Numerical Simulation Methods

Part 1 - Fundamental Theory

Lecture 8: Implicit Schemes

Part 1

Summary So Far

• Want to solve Navier-Stokes equations (any PDE).
• Use of simpler equation - Burgers equation
- has same features of N-S equations + has exact solutions.
• Approximated time and space derivatives by considering non-continuous discrete solution and Taylor expansions.
• The order of the terms neglected in the expansions determine the truncation error, and hence the order of accuracy of the scheme.
• Von-Neumann stability analysis used to give amplification factor and phase error.
• Schemes which use KNOWN values (level n) to update solution called EXPLICIT.
• Stability analysis shows schemes which violate physics of the real flow, i.e. have incorrect signal propagation, are unstable. Also shows explicit schemes are only stable for

$$CFL = \frac{c\Delta t}{\Delta x} \leq 1.$$

The amplification factor then depends on Δt and Δx. The higher order the scheme the smaller the amplification error. Normally $\lambda^2 = 1 - \mathrm{error}(\Delta x^p, \Delta t^q)$.

• Must have an 'upwind' scheme - where sign of the wavespeed determines which points are used in the finite-difference stencil. However, 'non-upwind' or unstable schemes can be stabilised by adding an 'artificial viscosity' term (in the form $\alpha \frac{\partial^2 u}{\partial x^2}$) to damp the solution.
• Non-conservative form gives incorrect signal speed.
• Considered higher-order schemes - Lax-Wendroff & McCormack's method.
• Considered explicit schemes for general flux functions and wavespeed implications.

TODAY

• Implicit schemes.

Implicit Methods

All the schemes considered so far have been explicit, i.e.

$$u_i^{n+1} = f(\dots, u_{i-2}^n, u_{i-1}^n, u_i^n, u_{i+1}^n, u_{i+2}^n, \dots)$$

only known values at the current time-level are required to update the solution at each point.
IMPLICIT schemes are fundamentally different, as they use 'unknown' values of the solution at the next time-level to compute the solution at the next time-level.
Consider the linear wave equation again. We approximated the derivatives for the explicit scheme at the current time-level as

$$\left.\frac{\partial u}{\partial t}\right|_i^n + c\left.\frac{\partial u}{\partial x}\right|_i^n = 0.$$

For implicit schemes the derivatives are approximated at the next time-level, i.e.

$$\left.\frac{\partial u}{\partial t}\right|_i^{n+1} + c\left.\frac{\partial u}{\partial x}\right|_i^{n+1} = 0.$$

Consider the first-order upwind scheme ($c > 0$). The explicit scheme gives

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

For implicit schemes we approximate derivatives at the next time level. But since a backward difference at the next time-level (n+1) is the same as a forward difference at the current time-level (n), we end up with the same temporal difference formula.
Expand about $i, n+1$:

$$u_i^n = u_i^{n+1} - \Delta t \left.\frac{\partial u}{\partial t}\right|_i^{n+1} + \frac{1}{2}(\Delta t)^2 \left.\frac{\partial^2 u}{\partial t^2}\right|_i^{n+1} + O(\Delta t^3)$$

and so negelcting terms of $O(\Delta t^2)$ we get

$$u_i^n = u_i^{n+1} - \Delta t \left.\frac{\partial u}{\partial t}\right|_i^{n+1} + O(\Delta t^2)$$

or

$$\left.\frac{\partial u}{\partial t}\right|_i^{n+1} = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

and so the implicit scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^{n+1} - u_{i-1}^{n+1}}{\Delta x} = 0.$$

This leads to a more complex equation to solve. The solution at each point cannot simply be updated using known values. At each time-step we get a set of equations which must be solved simultaneously. Consider the time-stepping scheme at each point. Again we use,

$$\nu_i = \frac{c\Delta t_i}{\Delta x_i}$$

and so the scheme is

$$u_i^{n+1} = u_i^n - \nu_i(u_i^{n+1} - u_{i-1}^{n+1})$$

To solve this for each point we get a matrix equation,

$$\begin{bmatrix} ? & ?, & 0, & 0, & .., & 0, & 0 & \\ -\nu_2, & (1+\nu_2), & 0, & 0, & .., & 0, & 0, & 0 \\ 0, & -\nu_3, & (1+\nu_3), & 0, & .., & 0, & 0, & 0 \\ 0, & 0, & -\nu_4, & (1+\nu_4), & .., & 0, & 0, & 0 \\ & & & . & & & & \\ & & & . & & & & \\ 0, & 0, & 0, & 0, & .., & -\nu_{NI-1}, & (1+\nu_{NI-1}), & 0 \\ 0, & 0, & 0, & 0, & .., & 0, & -\nu_{NI}, & (1+\nu_{NI}) \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ . \\ . \\ u_{NI-1}^{n+1} \\ u_{NI}^{n+1} \end{bmatrix} = \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ u_4^n \\ . \\ . \\ u_{NI-1}^n \\ u_{NI}^n \end{bmatrix}$$

This is a very simple matrix, bidiagonal. Most schemes lead to a much more complex matrix equation to solve, tri- or pentadiagonal or sometimes sparse. Exact matrix inversion is very expensive, and it is normally done using approximate inversion techniques. Normally have to consider operator splitting or approximate factorisation, and these will be considered later in the course. Remember, for 3-D equations the matrix is (NI.NJ.NK)x(NI.NJ.NK). For realistic mesh dimensions it would not be possible to construct the matrix on a single machine, as there would not be sufficient memory. Normally need multi-node parallelism just for the memory requirement.

So if they are so complex, why do we use implicit schemes ?

Stability of Implicit Methods

Using the Von-Neumann Fourier stability analysis again, with the implicit scheme

$$u_i^{n+1} = u_i^n - \nu(u_i^{n+1} - u_{i-1}^{n+1})$$

where as before

$$\nu = \frac{c\Delta t}{\Delta x}$$

Substituting our Fourier representation gives

$$u_i^{n+1} = u_i^n - \nu\left\{e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t} - e^{Ik(i-1)\Delta x}e^{-Ikc(n+1)\Delta t}\right\}$$

Expanding the highlighted term gives:

$$u_i^{n+1} = u_i^n - \nu\left\{e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t} - e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t}e^{-Ik\Delta x}\right\}$$

And then factorising out $\left(e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t}\right) = u_i^{n+1}$ gives:

$$u_i^{n+1} = u_i^n - \nu u_i^{n+1}\left\{1 - e^{-Ik\Delta x}\right\}$$

$$u_i^{n+1} = u_i^n - \nu u_i^{n+1} \left\{ 1 - e^{-Ik\Delta x} \right\}$$

Substitute $u_i^{n+1} = \lambda u_i^n$:

$$\lambda u_i^n = u_i^n - \nu \lambda u_i^n \left\{ 1 - e^{-Ik\Delta x} \right\}$$

Divide through by u_i^n to give

$$\lambda = 1 - \nu \lambda \left\{ 1 - e^{-Ik\Delta x} \right\}$$

Substitute with the complex trigonometric identity:

$$\lambda = 1 - \nu \lambda \left\{ 1 - \cos(k\Delta x) + I \sin(k\Delta x) \right\}$$

and rearrange:

$$1 = \lambda + \nu \lambda - \nu \lambda \cos(k\Delta x) + I \nu \lambda \sin(k\Delta x)$$

Easier to check the magnitude of the reciprocal:

$$\frac{1}{\lambda} = 1 + \nu - \nu \cos(k\Delta x) + I \nu \sin(k\Delta x)$$

$$\frac{1}{\lambda} = 1 + \nu - \nu\cos(k\Delta x) + I\nu\sin(k\Delta x)$$

Finally calculate the magnitude:

$$\left|\frac{1}{\lambda}\right|^2 = \{1 + \nu - \nu\cos(k\Delta x)\}^2 + \nu^2\sin^2(k\Delta x)$$

$$\left|\frac{1}{\lambda}\right|^2 = 1 + 2\nu + \nu^2 - 2\nu\cos(k\Delta x) - 2\nu^2\cos(k\Delta x)$$

$$+\nu^2\cos^2(k\Delta x) + \nu^2\sin^2(k\Delta x)$$

$$\left|\frac{1}{\lambda}\right| = \sqrt{1 + 2\left\{\nu + \nu^2\right\}\left\{1 - \cos(k\Delta x)\right\}}$$

$$\left|\frac{1}{\lambda}\right| = \sqrt{1 + 2\left\{\nu + \nu^2\right\}\left\{1 - \cos(k\Delta x)\right\}}$$

Hence, $\left|\frac{1}{\lambda}\right|$ can only be ≥ 1 and so

$$\lambda \leq 1$$

for all values of k, ν, i.e. the IMPLICIT method is unconditionally stable. Why is this ? Consider the difference between the stencils for explicit and implicit version of the first-order upwind scheme.

$$u_i^{n+1} = u_i^n - \nu(u_i^n - u_{i-1}^n) \quad \text{Explicit}$$

$$u_i^{n+1} = u_i^n - \nu(u_i^{n+1} - u_{i-1}^{n+1}) \quad \text{Implicit}$$

Explicit, each point only receives information from two points over one time step, i.e. i and $i-1$, so numerical domain of dependence is Δx. For implicit we have a matrix equation:

$$\mathbf{M}\underline{\mathbf{u}}^{n+1} = \underline{\mathbf{u}}^n$$

Premultiply by $\mathbf{M}^{-1}$,

$$\underline{\mathbf{u}}^{n+1} = \mathbf{M}^{-1}\underline{\mathbf{u}}^n$$

$$\underline{\mathbf{u}}^{n+1} = \mathbf{M}^{-1}\underline{\mathbf{u}}^{n}$$

Multiply out row by row,

$$u_1^{n+1} = M_{1,1}^{-1}u_1^n + M_{1,2}^{-1}u_2^n + M_{1,3}^{-1}u_3^n + ...M_{1,NI-1}^{-1}u_{NI-1}^n + M_{1,NI}^{-1}u_{NI}^n$$

$$u_2^{n+1} = M_{2,1}^{-1}u_1^n + M_{2,2}^{-1}u_2^n + M_{2,3}^{-1}u_3^n + ...M_{2,NI-1}^{-1}u_{NI-1}^n + M_{2,NI}^{-1}u_{NI}^n$$

$$u_3^{n+1} = M_{3,1}^{-1}u_1^n + M_{3,2}^{-1}u_2^n + M_{3,3}^{-1}u_3^n + ...M_{3,NI-1}^{-1}u_{NI-1}^n + M_{3,NI}^{-1}u_{NI}^n$$

ETC.

The stability limit for explicit schemes comes from the signal propagation. Each point only receives information from the points in the F.D. stencil, and this gives a limit on the time step. However, for the implicit scheme every point receives information from every other point at each time step, and so the numerical domain of dependence is infinite. There is no time-step limit for implicit schemes. Hence, they are usually used for steady flows, where we wish to get to the time-asymptotic solution as quickly as possible.

Lecture 8: Implicit Schemes

Part 2

Implicit Form of FTCS Scheme

Consider now an implicit form of the FTCS scheme. The explicit version is

$$u_i^{n+1} = u_i^n - \frac{\nu}{2}(u_{i+1}^n - u_{i-1}^n)$$

and this is unconditionally unstable. As done for the upwind scheme, we switch from a forward to a backward temporal derivative (so the approximation is the same as the explicit), and the spatial derivative is approximated implicitly, to give

$$u_i^{n+1} = u_i^n - \frac{\nu}{2}(u_{i+1}^{n+1} - u_{i-1}^{n+1})$$

Of course this is now Backward Time Centred Space at time-level n+1. Is this stable ? Same stability analysis as before

$$\lambda u_i^n = u_i^n - \frac{\nu}{2}\left\{e^{Ik(i+1)\Delta x}e^{-Ikc(n+1)\Delta t} - e^{Ik(i-1)\Delta x}e^{-Ikc(n+1)\Delta t}\right\}$$

$$\lambda u_i^n = u_i^n - \frac{\nu}{2}\left\{e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t}e^{Ik\Delta x} - e^{Iki\Delta x}e^{-Ikc(n+1)\Delta t}e^{-Ik\Delta x}\right\}$$

$$\lambda u_i^n = u_i^n - \frac{\nu}{2}\lambda u_i^n\left\{e^{Ik\Delta x} - e^{-Ik\Delta x}\right\}$$

Divide through by u_i^n to give

$$\lambda = 1 - \frac{\nu}{2}\lambda\left\{e^{Ik\Delta x} - e^{-Ik\Delta x}\right\}$$

or

$$\lambda = 1 - \frac{\nu}{2}\lambda\left\{\cos(k\Delta x) + I\sin(k\Delta x) - \cos(k\Delta x) + I\sin(k\Delta x)\right\}$$

$$\lambda = 1 - \frac{\nu}{2}\lambda 2I\sin(k\Delta x)$$

$$\frac{1}{\lambda} = 1 + I\nu\sin(k\Delta x)$$

Hence,

$$\left|\frac{1}{\lambda}\right| = \sqrt{1 + \nu^2\sin^2(k\Delta x)}$$

and so $\left|\frac{1}{\lambda}\right|$ can only be ≥ 1, or

$$\lambda \leq 1$$

for all values of k, ν. Hence, the IMPLICIT form of FTCS is unconditionally stable.

Implicit Form of General Equations

General equations are written in the form

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0.$$

This is approximated implicitly by

$$\left.\frac{\partial u}{\partial t}\right|_i^{n+1} + \left.\frac{\partial F(u)}{\partial x}\right|_i^{n+1} = 0.$$

Consider the implicit BTCS scheme for the general equation

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(F(u_{i+1}^{n+1}) - F(u_{i-1}^{n+1})\right)$$

which is normally written as

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(F_{i+1}^{n+1} - F_{i-1}^{n+1}\right)$$

So the problem now is, how do we approximate the flux function F^{n+1} ? The most common method is to use a Taylor series expansion for F about the nth time-level, i.e.

$$F_i^{n+1} = F_i^n + \Delta t \left.\frac{\partial F}{\partial t}\right|_i^n + O(\Delta t^2).$$

But to make this more useful we use

$$\left.\frac{\partial F}{\partial t}\right|_i^n = \left.\frac{\partial F}{\partial u}\right|_i^n \left.\frac{\partial u}{\partial t}\right|_i^n$$

Using the standard approximation

$$\left.\frac{\partial u}{\partial t}\right|_i^n = \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

and

$$\Delta u_i^n = u_i^{n+1} - u_i^n$$

we have

$$F_i^{n+1} = F_i^n + \Delta t \left.\frac{\partial F}{\partial u}\right|_i^n \frac{\Delta u_i^n}{\Delta t} + O(\Delta t^2).$$

or

$$F_i^{n+1} = F_i^n + \Delta u_i^n \left.\frac{\partial F}{\partial u}\right|_i^n + O(\Delta t^2).$$

Replacing this in the scheme gives Δu_i^n as the unknowns in a tridiagonal matrix equation. The term $\frac{\partial F}{\partial u}$ is the (scalar) Jacobian, J,

$$J_i^n = \left.\frac{\partial F}{\partial u}\right|_i^n$$

The scheme for each point is

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(F_{i+1}^{n+1} - F_{i-1}^{n+1}\right)$$

which becomes

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{2\Delta x}\left(F_{i+1}^n + J_{i+1}^n \Delta u_{i+1}^n - F_{i-1}^n - J_{i-1}^n \Delta u_{i-1}^n\right)$$

or

$$\Delta u_i^n = -\frac{\Delta t}{2\Delta x}\left(F_{i+1}^n + J_{i+1}^n \Delta u_{i+1}^n - F_{i-1}^n - J_{i-1}^n \Delta u_{i-1}^n\right)$$

This is reorganised as,

$$-\frac{\Delta t}{2\Delta x}J_{i-1}^n\Delta u_{i-1}^n + \Delta u_i^n + \frac{\Delta t}{2\Delta x}J_{i+1}^n\Delta u_{i+1}^n = -\frac{\Delta t}{2\Delta x}\left(F_{i+1}^n - F_{i-1}^n\right)$$

The right hand side term is normally known as the RESIDUAL. It is the effective error term. The matrix equation is then (written here for non-constant Δx and Δt)

$$\begin{bmatrix} ? & ?, & ?, & 0, & 0, & .., & 0, & 0, & 0 \\ a_2, & b_2, & c_2, & 0, & 0, & .., & 0, & 0, & 0 \\ 0, & a_3, & b_3, & c_3, & 0, & .., & 0, & 0, & 0 \\ 0, & 0, & a_4, & b_4, & c_4, & .., & 0, & 0, & 0 \\ & & & & . & & & & \\ & & & & . & & & & \\ & & & & . & & & & \\ 0, & 0, & 0, & 0, & 0, & .., & a_{NI-1}, & b_{NI-1} & c_{NI-1} \\ 0, & 0, & 0, & 0, & 0, & .., & ?, & ?, & ? \end{bmatrix} \begin{bmatrix} \Delta u_1^n \\ \Delta u_2^n \\ \Delta u_3^n \\ \Delta u_4^n \\ . \\ . \\ . \\ \Delta u_{NI-1}^n \\ \Delta u_{NI}^n \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} ? \\ \frac{\Delta t_2}{\Delta x_2}(F_3^n - F_1^n) \\ \frac{\Delta t_3}{\Delta x_3}(F_4^n - F_2^n) \\ \frac{\Delta t_4}{\Delta x_4}(F_5^n - F_3^n) \\ . \\ . \\ . \\ \frac{\Delta t_{NI-1}}{\Delta x_{NI-1}}(F_{NI}^n - F_{NI-2}^n) \\ ? \end{bmatrix}$$

where

$$a_i = -\frac{1}{2}\frac{\Delta t_i}{\Delta x_i}J_{i-1} \qquad b_i = 1.0 \qquad c_i = \frac{1}{2}\frac{\Delta t_i}{\Delta x_i}J_{i+1}$$

It is clear that the stencil has caused problems with the points at the boundaries. How do we treat them ?

General matrix equations, and their efficient solution, have been considered by many authors. The type of equation above was written by Robert MacCormack in the mid-eighties as,

$$\left[\quad NUMERICS \quad \right] . \left[\quad \Delta u \quad \right] = \left[\quad PHYSICS \quad \right]$$

The PHYSICS vector represents the 'forcing term'. It is the RESIDUAL term, the effective error. As the right hand side tends to zero, i.e. the error vanishes, then the changes to the soultion tend to zero. This term will normally be the same whether the scheme is explicit or implicit. This is because the residual (error) is always determined at the current time-level, i.e. using known values of F^n.

The NUMERICS matrix contains the discretisation terms, i.e. the elements depend on the discretisation we have chosen. The terms here will depend on whether the scheme is explicit or implicit. Generally, the more accurate the scheme, the more elements there are in the matrix. A second-order central-difference scheme in 1-D will be tridiagonal. A second-order upwind scheme will normally have a five-point stencil so will be pentadiagonal.

Summary

- Explicit schemes are cheap and easy to code, but suffer from the very restrictive CFL condition, limiting the time step that can be used.
- The alternative is to evaluate spatial and temporal gradients at the next time level $(n+1)$ instead of the current time level. This results in an IMPLICIT scheme.
- Implicit schemes are more complex, and require the solution of a matrix equation each time step, but the numerical domain of dependence is now infinite, as every point in the domain influences every other point every time step $\Rightarrow$ no time-step restriction.
- Implicit forms of first-order and FTCS schemes are both unconditionally stable.
- For general or systems of equations, the implicit scheme becomes more expensive, as each term in the matrix involves the Jacobian. This comes from the linearisation.
- The general implicit formulation results in a numerics matrix, including the implicit terms, and a RHS physics vector. The physics vector is the residual term at each point, and this is the same explicit or implicit.

END OF THEORY SECTION