Computation Fluid Dynamics

CFD I

Jitesh Gajjar

Maths Dept

Manchester University

Garbage In, Garbage Out

We will begin with a discussion of errors.
Useful to understand different types of errors which can arise when doing numerical computation.

- Roundoff errors

Garbage In, Garbage Out

We will begin with a discussion of errors.
Useful to understand different types of errors which can arise when doing numerical computation.

- Roundoff errors
- Errors in modelling

Garbage In, Garbage Out

We will begin with a discussion of errors.
Useful to understand different types of errors which can arise when doing numerical computation.

- Roundoff errors
- Errors in modelling
- Programming errors, ie bugs

Garbage In, Garbage Out

We will begin with a discussion of errors.
Useful to understand different types of errors which can arise when doing numerical computation.

- Roundoff errors
- Errors in modelling
- Programming errors, ie bugs
- Truncation, discretization errors.

Garbage In, Garbage Out

We will begin with a discussion of errors.
Useful to understand different types of errors which can arise when doing numerical computation.

- Roundoff errors
- Errors in modelling
- Programming errors, ie bugs
- Truncation, discretization errors.

Measures of error

Need some way to quantify errors. 2 useful measures.

Measures of error

Need some way to quantify errors. 2 useful measures.

- Absolute error
  If ϕ^* is an approximation to a quantity ϕ then the Absolute error is defined by

$$|\phi - \phi^*|$$

Measures of error

Need some way to quantify errors. 2 useful measures.

- Absolute error
  If ϕ^* is an approximation to a quantity ϕ then the Absolute error is defined by

$$|\phi - \phi^*|$$

- Relative error
  The Relative error is defined by

$$\frac{|\phi - \phi^*|}{|\phi|}, \quad \phi \neq 0$$

Roundoff errors

These arise when a computer is used for doing numerical computation.

Roundoff errors

These arise when a computer is used for doing numerical computation.

Example

Inexact representation of numbers eg, π, $\sqrt{2}$.

Roundoff errors

These arise when a computer is used for doing numerical computation.

Example

Inexact representation of numbers eg, π, $\sqrt{2}$.

Rounding and chopping errors.

Remember the way certain quantities are computed is under your control.

Errors in modelling

Example:

Replacing full N-S equations with Euler equations. Neglect of viscous terms means no matter how accurate the numerical solution, viscous effects will not be captured where important.

Programming errors, ie bugs

- These are all too familiar!

Programming errors, ie bugs

- These are all too familiar!
- The computer is only doing what you ask it to do.

Programming errors, ie bugs

- These are all too familiar!
- The computer is only doing what you ask it to do.
- Even NASA has made blunders.

Subtle errors

Suppose

$$\phi = O(10^{-8}), \quad \phi^* = O(10^{-8})$$

then something like

```
diff = MAX(ABS(phi-phistar))
tol= 1.e-6
IF(diff < tol) EXIT
```

in numerical codes will be wrong usage. The condition is always satisfied even though relative error is $O(1)$.

Truncation, discretization errors.

A pproximate

$$U_{xx} = f(x)$$

by

$$\frac{(u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} = f(x_i).$$

Truncation, discretization errors.

A pproximate

$$U_{xx} = f(x)$$

by

$$\frac{(u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} = f(x_i).$$

This gives rise to a truncation error ie

Truncation, discretization errors.

A pproximate

$$U_{xx} = f(x)$$

by

$$\frac{(u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} = f(x_i).$$

This gives rise to a truncation error ie

$$\tau(x_i) = U_{xx}(x_i) - f(x_i) = \frac{h^2}{12} U_{xxxx}(x_i) + \ldots$$

Initial value problems

Here we will look at the solution of ordinary differential equations of the type, say

$$\frac{dy}{dx} = f(x, y), \quad a \leq x \leq b,$$

subject to an initial condition

$$y(a) = \alpha$$

Example

Solve

$$\frac{dy}{dx} = y(1 - \frac{y}{4}), \quad 0 \leq x,$$

subject to an initial condition

$$y(0) = 1$$

Soln of ODE's

The methods also generalise to systems of equations i.e.

$$\frac{d\mathbf{Y}}{dx} = \mathbf{F}(x, \mathbf{Y}), \quad a \le x \le b,$$

where

$$\mathbf{Y} = (y_1(x), y_2(x), ..., y_N(x))^T,$$

$$\mathbf{F} = (f_1(x, \mathbf{Y}), f_2(x, \mathbf{Y}), ..., f_N(x, \mathbf{Y}))^T,$$

with initial data

$$\mathbf{Y}(a) = \alpha,$$

say, where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_N)^T$.

Example

Solve

$$y'' - 2xyy' + y^2 = 1, \quad y(1) = 1, \quad y'(1) = 2.$$

The equivalent first order system is obtained with

$$(y_1(x), y_2(x))^T = (y(x), y'(x))^T,$$

$$f_1(x, y_1, y_2) = y_2(x),$$

$$f_2(x, y_1, y_2) = 1 + 2xy_1(x)y_2(x) - y_1^2(x),$$

and initial condition

$$(y_1(1), y_2(1))^T = (1, 2)^T.$$

A mathematical result.

Suppose we define $\mathcal{D}$ to be the domain

$$\mathcal{D} = \{(x, y) \quad | \quad a \le x \le b, \quad -\infty < y < \infty\}$$

and $f(x, y)$ is continuous on $\mathcal{D}$. If $f(x, y)$ satisfies a Lipschitz condition on $\mathcal{D}$ then the ODE has a unique solution for $a \le x \le b$.

A mathematical result.

Suppose we define $\mathcal{D}$ to be the domain

$$\mathcal{D} = \{(x, y) \quad | \quad a \leq x \leq b, \quad -\infty < y < \infty\}$$

and $f(x, y)$ is continuous on $\mathcal{D}$. If $f(x, y)$ satisfies a Lipschitz condition on $\mathcal{D}$ then the ODE has a unique solution for $a \leq x \leq b$. Recall $f(x, y)$ satisfies a Lipschitz condition on $\mathcal{D}$ means that there exists a constant $L > 0$ (called the Lipschitz constant) such that

$$|f(x_1, y_1) - f(x_2, y_2)| \leq L|y_1 - y_2|$$

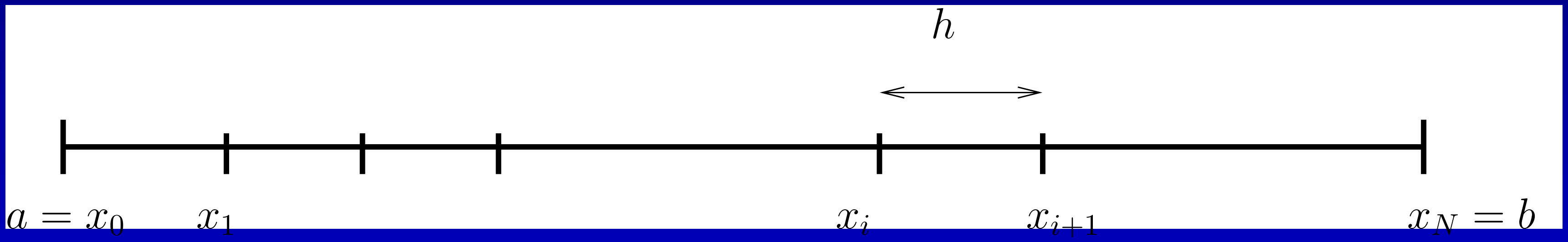
whenever $(x_1, y_1), (x_2, y_2)$ belong to $\mathcal{D}$.

Euler's Method

This is the simplest of techniques for the numerical solution of ODE's.
For simplicity define an equally spaced mesh

$$x_j = a + jh, \quad j = 0, .., N$$

where $h = (b - a)/N$ is called the step size.



We can derive Euler's method as follows.

Euler's Method

Suppose $y(x)$ is the unique solution to the ODE, and twice differentiable. Then by Taylor's theorem we have

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + y'(x_i)h + \frac{h^2}{2}y''(\xi)$$

where $x_i \leq \xi \leq x_{i+1}$.

Euler's Method

But from the differential equation $y'(x_i) = f(x_i)$, and $y_i = y(x_i)$.
This suggests the scheme

$$\begin{aligned} w_0 &= \alpha \\ w_{i+1} &= w_i + hf(x_i, w_i), \\ i &= 1, 2, .., N-1, \end{aligned}$$

for calculating the w_i.

This is Euler's method

Truncation error for Euler's method

Suppose that $y_i = y(x_i)$ is the exact solution at $x = x_i$.
Then the truncation error is defined by

$$\begin{aligned} \tau_{i+1}(h) &= \frac{y_{i+1} - (y_i + hf(x_i, y_i))}{h} \\ &= \frac{y_{i+1} - y_i}{h} - f(x_i, y_i), \end{aligned}$$

for $i = 0, 1, ..., N - 1$.

Truncation error

From the above we find that

$$\tau_{i+1}(h) = \frac{h}{2} y''(\xi_i)$$

for some ξ_i in (x_i, x_{i+1}). So if $y''(x)$ is bounded by a constant M in (a, b) then

$$|\tau_{i+1}(h)| \leq \frac{h}{2} M$$

.

Thus we see that the truncation error for Euler's method is $O(h)$.

In general if $\tau_{i+1} = h^p$ we say that the method is of order h^p.

In principle if h decreases, we should be able to achieve greater accuracy, although in practice round-off error limits the smallest size of h that we can take.

Higher order methods, Modified Euler

The modified Euler method is given by

$$
\begin{aligned}
w_0 &= \alpha \qquad k_1 = hf(x_i, w_i), \\
w_{i+1} &= w_i + \frac{h}{2}[f(x_i, w_i) + f(x_{i+1}, w_{i+1})], \\
i &= 1, 2, .., N-1,
\end{aligned}
$$

This has truncation error $O(h^2)$. Sometimes this is also called a Runge-Kutta method of order 2.

Modified Euler

Notice Euler's method is implicit

$$\begin{aligned} w_0 &= \alpha \qquad k_1 = hf(x_i, w_i), \\ w_{i+1} &= w_i + \frac{h}{2}[f(x_i, w_i) + f(x_{i+1}, w_{i+1})], \\ i &= 1, 2, .., N-1, \end{aligned}$$

Modified Euler

Notice Euler's method is implicit

$$
\begin{aligned}
w_0 &= \alpha \qquad k_1 = hf(x_i, w_i), \\
w_{i+1} &= w_i + \frac{h}{2}[f(x_i, w_i) + f(x_{i+1}, w_{i+1})], \\
i &= 1, 2, .., N-1,
\end{aligned}
$$

Thus some iteration may be necessary.

Runge-Kutta method of order 4

One of the most common Runge-Kutta methods of order 4 is given by

$$
\begin{aligned}
w_0 &= \alpha, \\
k_1 &= hf(x_i, w_i), \\
k_2 &= hf(x_i + \frac{h}{2}, w_i + \frac{1}{2}k_1) \\
k_3 &= hf(x_i + \frac{h}{2}, w_i + \frac{1}{2}k_2) \\
k_4 &= hf(x_{i+1}, w_i + k_3) \\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
\text{for} \quad i &= 0, 1, .., N - 1
\end{aligned}
$$

Systems of equations

All these methods generalise to a system of first order equations.

Thus for instance the RK(4) method above becomes

Runge-Kutta 4th order

$$
\begin{aligned}
\mathbf{w}_0 &= \alpha, \\
\mathbf{k}_1 &= h\mathbf{f}(x_i, \mathbf{w}_i), \\
\mathbf{k}_2 &= h\mathbf{f}(x_i + \frac{h}{2}, \mathbf{w}_i + \frac{1}{2}\mathbf{k}_1) \\
\mathbf{k}_3 &= h\mathbf{f}(x_i + \frac{h}{2}, \mathbf{w}_i + \frac{1}{2}\mathbf{k}_2) \\
\mathbf{k}_4 &= h\mathbf{f}(x_{i+1}, \mathbf{w}_i + \mathbf{k}_3) \\
\mathbf{w}_{i+1} &= \mathbf{w}_i + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \\
\text{for} \quad i &= 0, 1, .., N-1
\end{aligned}
$$

m-step multi-step method

The methods discussed above are called one-step methods.
Methods that use the approximate values at more than one previous mesh point are called multi-step methods.
There are two distinct types worth mentioning.

These are of the form

Two types

$$w_{i+1} = c_{m-1}w_i + c_{m-2}w_{i-1} + \ldots + c_0 w_{i+1-m}$$

$$+h[b_m f(x_{i+1}, w_{i+1}) + b_{m-1} f(x_i, w_i) +$$

$$\ldots + b_0 f(x_{i+1-m}, w_{i+1-m})]*$$

If $b_m = 0$ so that there is no term with w_{i+1} on the right hand side of (*), the method is **explicit.**

If $b_m \neq 0$ we have an **implicit** method.

Adams-Bashforth 4th order method (explicit)

Here we have

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3,$$

where these values are obtained using other methods such as RK(4) for instance. Then for $i = 3, 4, ..., N-1$ we use

$$\begin{aligned} w_{i+1} &= w_i + \frac{h}{24}[55f(x_i, w_i) \\ &- 59f(x_{i-1}, w_{i-1}) + 37f(x_{i-2}, w_{i-2}) - 9f(x_{i-3}, w_{i-3})]. \end{aligned}$$

Boundary Value Problems - Shooting Methods

Consider the differential equation

$$\frac{d^2y}{dx^2} + k\frac{dy}{dx} + xy = 0, \quad y(0) = 0, \quad y(1) = 1.$$

This is an example of a boundary value problem.

Why? Conditions have to satisfied at both ends.

BVP

If we write this as a system of first order equations we have

$$
\begin{aligned}
Y_1 &= y, \\
Y_2 &= \frac{dy}{dx}, \\
\frac{dY_1}{dx} &= Y_2, \\
\frac{dY_2}{dx} &= -kY_2 - xY_1.
\end{aligned}
$$

The boundary conditions give

$$Y_1(0) = 0, \quad Y_1(1) = 1.$$

We do not know the value of $Y_2(0)$.

BVP

Suppose we guess the value of $Y_2(0) = g$, say. Then we can integrate the system with the initial condition

$$\mathbf{Y}(0) = \begin{pmatrix} Y_1(0) \\ Y_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ g \end{pmatrix},$$

BVP

This will give us

$$\mathbf{Y}(1) = \begin{pmatrix} Y_1(1) \\ Y_2(1) \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix},$$

BVP

This will give us

$$\mathbf{Y}(1) = \begin{pmatrix} Y_1(1) \\ Y_2(1) \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix},$$

But β_1 will not necessarily satisfy the required condition $Y_1(1) = 1$,

So now need to iterative to try and get the correct value of g such that the required condition at $x = 1$ is satisfied.

BVP, shooting

To do this define

$$\phi(g) = Y_1(1; g) - 1,$$

We want to find the value of g such that $\phi(g) = 0$.

This gives rise to the idea of a *shooting method*.

Shooting Method

Suppose that we have a guess $\tilde{g}$ and we seek a correction dg such that $\phi(\tilde{g} + dg) = 0$.
By Taylor expansion we have

$$\phi(\tilde{g} + dg) = \phi(\tilde{g}) + \frac{d\phi}{dg}(\tilde{g})dg + O(dg^2).$$

This suggests that we take

$$dg = -\frac{\phi(\tilde{g})}{\phi'(\tilde{g})},$$

and hence a new value for g is $g + dg$.

Shooting- secant method

Hence

$$g_{n+1} = g_n - \frac{\phi(g_n)}{\phi'(g_n)},$$

Secant method

Now we are required to find $\phi'(g_n)$.
How can we do this? One way is to estimate $\phi'(g_n)$ by

$$\phi'(g_n) = \frac{\phi(g_n) - \phi(g_{n-1})}{g_n - g_{n-1}}.$$

This gives

$$g_{n+1} = g_n - \frac{\phi(g_n)(g_n - g_{n-1})}{\phi(g_n) - \phi(g_{n-1})},$$

which is known as the secant method.

Shooting- Newton's method

Consider again

$$\begin{aligned}\frac{dY_1}{dx} &= Y_2,\\ \frac{dY_2}{dx} &= -kY_2 - xY_1,\end{aligned}$$

with $\mathbf{Y}(0) = (0, g)^T$.
Now

$$\phi'(g_n) = \frac{\partial Y_1}{\partial g}(1; g),$$

Thus differentiate the original system of equations and boundary conditions with respect to g.

Shooting- Newton's method

$$\frac{d}{dx}\left(\frac{\partial \mathbf{Y}}{\partial g}\right) = \left(\begin{array}{c} \frac{\partial Y_2}{\partial g} \\ -k\frac{\partial Y_2}{\partial g} - x\frac{\partial Y_1}{\partial g} \end{array}\right),$$

$$\frac{d}{dx}\left(\frac{\partial \mathbf{Y}}{\partial g}\right)|_{x=0} = \left(\begin{array}{c} 0 \\ 1 \end{array}\right).$$

The system defines another initial value problem with given initial conditions.

Shooting- Newton's method

Note that

$$\phi'(g) = \frac{\partial Y_1}{\partial g}(1; g).$$

From the solution of the above we can extract $\frac{\partial Y_1}{\partial g}(x = 1)$ and hence compute dg update g.

This forms the basis of Newton's method combined with shooting, to solve boundary value problems.

Newton- augmented system

Can also use an augmented system where we define

$$Y_1 = y, \quad Y_2 = \frac{dy}{dx}, \quad Y_3 = \frac{\partial y}{\partial g} = \frac{\partial Y_1}{\partial g}, \quad Y_4 = \frac{\partial Y_2}{\partial g},$$

and then

$$\frac{d}{dx}\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix} = \begin{pmatrix} Y_2 \\ -kY_2 - xY_1 \\ Y_4 \\ -kY_4 - xY_3 \end{pmatrix}, \mathbf{Y}(0) = \begin{pmatrix} Y_1(0) \\ Y_2(0) \\ Y_3(0) \\ Y_4(0) \end{pmatrix} = \begin{pmatrix} 0 \\ g \\ 0 \\ 1 \end{pmatrix}$$

Multiple shooting

Consider

$$\frac{d^4y}{dx^4} = y^3 - (\frac{dy}{dx})^2,$$

$$y(0) = 1, \quad \frac{dy}{dx}(0) = 0, \quad y(1) = 2, \quad \frac{dy}{dx}(1) = 1,$$

We need two starting values at $x = 0$ and then we will have two conditions to satisfy at $x = 1$.

Multiple shooting

Define

$$Y_1 = y, Y_2 = y', Y_3 = y'', y_4 = y''',$$

We will need to guess for $Y_3(0) = e$, say, and $Y_4(0) = g$.

Multiple shooting

$$\phi_1(e, g) = Y_1(x = 1; e; g) - 2,$$

$$\phi_2(e, g) = Y_2(x = 1; e; g) - 1.$$

We need to iterative on both c and g to ensure that the remaining conditions are satisfied.

To find corrections, need Taylor expansion for function of two variables.

Multiple shooting

To obtain the corrections to guessed values $\tilde{e}$, $\tilde{g}$ we have

$$\begin{aligned}\phi_1(\tilde{e}+de,\tilde{g}+dg) &= 0=\phi_1(\tilde{e},\tilde{g})+de\frac{\partial\phi_1}{\partial e}(\tilde{e},\tilde{g})+dg\frac{\partial\phi_1}{\partial g}(\tilde{e},\\ \phi_2(\tilde{e}+de,\tilde{g}+dg) &= 0=\phi_2(\tilde{e},\tilde{g})+de\frac{\partial\phi_2}{\partial e}(\tilde{e},\tilde{g})+dg\frac{\partial\phi_2}{\partial g}(\tilde{e},\end{aligned}$$

Multiple shooting

Multidimensional case
Vector of guesses $\tilde{\mathbf{g}}$.
We can find the corrections $d\mathbf{g}$ as

$$d\mathbf{g} = -\mathbf{J}^{-1}(\tilde{\mathbf{g}})\phi(\tilde{\mathbf{g}}),$$

where $\mathbf{J}$ is the Jacobian $\frac{\partial \phi_i}{\partial g_k}$ and ϕ is the vector of conditions.

Richardson Extrapolation

Suppose that we use a method with truncation error of $O(h^m)$ to compute an approximation w_i.

We can use Richardson extrapolation to get an approximation with greater accuracy.

Richardson extrapolation

Suppose $w_i^{(1)}$ is approximation with step size h and $w_i^{(2)}$ with step size $2h$.
Then we can write

$$w_i^{(1)} = y_i + Eh^m + E_1 h^{m+1} + \ldots,$$

and

$$w_i^{(2)} = y_i + E(2h)^m + E_1 (2h)^{m+1} + \ldots.$$

Richardson extrapolation

Then we can eliminate the E term to get

$$2^m w_i^{(1)} - w_i^{(2)} = (2^m - 1)y_i + O(h^{m+1}).$$

Thus

$$w_i^* = \frac{2^m w_i^{(1)} - w_i^{(2)}}{2^m - 1}$$

is a more accurate approximationto the solution than $w_i^{(1)}$ or $w_i^{(2)}$.

Richardson extrapolation

Then we can eliminate the E term to get

$$2^m w_i^{(1)} - w_i^{(2)} = (2^m - 1)y_i + O(h^{m+1}).$$

Thus

$$w_i^* = \frac{2^m w_i^{(1)} - w_i^{(2)}}{2^m - 1}$$

is a more accurate approximationto the solution than $w_i^{(1)}$ or $w_i^{(2)}$. For a 4th order Runge-Kutta method the above gives

$$w_i^* = \frac{16 w_i^{(1)} - w_i^{(2)}}{15}.$$

Solution of BVP using finite-differences

Boundary value problems can also be tackled directly using finite-differences or some other technique such as spectral approximation.
We will look at one specific example with finite-differences.
Consider

$$\frac{d^2y}{dx^2} = \frac{1}{8}(32 + 2x^3 - y\frac{dy}{dx}), \qquad 1 \le x \le 3,$$

$$y(1) = 17, \qquad y(3) = \frac{43}{3}.$$

The exact solution is $y(x) = x^2 + (16/x)$.

Solution of BVP using finite-differences

Define a uniform grid $(x_0, x_1, ..., x_N)$ with $N + 1$ points.

Grid spacing $h = (x_N - x_0)/N$,

$x_j = x_0 + jh$, for $(j = 0, 1, .., N)$.

Solution of BVP using finite-differences

Approximate y at each of the nodes $x = x_i$ by w_i
The derivatives of y in the ode are approximated in finite-difference form as

$$\left(\frac{dy}{dx}\right)_{x=x_i} = \frac{w_{i+1} - w_{i-1}}{2h} + O(h^2),$$

$$\left(\frac{d^2y}{dx^2}\right)_{x=x_i} = \frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} + O(h^2).$$

Solution of BVP using finite-differences

These can be derived by making use of a Taylor expansion about the point $x = x_i$. Thus for example

$$y(x_{i+1}) \;=\; y(x_i) + h\frac{dy}{dx}(x_i) + \frac{h^2}{2}\frac{d^2y}{dx^2}(x_i) + \frac{h^3}{6}\frac{d^3y}{dx^3}(x_i) + \frac{h^4}{24}$$

$$y(x_{i-1}) \;=\; y(x_i) - h\frac{dy}{dx}(x_i) + \frac{h^2}{2}\frac{d^2y}{dx^2}(x_i) - \frac{h^3}{6}\frac{d^3y}{dx^3}(x_i) + \frac{h^4}{24}$$

By adding and subtracting and replacing $y(x_i)$ by w_i we obtain previous approximation.

Solution of BVP using finite-differences

Next replace y and its derivatives in ode by the above approximations to get

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} = 4 + \frac{x_i^3}{4} - w_i \left(\frac{w_{i+1} - w_{i-1}}{16h} \right),$$

$$\text{for} \quad (i = 1, 2, ...N - 1)$$

and

$$w_0 = 17, \qquad w_N = \frac{43}{3}.$$

The above equations are a set of nonlinear difference equations. We have $N + 1$ equations for $N + 1$ unknowns $w_0, ..., w_N$.

Solution of BVP using finite-differences

The nonlinear term above can now be tackled in many different ways. Thus for example we can replace it by

$$w_i^{(k-1)}\left(\frac{w_{i+1}^{(k)}-w_{i-1}^{(k)}}{16h}\right),$$

or

$$w_i^{(k)}\left(\frac{w_{i+1}^{(k-1)}-w_{i-1}^{(k-1)}}{16h}\right),$$

Newton linearization

Suppose that we have a guess for the solutions $w_i = W_i$.
We seek corrections δw_i such that the $w_i = W_i + \delta w_i$ satisfies the system.

Substituting $w_i = W_i + \delta w_i$ into equations and linearizing gives

BVP FD methods

$$\frac{\delta w_{i+1} - 2\delta w_i + \delta w_{i-1}}{h^2} = F_i -$$

$$\delta w_i \left(\frac{W_{i+1} - W_{i-1}}{16h} \right) - W_i \left(\frac{\delta w_{i+1} - \delta w_{i-1}}{16h} \right),$$

$$\text{for} \quad (i = 1, 2, ...N - 1)$$

and

$$\delta w_0 = F_0, \qquad \delta w_N = F_N.$$

BVP, FD methods

The techniques described above lead to the solution of a tridiagonal systems of linear equations of the form

$$\alpha_i w_{i-1} + \beta_i w_i + \gamma_i w_{i+1} = \delta_i, \quad i = 0, 1, .., N,$$

where the $\alpha_i, \beta_i, \gamma_i$ are coefficients obtainable from the difference equation.

BVP, FD methods

For example, we have

$$\alpha_i = \frac{1}{h^2} - \frac{W_i}{16h}, \quad \beta_i = -\frac{2}{h^2} + \frac{W_{i+1} - W_{i-1}}{16h},$$

$$\gamma_i = \frac{1}{h^2} + \frac{W_i}{16h}, \quad \delta_i = F_i, \quad i = 1, 2, .., N-1,$$

and

$$\beta_0 = 1, \gamma_0 = 0, \delta_0 = F_0,$$

$$\alpha_N = 0, \beta_N = 1, \delta_N = F_N.$$

Thomas's tridiagonal algorithm

This version of a tridiagonal solver is based on Gaussian elimination.
First we create zeros below the diagonal and then once we have a triangular matrix, we solve for the w_i using back substitution.

Thus the algorithm takes the form

Thomas's tridiagonal algorithm

$$\beta_j = \beta_j - \frac{\gamma_{j-1}\alpha_j}{\beta_{j-1}} \qquad j = 1, 2, 3, ..., N,$$

$$\delta_j = \delta_j - \frac{\delta_{j-1}\alpha_j}{\beta_{j-1}}, \qquad j = 1, 2, 3, ..., N,$$

$$w_N = \frac{\delta_N}{\beta_N}, \quad w_j = \frac{(\delta_j - \gamma_j w_{j+1})}{\beta_j},$$

$$j = N-1, ..., 1, 0.$$

Stability

In practice most initial value integrators should work reasonably well on standard problems. However certain types of problems (**stiff** problems) can cause difficulty and care needs to be exercised in the choice of the method.

Stability -Consistency

A method is said to be consistent if the local truncation error tends to zero as the step size $\to 0$, i.e

$$\lim_{h \to 0} \max_i |\tau_i(h)| = 0.$$

Stability -Convergence

A method is said to be convergent with respect to the equation it approximates if

$$\lim_{h\to 0} \max_i |w_i - y(x_i)| = 0,$$

where $y(x)$ is the exact solution and w_i an approximation produced by the method.

Stability, Theorem

It can be proven that if the difference method is consistent with the differential equation, then the method is stable if and only if the method is convergent.

Stability of m-step methods

If we consider an m-step method

$$w_0 = \alpha_0, \quad w_1 = \alpha_1, \quad \ldots, w_{m-1} = \alpha_{m-1},$$
$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + ... + a_0 w_{i+1-m}$$
$$+h[F(x_i, w_{i+1}, w_i, ..., w_{i+1-m})]$$

Stability of m-step methods

If we consider an m-step method

$$
\begin{aligned}
w_0 = \alpha_0, \quad w_1 = \alpha_1, \quad \ldots, w_{m-1} = \alpha_{m-1}, \\
w_{i+1} = a_{m-1} w_i + a_{m-2} w_{i-1} + \ldots + a_0 w_{i+1-m} \\
+ h[F(x_i, w_{i+1}, w_i, \ldots, w_{i+1-m})]
\end{aligned}
$$

Then ignoring the F term the homogenous part is just a difference equation.

Stability of m-step method

The stability is thus connected with the the roots of the characteristic polynomial

$$\lambda^m - a_{m-1}\lambda^{m-1} - \dots - a_1\lambda - a_0 = 0.$$

Why?

Stability of m-step method

Consider the ODE with $f(x, y) = 0$.

$$\frac{dy}{dx} = f(x, y), \quad a \leq x \leq b, \quad y(a) = \alpha$$

This has the solution $y(x) = \alpha$.

The difference equation has to produce the same solution, ie $w_n = \alpha$.

Stability of m-step method

Next consider

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \ldots + a_0 w_{i+1-m}.$$

Stability of m-step method

Next consider

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + ... + a_0 w_{i+1-m}.$$

If we look for solutions of the form $w_n = \lambda^n$ then this gives the characteristic polynomial equation

Stability of m-step method

Next consider

$$w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + ... + a_0 w_{i+1-m}.$$

If we look for solutions of the form $w_n = \lambda^n$ then this gives the characteristic polynomial equation

$$\lambda^m - a_{m-1}\lambda^{m-1} - ... - a_1\lambda - a_0 = 0.$$

Stability of m-step method

Suppose $\lambda_1, ...\lambda_m$ are distinct roots of characteristic polynomial.
Then we can write

Stability of m-step method

Suppose $\lambda_1, \ldots \lambda_m$ are distinct roots of characteristic polynomial.
Then we can write

$$w_n = \sum_{i=1}^{m} c_i \lambda_i^m.$$

Stability of m-step method

Suppose $\lambda_1, ...\lambda_m$ are distinct roots of characteristic polynomial.
Then we can write

$$w_n = \sum_{i=1}^{m} c_i \lambda_i^m.$$

Since $w_n = \alpha$ is a solution, the difference equation gives

$$\alpha - a_{m-1}\alpha - ... - a_0\alpha = 0,$$

or

$$\alpha(1 - a_{m-1} - ... - a_0) = 0.$$

Stability of m-step method

Thus

$$w_n = \alpha + \sum_{i=2}^{m} c_i \lambda_i^n.$$

Stability of m-step method

Thus

$$w_n = \alpha + \sum_{i=2}^{m} c_i \lambda_i^n.$$

In the absence of round-off error all the c_i would be zero.

Stability of m-step method

Thus

$$w_n = \alpha + \sum_{i=2}^{m} c_i \lambda_i^n.$$

In the absence of round-off error all the c_i would be zero.

If $|\lambda_i| \leq 1$ then the error due to roundoff will not grow.

Hence the method is stable if $|\lambda_i| \leq 1$.

Stability of m-step method

Is it enough just to have stability as defined above?

Stability of m-step method

Is it enough just to have stability as defined above? Consider the solution of

$$\frac{dy}{dx} = -30y, \quad y(0) = 1/3.$$

Satbility of m-step method

The RK(4) method, although stable, has difficulty in computing the accurate solution of this problem.

This means that we need something more than just the idea of stability defined above.

Absolute stability

Consider

$$\frac{dy}{dx} = ky, \quad y(0) = \alpha, \quad k < 0.$$

The exact solution of this is $y(x) = \alpha e^{kx}$.

Absolute stability

Consider

$$\frac{dy}{dx} = ky, \quad y(0) = \alpha, \quad k < 0.$$

The exact solution of this is $y(x) = \alpha e^{kx}$. If we take our one-step method and apply it to this equation we obtain

$$w_{i+1} = Q(hk)w_i.$$

Absolute Stability

Similarly a multi-step of the type used earlier, when applied to the test equation gives

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + ... + a_0 w_{i+1-m} \\ &+ h[b_m k w_{i+1} + b_{m-1} k w_i + ... + b_0 k w_{i+1-m}]. \end{aligned}$$

Absolute Stability

Thus if we seek solutions of the form $w_i = z^i$ this will give rise to the characteristic polynomial equation

$$Q(z, hk) = 0,$$

where

Absolute Stability

Thus if we seek solutions of the form $w_i = z^i$ this will give rise to the characteristic polynomial equation

$$Q(z, hk) = 0,$$

where

$$Q(z, hk) = (1 - hkb_m)z^m - (a_{m-1} + hkb_{m-1})z^{m-1}$$

$$-\ldots - (a_0 + hkb_0).$$

Absolute Stability

The **region R of absolute stability** for a one-step method is defined as the region in the complex plane R$= \{hk \in \mathcal{C}, |Q(hk)| < 1\}$.

Absolute Stability

The **region R of absolute stability** for a one-step method is defined as the region in the complex plane R$= \{hk \in \mathcal{C}, |Q(hk)| < 1\}$.

For a multi-step method $\mathrm{R} = \{hk \in \mathcal{C}, |\beta_j| < 1\}$, where β_j is a root of $Q(z, hk) = 0$.

Absolute Stability

A numerical method is A-stable if R contains the entire left half plane.

Consider the modified Euler method

$$w_0 = \alpha \qquad \begin{aligned} k_1 &= hf(x_i, w_i), \\ w_{i+1} &= w_i + \frac{h}{2}[f(x_i, w_i) + f(x_{i+1}, w_{i+1})], \end{aligned}$$

This is an A-stable method.

Numerical Solution of PDEs

Classification of PDE's

Partial differential equations can be classified as being of type elliptic, parabolic or hyperbolic. In some cases equations can be of mixed type.
Consider

$$A\frac{\partial^2\phi}{\partial x^2} + B\frac{\partial^2\phi}{\partial x\partial y} + C\frac{\partial^2\phi}{\partial y^2} + D\frac{\partial\phi}{\partial x} + E\frac{\partial\phi}{\partial y} + F\phi + G = 0,$$

where, in general, A, B, C, D, E, F, and G are functions of the independent variables x and y and of the dependent variables ϕ.

PDE's Classification

The equation is said to be

- elliptic if $B^2 - 4AC < 0$,

PDE's Classification

The equation is said to be

- elliptic if $B^2 - 4AC < 0$,
- parabolic if $B^2 - 4AC = 0$, or

PDE's Classification

The equation is said to be

- elliptic if $B^2 - 4AC < 0$,
- parabolic if $B^2 - 4AC = 0$, or
- hyperbolic if $B^2 - 4AC > 0$.

PDE's Classification

The equation is said to be

- elliptic if $B^2 - 4AC < 0$,
- parabolic if $B^2 - 4AC = 0$, or
- hyperbolic if $B^2 - 4AC > 0$.

Classification

An example of an elliptic equation is Poisson's equation

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = f(x, y).$$

Classification

The heat equation

$$\frac{\partial \phi}{\partial t} = k \frac{\partial^2 \phi}{\partial x^2}$$

is of parabolic type, and the wave equation

$$\frac{\partial^2 \phi}{\partial x^2} - \frac{\partial^2 \phi}{\partial y^2} = 0$$

is a hyperbolic pde.

Classification

An example of a mixed type equation is the transonic small disturbance equation given by

$$\left(K - \frac{\partial \phi}{\partial x}\right)\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0.$$

Classification 2

Consider a system of first order partial differential equations.

Unknowns $\mathbf{U} = (u_1, u_2, ..., u_n)^T$

Independent variables $\mathbf{x} = (x_1, x_2, ..., x_m)^T$.

Classification 2

Suppose that the equations can be written in quasi-linear form

$$\sum_{k=1}^{m} \mathbf{A}_k \frac{\partial \mathbf{U}}{\partial x_k} = \mathbf{Q}$$

where the $\mathbf{A}_k$ are $(n \times n)$ matrices and $\mathbf{Q}$ is an $(n \times 1)$ column vector, and both can depend on x_k and $\mathbf{U}$ but not on the derivatives of $\mathbf{U}$.

Classification 2

If we seek plane wave solutions of the homogeneous part of the above pde in the form

$$\mathbf{U} = \mathbf{U}_o e^{i\mathbf{x}.\mathbf{s}},$$

where $\mathbf{s} = (s_1, s_2, .., s_m)^T$, then

$$i\left[\sum_{k=1}^{m} \mathbf{A}_k s_k\right]\mathbf{U} = \mathbf{0}.$$

Classification 2

This will have a non-trivial solution only if the characteristic equation

$$\det \left| \sum_{k=1}^{m} \mathbf{A}_k s_k \right| = 0,$$

Classification 2

- The system is hyperbolic if n real characteristics exist.

Classification 2

- The system is hyperbolic if n real characteristics exist.
- If all the characteristics are complex, the system is elliptic.

Classification 2

- The system is hyperbolic if n real characteristics exist.
- If all the characteristics are complex, the system is elliptic.
- If some are real and some complex, the system is of mixed type.

Classification 2

- The system is hyperbolic if n real characteristics exist.
- If all the characteristics are complex, the system is elliptic.
- If some are real and some complex, the system is of mixed type.
- If the system is of rank less than n, then we have a parabolic system.

Classification 2

- The system is hyperbolic if n real characteristics exist.
- If all the characteristics are complex, the system is elliptic.
- If some are real and some complex, the system is of mixed type.
- If the system is of rank less than n, then we have a parabolic system.

Consistency, convergence and Lax equivalence theorem

Consistent

A discrete approximation to a partial differential equation is said to be consistent if in the limit of the stepsize(s) going to zero, the original pde system is recovered, ie the truncation error approaches zero.

Consistency, convergence and Lax equivalence theorem

Stability

If we define the error to be the difference between the computed solutions and the exact solution of the discrete approximation, then the scheme is stable if the error remains uniformly bounded for successive iterations.

Consistency, convergence and Lax equivalence theorem

Convergence

A scheme is stable if the solution of the discrete equations approaches the solution of the pde in the limit that the step-sizes approach zero.

Lax's Equivalence Theorem

For a well posed initial-value problem and a consistent discretization, stability is the necessary and sufficient condition for convergence.

Difference formulae

Suppose that we have a grid of points with equal mesh spacing Δ_x in the $x-$ direction and equal spacing Δ_y in the $y-$ direction.
Thus we can define points x_i, y_j by

$$x_i = x_0 + i\Delta_x, \quad y_j = y_0 + j\Delta_y.$$

Difference formulae

Suppose that we are trying to approximate a derivative of a function $\phi(x, y)$ at the points x_i, y_j.

Denote the approximate value of $\phi(x, y)$ at the point x_i, y_j by $w_{i,j}$ say.

Central Differences

The first and second derivatives in x or y may be approximated as before by

$$\left(\frac{\partial^2\phi}{\partial x^2}\right)_{ij} = \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta_x)^2} + O((\Delta_x)^2),$$

$$\left(\frac{\partial^2\phi}{\partial y^2}\right)_{ij} = \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta_y)^2} + O((\Delta_y)^2),$$

$$\left(\frac{\partial\phi}{\partial x}\right)_{ij} = \frac{w_{i+1,j} - w_{i-1,j}}{2\Delta_x} + O((\Delta_x)^2),$$

$$\left(\frac{\partial\phi}{\partial y}\right)_{ij} = \frac{w_{i,j+1} - w_{i,j-1}}{2\Delta_y} + O((\Delta_y)^2).$$

Central Differences

The approximations listed above are *centered* at the points (x_i, y_j), and are called central-difference approximations.

$$\begin{array}{ccc} \bullet & \times & \bullet \\ i-1,j & i,j & i+1,j \end{array}$$

One-sided approximations

We can also construct one-sided approximations to derivatives. Thus for example a second-order *forward* approximation to $\frac{\partial \phi}{\partial x}$ at the point (x_i, y_j) is given by

$$\left(\frac{\partial \phi}{\partial x}\right)_{ij} = \frac{-3w_{i,j} + 4w_{i+1,j} - w_{i+2,j}}{2\Delta_x}.$$

Weights for central differences

Order of Accuracy	$i-2$	$i-1$	i	$i+1$	$i+2$
	Node Points				
1st derivative					
$(\Delta_x)^2$		$-\frac{1}{2}$	0	$\frac{1}{2}$	
$(\Delta_x)^4$	$\frac{1}{12}$	$-\frac{2}{3}$	0	$\frac{2}{3}$	$-\frac{1}{12}$
2nd derivative					
$(\Delta_x)^2$		1	-2	1	
$(\Delta_x)^4$	$-\frac{1}{12}$	$\frac{4}{3}$	$-\frac{5}{2}$	$\frac{4}{3}$	$-\frac{1}{12}$

Weights for one-sided differences

Node Points					
Order of Accuracy	i	$i+1$	$i+2$	$i+3$	$i+4$
1st derivative					
(Δ_x)	-1	1			
$(\Delta_x)^2$	$-\frac{3}{2}$	2	$-\frac{1}{2}$		
$(\Delta_x)^3$	$-\frac{11}{6}$	3	$-\frac{3}{2}$	$\frac{1}{3}$	
$(\Delta_x)^4$	$-\frac{25}{12}$	4	-3	$\frac{4}{3}$	$-\frac{1}{4}$
2nd derivative					
(Δ_x)	1	-2	1		
$(\Delta_x)^2$	2	-5	4	-1	
$(\Delta_x)^3$	35	26	19	14	11

Mixed derivatives

For finding suitable discrete approximations for mixed derivatives use a multidimensional Taylor expansion.

Thus for example second order approximations to $\partial^2\phi/\partial x\partial y$ at the point i, j are given

$$\frac{\partial^2\phi}{\partial x\partial y} = \frac{w_{i+1,j+1} - w_{i-1,j+1} + w_{i-1,j-1} - w_{i+1,j-1}}{4\Delta_x\Delta_y} + O((\Delta_x)^2, (\Delta$$

In stencil form we can express this as

$$\frac{1}{4\Delta_x\Delta_y}\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}.$$

Mixed derivatives

Alternatively,

$$\frac{\partial^2 \phi}{\partial x \partial y} =$$

$$\frac{w_{i+1,j+1} - w_{i+1,j} - w_{i,j+1} + w_{i-1,j-1} - w_{i-1,j} - w_{i,j-1} + 2w_{i,j}}{2\Delta_x \Delta_y} +$$

or

$$\frac{1}{2\Delta_x \Delta_y} \begin{pmatrix} 0 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 0 \end{pmatrix}.$$

Central, one-sided differences

Consider the approximation

$$\left(\frac{\partial \phi}{\partial x}\right)_{ij} = \frac{w_{i+1,j} - w_{i,j}}{\Delta_x}.$$

By Taylor expansion we see that this gives rise to a truncation error of $O(\Delta_x)$. In addition this approximation is centered at the point $x_{i+\frac{1}{2},j}$.

Solution of elliptic pde's

A prototype elliptic pde is Poisson's equation given by

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = f(x, y),$$

where $f(x, y)$ is a known/given function. The equation has to be solved in a domain $\mathcal{D}$

Boundary Conditions

Boundary conditions are given on the boundary $\delta\mathcal{D}$ of $\mathcal{D}$.

Boundary Conditions

These can be of three types:

- Dirichlet $\quad \phi = g(x, y)$ on $\delta\mathcal{D}$.

Boundary Conditions

These can be of three types:

- Dirichlet $\phi = g(x, y)$ on $\delta\mathcal{D}$.
- Neumann $\dfrac{\partial\phi}{\partial n} = g(x, y)$ on $\delta\mathcal{D}$.

Boundary Conditions

These can be of three types:

- Dirichlet $\phi = g(x, y)$ on $\delta\mathcal{D}$.
- Neumann $\dfrac{\partial \phi}{\partial n} = g(x, y)$ on $\delta\mathcal{D}$.
- Robin/Mixed $\mathcal{B}(\phi, \dfrac{\partial \phi}{\partial n}) = 0$ on $\delta\mathcal{D}$.

  Robin boundary conditions involve a linear combination of ϕ and its normal derivative on the boundary.

  Mixed boundary conditions involve different conditions for one part of the boundary, and another type for other parts of the boundary.

Solution of model problem

Let us consider a model problem with

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = f(x, y), \quad 0 < x, y < 1$$

$$\phi = 0 \quad \text{on} \quad \delta\mathcal{D}.$$

Here the domain $\mathcal{D}$ is the square region $0 < x < 1$ and $0 < y < 1$.

Solution of model problem

Construct a finite difference mesh with points (x_i, y_j), say where

$$x_i = i\Delta_x, \quad i = 0, 1, ..., N, \quad y_j = j\Delta_y, \quad j = 0, 1, ...$$

where $\Delta_x = 1/N$, and $\Delta_y = 1/M$ are the step sizes in the x and y directions.

Solution of model problem

Next replace the derivatives in Poisson equation by the discrete approximations to get

$$\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta_x)^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta_y)^2} = f_{i,j},$$

$$1 \leq i \leq N-1, \quad 1 \leq j \leq M-1$$

and

$$w_{i,j} = 0, \quad \text{if} \quad i = 1, N, \quad 1 \leq j \leq M,$$

$$w_{i,j} = 0, \quad \text{if} \quad j = 1, M, \quad 1 \leq i \leq N.$$

Solution of model problem

Thus we have $(N-1)\times(M-1)$ unknown values $w_{i,j}$ to find at the interior points of the domain.

If we write

$$\mathbf{w}_i = (w_{i,1}, w_{i,2}, ..., w_{i,M-1})^T$$

and

$$\mathbf{f}_i = (f_{i,1}, f_{i,2}, ..., f_{i,M-1})^T$$

we can write the above system of equations in matrix form as

Solution of model problem

$$\begin{bmatrix} \mathbf{B} & \mathbf{I} & & & & \\ \mathbf{I} & \mathbf{B} & \mathbf{I} & & & \\ & \mathbf{I} & \mathbf{B} & \mathbf{I} & & \\ & & & & & \\ & & & & & \\ & & & & \mathbf{I} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \\ \\ \mathbf{w}_{N-1} \end{bmatrix} = (\Delta_x)^2 \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \\ \\ \mathbf{f}_{N-1} \end{bmatrix}$$

Solution of model problem

In the above $\mathbf{I}$ is the $(M-1)\times(M-1)$ identity matrix and

$$\mathbf{B} = \begin{bmatrix} b & c & & & & & \\ c & b & c & & & & \\ 0 & c & b & c & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \cdot & \\ & & & & & c & b \end{bmatrix},$$

with

$$b = -2(\frac{1}{(\Delta_x)^2} + \frac{1}{+(\Delta_y)^2}), \quad c = 1.$$

Solution of model problem

Let us write the linear system as

$$\mathbf{A}\mathbf{w} = \mathbf{f}$$

What we observe is that the matrix $\mathbf{A}$ is very sparse.
The matrix $\mathbf{A}$ is very large.

For instance with $N = M = 101$ the linear system is of size $10^4 \times 10^4$ and we have 10^4 unknowns to find.

Solution of linear system

The linear system can be solved using

- Direct Methods These are not efficient if $\mathbf{A}$ is large. Also very expensive, and require large storage.

Solution of linear system

The linear system can be solved using

- Direct Methods These are not efficient if **A** is large. Also very expensive, and require large storage.
- Iterative Methods The method of choice for most applications.

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR
- Line Relaxation

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR
- Line Relaxation
- Conjugate gradient and PCG methods

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR
- Line Relaxation
- Conjugate gradient and PCG methods
- Minimal residual methods

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR
- Line Relaxation
- Conjugate gradient and PCG methods
- Minimal residual methods
- Multigrid

Iterative methods

- Point relaxation. Jacobi, Gauss-Seidel, SOR
- Line Relaxation
- Conjugate gradient and PCG methods
- Minimal residual methods
- Multigrid
- Fast Direct Methods (FFT)

Iterative methods- Jacobi method

Rewrite the discrete equations

$$\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta_x)^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta_y)^2} = f_{i,j},$$

$$1 \le i \le N-1, \quad 1 \le j \le M-1$$

as

Iterative methods- Jacobi method

Rewrite the discrete equations

$$\frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{(\Delta_x)^2} + \frac{w_{i,j+1} - 2w_{i,j} + w_{i,j-1}}{(\Delta_y)^2} = f_{i,j},$$

$$1 \leq i \leq N-1, \quad 1 \leq j \leq M-1$$

as

$$w_{i,j} = \frac{1}{2(1+\beta^2)} \left(w_{i+1,j} + w_{i-1,j} + \beta^2 (w_{i,j+1} + w_{i,j-1}) - (\Delta_x)^2 \right.$$

with $\beta = \Delta_x / \Delta_y$.

Iterative methods- Jacobi method

This suggests the iterative scheme

$$w_{i,j}^{\text{new}} = \frac{1}{2(1+\beta^2)}\left(w_{i+1,j}^{\text{old}} + w_{i-1,j}^{\text{old}} + \beta^2(w_{i,j+1}^{\text{old}} + w_{i,j-1}^{\text{old}}) - (\Delta_x)^2\right.$$

Iterative methods- Jacobi method

What are suitable convergence criteria?

Iterative methods- Jacobi method

What are suitable convergence criteria? Suppose we write the linear system as

$$\mathbf{A}\mathbf{v} = \mathbf{f}$$

where $\mathbf{v}$ is the exact solution of the linear system. ’ If $\mathbf{w}$ is an approximate solution, the error $\mathbf{e}$ is defined by

$$\mathbf{e} = \mathbf{v} - \mathbf{w}.$$

Thus

$$\mathbf{A}\mathbf{e} = \mathbf{A}(\mathbf{v} - \mathbf{w}) = \mathbf{f} - \mathbf{A}\mathbf{w}.$$

Iterative methods- Jacobi method

Continue iterating until residual is small enough. The residual is defined by

$$\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{w}$$

which can be computed.

Iterative methods- Jacobi method

Continue iterating until residual is small enough. The residual is defined by

$$\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{w}$$

which can be computed. For the Jacobi scheme the residual is given by

$$r_{i,j} = (\Delta_x)^2 f_{i,j} + 2(1+\beta^2) w_{i,j} - (w_{i+1,j} + w_{i-1,j} + \beta^2 (w_{i,j+1} + w_{i,j-$$

Iterative methods- Jacobi method

Therefore a suitable stopping condition might be

$$\max_{i,j} |r_{i,j}| < \epsilon_1, \qquad \text{or} \qquad \sqrt{\sum_{i,j} r_{i,j}^2} < \epsilon_2.$$

Gauss-Siedel iteration

This is given by

$$w_{i,j}^{\text{new}} = \frac{1}{2(1+\beta^2)}\left(w_{i+1,j}^{\text{old}} + w_{i-1,j}^{\text{new}} + \beta^2(w_{i,j+1}^{\text{old}} + w_{i,j-1}^{\text{new}}\right) - (\Delta_x)^2$$

where the new values overwrite existing values.

Relaxation and the SOR method

Instead of updating the new values as indicated above, it is better to use relaxation. Here we compute

$$w_{i,j} = (1-\omega)w_{i,j}^{\mathrm{old}} + \omega w_{i,j}^{*},$$

where ω is called the relaxation factor, and $w_{i,j}^{*}$ denotes the value as computed by the Jacobi, or Gauss-Seidel scheme. $\omega = 1$ reduces to the Jacobi or Gauss-Seidel scheme.

The Gauss-Seidel scheme with $\omega \neq 1$ is called the method of successive overrelaxation or SOR scheme.

Line relaxation

The Jacobi, Gauss-Seidel and SOR schemes are called point relaxation methods.

On the other hand we may compute a whole line of new values at a time, leading to the line-relaxation methods.

Line Relaxation

For instance suppose we write the equations as

$$w_{i+1,j}^{\text{new}} - 2(1+\beta^2) w_{i,j}^{\text{new}} + w_{i-1,j}^{\text{new}} = -\beta^2 (w_{i,j+1} + w_{i,j-1}^{\text{old}})) + (\Delta_x)^2 f_{i,j}$$

then starting from $j = 1$ we may compute the values $w_{i,j}$, for $i = 1, ..., N-1$ in one go.

To solve for a line we need a tridiagonal solver.

Convergence properties of basic iteration schemes

Consider the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{-32}$$

where $\mathbf{A} = (a_{i,j})$ is an $n \times n$ matrix, and $\mathbf{x}$, $\mathbf{b}$ are $n \times 1$ column vectors.

Convergence properties

Suppose we write the matrix $\mathbf{A}$ in the form $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$ where $\mathbf{D}$, $\mathbf{L}$, $\mathbf{U}$ are the diagonal matrix, lower and upper triangular parts of $\mathbf{A}$, ie

$$\mathbf{D} = \begin{bmatrix} a_{1,1} & 0 & & & & & \\ 0 & a_{2,2} & 0 & & & & \\ 0 & 0 & a_{3,3} & 0 & & & \\ & & & . & & & \\ & & & & . & & \\ & & & & & . & \\ & & & & & 0 & a_{n,n} \end{bmatrix},$$

Convergence properties

$$-\mathbf{L} = \begin{bmatrix} 0 & 0 & & & & & \\ a_{2,1} & 0 & 0 & & & & \\ a_{3,1} & a_{3,2} & 0 & 0 & & & \\ & & & . & & & \\ & & & & . & & \\ & & & & & . & \\ a_{n,1} & a_{n,2} & & & & a_{n,n-1} & 0 \end{bmatrix},$$

$$-\mathbf{U} = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & . & . & . & a_{1,n} \\ 0 & 0 & a_{2,3} & . & . & . & a_{2,n} \\ 0 & 0 & 0 & a_{3,4} & . & . & a_{3,n} \\ & & & . & & & \\ & & & & . & & \\ & & & & & 0 & a_{n-1,n} \\ 0 & 0 & & & & & 0 \end{bmatrix}.$$

Convergence properties

Then

$$\mathbf{Ax} = \mathbf{b}$$

can be written as

$$\mathbf{Dx} = (\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{b}.$$

Convergence properties

The Jacobi iteration is then defined as

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}.$$

Convergence properties

The Jacobi iteration is then defined as

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}.$$

The Gauss-Seidel iteration is defined by

$$(\mathbf{D} - \mathbf{L})\mathbf{x}^{(k+1)} = \mathbf{U}\mathbf{x}^{(k)} + \mathbf{b},$$

or

$$\mathbf{x}^{(k+1)} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}\mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}.$$

Convergence properties

In general an iteration scheme may be written as

$$\mathbf{x}^{(k+1)} = \mathbf{P}\mathbf{x}^{(k)} + \mathbf{P}\mathbf{b}$$

where $\mathbf{P}$ is called the iteration matrix.

Convergence properties

In general an iteration scheme may be written as

$$\mathbf{x}^{(k+1)} = \mathbf{P}\mathbf{x}^{(k)} + \mathbf{P}\mathbf{b}$$

where $\mathbf{P}$ is called the iteration matrix.
For the Jacobi scheme scheme we have

$$\mathbf{P} = \mathbf{P}_J = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$$

Convergence properties

In general an iteration scheme may be written as

$$\mathbf{x}^{(k+1)} = \mathbf{P}\mathbf{x}^{(k)} + \mathbf{P}\mathbf{b}$$

where $\mathbf{P}$ is called the iteration matrix.
For the Jacobi scheme scheme we have

$$\mathbf{P} = \mathbf{P}_J = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$$

For the Gauss-Seidel scheme

$$\mathbf{P} = \mathbf{P}_G = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}.$$

Convergence properties

The exact solution to the linear system satisfies

$$\mathbf{x} = \mathbf{P}\mathbf{x} + \mathbf{P}\mathbf{b}.$$

Convergence properties

The exact solution to the linear system satisfies

$$\mathbf{x} = \mathbf{P}\mathbf{x} + \mathbf{P}\mathbf{b}.$$

Define the error at the kth iteration by

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$$

then the error satisfies the equation

Convergence properties

The exact solution to the linear system satisfies

$$\mathbf{x} = \mathbf{P}\mathbf{x} + \mathbf{P}\mathbf{b}.$$

Define the error at the kth iteration by

$$\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$$

then the error satisfies the equation

$$\mathbf{e}^{(k+1)} = \mathbf{P}\mathbf{e}^{(k)} = \mathbf{P}^2\mathbf{e}^{(k-1)} = \mathbf{P}^{k+1}\mathbf{e}^{(0)}.$$

Convergence properties

In order that the error diminishes as $k \to \infty$ we must have

$$||\mathbf{P}^k|| \to 0 \quad \text{as} \quad k \to \infty,$$

Convergence properties

In order that the error diminishes as $k \to \infty$ we must have

$$||\mathbf{P}^k|| \to 0 \quad \text{as} \quad k \to \infty,$$

Since

$$||\mathbf{P}^k|| = ||\mathbf{P}||^k$$

we see that we require

$$||\mathbf{P}|| < 1.$$

Convergence properties

From linear algebra

$$||\mathbf{P}|| < 1$$

is equivalent to the requirement that

$$\rho(\mathbf{P}) = \max_i |\lambda_i| < 1$$

where λ_i are the eigenvalues of the matrix $\mathbf{P}$. $\rho(\mathbf{P})$ is called the spectral radius of $\mathbf{P}$.

Convergence properties

Note also that for large k

$$||\mathbf{e}^{(k+1)}|| = \rho||\mathbf{e}^{(k)}||.$$

Convergence properties

Note also that for large k

$$||\mathbf{e}^{(k+1)}|| = \rho||\mathbf{e}^{(k)}||.$$

How many iterations does it takes to reduce the initial error by a factor ϵ?

Convergence properties

Note also that for large k

$$||\mathbf{e}^{(k+1)}|| = \rho||\mathbf{e}^{(k)}||.$$

How many iterations does it takes to reduce the initial error by a factor ϵ?
We need q iterations where q is the smallest value for which

$$\rho^q < \epsilon$$

giving

$$q \geq q_d = \frac{\ln \epsilon}{\ln \rho}.$$

Convergence properties

Thus iteration matrices where the spectral radius is close to 1 will converge slowly.
For the model problem it can be shown that for Jacobi iteration

$$\rho = \rho(\mathbf{P}_J) = \frac{1}{2}\left(\cos\frac{\pi}{N} + \cos\frac{\pi}{M}\right),$$

and for Gauss-Seidel

$$\rho = \rho(\mathbf{P}_G) = [\rho(\mathbf{P}_J)]^2.$$

Convergence properties

If we take $N = M$ and $N >> 1$ then for Jacobi iteration we have

$$q_d = \frac{\ln \epsilon}{\ln(1 - \frac{\pi^2}{2N^2})} = -\frac{2N^2}{\pi^2} \ln \epsilon.$$

Convergence properties

If we take $N = M$ and $N >> 1$ then for Jacobi iteration we have

$$q_d = \frac{\ln \epsilon}{\ln(1 - \frac{\pi^2}{2N^2})} = -\frac{2N^2}{\pi^2} \ln \epsilon.$$

Gauss-Seidel converges twice as fast as Jacobi.

Convergence properties

For point SOR the spectral radius depends on the relaxation factor ω, but for the model problem with optimum values and $N = M$ it can be shown that

$$\rho = \frac{1 - \sin\frac{\pi}{N}}{1 + \sin\frac{\pi}{N}}$$

giving

$$q_d = -\frac{N}{2\pi}\ln\epsilon.$$

Convergence properties

For line SOR, and it can be shown that using optimum values

$$\rho = \left(\frac{1 - \sin \frac{\pi}{2N}}{1 + \sin \frac{\pi}{2N}} \right)^2, \quad q_d = -\frac{N}{2\sqrt{2}\pi} \ln \epsilon.$$

Parabolic Equations

In this section we will look at the solution of parabolic partial differential equations. The techniques introduced earlier apply equally to parabolic pdes.

Parabolic Equations

One of the simplest parabolic pde is the diffusion equation which in one space dimensions is

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

Parabolic Equations

One of the simplest parabolic pde is the diffusion equation which in one space dimensions is

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

For two or more space dimensions we have

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u.$$

In the above κ is some given constant.

Parabolic Equations

Another familiar set of parabolic pdes is the boundary layer equations

$$\begin{aligned} u_x + v_y &= 0, \\ u_t + uu_x + vu_y &= -p_x + u_{yy}, \\ 0 &= -p_y. \end{aligned}$$

Parabolic Equations

With a parabolic pde we expect, in addition to boundary conditions, an initial condition at say, $t = 0$.

Parabolic Equations

Let us consider

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

in the region $a \le x \le b$.

Parabolic Equations

Let us consider

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

in the region $a \leq x \leq b$. Take a uniform mesh in x with

$$x_j = a + j\Delta_x, \quad j = 0, 1, .., N$$

and

$$\Delta_x = (b - a)/N$$

.

Parabolic Equations

Let us consider

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

in the region $a \leq x \leq b$. Take a uniform mesh in x with

$$x_j = a + j\Delta_x, \quad j = 0, 1, .., N$$

and

$$\Delta_x = (b - a)/N$$

.

For the differencing in time we assume a constant step size Δ_t so that

$$t = t_k = k\Delta_t$$

1st order central difference approximation

We may approximate our equation by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

1st order central difference approximation

We may approximate our equation by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Here w_j^k denotes an approximation to the exact solution $u(x, t)$ of the pde at $x = x_j, t = t_k$.

1st order central difference approximation

We may approximate our equation by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Here w_j^k denotes an approximation to the exact solution $u(x,t)$ of the pde at $x = x_j, t = t_k$. The above scheme is first order in time $O(\Delta_t)$ and second order in space $O(\Delta_x)^2$.

1st order central difference approximation

We may approximate our equation by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Here w_j^k denotes an approximation to the exact solution $u(x,t)$ of the pde at $x = x_j, t = t_k$. The above scheme is first order in time $O(\Delta_t)$ and second order in space $O(\Delta_x)^2$. The scheme is *explicit* because the unknowns at level $k+1$ can be computed directly.

Parabolic pde

Let us assume that we are given a suitable initial condition, and boundary conditions of the form

$$u(a,t) = f(t), \quad u(b,t) = g(t).$$

Notice that there is a time lag before the effect of the boundary data is felt on the solution.

Parabolic pde

As we will see later this scheme is conditionally stable for

$$\beta \leq 1/2$$

where

$$\beta = \frac{\kappa \Delta_t}{\Delta_x^2}.$$

Note that β is sometimes called the Peclet or diffusion number.

Fully implicit, first order

A better approximation is one which makes use of an *implicit* scheme. Then we have

Fully implicit, first order

A better approximation is one which makes use of an *implicit* scheme. Then we have

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} \right].$$

Fully implicit, first order

A better approximation is one which makes use of an *implicit* scheme. Then we have

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} \right]$$

The unknowns at level $k+1$ are coupled together and we have a set of implicit equations to solve.

Fully implicit, first order

Rearrange to get

$$\beta w_{j-1}^{k+1} - (1+2\beta) w_j^{k+1} + \beta w_{j-1}^{k+1} = -w_j^k, \quad 1 \le j \le N-$$

Approximation of the boundary conditions gives

$$w_0^{k+1} = f(t_{k+1}), \quad w_N^{k+1} = g(t_{k+1}).$$

Fully implicit, first order

The discrete equations are of tridiagonal form and thus easily solved.

Fully implicit, first order

The discrete equations are of tridiagonal form and thus easily solved.
The scheme is unconditionally stable .

Fully implicit, first order

The discrete equations are of tridiagonal form and thus easily solved.
The scheme is unconditionally stable .

The accuracy of the above fully implicit scheme is only first order in time. We can try and improve on this with a second order scheme.

Richardson method

Consider

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Richardson method

Consider

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

This uses three time levels and has accuracy $O(\Delta_t^2, \Delta_x^2)$.

Richardson method

Consider

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

This uses three time levels and has accuracy $O(\Delta_t^2, \Delta_x^2)$.

The scheme was devised by a meteorologist and is unconditionally unstable!

Du-Fort Frankel

This uses the approximation

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - w_j^{k+1} - w_j^{k-1} + w_{j-1}^k}{\Delta_x^2} \right].$$

Du-Fort Frankel

This uses the approximation

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - w_j^{k+1} - w_j^{k-1} + w_{j-1}^k}{\Delta_x^2} \right].$$

This has truncation error $O(\Delta_t^2, \Delta_x^2, (\frac{\Delta_t^4}{\Delta_x^2}))$, and is an explicit scheme.

Du-Fort Frankel

This uses the approximation

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - w_j^{k+1} - w_j^{k-1} + w_{j-1}^k}{\Delta_x^2} \right].$$

This has truncation error $O(\Delta_t^2, \Delta_x^2, (\frac{\Delta_t^4}{\Delta_x^2}))$, and is an explicit scheme.

The scheme is unconditionally stable, but is inconsistent if $\Delta_t \to 0, \Delta_x \to 0$ but with Δ_t/Δ_x remaining fixed.

Crank-Nicolson

A popular scheme is the Crank-Nicolson scheme given by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \frac{\kappa}{2}\left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} + \frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2}\right].$$

Crank-Nicolson

A popular scheme is the Crank-Nicolson scheme given by

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \frac{\kappa}{2}\left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} + \frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2}\right].$$

This is second order accurate $O(\Delta_t^2, \Delta_x^2)$ and is unconditionally stable. (Taking very large time steps can however cause problems). As can be seen it is also an implicit scheme.

Multi-space dimensions

The schemes outlined above are easily extended to multi-dimensions. Thus in two space dimensions a first order explict approximation to

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u,$$

is

Multi-space dimensions

The schemes outlined above are easily extended to multi-dimensions. Thus in two space dimensions a first order explicit approximation to

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u,$$

is

$$\frac{w_{i,j}^{k+1} - w_{i,j}^{k}}{\Delta_t} = \kappa \left[\frac{w_{i+1,j}^{k} - 2w_{i,j}^{k} + w_{i-1,j}^{k}}{\Delta_x^2} + \frac{w_{i,j+1}^{k} - 2w_{i,j}^{k} + w_{i,j-1}^{k}}{\Delta_y^2} \right].$$

Multi-space dimensions

The schemes outlined above are easily extended to multi-dimensions. Thus in two space dimensions a first order explicit approximation to

$$\frac{\partial u}{\partial t} = \kappa \nabla^2 u,$$

is

$$\frac{w_{i,j}^{k+1} - w_{i,j}^{k}}{\Delta_t} = \kappa \left[\frac{w_{i+1,j}^{k} - 2w_{i,j}^{k} + w_{i-1,j}^{k}}{\Delta_x^2} + \frac{w_{i,j+1}^{k} - 2w_{i,j}^{k} + w_{i,j-1}^{k}}{\Delta_y^2} \right].$$

This is first order in Δ_t and second order in space. It is conditionally stable for

$$\frac{\kappa \Delta_t}{(\Delta_x)^2} + \frac{\kappa \Delta_t}{(\Delta_y)^2} \leq \frac{1}{2}.$$

Multi-dimensional schemes

If we use a fully implicit scheme we would obtain

$$\frac{w_{i,j}^{k+1} - w_{i,j}^{k}}{\Delta_t} = \kappa \left[\frac{w_{i+1,j}^{k+1} - 2w_{i,j}^{k+1} + w_{i-1,j}^{k+1}}{\Delta_x^2} + \frac{w_{i,j+1}^{k+1} - 2w_{i,j}^{k+1} + w_{i,j-1}^{k+1}}{\Delta_y^2} \right].$$

Multi-dimensional schemes

This leads to an implicit system of equations of the form

$$\alpha w_{i+1,j}^{k+1} + \alpha w_{i-1,j}^{k+1} - (2\alpha + 2\beta + 1) w_{i,j}^{k+1} + \beta w_{i,j-1}^{k+1} + \beta w_{i,j+1}^{k+1} = -w_{i,j}^{k},$$

where

$$\alpha = \kappa \Delta_t / \Delta_x^2$$

$$\beta = \kappa \Delta_t / \Delta_y^2$$

The form of the discrete equations is very much like the system of equations arising in elliptic pdes.

Multi-dimensional schemes

From the computational point of view a better scheme is

$$\frac{w_{i,j}^{k+\frac{1}{2}} - w_{i,j}^{k}}{\Delta_t/2} = \kappa \left[\frac{w_{i+1,j}^{k+\frac{1}{2}} - 2w_{i,j}^{k+\frac{1}{2}} + w_{i-1,j}^{k+\frac{1}{2}}}{\Delta_x^2} + \frac{w_{i,j+1}^{k} - 2w_{i,j}^{k} + w_{i,j-1}^{k}}{\Delta_y^2} \right],$$

$$\frac{w_{i,j}^{k+1} - w_{i,j}^{k+\frac{1}{2}}}{\Delta_t/2} = \kappa \left[\frac{w_{i+1,j}^{k+\frac{1}{2}} - 2w_{i,j}^{k+\frac{1}{2}} + w_{i-1,j}^{k+\frac{1}{2}}}{\Delta_x^2} + \frac{w_{i,j+1}^{k+1} - 2w_{i,j}^{k+1} + w_{i,j-1}^{k+1}}{\Delta_y^2} \right] \quad \text{(-34)}$$

which leads to a tridiagonal system of equations similar to the ADI scheme. The above scheme is second order in time and space and also unconditionally stable.

Consistency revisited

Let us consider the truncation error for the first order central (explicit) scheme, and also the Du-Fort Frankel scheme.

If $u(x,t)$ is the exact solution then we may write $u_j^k = u(x_j, t_k)$ and thus from a Taylor series expansion

Consistency revisited

$$
\begin{aligned}
u_j^{k+1} \quad &= \quad u(x_j, t_k + \Delta_t) = \\
u_j^k \quad &+\Delta_t \left(\frac{\partial u}{\partial t}\right)_{j,k} + \frac{\Delta_t^2}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j,k} + O(\Delta_t)^3,
\end{aligned}
\qquad (-35)
$$

and

$$
\begin{aligned}
u_{j+1}^k \quad &= \quad u(x_j + \Delta_x, t_k) = \\
u_j^k \quad &+ \quad \Delta_x \left(\frac{\partial u}{\partial x}\right)_{j,k} + \frac{\Delta_x^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,k} + \frac{\Delta_x^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k} + \frac{\Delta_x^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j,k} + O(\Delta
\end{aligned}
$$

(-

and

$$
\begin{aligned}
u_{j-1}^k \quad &= \quad u(x_j - \Delta_x, t_k) = \\
u_j^k \quad &- \quad \Delta_x \left(\frac{\partial u}{\partial x}\right)_{j,k} + \frac{\Delta_x^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,k} - \frac{\Delta_x^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k} + \frac{\Delta_x^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j,k} + O(\Delta
\end{aligned}
$$

Consistency revisited

Substituting into the pde gives

$$\frac{1}{\Delta_t}\left[u_j^k + \Delta_t\left(\frac{\partial u}{\partial t}\right)_{j,k} + \frac{\Delta_t^2}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j,k} - u_j^k + O(\Delta_t)^3\right] =$$

$$\frac{\kappa}{\Delta_x^2}\left[u_j^k + \Delta_x\left(\frac{\partial u}{\partial x}\right)_{j,k} + \frac{\Delta_x^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,k} + \frac{\Delta_x^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k} + \frac{\Delta_x^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j,k} +\right.$$

$$+ \quad u_j^k - \Delta_x\left(\frac{\partial u}{\partial x}\right)_{j,k} + \frac{\Delta_x^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,k} - \frac{\Delta_x^3}{6}\left(\frac{\partial^3 u}{\partial x^3}\right)_{j,k} + \frac{\Delta_x^4}{24}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j,k} + O(\Delta$$

Consistency revisited

from which we obtain

$$\left[\frac{\partial u}{\partial t} - \kappa\frac{\partial^2 u}{\partial x^2}\right]_{j,k} = -\frac{\Delta_t}{2}\left(\frac{\partial^2 u}{\partial t^2}\right)_{j,k} + \frac{\kappa\Delta_x^2}{12}\left(\frac{\partial^4 u}{\partial x^4}\right)_{j,k}.$$

This shows that as $\Delta_t \longrightarrow 0$ and $\Delta_x \longrightarrow 0$ the original pde is satisfied, and the right hand side implies a truncation error $O(\Delta_t, \Delta_x^2)$.

Consistency revisited

If we do the same for the Du-Fort Frankel scheme we find that

$$\frac{u_j^{k+1} - u_j^{k-1}}{2\Delta_t} - \kappa \left[\frac{u_{j+1}^k - u_j^{k+1} - u_j^{k-1} + u_{j-1}^k}{\Delta_x^2} \right]$$

$$= \left[\frac{\partial u}{\partial t} - \kappa \frac{\partial^2 u}{\partial x^2} + \kappa \frac{\Delta_t^2}{\Delta_x^2} \frac{\partial^2 u}{\partial t^2} \right]_{j,k} + O(\Delta_t^2, \Delta_x^2, \frac{\Delta_t^4}{\Delta_x^2}).$$

Consistency revisited

This shows that the Du-Fort scheme is only consistent if the step sizes approach zero and also $\frac{\Delta_t}{\Delta_x} \longrightarrow 0$ simultaneously. Otherwise if we take step sizes such that $\frac{\Delta_t}{\Delta_x}$ remains constant as both step sizes approach zero, then the above shows that we are solving the wrong equation.

Stability

Consider the first order explicit scheme which can be written as

$$w_j^{k+1} = \beta w_{j-1}^k + (1-2\beta) w_j^k + \beta w_{j+1}^k, \quad 1 \le j \le N-1,$$

with w_0^k, w_N^k given. We can write the above in matrix form as

Stability

$$\mathbf{w}^{k+1} = \begin{bmatrix} (1-2\beta) & \beta & & & \\ \beta & (1-2\beta) & \beta & & \\ 0 & \beta & (1-2\beta) & \beta & \\ & & & & \\ & & & . & \\ & & & & . \\ & & & \beta & (1-2\beta) \end{bmatrix} \mathbf{w}^k,$$

where $\mathbf{w}^k = (w_1^k, w_2^k, ..., w_{N-1}^k)^T$.

Stability

Thus we have

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k}.$$

Stability

Thus we have

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k}.$$

Recall convergence of iterative methods.

Stability

Thus we have

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^k.$$

Recall convergence of iterative methods.

The above scheme is stable if and only if $||\mathbf{A}|| \leq 1$.

Stability

Now the infinity norm $||\mathbf{A}||_\infty$ is defined by

$$||\mathbf{A}||_\infty = \max_j \sum_i^N |a_{i,j}|$$

for an $N \times N$ matrix $\mathbf{A}$.

Stability

For the above matrix we we have

$$||\mathbf{A}||_\infty = \beta + |1 - 2\beta| + \beta = 2\beta + |1 - 2\beta|.$$

Stability

For the above matrix we we have

$$||\mathbf{A}||_\infty = \beta + |1 - 2\beta| + \beta = 2\beta + |1 - 2\beta|.$$

So if $(1 - 2\beta) \geq 0$ ie $\beta \leq 1/2$ then

Stability

For the above matrix we we have

$$||\mathbf{A}||_\infty = \beta + |1 - 2\beta| + \beta = 2\beta + |1 - 2\beta|.$$

So if $(1 - 2\beta) \geq 0$ ie $\beta \leq 1/2$ then

$$||\mathbf{A}||_\infty = 2\beta + 1 - 2\beta = 1.$$

Stability

For the above matrix we we have

$$||\mathbf{A}||_\infty = \beta + |1-2\beta| + \beta = 2\beta + |1-2\beta|.$$

So if $(1-2\beta) \geq 0$ ie $\beta \leq 1/2$ then

$$||\mathbf{A}||_\infty = 2\beta + 1 - 2\beta = 1.$$

If $(1-2\beta) < 0$ then

Stability

For the above matrix we we have

$$||\mathbf{A}||_\infty = \beta + |1-2\beta| + \beta = 2\beta + |1-2\beta|.$$

So if $(1-2\beta) \geq 0$ ie $\beta \leq 1/2$ then

$$||\mathbf{A}||_\infty = 2\beta + 1 - 2\beta = 1.$$

If $(1-2\beta) < 0$ then

$$||\mathbf{A}||_\infty = 2\beta + 2\beta - 1 = 4\beta - 1 > 1.$$

Thus we have proved that the explicit scheme is unstable if $\beta > 1/2$.

Stability of Crank-Nicolson scheme

The Crank-Nicolson scheme may be written as

$$\begin{aligned} -\beta w_{j-1,k+1} + (2+2\beta) w_{j,k+1} - \beta w_{j+1,k+1} = \\ \beta w_{j-1,k} + (2-2\beta) w_{j,k} + \beta w_{j+1,k}, \\ j = 1, 2, ..., N-1 \end{aligned}$$

where $\beta = \frac{\Delta_t \kappa}{(\Delta_x)^2}$.

In matrix form with $\mathbf{w}^k = (w_1^k, w_2^k, ..., w_{N-1}^k)^T$.

$$\begin{bmatrix} (2+2\beta) & -\beta & & & \\ -\beta & (2+2\beta) & -\beta & & \\ 0 & -\beta & (2+2\beta) & -\beta & \\ & & & & \\ & & & . & \\ & & & & . \\ & & & -\beta & (2+2\beta) \end{bmatrix} \mathbf{w}^{k+1} =$$

$$\begin{bmatrix} (2-2\beta) & \beta & & & \\ \beta & (2-2\beta) & \beta & & \\ 0 & \beta & (2-2\beta) & \beta & \\ & & & & \\ & & & . & \\ & & & & . \\ & & & \beta & (2-2\beta) \end{bmatrix} \mathbf{w}^k,$$

Stability

This is of the form

$$\mathbf{B}\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k},$$

where

$$\mathbf{B} = 2\mathbf{I}_{N-1} - \beta\mathbf{S}_{N-1},$$

and

$$\mathbf{A} = 2\mathbf{I}_{N-1} + \beta\mathbf{S}_{N-1}$$

and $\mathbf{I_N}$ is the $N \times N$ identity matrix.

Stability

Also $\mathbf{S}_{N-1}$ is the $(N-1)\times(N-1)$ matrix

$$\mathbf{S}_{N-1} = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ 0 & 1 & -2 & 1 & \\ & & & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & 1 & -2 \end{bmatrix}.$$

Stability

Hence

$$\mathbf{w}^{k+1} = \mathbf{B}^{-1}\mathbf{A}\mathbf{w}^k.$$

Stability

Hence

$$\mathbf{w}^{k+1} = \mathbf{B}^{-1}\mathbf{A}\mathbf{w}^{k}.$$

Thus the Crank-Nicolson scheme will be stable if the spectral radius of the matrix $\mathbf{B}^{-1}\mathbf{A}$ is less than unity, ie

Stability

Hence

$$\mathbf{w}^{k+1} = \mathbf{B}^{-1}\mathbf{A}\mathbf{w}^{k}.$$

Thus the Crank-Nicolson scheme will be stable if the spectral radius of the matrix $\mathbf{B}^{-1}\mathbf{A}$ is less than unity, ie

$$\rho(\mathbf{B}^{-1}\mathbf{A}) < 1.$$

Stability

Hence

$$\mathbf{w}^{k+1} = \mathbf{B}^{-1}\mathbf{A}\mathbf{w}^{k}.$$

Thus the Crank-Nicolson scheme will be stable if the spectral radius of the matrix $\mathbf{B}^{-1}\mathbf{A}$ is less than unity, ie

$$\rho(\mathbf{B}^{-1}\mathbf{A}) < 1.$$

We therefore need the eigenvalues of the matrix

$$\mathbf{B}^{-1}\mathbf{A}.$$

Stability

Recall that λ is an eigenvalue of the matrix $\mathbf{S}$, and $\mathbf{x}$ a corresponding eigenvector if

$$\mathbf{S}\mathbf{x} = \lambda \mathbf{x}.$$

Stability

Recall that λ is an eigenvalue of the matrix $\mathbf{S}$, and $\mathbf{x}$ a corresponding eigenvector if

$$\mathbf{S}\mathbf{x} = \lambda\mathbf{x}.$$

Thus for any integer p

$$\mathbf{S}^p\mathbf{x} = \mathbf{S}^{p-1}\mathbf{S}\mathbf{x} = \mathbf{S}^{p-1}\lambda\mathbf{x} = \ldots = \lambda^p\mathbf{x}.$$

Stability

Recall that λ is an eigenvalue of the matrix $\mathbf{S}$, and $\mathbf{x}$ a corresponding eigenvector if

$$\mathbf{S}\mathbf{x} = \lambda\mathbf{x}.$$

Thus for any integer p

$$\mathbf{S}^p\mathbf{x} = \mathbf{S}^{p-1}\mathbf{S}\mathbf{x} = \mathbf{S}^{p-1}\lambda\mathbf{x} = ... = \lambda^p\mathbf{x}.$$

Hence the eigenvalues of $\mathbf{S}^p$ are λ^p with eigenvector $\mathbf{x}$.

Stability

Extending this result, if $P(\mathbf{S})$ is the matrix polynomial

$$P(\mathbf{S}) = a_0\mathbf{S}^n + a_1\mathbf{S}^{n-1} + ... + a_n\mathbf{I}$$

then

Stability

Extending this result, if $P(\mathbf{S})$ is the matrix polynomial

$$P(\mathbf{S}) = a_0\mathbf{S}^n + a_1\mathbf{S}^{n-1} + ... + a_n\mathbf{I}$$

then

$$P(\mathbf{S})\mathbf{x} = P(\lambda)\mathbf{x}, \quad \text{and} \quad P^{-1}(\mathbf{S})\mathbf{x} = \frac{1}{P(\lambda)}\mathbf{x}.$$

Stability

Extending this result, if $P(\mathbf{S})$ is the matrix polynomial

$$P(\mathbf{S}) = a_0\mathbf{S}^n + a_1\mathbf{S}^{n-1} + ... + a_n\mathbf{I}$$

then

$$P(\mathbf{S})\mathbf{x} = P(\lambda)\mathbf{x}, \quad \text{and} \quad P^{-1}(\mathbf{S})\mathbf{x} = \frac{1}{P(\lambda)}\mathbf{x}.$$

Finally if $Q(\mathbf{S})$ is any other polynomial in $\mathbf{S}$ then we see that

$$P^{-1}(\mathbf{S})Q(\mathbf{S})\mathbf{x} = \frac{Q(\lambda)}{P(\lambda)}\mathbf{x}.$$

Stability

Extending this result, if $P(\mathbf{S})$ is the matrix polynomial

$$P(\mathbf{S}) = a_0\mathbf{S}^n + a_1\mathbf{S}^{n-1} + ... + a_n\mathbf{I}$$

then

$$P(\mathbf{S})\mathbf{x} = P(\lambda)\mathbf{x}, \quad \text{and} \quad P^{-1}(\mathbf{S})\mathbf{x} = \frac{1}{P(\lambda)}\mathbf{x}.$$

Finally if $Q(\mathbf{S})$ is any other polynomial in $\mathbf{S}$ then we see that

$$P^{-1}(\mathbf{S})Q(\mathbf{S})\mathbf{x} = \frac{Q(\lambda)}{P(\lambda)}\mathbf{x}.$$

Stability

If we let

$$\mathbf{P} = \mathbf{B}(\mathbf{S}_{N-1}) = 2\mathbf{I}_{N-1} - \beta\mathbf{S}_{N-1},$$

and

$$\mathbf{Q} = \mathbf{A}(\mathbf{S}_{N-1}) = 2\mathbf{I}_{N-1} + \beta\mathbf{S}_{N-1}$$

Stability

If we let

$$\mathbf{P} = \mathbf{B}(\mathbf{S}_{N-1}) = 2\mathbf{I}_{N-1} - \beta\mathbf{S}_{N-1},$$

and

$$\mathbf{Q} = \mathbf{A}(\mathbf{S}_{N-1}) = 2\mathbf{I}_{N-1} + \beta\mathbf{S}_{N-1}$$

then the eigenvalues of the matrix $\mathbf{B}^{-1}\mathbf{A}$ are given by

$$\mu = \frac{2 + \beta\lambda}{2 - \lambda\beta}$$

where λ is an eigenvalue of the matrix $\mathbf{S}_{N-1}$.

Stability

Now the eigenvalues of the $N \times N$ matrix

$$\mathbf{T} = \begin{bmatrix} a & b & & & \\ c & a & b & & \\ 0 & c & a & b & \\ & . & . & . & \\ & & . & . & . \\ & & & c & a \end{bmatrix}$$

can be shown to be given by

$$\lambda = \lambda_n = a + 2\sqrt{bc}\cos\frac{n\pi}{N+1}, \quad n = 1, 2, .., N.$$

Stability

Hence the eigenvalues of $\mathbf{S}_{N-1}$ are

$$\lambda_n = -4\sin^2\frac{n\pi}{2N}, \quad n = 1, 2, ..., N-1$$

and so the eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$ are

$$\mu_n = \frac{2 - 4\beta\sin^2\frac{n\pi}{N}}{2 + 4\beta\sin^2\frac{n\pi}{N}} \quad n = 1, 2, .., N-1.$$

Stability

Clearly

$$\rho(\mathbf{B}^{-1}\mathbf{A}) = \max_n |\mu_n| < 1 \quad \forall\beta > 0.$$

This proves that the Crank-Nicolson scheme is unconditionally stable.

Stability condition allowing exponential growth

In the above discussion of stability we have said that the solution of

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k}$$

is stable if $||\mathbf{A}|| \leq 1$.

Stability condition allowing exponential growth

In the above discussion of stability we have said that the solution of

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k}$$

is stable if $||\mathbf{A}|| \leq 1$.
This condition does not make allowance for solutions of the pde which may be growing exponentially in time.

Stability condition allowing exponential growth

In the above discussion of stability we have said that the solution of

$$\mathbf{w}^{k+1} = \mathbf{A}\mathbf{w}^{k}$$

is stable if $||\mathbf{A}|| \leq 1$.
This condition does not make allowance for solutions of the pde which may be growing exponentially in time.
A necessary and sufficient condition for stability when the solution of the pde is increasing exponentially in time is that

$$||\mathbf{A}|| \leq 1 + M\Delta_t = 1 + O(\Delta_t)$$

where M is a constant independent of Δ_x and Δ_t.

Von-Neumann stability analysis

A very versatile tool for analysing stability is the Fourier method developed by von Neumann.

Here initial values at mesh points are expressed in terms of a finite Fourier series, and we consider the growth of individual Fourier components.

Von-Neumann stability analysis

A finite sine or cosine series expansion in the interval $a \leq x \leq b$ takes the form

$$\sum_n a_n \sin(\frac{n\pi x}{L}), \quad \text{or} \quad \sum_n b_n \cos(\frac{n\pi x}{L}),$$

where $L = b - a$.

Von-Neumann stability analysis

A finite sine or cosine series expansion in the interval $a \leq x \leq b$ takes the form

$$\sum_n a_n \sin(\frac{n\pi x}{L}), \quad \text{or} \quad \sum_n b_n \cos(\frac{n\pi x}{L}),$$

where $L = b - a$. Now consider an individual component written in complex exponential form at a mesh point $x = x_j = a + j\Delta_x$

$$A_n e^{\frac{inx\pi}{L}} = A_n e^{\frac{ina\pi}{L}} e^{\frac{inj\pi\Delta_x}{L}} = \tilde{A}_n e^{i\alpha_n j\Delta_x}$$

where $\alpha_n = n\pi/L$.

Von-Neumann stability analysis

Given initial data we can express the initial values as

$$w_p^0 = \sum_{n=0}^{N} \tilde{A}_n e^{i\alpha_n p\Delta_x} \qquad p = 0, 1, ..., N,$$

and we have $N + 1$ equations to determine the $N + 1$ unknowns $\tilde{A}$.

Von-Neumann stability

To find how each Fourier mode develops in time, assume a simple separable solution of the form

Von-Neumann stability

To find how each Fourier mode develops in time, assume a simple separable solution of the form

$$w_p^k = e^{i\alpha_n p\Delta_x} e^{\Omega t_k} = e^{i\alpha_n p\Delta_x} e^{\Omega k\Delta_t} = e^{i\alpha_n p\Delta_x} \xi^k,$$

where $\xi = e^{\Omega\Delta_t}$.

Von-Neumann stability

To find how each Fourier mode develops in time, assume a simple separable solution of the form

$$w_p^k = e^{i\alpha_n p\Delta_x} e^{\Omega t_k} = e^{i\alpha_n p\Delta_x} e^{\Omega k\Delta_t} = e^{i\alpha_n p\Delta_x} \xi^k,$$

where $\xi = e^{\Omega\Delta_t}$.
Here ξ is called the amplification factor.

Von-Neumann stability

To find how each Fourier mode develops in time, assume a simple separable solution of the form

$$w_p^k = e^{i\alpha_n p\Delta_x} e^{\Omega t_k} = e^{i\alpha_n p\Delta_x} e^{\Omega k\Delta_t} = e^{i\alpha_n p\Delta_x} \xi^k,$$

where $\xi = e^{\Omega\Delta_t}$.
Here ξ is called the amplification factor.
For stability we thus require $|\xi| \leq 1$. If the exact solution of the pde grows exponentially, then the difference scheme will allow such solutions if

$$|\xi| \leq 1 + M\Delta_t$$

where M does not depend on Δ_x or Δ_t.

Von-Neumann stability analysis

Consider the fully implicit scheme

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} \right].$$

Von-Neumann stability analysis

Consider the fully implicit scheme

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} \right].$$

Let

$$w_j^k = \xi^k e^{i\alpha_n j \Delta_x}.$$

Then substituting into the above gives

Von-Neumann stability analysis

Consider the fully implicit scheme

$$\frac{w_j^{k+1} - w_j^k}{\Delta_t} = \kappa \left[\frac{w_{j+1}^{k+1} - 2w_j^{k+1} + w_{j-1}^{k+1}}{\Delta_x^2} \right].$$

Let

$$w_j^k = \xi^k e^{i\alpha_n j \Delta_x}.$$

Then substituting into the above gives

$$\frac{1}{\Delta_t}\xi^k(\xi - 1)e^{i\alpha_n j\Delta_x} = \frac{\kappa\xi^{k+1}}{\Delta_x^2}(e^{-i\alpha_n\Delta_x} - 2 + e^{i\alpha_n\Delta_x})e^{i\alpha_n j\Delta_x}.$$

Stability - fully implicit scheme

Thus with $\beta = \Delta_t \kappa / \Delta_x^2$

$$\xi - 1 = \beta\xi(2\cos(\alpha_n \Delta_x) - 2) = -4\beta\xi \sin^2(\frac{\alpha_n \Delta_x}{2}).$$

Stability - fully implicit scheme

Thus with $\beta = \Delta_t \kappa / \Delta_x^2$

$$\xi - 1 = \beta\xi(2\cos(\alpha_n \Delta_x) - 2) = -4\beta\xi \sin^2(\frac{\alpha_n \Delta_x}{2}).$$

This gives

$$\xi = \frac{1}{1 + 4\beta \sin^2(\frac{\alpha_n \Delta_x}{2})},$$

and clearly $0 < \xi \leq 1$ for all $\beta > 0$ and for all α_n.

Stability - fully implicit scheme

Thus with $\beta = \Delta_t \kappa / \Delta_x^2$

$$\xi - 1 = \beta\xi(2\cos(\alpha_n \Delta_x) - 2) = -4\beta\xi \sin^2(\frac{\alpha_n \Delta_x}{2}).$$

This gives

$$\xi = \frac{1}{1 + 4\beta \sin^2(\frac{\alpha_n \Delta_x}{2})},$$

and clearly $0 < \xi \leq 1$ for all $\beta > 0$ and for all α_n. Thus the fully implicit scheme is unconditionally stable.

Stability - Richardson's scheme

The Richardson scheme is given by

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Stability - Richardson's scheme

The Richardson scheme is given by

$$\frac{w_j^{k+1} - w_j^{k-1}}{2\Delta_t} = \kappa \left[\frac{w_{j+1}^k - 2w_j^k + w_{j-1}^k}{\Delta_x^2} \right].$$

Using a von-Neumann analysis and writing

$$w_p^k = \xi^k e^{i\alpha_n p\Delta_x},$$

gives after substitution

Stability - Richardson’s scheme

$$e^{i\alpha_n p\Delta_x}\xi^{k-1}(\xi^2-1) = \beta\xi^k(e^{-i\alpha_n\Delta_x}-2+e^{i\alpha_n\Delta_x})e^{i\alpha_n p\Delta_x}.$$

This gives

Stability - Richardson's scheme

$$e^{i\alpha_n p\Delta_x}\xi^{k-1}(\xi^2-1) = \beta\xi^k(e^{-i\alpha_n\Delta_x}-2+e^{i\alpha_n\Delta_x})e^{i\alpha_n p\Delta_x}.$$

This gives

$$\xi^2 - 1 = -4\xi\beta\sin^2(\frac{\alpha_n\Delta_x}{2}),$$

where $\beta = 2\Delta_t\kappa/\Delta_x^2$.

Stability analysis, Richardson

Thus

$$\xi^2 + 4\xi\beta\sin^2(\frac{\alpha_n\Delta_x}{2}) - 1 = 0.$$

This quadratic has two roots ξ_1, ξ_2. The sum and product of the roots is given by

$$\xi_1 + \xi_2 = -4\xi\beta\sin^2(\frac{\alpha_n\Delta_x}{2}), \quad \xi_1\xi_2 = -1.$$

Stability analysis, Richardson

For stability we require $|\xi_1| \leq 1$ and $|\xi_2| \leq 1$ and the above shows that if $|\xi_1| < 1$ then $|\xi_2| > 1$, and vice-versa. Also if $\xi_1 = 1$ and $\xi_2 = -1$ then again we must have $\beta = 0$.

Thus the Richardson scheme is unconditionally unstable.