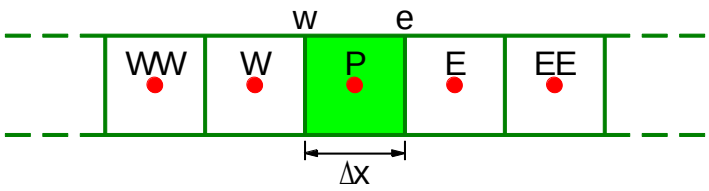
where

$$\psi(r) = \max[0, \min\{2, 2r, \tfrac{1}{4}(1+3r), \tfrac{1}{4}(3+r)\}], \qquad r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U} .$$

Sketch a graph of ψ against *r*, indicating key points. Show that the UMIST scheme reduces to the QUICK scheme when $1 \le r \le 5$.

(e) Find the values of ϕ on faces *w* and *e* using the UMIST scheme.

Q11. (Exam 2008)
The local arrangement of nodes and faces in a 1-d finite-volume mesh with standard geographical notation and uniform mesh spacing Δx is shown in the figure below.



(a) The 1-d advection-diffusion equation for a transported variable ϕ is

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\rho u\phi - \Gamma \frac{\mathrm{d}\phi}{\mathrm{d}x} \right) = s ,$$

where ρ is density, u is velocity, Γ is diffusivity, s is source density. Write this in a corresponding integral form for the shaded cell.

(b) Write finite-difference expressions for $\mathrm{d}\phi/\mathrm{d}x$ on w and e faces.

(c) Write expressions for ϕ on w and e faces of the shaded cell if u is positive for:
(i) first-order upwind
(ii) central
advection schemes.

(d) Define the term *bounded* when applied to flux-differencing schemes and deduce a condition on ρ, u, Γ and Δx for the central scheme to be bounded.

A general expression for ϕ_e when u is positive is

$$\phi_e = \phi_P + \tfrac{1}{2}\psi(r)(\phi_E - \phi_P) \qquad \text{where} \qquad r = \frac{\phi_P - \phi_W}{\phi_E - \phi_P}$$

(e) Write the (constant) values of ψ for
(i) first-order upwind
(ii) central
advection schemes.

(f) The Van Leer flux limiter is

$$\psi(r) = \begin{cases} \dfrac{2r}{1+r} & \text{if } r > 0 \\ 0 & \text{otherwise} \end{cases}$$

In a particular case, $\rho = 1$, $u = 1$, $\Gamma = 0.02$ and $\Delta x = 0.1$ in consistent units. In a solution with the Van Leer advection scheme, ϕ takes the following values:

$$\phi_{WW} = 1, \qquad \phi_W = 2, \qquad \phi_P = 4, \qquad \phi_E = 2, \qquad \phi_{EE} = 3,$$

Find the values of ϕ and $\mathrm{d}\phi/\mathrm{d}x$ on w and e faces, and hence the mean source density s for cell P.

Q12.

(a) Define the following properties in the context of a flux-differencing scheme in a finite-volume discretisation:

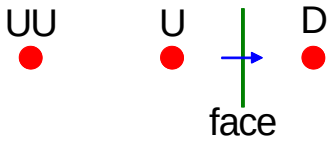
transportive;

bounded.

(b) What is meant by the statement that an advection scheme is *Total-Variation-Diminishing* (TVD)?

The general three-point scheme for the cell-face value of a transported scalar ϕ is

$$\phi_{face} = \phi_U + \tfrac{1}{2}\psi(r)(\phi_D - \phi_U), \qquad r = \frac{\phi_U - \phi_{UU}}{\phi_D - \phi_U},$$



where nodes *UU*, *U* and *D* are determined by flow direction as shown.

Conditions for an advection scheme to be TVD are:

$\psi = 0$ if $r \le 0$;

$\psi \le \min(2r, 2)$ if $r \ge 0$.

A scheme is at least 2nd-order accurate if $\psi(r)$ passes through the point (1,1) and 3rd-order accurate if it does so with slope ¼.

(c) For each of the following advection schemes use the conditions above to determine:

(i) whether the scheme is TVD;

(ii) the order of the scheme.

Upwind: $\psi(r) = 0$

Central: $\psi(r) = 1$

QUICK: $\psi(r) = \tfrac{1}{4}(3 + r)$

Min-mod: $\psi(r) = \max(0, \min(r, 1))$

Van Leer: $\psi(r) = \begin{cases} 2r/(1+r) & \text{if } r > 0 \\ 0 & \textit{otherwise} \end{cases}$

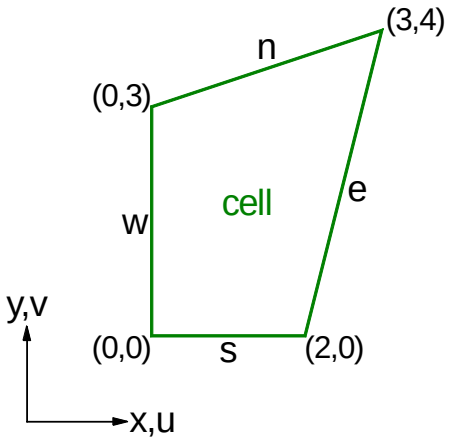
Q13. (Exam 2010 – part)

The figure below and the accompanying table show the velocity components and values of a conserved scalar ϕ on the faces of a cell in a 2-d finite-volume simulation of steady, incompressible flow. Coordinates of cell vertices are given in the figure. Assume units such that the density $\rho = 1$.

(a) Calculate the mass flux (per unit span) *out of* each of the *w*, *n*, *s* faces.

(b) Use continuity to find the velocity component *u* on the east (*e*) face.

(c) Assuming no source term or diffusion, calculate ϕ on the east face.



face	velocity *u*	velocity *v*	scalar ϕ
e	?	0	?
n	6	3	4
w	5	2	2
s	5	2	3.5

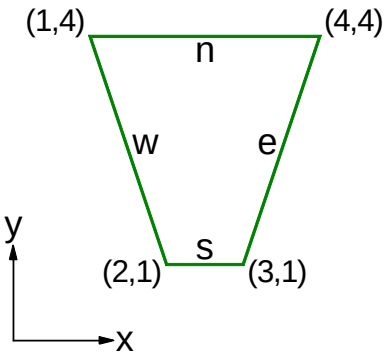
Q14. (*** *Advanced* ***)

The figure below and the accompanying table show the velocity components and values of a conserved scalar ϕ on the faces of a cell in a 2-d finite-volume simulation of steady, incompressible, irrotational flow. The coordinates of the cell vertices are given in the figure.

(a) Calculate the volume flux (per unit span) *out of* each of the *n*, *w*, *s* faces.

(b) Use the incompressibility and irrotationality conditions to find the velocity components *u* and *v* on the east (*e*) face.

(c) Assuming no source or diffusion of the scalar, calculate ϕ on the east face.

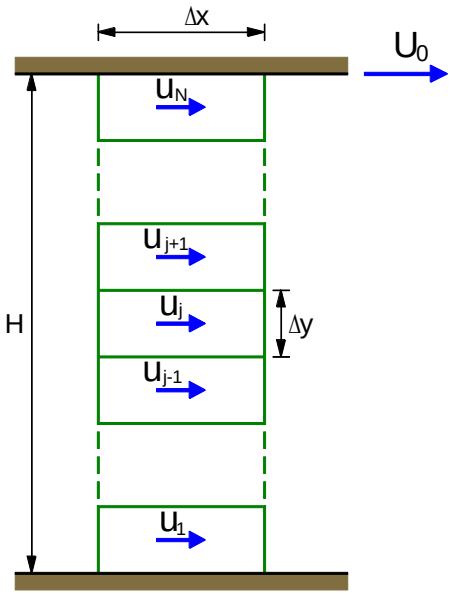


face	velocity *u*	velocity *v*	scalar ϕ
e	?	?	?
n	9	–3	0
w	4	-2	6
s	3	3	2

Q15.

A 2-d finite-volume calculation is to be undertaken for fully-developed, laminar flow between plane, parallel walls. The upper wall is moving at speed U_0, whilst the lower wall is stationary. A streamwise pressure gradient $\mathrm{d}p/\mathrm{d}x = -G$ is imposed.

The depth of the channel, H, is divided into N equally-sized cells of dimension $\Delta x \times \Delta y \times 1$ as shown, with the velocity u stored at the centre of each cell.



(a) What are the boundary conditions on upper and lower walls?

(b) What is the net pressure force on a single cell?

(c) Write down a second-order approximation for the shear stress τ on the upper face of the j^{th} internal cell, in terms of the velocities u_j and u_{j+1}.

(d) Write down similar approximations for τ on upper and lower walls.

(e) By balancing pressure and viscous forces set up the finite-volume equations for velocity. (Separate equations are required for internal and boundary cells).

(f) Solve your equations for the nodal velocities in the case

$$N = 4, \qquad \frac{GH^2}{\mu U_0} = 2$$

leaving your answers in terms of U_0.

(g) Using your answer to part (f), find the volume flow rate per unit span (leaving your answer as a multiple of $U_0 H$).

(h) Solve the Navier-Stokes equation analytically for this case and compare with your answers in parts (f) and (g).

Q16. (Exam 2010 – part)
Use the Gauss-Seidel iterative method to solve the system of equations

$$\begin{pmatrix} 5 & -1 & 0 & -1 \\ -3 & 8 & -1 & 0 \\ 0 & -4 & 10 & -1 \\ -2 & 0 & -3 & 10 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \begin{pmatrix} 9.5 \\ -3.5 \\ 30.5 \\ -37 \end{pmatrix},$$

giving solutions for *A, B, C,* and *D* correct to 2 decimal places.

Q17. (Exam 2013 – part)
The Gauss-Seidel iterative method may be used to solve systems of coupled nonlinear equations. Use this method to find a solution of the following set of equations for *A, B* and *C*. (You should rearrange judiciously to ensure stability.) Give your final answers to 2 significant figures.

$$\begin{aligned} 5A - B &= 1 - 9|A|A \\ -2A + 5B - C &= 2 - 9|B|B \\ -2B + 5C &= 3 - 9|C|C \end{aligned}$$

Q18.
The discretisation of a 1-d scalar-transport equation results in a set of simultaneous equations for the nodal values $\{\phi_i\}$ of the form

$$-2\phi_{i-1} + 6\phi_i - \phi_{i+1} = 2 - 2|\phi_i|\phi_i \ , \qquad i = 1,2,\ldots,N$$

where ϕ_0 and ϕ_{N+1} are given. Use the Gauss-Seidel method to solve this iteratively for the case $N = 3$ with boundary conditions $\phi_0 = \phi_4 = 0$.

Q19.
The linear system of equations

$$-a_i\phi_{i-1} + b_i\phi_i - c_i\phi_{i+1} = d_i \ , \qquad i = 1,2,\ldots,N$$

(where a_i, b_i, c_i and d_i are constants, and ϕ_0 and ϕ_{N+1} are fixed) can be solved by the tri-diagonal matrix algorithm as

$$\phi_i = P_i\phi_{i+1} + Q_i \ , \qquad i = N,\ldots,1$$

where $\{P_i\}$ and $\{Q_i\}$are determined from an initial forward pass. Show that

$$P_i = \frac{c_i}{b_i - a_i P_{i-1}} \ , \qquad i = 1,\ldots,N; \qquad P_0 = 0$$

and derive a similar recurrence relation for the $\{Q_i\}$.

(You may assume here that $a_i > 0$, $c_i > 0$, $a_i + c_i \le b_i$ for all i.)