
9.1 Stages of a CFD Analysis

Pre-Processing

The pre-processing stage consists of:

- determining the equations to be solved (flow physics);
- specifying boundary conditions;
- generating a mesh.

It depends upon:

- the desired outputs of the simulation (e.g. force coefficients, heat transfer, ...);
- the capabilities of the solver.

Solving

With commercial CFD packages the solver is often operated as a "black box". Nevertheless, user intervention is necessary – to set under-relaxation factors and input parameters, for example – whilst an understanding of discretisation methods and internal data structures is required in order to supply mesh data in an appropriate format and to analyse output.

Post-Processing

The raw output of the solver is the set of values of each field variable (u, v, w, p, …) at each node. This huge dataset must be reduced and/or manipulated to obtain the desired outputs. For example, a subset of surface pressures, shear stresses and cell-face areas is required in order to compute a drag coefficient.

Commercial packages routinely provide:

- plotting tools to visualise the flow;
- analysis tools to extract and manipulate data.

Most commercial CFD vendors supplement their flow solvers with grid-generation and flow-visualisation tools, as well as graphical user interfaces (GUIs) to simplify the setting-up of a CFD analysis. Some of the more popular commercial CFD packages are given below.

Vendor	Code(s)	Web address (liable to change!)
CD-adapco	STAR CD STAR CCM+	http://www.cd-adapco.com/
Ansys	FLUENT CFX	http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics
Flow Science	FLOW3D	http://www.flow3d.com/
EXA	PowerFLOW	http://www.exa.com/powerflow.html

A popular non-commercial CFD solver is OpenFOAM (http://www.openfoam.com/). An excellent web portal for all things CFD is http://www.cfd-online.com/.

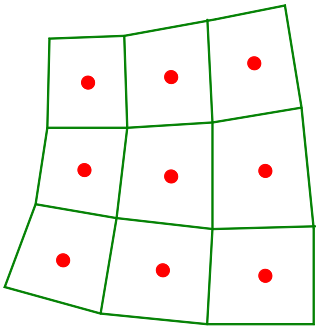
9.2 The Computational Mesh

9.2.1 Mesh Structure

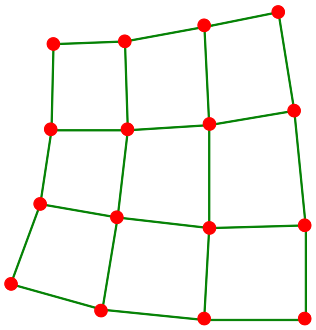
The purpose of the grid generator is to decompose the flow domain into control volumes.

The primary outputs are:
- cell vertices;
- connectivity information.

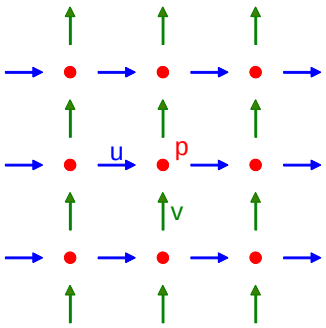
Various arrangements of nodes and vertices may be employed.



cell-centred storage

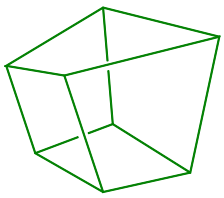


cell-vertex storage

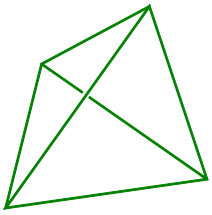


staggered velocity mesh

The shapes of control volumes depend on the capabilities of the solver. Structured-grid codes use quadrilaterals in 2-d and hexahedra in 3-d flows. Unstructured-grid solvers often use triangles (2-d) or tetrahedra (3-d), but newer codes can use arbitrary polyhedra.



hexahedron



tetrahedron

In all cases it is necessary to specify *connectivity*: that is, which cells are adjacent to each other, and which faces and vertices they share. For structured grids with (*i*,*j*,*k*) numbering this is straightforward, but for unstructured grids advanced data structures are required.

9.2.2 Areas and Volumes

To calculate a flux requires the *vector area* **A** of a cell face. This has magnitude equal to the area and direction normal to the surface. Its orientation will also distinguish in or out.

$$\textit{mass flux} = \rho \mathbf{u} \bullet \mathbf{A}$$

$$\textit{diffusive flux} = -\Gamma \nabla\phi \bullet \mathbf{A}$$

To find the total amount of some property in a cell requires its volume V:

$$\textit{amount} = \rho V \phi$$

Vector techniques are very useful for computing areas and volumes of non-Cartesian cells.

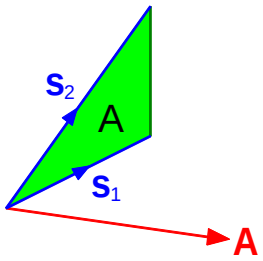
Areas

Triangles.

The vector area of a triangle with side vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ is the cross product

$$\mathbf{A} = \tfrac{1}{2}\mathbf{s}_1 \wedge \mathbf{s}_2$$

The orientation depends on the order of vectors in the cross product.

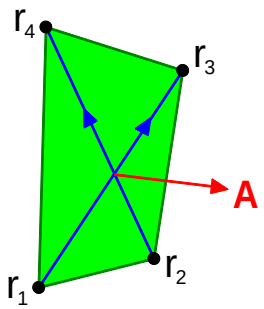


Quadrilaterals

4 (or more) points do not, in general, lie in a plane. However, since the sum of the outward vector areas from any closed surface is zero, i.e.

$$\oint_{\partial V} \mathrm{d}\mathbf{A} = 0 \qquad \text{or} \qquad \sum_{faces} \mathbf{A}_f = 0,$$

any surface spanning the same set of points has the same vector area. Hence, it can be divided into planar triangles and their vector areas summed.



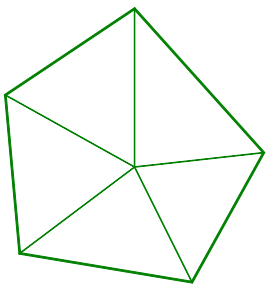
By adding vector areas of, e.g., triangles 123 and 134, the vector area of *any* surface spanned by these points and bounded by the side vectors is found (see the Examples) to be half the cross product of the diagonals:

$$\mathbf{A} = \tfrac{1}{2}\mathbf{d}_{13} \wedge \mathbf{d}_{24} = \tfrac{1}{2}(\mathbf{r}_3 - \mathbf{r}_1) \wedge (\mathbf{r}_4 - \mathbf{r}_2)$$

The order of points determines the orientation (outward or inward) of the area vector.

General polygons

The vector area of an arbitrary polygonal face may be found by breaking it up into triangles and summing the individual vector areas. The overall vector area is independent of how it is decomposed.



Volumes

If $\mathbf{r} \equiv (x, y, z)$ is the position vector, then $\nabla \bullet \mathbf{r} = \frac{\partial x}{\partial x} + \frac{\partial y}{\partial y} + \frac{\partial z}{\partial z} = 3$. Integrating over an arbitrary control volume:

$$\int_V \nabla \bullet \mathbf{r} \; \mathrm{d}V = 3V$$

Dividing by 3 and using the divergence theorem (see Appendix) gives the volume of a cell:

$$V = \frac{1}{3} \oint_{\partial V} \mathbf{r} \bullet \mathrm{d}\mathbf{A}$$

If the cell has **plane** faces this can be evaluated as

$$V = \frac{1}{3} \sum_{faces} \mathbf{r}_f \bullet \mathbf{A}_f$$

where $\mathbf{A}_f$ is a vector area of face f and $\mathbf{r}_f$ is the position vector of any point on that face. (It doesn't matter which point is chosen since, for any other vector $\mathbf{r}$ in that face,

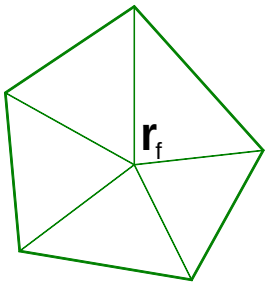
$$\mathbf{r} \bullet \mathbf{A}_f - \mathbf{r}_f \bullet \mathbf{A}_f = (\mathbf{r} - \mathbf{r}_f) \bullet \mathbf{A}_f \quad = 0$$

The last scalar product vanishes because $\mathbf{r} - \mathbf{r}_f$ lies in the plane and $\mathbf{A}_f$ is perpendicular to it.)

If the cell faces are not planar, then the volume depends on how each face is broken down into triangles. A consistent approach is to connect the vertices surrounding a particular face with a common central reference point: for example the average of the vertices

$$\mathbf{r}_f = \frac{1}{N} \sum_{vertices} \mathbf{r}_i$$

or (with a little more work), the centroid of a face.

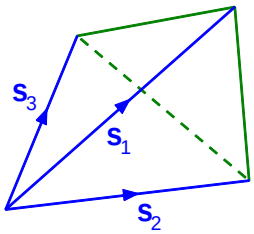


Volumes for the commonest shape of cell include the following.

Tetrahedra

The volume of a tetrahedron formed from side vectors $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ (taken in a right-handed sense) is

$$V = \tfrac{1}{6} \mathbf{s}_1 \bullet \mathbf{s}_2 \wedge \mathbf{s}_3$$



Hexahedra

For hexahedra, on each face the reference point and vector area are:

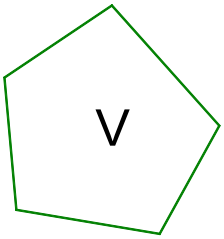
$$\mathbf{r}_f = \tfrac{1}{4}(\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3 + \mathbf{r}_4)$$

$$\mathbf{A}_f = \tfrac{1}{2}(\mathbf{r}_1 - \mathbf{r}_3) \wedge (\mathbf{r}_2 - \mathbf{r}_4)$$

To obtain an **outward**-directed face vector, the points with position vectors $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, $\mathbf{r}_4$ should be in clockwise order when viewed along the outward normal (right-hand screw). Volumes are then calculated by the formula for arbitrary polyhedral:

$$V = \frac{1}{3} \sum_{faces} \mathbf{r}_f \bullet \mathbf{A}_f$$

2-d Cases

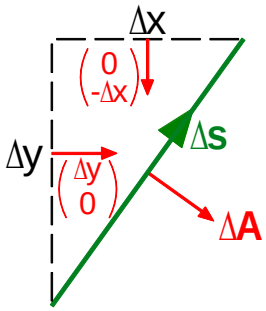


In 2-d cases consider 3-d cells to be of unit depth. The "volume" of the cell is then numerically equal to its planar area.

The side "area" vectors are most easily obtained from their Cartesian projections:

$$\Delta\mathbf{A} = \begin{pmatrix} \Delta y \\ -\Delta x \end{pmatrix}$$

where, to obtain an outward-directed vector area, the edge increments (Δx, Δy) are taken anticlockwise around the cell.



9.2.3 Cell-Averaged Derivatives

The average value of any function f over a cell is defined by

$$f_{av} \equiv \frac{1}{V}\int_V f \;\; \mathrm{d}V$$

By applying the divergence theorem to $\phi\mathbf{e}_x$, where $\mathbf{e}_x$ is the unit vector in the x direction, the volume-averaged x-derivative of a scalar field ϕ is

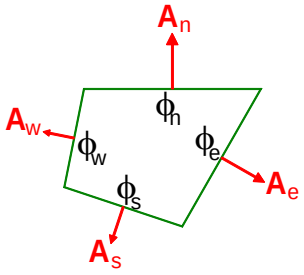
$$\left(\frac{\partial\phi}{\partial x}\right)_{av} \equiv \frac{1}{V}\int_V \frac{\partial\phi}{\partial x}\,\mathrm{d}V \quad = \frac{1}{V}\int_V \nabla\bullet(\phi\mathbf{e}_x)\;\; \mathrm{d}V \quad = \frac{1}{V}\oint_{\partial V} \phi\mathbf{e}_x\bullet\mathrm{d}\mathbf{A} \quad = \frac{1}{V}\oint_{\partial V} \phi\;\; \mathrm{d}A_x$$

Considering all three derivatives, and assuming polyhedral cells:

$$\begin{pmatrix} \partial\phi/\partial x \\ \partial\phi/\partial y \\ \partial\phi/\partial z \end{pmatrix}_{av} = \frac{1}{V}\sum_{faces}\phi_f\mathbf{A}_f$$

e.g. for hexahedra:

$$\begin{pmatrix} \partial\phi/\partial x \\ \partial\phi/\partial y \\ \partial\phi/\partial z \end{pmatrix}_{av} = \frac{1}{V}(\phi_w\mathbf{A}_w + \phi_e\mathbf{A}_e + \phi_s\mathbf{A}_s + \phi_n\mathbf{A}_n + \phi_b\mathbf{A}_b + \phi_t\mathbf{A}_t)$$



If the cell happens to be Cartesian this reduces to the expected form since, e.g.,

$$\frac{\partial\phi}{\partial x} = \frac{\phi_e - \phi_w}{\Delta x} \quad = \frac{\phi_e A - \phi_w A}{A\Delta x} \quad = \frac{\phi_e A_{ex} + \phi_w A_{wx}}{V}$$

Here, the relevant outward area vectors are $\mathbf{A}_e = (A,0,0)$, $\mathbf{A}_w = (-A,0,0)$ and none of $\mathbf{A}_s$, $\mathbf{A}_n$, $\mathbf{A}_b$, $\mathbf{A}_t$ has any component in the x direction.

Classroom Example 1

A tetrahedral cell has vertices at A(2, –1, 0), B(0, 1, 0), C(2, 1, 1) and D(0, –1, 1).

(a) Find the outward vector areas of all faces. Check that they sum to zero.

(b) Find the volume of the cell.

(c) If the values of ϕ at the centroids of the faces (indicated by their vertices) are

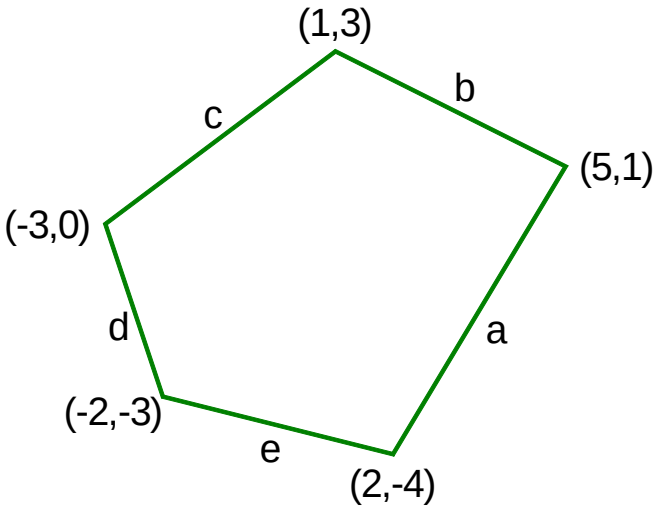
$$\phi_{BCD} = 5, \quad \phi_{ACD} = 3, \quad \phi_{ABD} = 4, \quad \phi_{ABC} = 2,$$

find the volume-averaged derivatives $\left(\frac{\partial \phi}{\partial x}\right)_{av}$, $\left(\frac{\partial \phi}{\partial y}\right)_{av}$, $\left(\frac{\partial \phi}{\partial z}\right)_{av}$.

Classroom Example 2

In a 2-dimensional unstructured mesh, one cell has the form of a pentagon. The coordinates of the vertices are as shown in the figure, whilst the average values of a scalar ϕ on edges *a* – *e* are:

$$\phi_a = -7, \quad \phi_b = 8, \quad \phi_c = -2, \quad \phi_d = 5, \quad \phi_e = 0$$



Find:

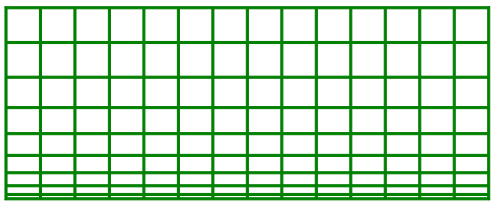
(a) the area of the pentagon;

(b) the cell-averaged derivatives $\left(\frac{\partial \phi}{\partial x}\right)_{av}$ and $\left(\frac{\partial \phi}{\partial y}\right)_{av}$.

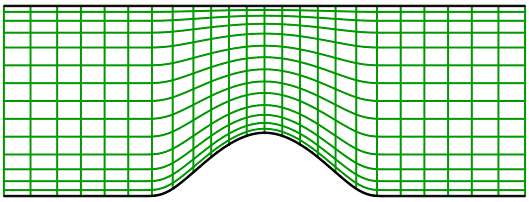
9.2.4 Classification of Meshes

Structured meshes are those whose control volumes can be indexed by (i,j,k) for $i = 1,..., n_i$, $j = 1,.., n_j$, $k = 1,..., n_k$, or by a group of such blocks (*multi-block structured grids*).

Structured grids can be *Cartesian* (lines parallel to coordinate axes) or *curvilinear* (usually curved to fit boundaries). The grid is *orthogonal* if all grid lines cross at 90°; examples include Cartesian, cylindrical polar or spherical polar grids. Some flows can be treated as *axisymmetric*, and in these cases, the flow equations can be expressed in terms of *polar coordinates* (r,θ), rather than Cartesian coordinates (x,y), with minor modifications.

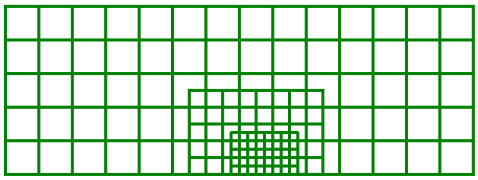


Cartesian mesh

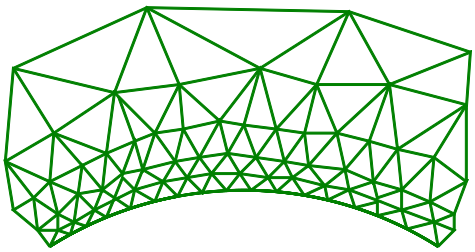


Curvilinear mesh

Unstructured meshes can accommodate completely arbitrary geometries. However, connectivity data structures, solution algorithms, grid generators and plotting routines for such meshes are very complex.



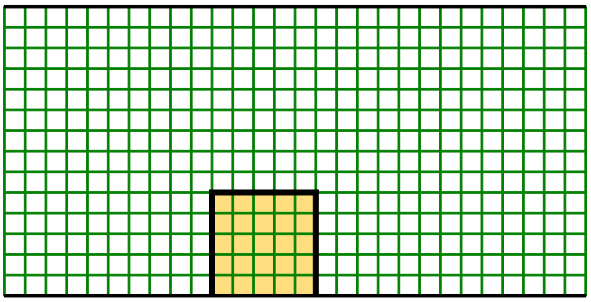
Unstructured Cartesian mesh



Unstructured triangle mesh

9.2.5 Fitting Complex Boundaries With Structured Meshes

Blocking Out Cells



Some important bluff-body flows can be computed on a single-block Cartesian mesh by *blocking out* cells. Solid-surface boundary conditions are applied to cell faces abutting the blocked-out region, whilst values of velocity and other flow variables are forced to zero by a modification of the source term for those cells. If the scalar-transport equations for a single cell are discretised as

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

then the source-term coefficients are modified:

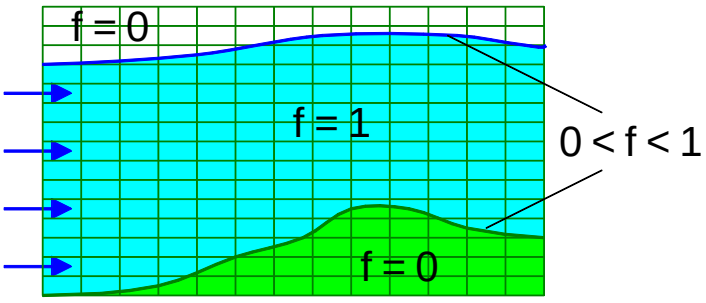
$$b_P \to 0, \quad s_P \to -\gamma$$

where γ is a large number (e.g. 10^{30}). Rearranging for ϕ_P, this gives

$$\phi_P = \frac{\sum a_F \phi_F}{large\ number}$$

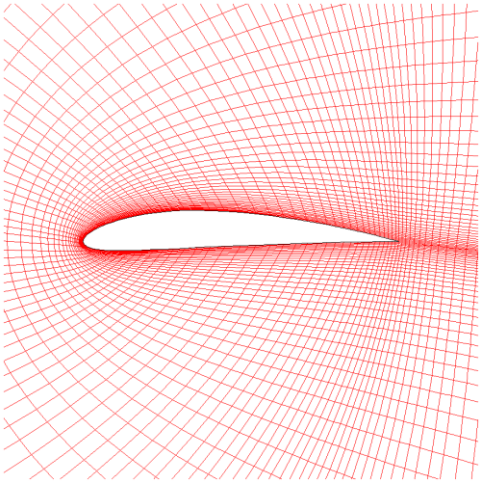
ensuring that the computational variable ϕ_P is effectively forced to zero in these cells. However, the computer still stores values and carries out operations for these points, so that it is essentially performing a lot of redundant work. A better approach is to fit several structured-mesh blocks around the body. Multi-block grids are discussed below.

Volume-of-Fluid (VOF) Approach



In the *volume-of-fluid* approach the fraction *f* of each cell filled with fluid is stored: $f = 0$ for cells with no fluid in, $f = 1$ for cells completely within the interior of the fluid and $0 < f < 1$ for cells which are cut by a boundary. At solid boundaries *f* is determined by the surface contour. At moving free surfaces an equation is solved for *f*, with the surface being reconstructed from its values. A related technique is the *level-set* method where each phase-change surface (including an air-water free-surface interface) is a contour of an indicator variable *f*.

Body-Fitted Meshes



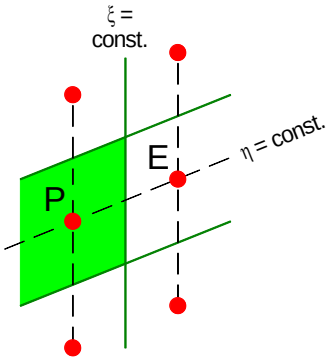
The majority of general-purpose codes employ body-fitted curvilinear grids. The mesh lines are distorted so as to fit curved boundaries. Accuracy in turbulent-flow calculations demands a high density of grid cells close to solid surfaces and the use of body-fitted meshes means that the grid need only be refined in the direction normal to the surface.

However, the use of body-fitted meshes has important consequences.

- It is necessary to store a lot of geometric data for each control volume; for example, in our multi-block structured code STREAM we need to store (x,y,z) components of the cell-face-area vectors for "east", "north" and "top" faces of each cell, plus the volume of the cell itself – a total of 10 arrays.

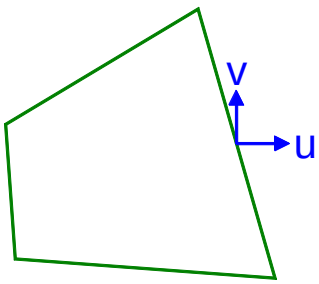
- Unless the mesh is orthogonal, the diffusive flux is no longer dependent only on the nodes immediately on either side of a face; e.g. for the east face:

$$-\Gamma \frac{\partial \phi}{\partial n} A \neq -\Gamma \left(\frac{\phi_E - \phi_P}{\Delta_{PE}} \right) A$$



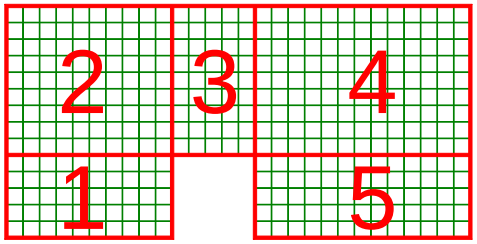
The derivative of ϕ normal to the face involves *cross-derivative* terms parallel to the cell face and nodes other than *P* and *E*. These extra terms are usually transferred to the source term and have to be treated explicitly, which tends to reduce stability.

- Interpolated values of all three velocity components are needed to evaluate the mass flux through a single face. This requires approximations in the pressure-correction equation that can slow down convergence.



Multi-Block Structured Meshes

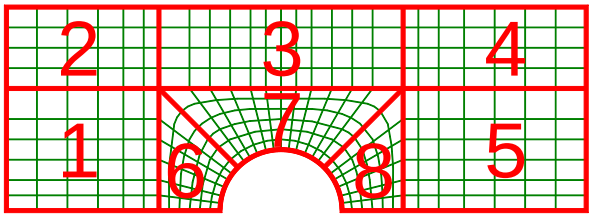
In multi-block structured meshes the domain is decomposed into a small number of regions, in each of which the mesh is structured.



Grid lines may match at the interfaces between blocks, so that the edge vertices are common to two blocks. Alternatively, *arbitrary interfaces* permit block boundaries where vertices do not align and interpolation is required. An important example is a sliding interface, used in rotating machinery.

On each iteration of a scalar-transport equation the discretised equations are solved implicitly within each block, with values from the adjacent blocks providing internal boundary

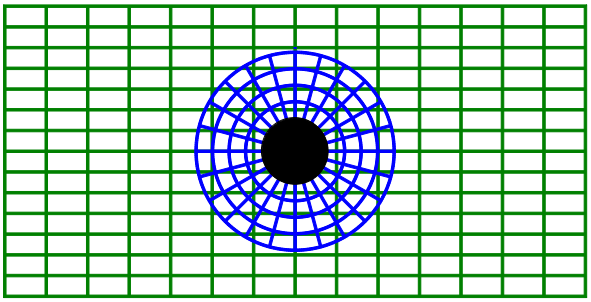
conditions which are updated explicitly at the end of the iteration. (In an arrangement where cell vertices match at block boundaries and blocks also meet "whole-face to whole-face", this can be done more accurately by extending and overlapping adjacent blocks.)



It is generally desirable to avoid sharp changes in grid direction and/or non-orthogonality (which lead to lower accuracy and stability problems) in rapidly-changing regions of the flow.

Chimera (or Overset) Meshes

Some solvers allow overlapping blocks (*chimera*, or *overset*, meshes). Here there is usually a simple background mesh with one or more overset meshes that exchange information with it. These are particularly useful for moving objects, since the overset meshes can move with the object to which they are attached, whilst the background mesh is stationary.



9.2.6 Disposition of Grid Cells

To resolve detail more points are needed in rapidly-changing regions of the flow such as:

- solid boundaries;
- separation, reattachment and impingement points;
- flow discontinuities (shocks, hydraulic jumps).

Simulations must demonstrate *grid-independence*, i.e. that a finer-resolution grid would not significantly modify the solution. This generally requires a sequence of calculations on successively finer grids.

Boundary conditions for turbulence modelling impose some limitations on the cell size near walls. Low-Reynolds-number models (resolving the near-wall viscous sublayer) generally require that $y^+ < 1$ for the near-wall node, whilst high-Reynolds-number models relying on wall functions require (in principle) the near-wall node to lie in the log-law region, ideally $30 < y^+ < 150$.

9.2.7 Multiple Grids

Multiple grids – combining cells so that there are 1, 1/2, 1/4, 1/8, ... times the number of cells in each direction compared with a basic fine grid – are used in *multigrid* methods. A moderately accurate solution is obtained quickly on the coarsest grid, where the number of cells is small and changes propagate rapidly across the domain. This is then interpolated and iterated to solution on successively finer grids.

If two levels of grid are used then *Richardson extrapolation* may be used both to estimate the error and refine the solution. If the basic discretisation is known to be of order n and the exact (but unknown) solution of some property ϕ is denoted ϕ^*, then the error $\phi - \phi^*$ may be taken as proportional to Δ^n, where Δ is the mesh spacing. For solutions ϕ_Δ and $\phi_{2\Delta}$ respectively on two grids with mesh spacing Δ and 2Δ,

$$\phi_\Delta - \phi^* = C\Delta^n$$
$$\phi_{2\Delta} - \phi^* = C(2\Delta)^n$$

These two equations for two unknowns, ϕ^* and C, can be solved to get a better estimate of the exact solution:

$$\phi^* = \frac{2^n \phi_\Delta - \phi_{2\Delta}}{2^n - 1}$$

and the error in the fine-grid solution:

$$C\Delta^n = \frac{\phi_{2\Delta} - \phi_\Delta}{2^n - 1}$$

Classroom Example 3 (Computational Hydraulics Exam, May 2010 – part)

A numerical scheme known to be second-order accurate is used to calculate a steady-state solution on two regular Cartesian meshes A and B, where the finer mesh A has half the grid spacing of mesh B. The values of the solution ϕ at a particular point are found to be 0.74 using mesh A and 0.78 using mesh B. Use *Richardson extrapolation* to:

(a) estimate an improved value of the solution at this point;
(b) estimate the error at this point using the mesh-A solution.

9.3 Boundary Conditions

INLET

- *Velocity inlet*
  transported variables specified on the boundary, either by a predefined profile or by doing an initial fully-developed-flow calculation.
- *Stagnation* (or *reservoir*) *boundary*
  total pressure and total temperature (in compressible flow) or total head (in incompressible flow) fixed; this is a common inlet condition in compressible flow.

OUTLET

- *Standard outlet*
  zero normal gradient ($\partial\phi/\partial n = 0$) for all variables.
- *Pressure boundary*
  as for standard outlet, except fixed value of pressure; this is the usual outlet condition in compressible flow if the exit flow is subsonic (Ma < 1) and in free-surface flow if the exit flow is subcritical (Fr < 1).
- *Radiation* (or *convection*)
  prevent reflection of wave-like motions at outflow boundaries by solving a simplified first-order wave equation $\frac{\partial\phi}{\partial t} + c\frac{\partial\phi}{\partial x} = 0$ with wave velocity *c*.

WALL

- *Non-slip wall*
  the default case for solid boundaries (zero velocity relative to the wall; wall stress computed by viscous stress or wall functions).
- *Slip wall*
  only the velocity component normal to the wall vanishes; used if it is not necessary to resolve a thin boundary layer on an unimportant wall boundary.

OTHER

- *symmetry plane*
  $\partial\phi/\partial n = 0$, except for the velocity component normal to the boundary, which is zero; used where there is a geometric plane of symmetry, but also as a far-field boundary condition, because it ensures that there is no flow through, nor viscous stresses on, the boundary.
- *periodic*
  used in repeating flow; e.g. rotating machinery, regular arrays.
- *free-surface*
  pressure fixed (dynamic boundary condition); no net mass flux through the surface (kinematic boundary condition).

9.4 Flow Visualisation

CFD has a reputation for producing colourful output and, whilst some of it is promotional, the ability to display results effectively may be an invaluable design tool.

9.4.1 Available Packages

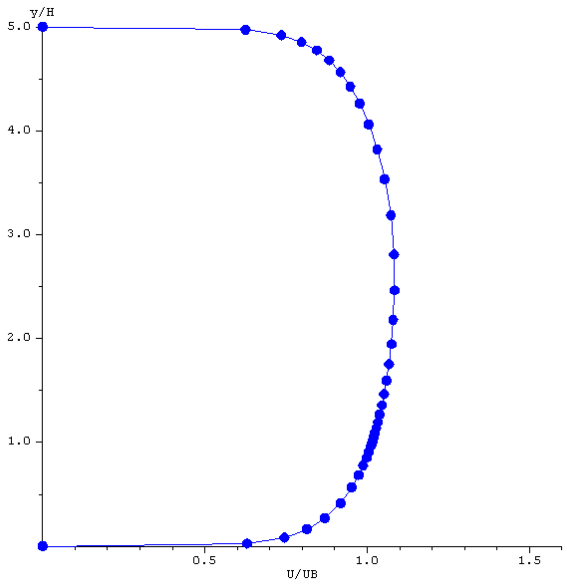
Visualisation tools are often packaged with commercial CFD products. However, many excellent stand-alone applications or libraries are available, including the following.

- TECPLOT (http://www.tecplot.com)
- EnSight (http://www.ensight.com)
- ParaView (http://www.paraview.org)
- AVS (http://www.avs.com)
- Iris Explorer (http://www.nag.co.uk/visualisation_graphics.asp)
- Dislin (http://www.mps.mpg.de/dislin)
- Gnuplot (http://www.gnuplot.info)

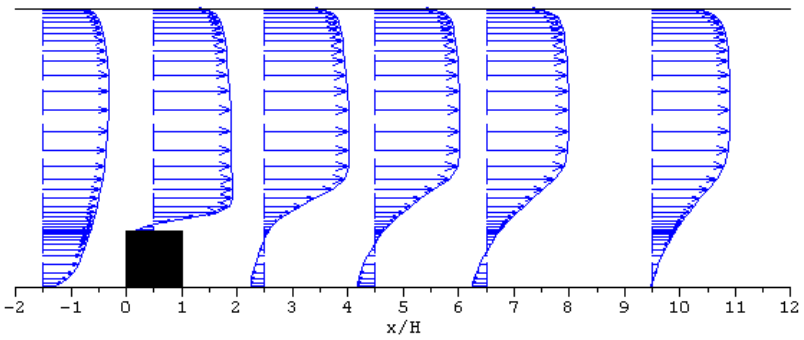
9.4.2 Types of Plot

x-y plots

Simple two-dimensional graphs can be drawn by hand or by many plotting packages. They are the most precise and quantitative way to present numerical data and, since laboratory data is often gathered by straight-line traverses, they are a common way of comparing experimental and computational data. Logarithmic scales allow the identification of important effects occurring at very small scales, particularly near solid boundaries. Line graphs are widely used for profiles of flow variables and for plots of surface quantities such as pressure or skin-friction coefficients.

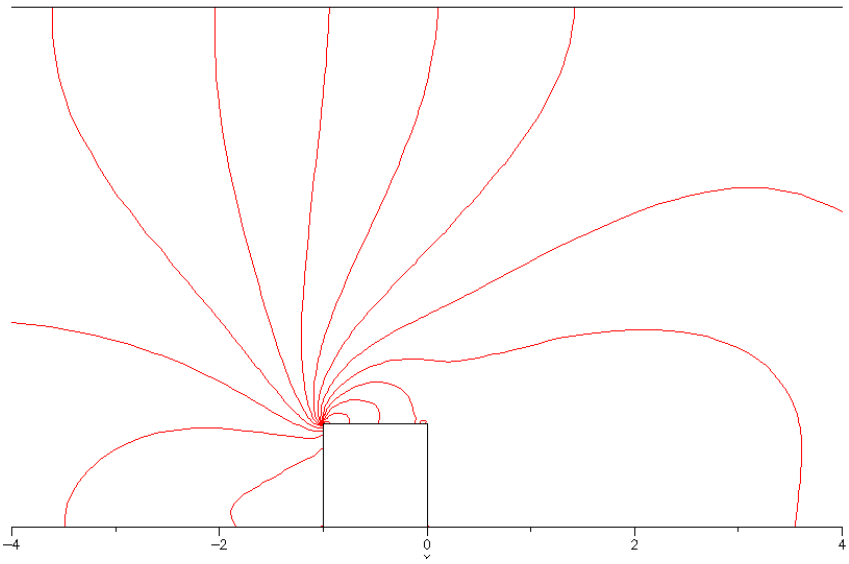


One way of visualising flow development is to use several successive profiles.



Line Contour Plots

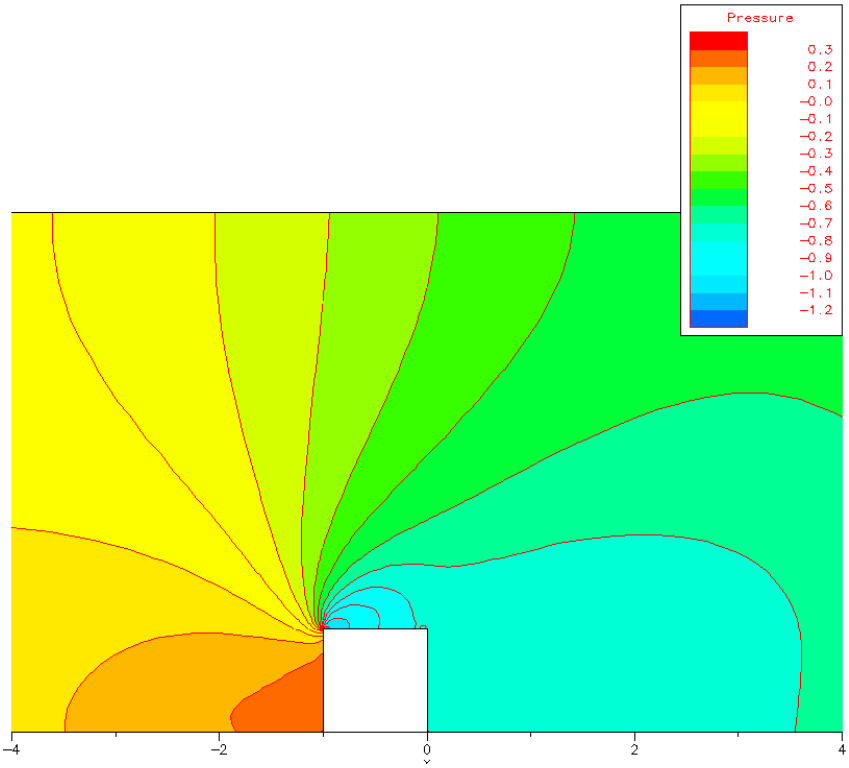
A contour line (*isoline*) is a curve along which some property is constant. The equivalent in 3d is an *isosurface*. Any field variable may be contoured. In contrast to line graphs, contour plots give a global view of the flow field, but are less useful for reading off precise numerical values. If the domain is linearly scaled then detail occurring in small regions is often obscured.



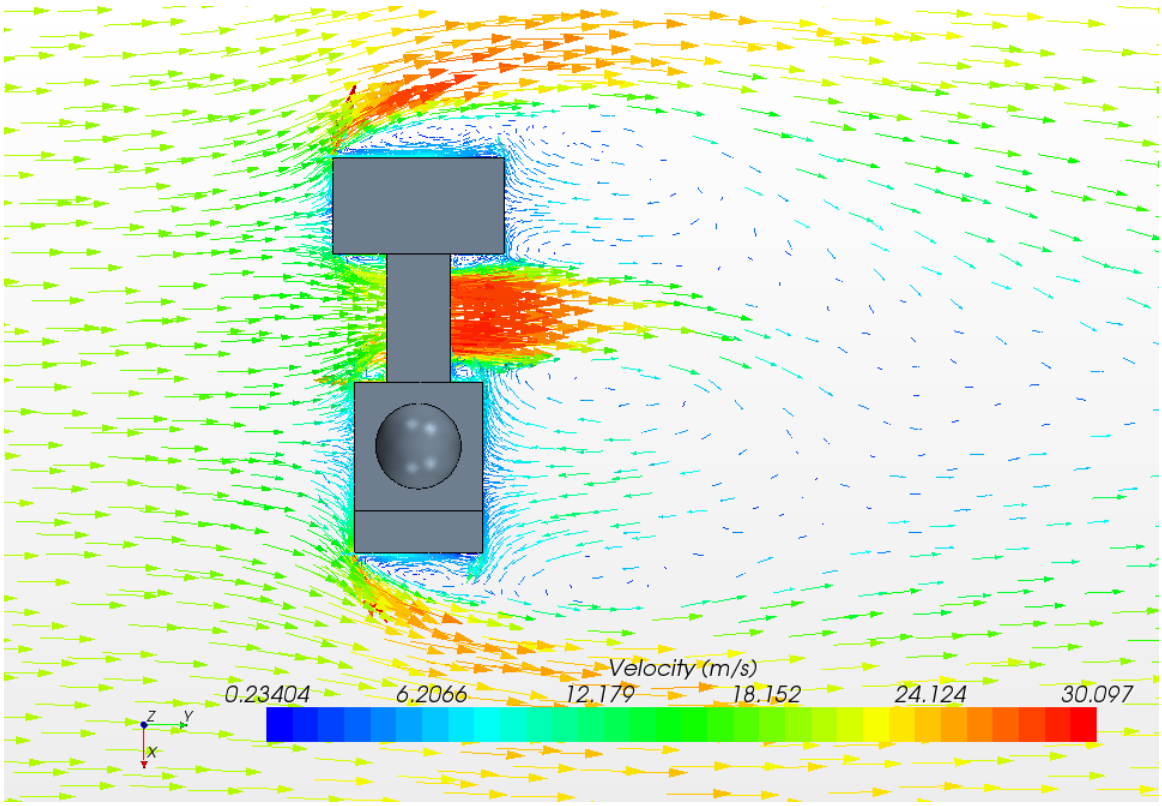
If contour intervals are equal then clustering of lines indicates rapid changes in flow quantities (e.g. shocks and discontinuities).

Shaded Contour Plots

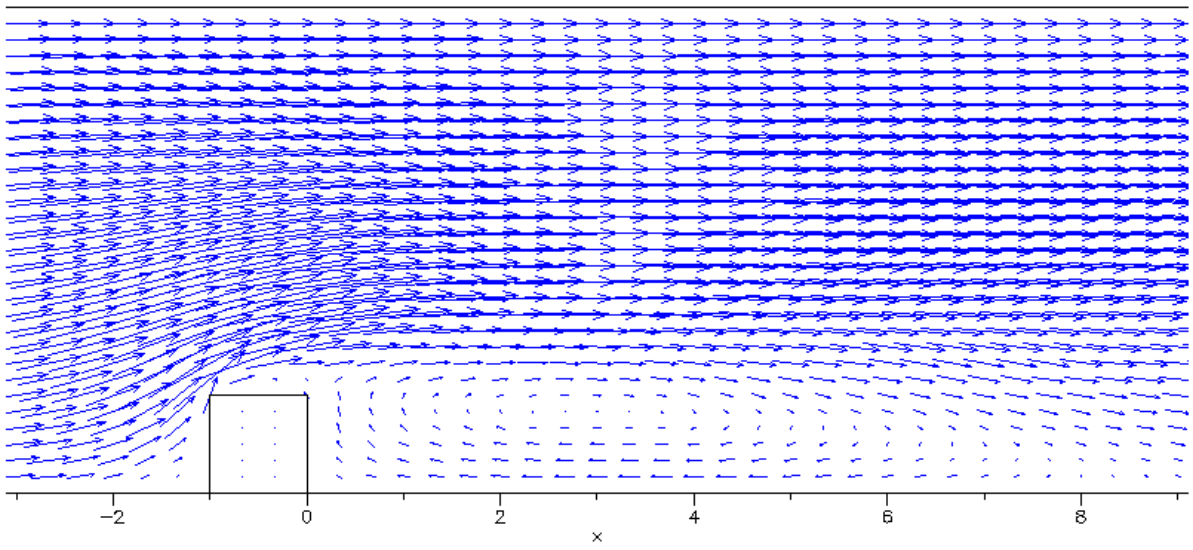
Colour is an excellent medium for conveying information and good for on-screen and presentational analysis of data. Simple packages flood the region between isolines with a fixed colour for that interval. The most advanced packages allow a pixel-by-pixel gradation of colour between values specified at the cell vertices, together with lighting and other special effects such as translucency.



Vector Plots



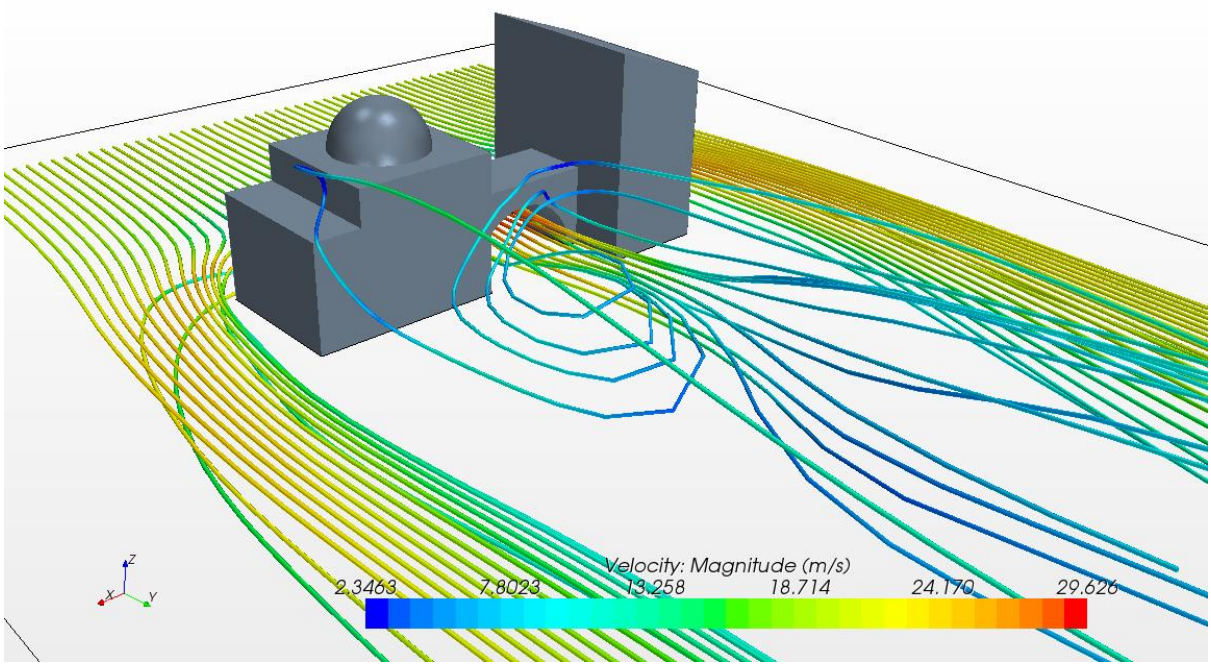
Vector plots display vector quantities (usually velocity, occasionally stress) with arrows whose orientation indicates direction and whose size (and/or colour) indicates magnitude. They are a popular and informative means of illustrating the flow field in two dimensions, although if grid densities are high then either interpolation to a uniform grid or a reduced set of output positions is necessary to prevent the number density of arrows obscuring plots. It can be difficult to selecting scales for arrows when large velocity differences are present, especially in recirculation zones where the mean flow speed is low. In three dimensions, vector plots can be deceptive because of the angle from which they are viewed.



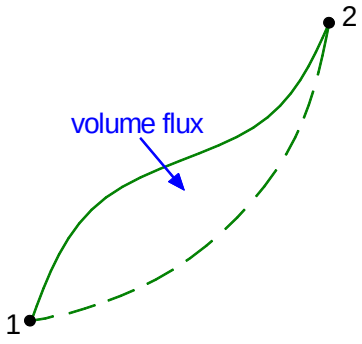
Streamline Plots

Streamlines are everywhere parallel to the local mean velocity vector. In 3-d they must be obtained by integration:

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}$$



In 2-d incompressible flow a more accurate method is to contour the *streamfunction* ψ. This function is defined by fixing ψ arbitrarily at one point and then setting the change in ψ between two points as the volume flow rate (per unit depth) across any curve joining them:



$$\psi_2 - \psi_1 = \int_1^2 \mathbf{u} \bullet \mathbf{n} \; ds$$

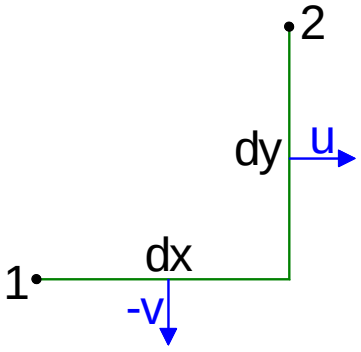
(The sense used here is clockwise about the start point, although the opposite sign convention is equally valid). ψ is well-defined in incompressible flow because, by continuity, the flow rate across any curve connecting two points must be the same. Contouring ψ produces streamlines because a curve of constant ψ is one across which there is no flow.

For a short path consisting of small increments dx and dy parallel to the coordinate axes,

$$d\psi = u\,dy - v\,dx$$

and hence the velocity components are related to ψ by

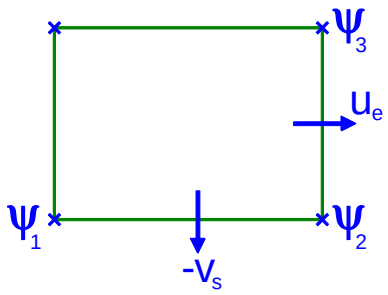
$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}$$



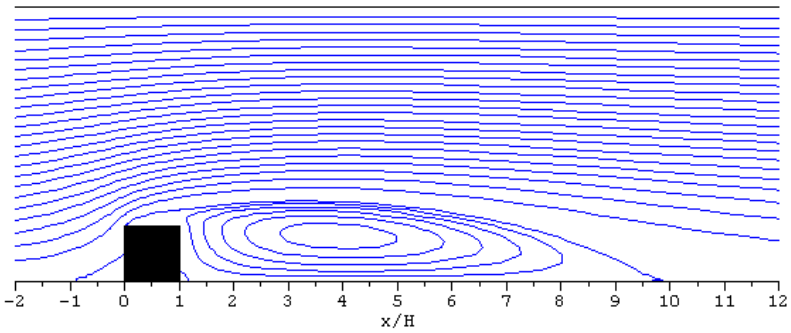
Computationally, ψ is stored at cell *vertices*. This is convenient because the flow rates across the cell edges are already stored as part of the calculation. In the Cartesian cell shown:

$$\psi_2 = \psi_1 - v_s \Delta x$$

$$\psi_3 = \psi_2 + u_e \Delta y$$

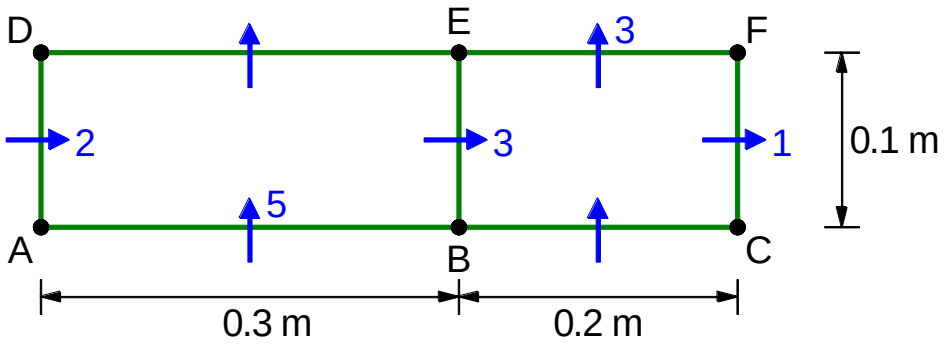


If isolines are at equal increments in ψ, then clustering of lines corresponds to high velocities, whilst regions where the streamlines are further apart signify low velocities. As with vector plots, this has the effect of making it difficult to visualise the flow pattern in low-velocity regions such as recirculation zones and a smaller increment in ψ is needed here.



Classroom Example 4

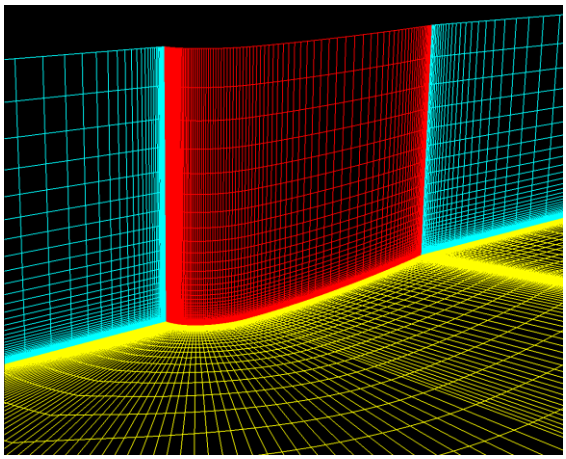
(a) Two adjacent cells in a 2-dimensional Cartesian mesh are shown below, along with the cell dimensions and some of the velocity components (in m s^{-1}) normal to cell faces. The value of the stream function ψ at the bottom left corner is $\psi_A = 0$. Find the value of the stream function at the other vertices B – F. (You may use either sign convention for the stream function).

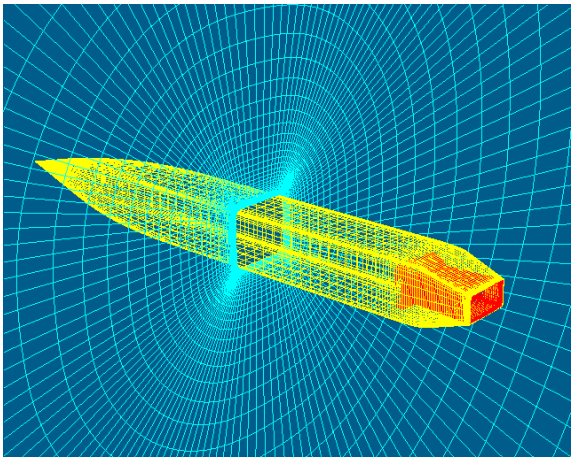


(b) Sketch the pattern of streamlines across the two cells in part (a).

Mesh Plots

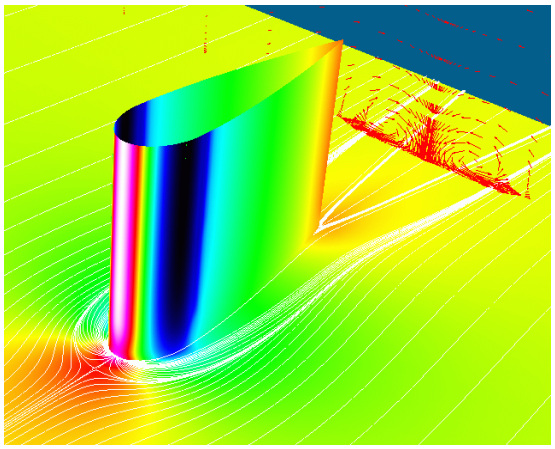
The computational mesh is usually visualised by plotting the edges of control volumes. It can be difficult to visualise fully-unstructured 3-d meshes, and usually only surface meshes or mesh projections onto a plane section is portrayed.

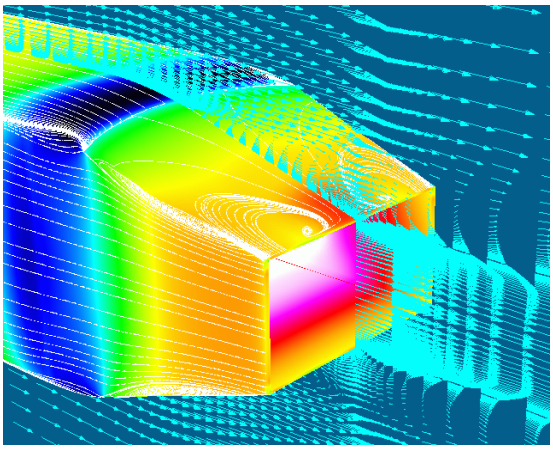




Composite Plots

To maximise information it is common to combine plots of the above types, emphasising the behaviour of several important quantities in a single scene.





Appendix: Summary of Vector Calculus (*Optional*)

This is not examinable – it is here so that you can see where some of the notation and results of earlier sections come from.

The operator

$$\nabla \equiv (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$$

(called *del* or *nabla*) is both a vector and a differential operator.

It is used to define:

gradient: $grad\ \phi \equiv \nabla\phi \equiv (\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z})$ acting on scalar field ϕ

divergence: $div\ \mathbf{f} \equiv \nabla \bullet \mathbf{f} \equiv \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} + \frac{\partial f_z}{\partial z}$ acting on vector field $\mathbf{f} = (f_x, f_y, f_z)$

curl: $curl\ \mathbf{f} \equiv \nabla \wedge \mathbf{f} \equiv \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ f_x & f_y & f_z \end{vmatrix}$ acting on vector field $\mathbf{f} = (f_x, f_y, f_z)$

$div(grad\ \phi) \equiv \nabla \bullet \nabla\phi \equiv \nabla^2\phi \equiv \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2}$ is called the **Laplacian**.

Integral Theorems

Gauss' Divergence Theorem
For arbitrary volume V, with bounding surface ∂V:

$$\int_V \nabla \bullet \mathbf{f}\ \mathrm{d}V = \oint_{\partial V} \mathbf{f} \bullet \mathrm{d}\mathbf{A}$$

Stokes' Theorem
For arbitrary open surface A, with boundary ∂A:

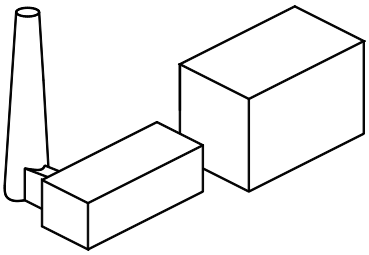
$$\int_A \nabla \wedge \mathbf{f} \bullet \mathrm{d}\mathbf{A} = \oint_{\partial A} \mathbf{f} \bullet \mathrm{d}\mathbf{s}$$

Examples

Q1.
(a) Explain what is meant, in the context of computational meshes, by:
(i) structured;
(ii) multi-block structured;
(iii) unstructured;
(iv) chimera.

(b) Define the following terms when applied to structured meshes:
(i) Cartesian;
(ii) curvilinear;
(iii) orthogonal.

Q2. (Exam 2011 – part)
(a) For wind-loading calculations a CFD calculation of airflow is to be performed about the building complex shown. Sketch a suitable computational domain indicating the specific types of boundary condition that are applied at each boundary of the fluid domain. For each boundary type summarise the mathematical conditions that are applied to each flow variable (velocity, pressure, turbulent scalar).



(b) Define the *drag coefficient* for an object and explain how it would be calculated from the raw data obtained in a CFD simulation.

Q3.
Show that the vector area of a (possibly non-planar) quadrilateral is half the cross product of its diagonals; i.e.

$$\mathbf{A} = \tfrac{1}{2}\mathbf{d}_{13} \wedge \mathbf{d}_{24} = \tfrac{1}{2}(\mathbf{r}_3 - \mathbf{r}_1) \wedge (\mathbf{r}_4 - \mathbf{r}_2)$$

Q4.
Derive the following formulae for the volume of a cell and the cell-averaged derivative of a scalar field:

(a) $V = \frac{1}{3}\oint_{\partial V} \mathbf{r} \bullet \mathrm{d}\mathbf{A}$;

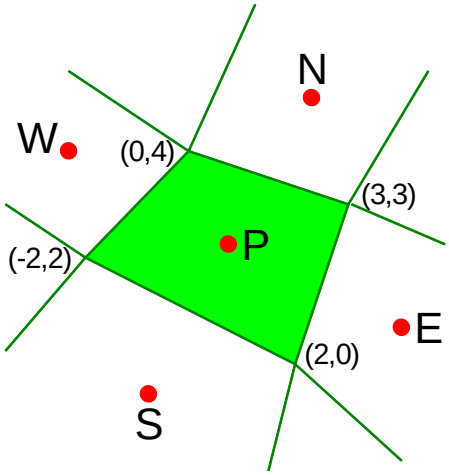
(b) $\left(\frac{\partial \phi}{\partial x}\right)_{av} = \frac{1}{V}\oint_{\partial V} \phi \ \ \mathrm{d}A_x$;

where $\mathbf{r}$ is a position vector and $\mathrm{d}\mathbf{A}$ a small element of (outward-directed) area on the closed surface ∂V.

Q5.

The figure shows part of a non-Cartesian 2-d mesh. A single quadrilateral cell is highlighted and the coordinates of the corners marked. The values of a variable ϕ at the cell-centre nodes (labelled geographically) are:

$\phi_P = 2, \quad \phi_E = 5, \quad \phi_W = 0, \quad \phi_N = 3, \quad \phi_S = 1.$



Find:

(a) the area of the cell;

(b) the cell-averaged derivatives $(\partial\phi/\partial x)_{av}$ and $(\partial\phi/\partial y)_{av}$, assuming that cell-face values are the average of those at the nodes either side of that cell face.

Q6.

One quadrilateral face of a hexahedral cell in a finite-volume mesh has vertices at

(2, 0, 1), (2, 2, –1), (0, 3, 1), (–1, 0, 2)

(a) Find the vector area of this face (in either direction).

(b) Determine whether or not the vertices are coplanar.

Q7.

(a) The vertices of a particular tetrahedral cell in a finite-volume mesh are

(0,0,0), (4,0,0), (1,4,0), (1,2,4)

Find:

(i) the *outward* vector area of each face;

(ii) the volume of the cell.

(b) For this cell a scalar ϕ has value 6 on the largest face, 2 on the smallest face and 3 on the other two faces. Find the cell-averaged derivatives $(\partial\phi/\partial x, \partial\phi/\partial y, \partial\phi/\partial z)_{av}$.

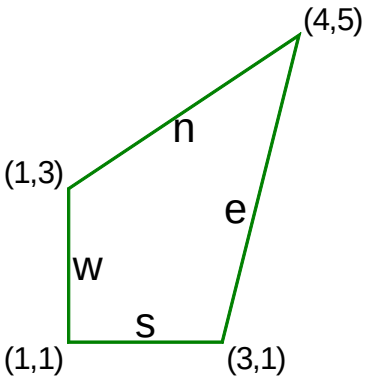
(c) Determine whether the point (1,2,1) lies inside, outside or on the boundary of the cell.

Q8. (Exam 2009 – part)

The figure below shows a quadrilateral cell, together with the coordinates of its vertices and the velocity components on each face. If the value of the stream function at the bottom left corner is $\psi = 0$, find:

(a) the values of ψ at the other vertices;

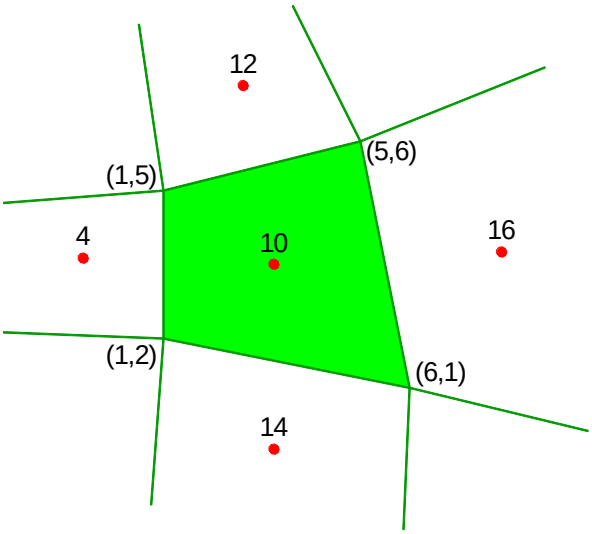
(b) the unknown velocity component v_n.



face	velocity	
	u	*v*
w	3	–3
s	0	2
e	3	1
n	1	v_n

Q9. (Exam 2012)

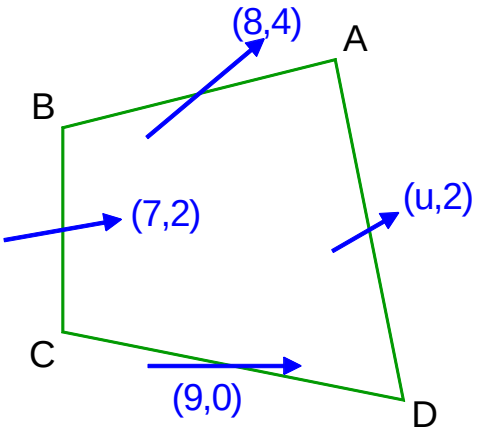
The figure right shows a region of a 2-d structured mesh. The coordinates of the vertices of one cell are marked, together with the pressure at nearby nodes.



(a) Assuming that pressure on an edge is the average of that at nodes on either side of the edge, find the (x,y) components of the pressure force (per unit depth) on each edge of the shaded cell and hence the net pressure force vector on the cell.

(b) Find the area of the cell and the cell-averaged pressure gradients $\partial p/\partial x$ and $\partial p/\partial y$.

(c) The figure right shows the velocity components on each edge of the same cell. Find the outward volume flow rate (per unit depth) from each edge and the missing velocity component *u*.



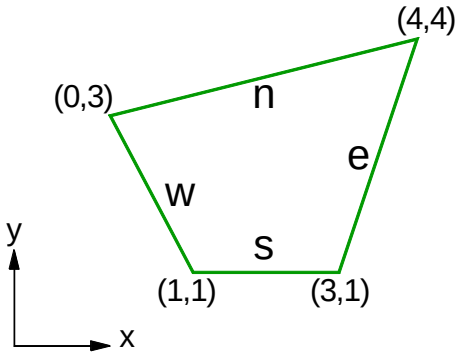
(d) If the value of the streamfunction ψ at vertex A is 3, calculate the streamfunction at the other vertices. (You may use either sign convention.)

Q10. (*** *Advanced* ***)

(a) Give a physical definition of the stream function ψ in a 2-d incompressible flow, and show how it is related to the velocity components.

(b) Show that if the flow is irrotational then ψ satisfies Laplace's equation. If the flow is not irrotational how is $\nabla^2\psi$ related to the vorticity?

The figure below and the accompanying table show a 2-d quadrilateral cell, with coordinates at the vertices and velocity components on the cell edges. The flow is incompressible. All quantities are non-dimensional.



face	velocity	
	u	*v*
w	3	2
s	0	2
e	3	–1
n	1	v_n

(c) If the value of the stream function at the bottom left (SW) corner is 0, calculate the stream function at the other vertices.

(d) Find the *y*-velocity component v_n, whose value is not given in the table.

(e) By calculating $\oint \mathbf{u} \bullet d\mathbf{s}$ for the quadrilateral cell estimate the cell vorticity.

Q11. (*** *Advanced* ***)
A quadrilateral cell in a 2-d finite-volume mesh is shown in the figure below. The coordinates of the vertices are marked in the figure and the velocities on the edges are given in the adjacent table.

(a) Find the area A of the quadrilateral.

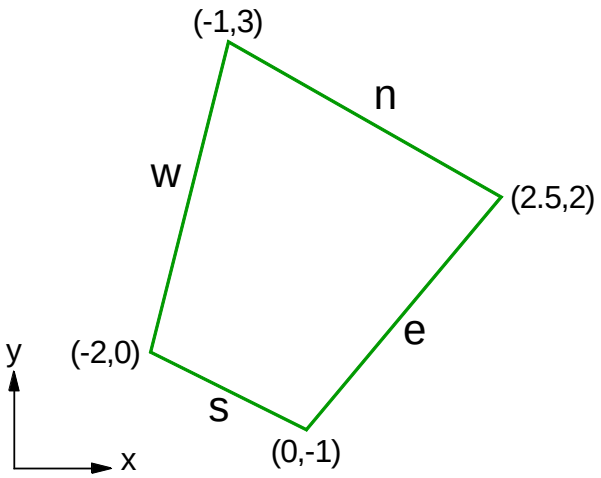
(b) Find the cell-averaged derivatives $\left(\frac{\partial u}{\partial x}\right)_{av}$, $\left(\frac{\partial u}{\partial y}\right)_{av}$, $\left(\frac{\partial v}{\partial x}\right)_{av}$, $\left(\frac{\partial v}{\partial y}\right)_{av}$.

(c) By calculating the line integral from the velocity and geometric data, confirm that both the cell-averaged velocity derivatives and the discrete form of Stokes' law,

$$(\omega_z)_{av} = \frac{1}{A}\oint_{\partial A} \mathbf{u} \bullet d\mathbf{s},$$

give the same value for the cell-averaged vorticity, $(\omega_z)_{av}$.

(Note: this is, in fact, true in general.)



edge	u	v
w	9	–6
s	1	0
e	10	12
n	13	6