Lecture 3: Time Marching Example & Model Equation

University of Bristol

Numerical and Simulation Methods

- Objective is to develop and understand numerical methods to solve (Navier-Stokes) eqns.
- Considered forms of the equations, i.e. conservative v non-conservative.
- Introduced the fundamental concepts of discretisation and 'time-marching'.

TODAY

- Consider an example aerodynamic simulation using 'time-marching'.

- Significance of boundary conditions.

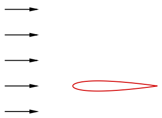
- Introduce a simpler model equation.

Example of Steady Flow Time-Marching

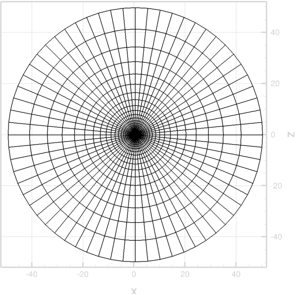
Two-Dimensional Euler Equations

Consider computation of flow around a two-dimensional aerofoil. We have a solid surface, and upstream flow. We set farfield flow as:

$$P = P_\infty \quad \rho = \rho_\infty \quad u = M_\infty c_\infty \quad v = 0.$$



Generate a mesh to fill domain, to 25-50 chords. What are the flow conditions at the edge of the mesh? What are the flow conditions at infinity? $\rightarrow$ farfield boundary conditions. These require very substantial care and simple options often either don't work or introduce insidious errors (consider two-dimension Kutta-Jukowski theorem!).



$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial P + \rho u^2}{\partial x} + \frac{\partial \rho u v}{\partial y} = 0$$

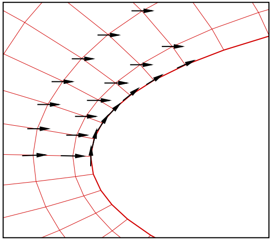
$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho u v}{\partial x} + \frac{\partial P + \rho v^2}{\partial y} = 0$$

$$\frac{\partial E}{\partial t} + \frac{\partial (P + E) u}{\partial x} + \frac{\partial (P + E) v}{\partial y} = 0$$

Consider the continuity equation, and the time-marching scheme for it. When the iteration first starts, flow is uniform everywhere, and so all gradients are zero. However, it is the *boundary conditions* that determine what flow evolves from the equations.

For an inviscid flow, the solid surface boundary condition is tangential flow, i.e. the flow cannot go through the surface, but needs to go around it. Hence, consider first iteration in time-marching; flow is uniform everywhere, except at the surface, where the flow vector must be turned to be tangential to the surface $\Rightarrow$ spatial gradients of the velocity vector not zero any more. We have $\mathbf{v} \cdot \mathbf{n} = 0$, so the convective fluxes vanish, but what is the pressure?

Commonly wall pressure is set equal to the pressure in the adjoining cell (first order accurate). Or, a linear extrapolation of the pressure to the wall can be used.



Farfield conditions are a challenge. The flow at infinity is defined by freestream, but the edge of the mesh is never far enough away for that to be suitable. Also, because $\Gamma = \int u \cdot ds$ if you set a constant velocity, the implied lift would be zero at the farfield, but then nonzero at the aerofoil wall, which is contradictory.

One solution is to linearise about the freestream conditions, and then assume the farfield conditions are a small perturbation about freestream conditions. This allows nonzero lift and for error waves to escape. The only limit is the farfield should not be too close, otherwise linearisation errors may be large, or nonlinear instability may occur. Other options based on entropy arguments exist.

Just don't set farfield to freestream - unless it's a supersonic inflow...

- 1. Start with guessed solution at $t = 0$ (Normally freestream values).
- 2. Enforce boundary conditions (no flow through surface, constant upstream pressure, etc.).
- 3. Evaluate error in spatial gradients at every point: $ERROR = \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y}$ (This is the 'hard' part of numerical methods.)
- 4. If $ERROR \rightarrow 0$ FINISH. Solution is converged.
- 5. Otherwise, evaluate update to conserved variables (ρ in continuity equation):

$$\rho(t + \Delta t) = \rho(t) - \Delta t \times ERROR$$

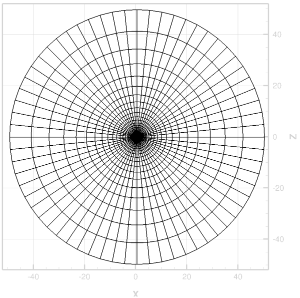
- 6. $t = t + \Delta t$. GO TO 2.

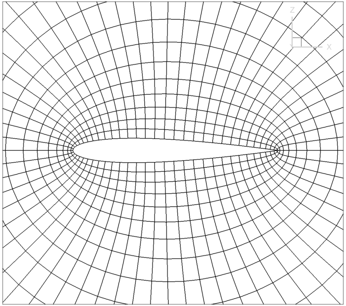
This scheme must be applied to each of the four equations, and at every point in the mesh where the solution is stored. This can be millions of cells, so lots of computations.

Example of 'Time-Marching'

- Mach 0.8 inviscid flow onto symmetric 2-D aerofoil. 65×33 mesh.

Incidence $= 1.25^o$.



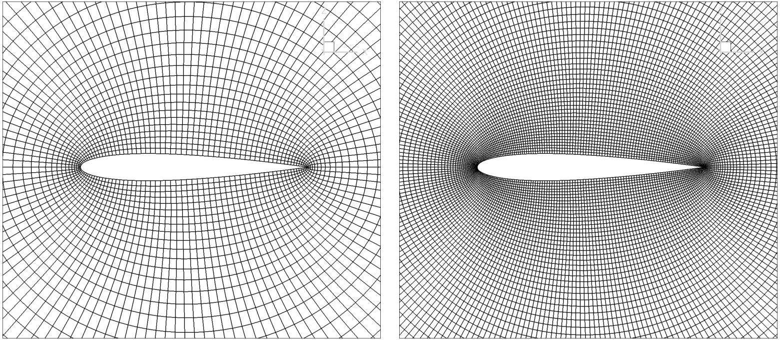


- Initial conditions, $\underline{\mathbf{U}}(x, y, 0)$, Uniform freestream

- constant P, ρ, E, $u = 0.8c, v = 0$.

- Solution 'driven' by solid surface boundary condition.

- Also, 129×65 mesh and 257×129 mesh.



- Time-marched to convergence.
- Error in each cell (shown in animations) is defined as

$$\mathrm{ERROR} = \frac{|\underline{\mathbf{U}}(t+\Delta t) - \underline{\mathbf{U}}(t)|}{\Delta t} = \frac{1}{\Delta t}\{|\rho(t+\Delta t) - \rho(t)| + |\rho u(t+\Delta t) - \rho u(t)|$$
$$+ |\rho v(t+\Delta t) - \rho v(t))| + |E(t+\Delta t) - E(t)|\}$$

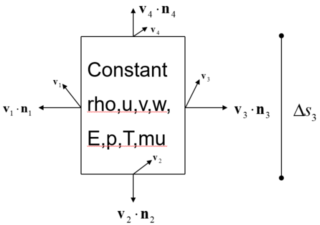


Figure: A finite volume

$$\int_\Omega \frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} dv = \int_\Omega \frac{\partial \rho}{\partial t} dv + \int_{\partial \Omega} \rho \mathbf{v} \cdot \mathbf{n} ds$$

$$\int_\Omega \nabla \cdot \rho \mathbf{v} dv = \int_{\partial \Omega} \rho \mathbf{v} \cdot \mathbf{n} ds$$

$$\int_\Omega \frac{\partial \rho}{\partial t} dv = -\int_{\partial \Omega} \rho \mathbf{v} \cdot \mathbf{n} ds \approx -\sum_{i=1}^{i=4} \rho_i \mathbf{v}_i \cdot \mathbf{n}_i \Delta S_i$$

Can do a similar approach for momentum (here just in x)

$$\int_{\Omega} \frac{\partial \rho u}{\partial t} + \nabla \cdot \rho u \mathbf{v} + \nabla p_x dv = \int_{\Omega} \frac{\partial \rho u}{\partial t} dv + \int_{\partial\Omega} \rho u \mathbf{v} \cdot \mathbf{n} ds + \int_{\partial\Omega} p \mathbf{n} \cdot \mathbf{i} ds$$

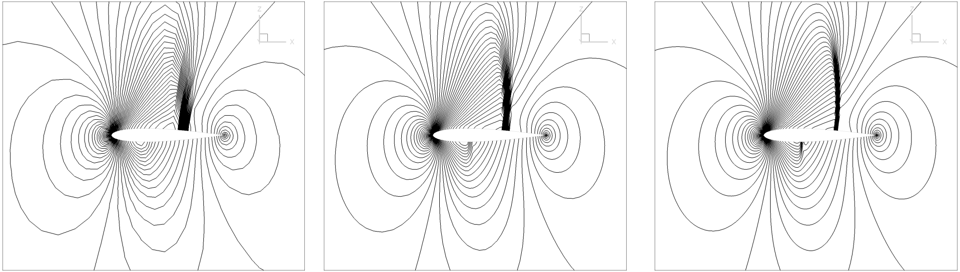
and can also write for all coordinate directions in one go as

$$\int_{\Omega} \frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v}\mathbf{v} + \nabla p dv = \int_{\Omega} \frac{\partial \rho \mathbf{v}}{\partial t} dv + \int_{\partial\Omega} \rho \mathbf{v} \left(\mathbf{v} \cdot \mathbf{n}\right) ds + \int_{\partial\Omega} p \mathbf{n} ds$$

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{V_i} \sum_{i=1}^{i=4} \rho_i \mathbf{v}_i \cdot \mathbf{n}_i \Delta S_i$$

This is exactly the update used in any finite volume explicit method. The momentum and energy equations are handled in a similar way. $\mathbf{v} \cdot \mathbf{n}$ is known as the 'convective velocity' because it is the net flow quantity that moves mass, momentum and energy from place to place (cell to cell). The term $\rho_i \mathbf{v}_i \cdot \mathbf{n}_i \Delta S_i$ is known as the 'face flux' because it is the amount of 'stuff' (in this case mass) crossing that face. A huge amount of effort goes in to finding $\rho_i \mathbf{v}_i \cdot \mathbf{n}_i$ (central difference schemes, upwind schemes - see later). Very rarely does anyone use a first order time update though - 4th order Runge-Kutta is most common in some form.

Steady Solutions



Solution Isobars

- $C_L = 0.3621, 0.3663, 0.3651.$
- Solutions after 5000, 10000, 20000 simple explicit time-steps.
- Run-times 1, 8, 64 cost units

- methods to reduce this significantly $\rightarrow$ cost proportional to number mesh points

Unsteady solutions

What to do if the flow is unsteady (eg. aerofoil moving up and down)?

$$\int_\Omega \frac{\partial \rho}{\partial t^*} dv = -\int_\Omega \frac{\partial \rho}{\partial t} dv - \int_{\partial\Omega} \rho \mathbf{v} \cdot \mathbf{n} ds \approx -\int_\Omega \frac{\partial \rho}{\partial t} dv - \sum_{i=1}^{i=4} \rho_i \mathbf{v}_i \cdot \mathbf{n}_i \Delta S_i$$

$$\rho_c^{m+1} = \rho_c^m - \frac{\Delta t^*}{V_c} \left[\frac{\rho_c^{n+1} - \rho_c^n}{\Delta t} + \sum_{i=1}^{i=4} \rho_i^{n+1} \mathbf{v}_i^{n+1} \cdot \mathbf{n}_i^{n+1} \Delta S_i \right]$$

The strategy is the same, except we introduce a pseudo time variable t^* and include the real contribution of the actual time derivative alongside the spatial residual on the right hand side. This is implicit in real time, but explicit in pseudo time. Other unsteady methods exist (eg. explicit in real time, and implicit in real time with a linear solve). Usually second order backward differences are used.

Pseudo time is really useful, but not efficient for all PDE problems even though it can nearly always be used.

Numerical Solution Procedures for "Navier-Stokes Like" Equation

Navier-Stokes equations are insoluble analytically. They are a set of five non-linear, coupled, parabolic P.D.E.s, plus the equation of state. Classical aerodynamics involves making simplifying assumptions about the flow leading to reduction of the equations until we obtain a simple linear equation that we can solve analytically. However, this places severe restrictions on the types of flow we can consider. Most flow phenomena present in real aerodynamic flows cannot be modelled with this approach. Hence, we turn to numerical solution of the equations. In examining numerical methods it is not sensible to apply them to the full Navier-Stokes equations initially. More sensible is to consider a simpler equation which exhibits the same behaviour as the N-S equations. A commonly used equation is the 1-D non-linear Burgers' equation (after J.M. Burgers)

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{1}{2} u^2}{\partial x} = \alpha \frac{\partial^2 u}{\partial x^2}$$

(u is any scalar.) This equation is a good test equation as it contains the same essential features as the N-S equations, i.e.

1) a temporal derivative
2) a non-linear spatial derivative
3) a diffusive term

But also there are exact analytical solutions available for some forms of the equation, so we can compare our numerical solutions with exact ones.
In the equation $u = u(x,t)$ is some scalar quantity, and α is the viscosity coefficient.
The above equation is the conservative form. This can also of course be written in non-conservative form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2}.$$

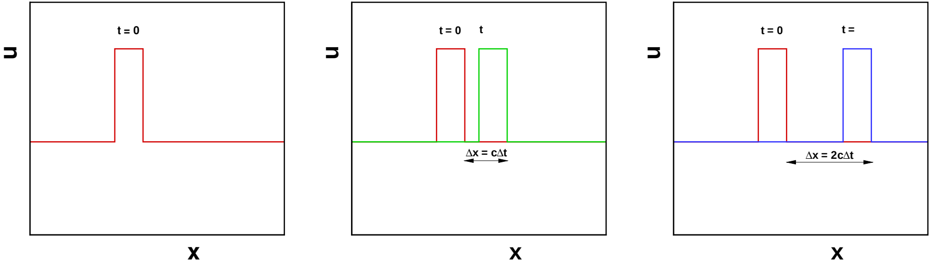
These two equations are mathematically equivalent, but as will be seen later do not lead to the same result numerically. The inviscid form of Burgers' equation is obtained by simply setting the viscosity coefficient to zero,

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{1}{2}u^2}{\partial x} = 0.$$

This can be simplified even further, by considering the non-conservative form and setting the 'wavespeed' (term outside the derivative) to a constant value,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0.$$

This is now a linear equation, and is known as the linear advection (or wave) equation. c is the wave or propagation speed. This last equation has a very simple solution.
If at time $t = 0$ we have the initial condition $u(x, 0) = f(x)$, then $u(x, t) = f(x - ct)$ and $-cf' + cf' = 0$ (hence linear wave equation solved). This represents a wave of fixed shape moving at a constant speed c. It is essentially a one dimensional cloud drifting in the wind, but where the cloud shape is not changing.



Thought for the day

Figure: Cloud advection - imagine a cloud that moves without changing shape - this is what we have with the linear wave equation

Time-Marching of Model Equation

We will consider our model equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

This is a continuous equation, i.e. it gives u as a function of x and t. The equation also gives the "time-evolution" of u, i.e. given $u(x, t_1)$ we can use the equation to predict $u(x, t_2)$. This can be used to derive a solution procedure for steady flows. We are usually interested in steady flows, i.e. the asymptotic solution as $t \to \infty$.

Consider the ways in which a solution might 'go wrong'. There are only really two - either the solution explodes in an unstable manner, or the wave is diffused to nothing over a long period of time. The first is instability and easily avoided, the second is a diffusive error much harder to deal with in general.

- 1. Guess $u(x, 0)$.
- 2. Enforce boundary conditions (i.e. no flow through body, or constant upstream pressure).
- 3. Let solution "evolve" (time-march) over small time step Δt to new solution $u(x, t + \Delta t)$.
- 4. Evaluate $\frac{\partial u}{\partial t}(= -c\frac{\partial u}{\partial x}$ for our problem).
- 5. If $\frac{\partial u}{\partial t}$ is small we have a steady state solution.
- 6. If not, set $t = t + \Delta t$ and goto 2.

If this converges ($\frac{\partial u}{\partial t} \to 0$) then we can use it to find steady solution.
Sometimes, an apparently steady problem has an unsteady solution ! e.g. transonic buzz.
With a symmetric aerofoil at zero incidence, with constant onflow, the shock waves can oscillate back and forward on both surfaces.

Summary

• Discretisation is the concept of representing a real continuous funtion, $u(x, t)$, with a discrete solution, i.e. only stored/known at discrete points in time and space, $u(i\Delta x, n\Delta t)$.

• Equations we want to solve are non-linear PDEs with a temporal derivative, and this is exploited to produce a time-marching approach to steady flows.

Begin with a guessed (wrong) initial value, $u(x, 0)$, continue to update u until it stops changing, i.e. $\frac{\partial u}{\partial t} \to 0$. $\frac{\partial u}{\partial t}$ is the 'error' in the spatial gradients.

- Boundary conditions applied at domain boundaries determines solution.

• Conservative form of the equations must be solved in numerical methods, due to difficulty in representing large gradients discretely.

• Considered a two-dimensional example, and the accuracy dependence on the mesh density clearly demonstrated.

• Introduced a model equation with exact solution.

NEXT LECTURE: How do we approximate the spatial gradients in a discrete solution ?