Stability Analysis - Von Neumann Method

There are several ways to determine ahead of time whether a numerical scheme will be stable. One way is to look what effect the numerical method has on a wave of the form

$$a(x,t) = A(t)e^{ikx}$$

In a discretized form we write the above as

$$a(x_j, t_n) = a_j^n = A^n e^{ikjh}$$

If we advance the solution by one step

$$a_j^{n+1} = A^{n+1} e^{ikjh} = \xi A^n e^{ikjh}$$

If the solution technique were exact that $|\xi|=1$

- if $|\xi|>1$, then the solution is unstable
- if $|\xi| < 1$ the the solution is stable, but the wave is diffused
- In general will be complex with some phase so $\xi = |\xi| e^{i\phi}$
  - A nonzero ϕ is an indicator of dispersion

Von Neumann Stability Analysis for the FTCS method for the Diffusion Equation

Recall that the equation

$$\frac{\partial T(x,t)}{\partial t} = \kappa \frac{\partial^2 T(x,t)}{\partial x^2}$$

in the FTCS approach is written as

$$T_i^{n+1} = T_i^n + \frac{\tau\kappa}{h^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

Lets look at what effect this has on a wave of the form

$$T(x_j, t_n) = T_j^n = A^n e^{ikjh}$$

Plugging into the above we get

$$A^{n+1} e^{ikjh} = A^n e^{ikjh} + \frac{\tau\kappa}{h^2} A^n e^{ikjh} (e^{ikh} - 2 + e^{-ikh})$$

Simplifying

$$\xi = 1 + \frac{\tau\kappa}{h^2}(e^{ikh} - 2 + e^{-ikh})$$

FTCS method for the Diffusion Equation

Since

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i} \quad \text{and} \quad \sin^2 x = -\frac{e^{2ix} - 2 + e^{-2ix}}{4}$$

we get $\xi = 1 - 4\frac{\tau\kappa}{h^2}\sin^2\left(\frac{kh}{2}\right)$

Since we want $|\xi| \le 1$ $\quad \left|1 - 4\frac{\tau\kappa}{h^2}\sin^2\left(\frac{kh}{2}\right)\right| \le 1$

$\sin^2 = 1$

This can be true if $\quad 4\frac{\tau\kappa}{h^2} - 1 \le 1$

or

$$\boxed{\tau \le \frac{h^2}{2\kappa}}$$

Von Neumann Stability Analysis for the implicit method for the Diffusion Equation

$$\frac{\partial T(x,t)}{\partial t} = \kappa \frac{\partial^2 T(x,t)}{\partial x^2}$$

Lets look at what effect a method that uses the future value of the function, i.e.,

$$T_i^{n+1} = T_i^n + \frac{\tau\kappa}{h^2}(T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1})$$

Plugging in $\quad T(x_j, t_n) = T_j^n = A^n e^{ikjh}$

We get $\quad A^{n+1}e^{ikjh} = A^n e^{ikjh} + \frac{\tau\kappa}{h^2}A^{n+1}e^{ikjh}(e^{ikh} - 2 + e^{-ikh})$

gives

$$\boxed{\xi = \frac{1}{1 + \frac{4\tau\kappa}{h^2}\sin^2\frac{kh}{2}}}$$

This scheme is *unconditionally stable*.

Implicit method

If we assume a Dirichlet boundary where T_1 and T_N are specified and define $\alpha = \tau\kappa/h^2$, then one gets a system of equations that can be written in matrix form as

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & & & 0 \\ -\alpha & 1+2\alpha & -\alpha & & & & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \\ & & -\alpha & 1+2\alpha & -\alpha & & \vdots \\ & & & \ddots & \ddots & \ddots & \\ 0 & & & & -\alpha & 1+2\alpha & -\alpha \\ 0 & & & & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} T_1^{n+1} \\ T_2^{n+1} \\ \\ T_i^{n+1} \\ \\ T_{N-1}^{n+1} \\ T_N^{n+1} \end{pmatrix} = \begin{pmatrix} T_1^n \\ T_2^n \\ \\ T_i^n \\ \\ T_{N-1}^n \\ T_N^n \end{pmatrix}$$

This is a tridiagonal system of equations.

Von Neumann Stability Analysis for the FTCS method for the Advection Equation

Recall that the equation

$$\frac{\partial a(x,t)}{\partial t} = -c\frac{\partial a(x,t)}{\partial x}$$

in the FTCS approach is written as

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n)$$

Lets look at what effect this has on a wave of the form

$$A(x_j, t_n) = A_j^n = A^n e^{ikjh}$$

Plugging into the above we get

$$A^{n+1}e^{ikjh} = A^n e^{ikjh} - \frac{c\tau}{2h}A^n e^{ikjh}(e^{ikh} - e^{-ikh})$$

Simplifying

$$\xi = 1 - i\frac{c\tau}{h}\sin kh$$

FTCS Wave Equation (cont.)

Since we have

$$\xi = 1 - i\frac{c\tau}{h}\sin kh$$

$$|\xi| = \sqrt{1 + \left(\frac{c\tau}{h}\right)^2 \sin^2 kh}$$

In this case |ξ| is always greater than 1, and is therefore unconditionally unstable. Note also in this case there is also a phase error introduced by the method.

Von Neumann Stability Analysis for the implicit method for the Advection Equation

Recall that the equation

$$\frac{\partial a(x,t)}{\partial t} = -c\frac{\partial a(x,t)}{\partial x}$$

in the implicit approach is written as

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}(a_{i+1}^{n+1} - a_{i-1}^{n+1})$$

Again, lets look at what effect this has on a wave of the form

$$A(x_j, t_n) = A_j^n = A^n e^{ikjh}$$

Plugging into the above we get

$$A^{n+1}e^{ikjh} = A^n e^{ikjh} - \frac{c\tau}{2h}A^n e^{ikjh}(e^{ikh} - e^{-ikh})$$

Simplifying

$$A^{n+1}e^{ikjh} + \frac{c\tau}{2h}A^{n+1}e^{ikjh}(e^{ikh} - e^{-ikh}) = A^n e^{ikjh}$$

$$\xi = \frac{1}{1 + i\frac{c\tau}{h}\sin kh}$$

Implicit Wave Equation (cont.)

Since we have

$$\xi = \frac{1}{1 + i\frac{c\tau}{h}\sin kh} = \frac{1 - i\frac{c\tau}{h}\sin kh}{1 + \left(\frac{c\tau}{h}\right)^2 \sin^2 kh}$$

$$|\xi| = \frac{1}{\sqrt{1 + \left(\frac{c\tau}{h}\right)^2 \sin^2 kh}}$$

In this case $|\xi|$ is always less than 1, and is therefore unconditionally stable. Note also in this case there is also significant amount of damping - the larger the timestep the larger the amount of damping.

Matrix form of the implicit wave equation

Since we have the equation

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}(a_{i+1}^{n+1} - a_{i-1}^{n+1})$$

We can rewrite this as

$$a_i^{n+1} + \frac{c\tau}{2h}(a_{i+1}^{n+1} - a_{i-1}^{n+1}) = a_i^n$$

In matrix/vector form it is

$$(\mathbf{I} + \frac{c\tau}{2h}\mathbf{A})\vec{a}^{n+1} = \vec{a}^n$$

or

$$\vec{a}^{n+1} = (\mathbf{I} + \frac{c\tau}{2h}\mathbf{A})^{-1}\vec{a}^n$$

Matrix form of wave vector equation

Where the matrix A takes the form

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & -1 \\ -1 & 0 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 0 & 1 & \cdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & 0 & -1 & 0 & 1 \\ 1 & 0 & \cdots & 0 & -1 & 0 \end{bmatrix}$$

Crank-Nicholson Method

Better accuracy can be achieved, yielding second order in time accuracy, by time averaging the spatial derivatives

$$a_i^{n+1} = a_i^n - \frac{1}{2}\left[\frac{c\tau}{2h}(a_{i+1}^{n+1} - a_{i-1}^{n+1}) - \frac{c\tau}{2h}(a_{i+1}^{n} - a_{i-1}^{n})\right]$$

Stability analysis shows that

$$\xi = \frac{1 - i\frac{c\tau}{2h}\sin kh}{1 + i\frac{c\tau}{2h}\sin kh}$$

And that $|\xi| = 1$

Matrix form of the Crank Nicolson method for the wave equation

Since we have the equation

$$a_i^{n+1} = a_i^n - \frac{c\tau}{2h}\frac{(a_{i+1}^n - a_{i-1}^n)}{2} - \frac{c\tau}{2h}\frac{(a_{i+1}^{n+1} - a_{i-1}^{n+1})}{2}$$

We can rewrite this as

$$a_i^{n+1} + \frac{c\tau}{4h}(a_{i+1}^{n+1} - a_{i-1}^{n+1}) = a_i^n - \frac{c\tau}{4h}(a_{i+1}^n - a_{i-1}^n)$$

In matrix/vector form it is

$$(\mathbf{I} + \frac{c\tau}{4h}\mathbf{A})\vec{a}^{\,n+1} = (\mathbf{I} - \frac{c\tau}{4h}\mathbf{A})\vec{a}^{\,n}$$

Where A is defined as on page 11, the time step is then

$$\vec{a}^{\,n+1} = (\mathbf{I} + \frac{c\tau}{4h}\mathbf{A})^{-1}(\mathbf{I} - \frac{c\tau}{4h}\mathbf{A})\vec{a}^{\,n}$$

Von Neumann Stability Analysis for the Lax method for the Advection Equation

In this case the equation $\frac{\partial a(x,t)}{\partial t} = -c\frac{\partial a(x,t)}{\partial x}$

in the Lax approach is written as

$$a_i^{n+1} = \frac{1}{2}\left(a_{i+1}{}^n + a_{i-1}{}^n\right) - \frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n)$$

Plugging into the above and simplifying we get

so that $$A^{n+1}e^{ikjh} = \frac{1}{2}A^n e^{ikjh}\left[(e^{ikh} + e^{-ikh}) - \frac{c\kappa}{h}(e^{ikh} - e^{-ikh})\right]$$

$$\xi = \cos(kh) - i\frac{c\tau}{h}\sin kh$$

$$|\xi|^2 = \sqrt{\cos^2(kh) + \left(\frac{c\tau}{h}\right)^2\sin^2 kh}$$

thus $|\xi| \leq 1$ if $|c\tau/h| \leq 1$, which is the CFL condition.

Lax method for the Advection Equation

Some things to note about the Lax Method

$$\xi = \cos(kh) - i\frac{c\tau}{h}\sin kh$$

$$|\xi| = \sqrt{\cos^2(kh) + \left(\frac{c\tau}{h}\right)^2 \sin^2(kh)}$$

- When $|c\tau/h| = 1$ there is no damping, i.e., $|\xi| = 1$
- When $|c\tau/h| < 1$ the shortest wavelength modes are damped
- If we subtract a_i^n from both sides we get

$$\frac{a_i^{n+1} - a_i^n}{\tau} = -\frac{c\tau}{2h}(a_{i+1}^n - a_{i-1}^n) + \frac{h^2}{2\tau}\frac{\left(a_{i+1}{}^n - 2a_i^n + a_{i-1}{}^n\right)}{h^2}$$

  - The last term adds on an effective "diffusion" that stabilizes the method

Von Neumann Stability Analysis for the Leapfrog method for the Advection Equation

In this case the equation $\frac{\partial a(x,t)}{\partial t} = -c\frac{\partial a(x,t)}{\partial x}$
in the leapfrog approach is written as

$$a_i^{n+1} = a_i^{n-1} - \frac{c\tau}{h}(a_{i+1}^n - a_{i-1}^n)$$

Plugging into the above and simplifying we get

$$A^{n+1}e^{ikjh} = A^{n-1}e^{ikjh} - \frac{c\tau}{2h}A^n e^{ikjh}(e^{ikh} - e^{-ikh})$$

If we assume that $\xi = \frac{A^{n+1}}{A^n} = \frac{A^n}{A^{n-1}}$

$$\xi = \xi^{-1} - \frac{c\tau}{2h}(e^{ikh} - e^{-ikh}) \quad \Rightarrow \xi^2 + i\xi\frac{c\tau}{h}\sin kh - 1 = 0$$

Leap-Frog - 2

In this case we have
$$\xi = -i\frac{c\tau}{h}\sin(kh) \pm \sqrt{1-\left(\frac{c\tau}{h}\sin kh\right)^2}$$

so that $|\xi|=1$ independent of k provided that the term $|c\tau/h| \le 1$. Note however that there is significant dispersion, since

$$\tan\phi = \frac{\mp\frac{c\tau}{h}\sin(kh)}{\sqrt{1-\left(\frac{c\tau}{h}\sin kh\right)^2}}$$

if $|c\tau/h| = 1$, we get

$$\phi = \pm kh$$

So that the shorter the wavelength, the greater the dispersion.

Leapfrog method splits the grid into 2 interpenetrating lattices that do not influence each other. Roundoff errors can lead to divergence of the two lattices.

Matrix Stability

Another way to look at stability, that also includes the effects of boundary conditions is via *Matrix Stability Analysis*. Lets go back to the diffusion problem $T_i^{n+1} = T_i^n + \frac{\tau\kappa}{h^2}(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$

We can write this as

$$\mathbf{T}^{n+1} = \mathbf{T}^n + \frac{\tau\kappa}{h^2}\mathbf{D}\mathbf{T}^n$$

$$= (\mathbf{I} + \frac{\tau\kappa}{h^2}\mathbf{D})\mathbf{T}^n = \mathbf{A}\mathbf{T}^n$$

where

$$\mathbf{T}^n = \begin{bmatrix} T_1^n \\ T_2^n \\ T_3^n \\ \vdots \\ T_{N-1}^n \\ T_N^n \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Matrix Stability - 2

Stability is determined by looking at the eigenvalues of the Matrix **A**, i.e., lets look at

$$\mathbf{A}\mathbf{v}_k = \lambda \mathbf{v}_k$$

where $\mathbf{v}_k$ is the eigenvector for eigenvalue λ_k. The eigenvalues are labeled in decreasing order so that $|\lambda_1| > |\lambda_2| > |\lambda_3| \cdots > |\lambda_N|$

We can use these eigenvectors to form a basis so that the initial condition can be written as

$$\mathbf{T}^1 = \sum_{k=1}^{N} b_k \mathbf{v}_k$$

So that $\mathbf{T}^{n+1} = \mathbf{A}\mathbf{T}^n = \mathbf{A}^n \mathbf{T}^1$

$$\mathbf{T}^{n+1} = \sum_{k=1}^{N} b_k \mathbf{A}^n \mathbf{v}_k = \sum_{k=1}^{N} b_k (\lambda_k)^n \mathbf{v}_k$$

If $|\lambda_k| > 1$ for any eigenvalue, then $|T^n| \to \infty$ as $n \to \infty$. The spectral radius of A is $|\lambda_1|$. A scheme is stable if $|\lambda_1| \le 1$.

The Power Method

A reasonably quick way of determining the largest eigenvalue and eigenvector of a matrix is through the power method. Take any vector $\mathbf{x}_0$ and write it as

$$\mathbf{x}_0 = \sum_{k=1}^{N} b_k \mathbf{v}_k$$

where $\mathbf{v}_k$ are the normalized eigenvectors of the matrix **M** and $\forall b_k \neq 0$. If we then compute

$$\mathbf{M}^n \mathbf{x}_0 = \mathbf{M}^n \sum_{k=1}^{N} b_k \mathbf{v}_k$$

$$= \sum_{k=1}^{N} b_k \mathbf{M}^n \mathbf{v}_k = \sum_{k=1}^{N} b_k \lambda_k^{\,n} \mathbf{v}_k$$

Power Method -2

As n→∞ we get $\mathbf{M}^n\mathbf{x} = b_1\lambda_1^n\mathbf{v}_1$

where λ_1 is the dominant eigenvalue, ie, $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$

A more efficient approach is the following, is to iterate, so that at the k+1-th step we have

$$\mathbf{x}_{k+1} = \frac{\mathbf{M}\mathbf{x}_k}{c_{k+1}}$$

where $c_{k+1} = \max(|\mathbf{x}_k|)$

It can be shown that if **x** is chosen appropriately and that the matrix M has n distinct eigenvalues and that they are ordered in decreasing magnitude as and we iterate recursively, then

$$\lim_{k\to\infty}\mathbf{x}_k = \mathbf{v}_1$$

And that

$$\lim_{k\to\infty}c_k = \lambda_1$$

In other words the power method converges to the maximum eigenvalue and eigenvector

Power Method - 3

To show this, note that

$$\lim_{k\to\infty}\mathbf{x}_k = \lim_{k\to\infty}\frac{b_1\lambda_1^k}{c_1 \ldots c_k}\mathbf{v}_1$$

Since it was required that $\mathbf{x}_k$ and $\mathbf{v}_1$ be normalized and that their largest component be 1, the above vector must be normalized. So that

$$\lim_{k\to\infty}\frac{b_1\lambda_1^k}{c_1 \ldots c_k} = 1$$

There are a couple of caveats when using the power method:

1. It only converges if the initial guess of x is chosen so that $b_1 \neq 0$
2. The method can have problems when the eigenvalues are complex
3. It converges slowly if the maximum eigenvalues are close together

```
function lamda=power2(A,X,eps,max1)

lamda=0;
cnt=1;
xnew=X;
xold=0*X;
err=norm(X);
xnew = xnew/max(abs(xnew));

while err>eps & cnt < max1
    xold=xnew;
    ck = max(abs(xnew));
    xnew=A*xnew/ck;;
    cnt = cnt+1;
    err = norm(xold-xnew);
end

lamda = ck;
```


Other Norms

Other norms that can be used include the 1-norm

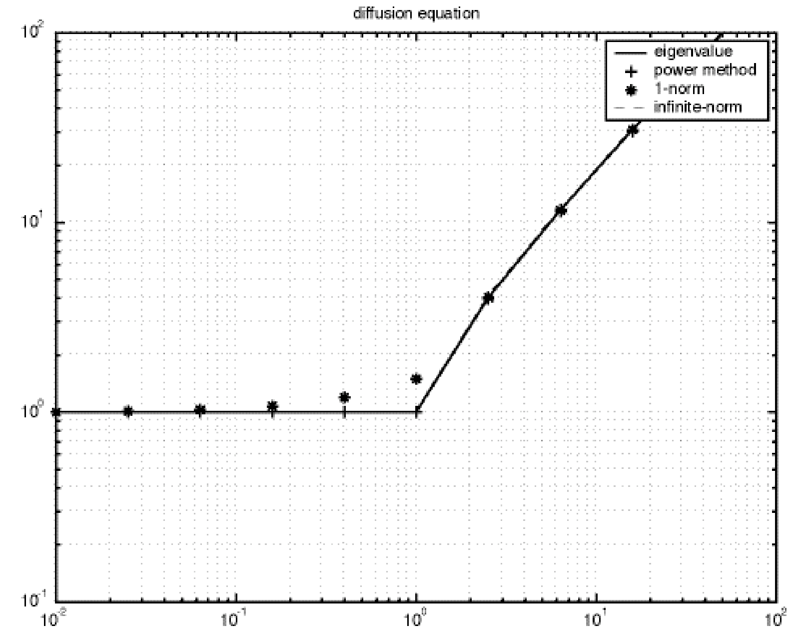
$$\|A\|_1 = \max\left\{\sum_{j=1}^{N} \left|A_{i,j}\right|\right\}$$

and the infinity norm

$$\|A\|_1 = \max\left\{\sum_{i=1}^{N} \left|A_{i,j}\right|\right\}$$

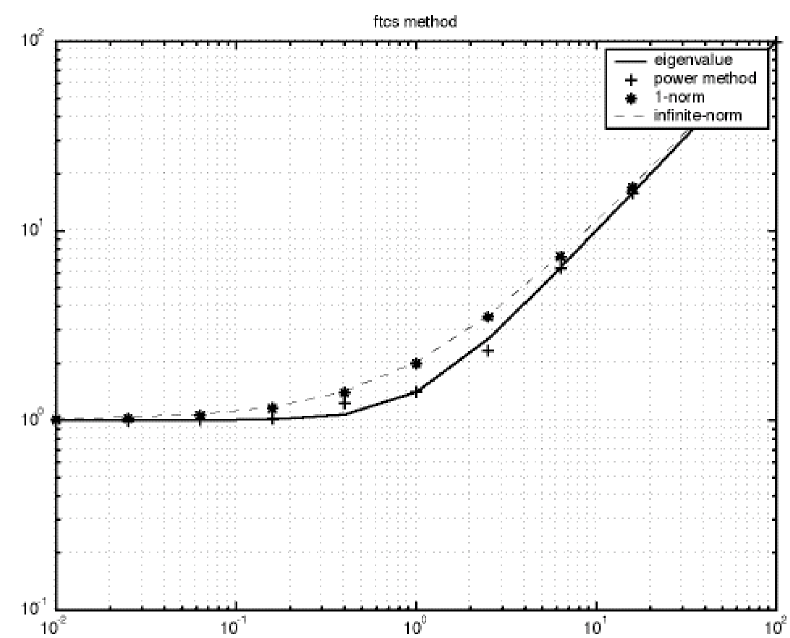
Example

For the example shown earlier for the diffusion equation



FTCS Example

(Question 8, page 286)



Solution of Schrödingers Equation

The standard 1D Schrödingers equation is usually written in the form

$$i\hbar\frac{\partial}{\partial t}\Psi(x,t) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right)\Psi(x,t)$$

$$= \mathrm{H}\ \Psi(x,t)$$

Where Ψ(x,t) is the wave function and V(x) is the potential and H is the Hamiltonian. The FTCS scheme for this equation, using the usual syntax

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = -\frac{\hbar^2}{2m}\frac{\psi_{j-1}^n - 2\psi_j^n + \psi_{j+1}^n}{h^2} + V_j^n\psi_j^n$$

Which can be written in matrix form as

$$i\hbar\frac{\psi_j^{n+1} - \psi_j^n}{\tau} = \sum_{k=1}^{N} H_{jk}\psi_k^n$$

Schrödingers Equation - FTCS

Where H is a matrix operator

$$H_{jk} = -\frac{\hbar^2}{2m}\frac{\delta_{j-1,k} - 2\delta_{j,k} + \delta_{j+1,k}}{h^2} + V_j^n\delta_{j,k}$$

This gives the numerical scheme to be

$$\Psi^{n+1} = \left(I - \frac{i\tau}{\hbar}\mathbf{H}\right)\Psi^n$$

Where Ψ is a column vector and I is the identity matrix.

Schrödingers Equation - Implicit/Crank Nicolson

The implicit scheme is then

$$i\hbar \frac{\psi_j^{n+1} - \psi_j^n}{\tau} = \sum_{k=1}^{N} H_{jk} \psi_k^{n+1}$$

Which has the form,

$$\Psi^{n+1} = \left(I + \frac{i\tau}{\hbar} \mathbf{H} \right)^{-1} \Psi^n$$

And the Crank-Nicolson

$$\Psi^{n+1} = \left(I + \frac{i\tau}{2\hbar} \mathbf{H} \right)^{-1} \left(I - \frac{i\tau}{2\hbar} \mathbf{H} \right) \Psi^n$$

Solving a Tridiagonal Matrix

The system of equations correspond to the solution of the system of equations $b_1 T_1 + c_1 T_2 = d_1$

$$a_i T_{i-1} + b_i T_i + c_i T_{i+1} = d_i \quad (2 \le i \le N-1)$$

$$a_N T_{N-1} + b_N T_N = d_N$$

where $a=c=-\alpha$ and $b=1+2\ \alpha$ and $d_i = T_i^n$.

- If we multiply the first row by $-a_2/b_1$ and add it to the second row the a_2 term will be eliminated

$$b_1 T_1 + c_1 T_2 = d_1 \qquad \times a_2 / b_1$$

$$a_2 T_1 + b_2 T_2 + c_2 T_3 = d_2$$

- Now divide the second row by the new b_2 term and repeat
- One eventually gets a new matrix with zeros below the diagonal and 1's on the diagonal - *upper diagonal* matrix
- A reverse pass will yield a diagonal matrix

Tridiagonal Matrix

Given the following tridiagonal matrix,

$$\begin{bmatrix} \beta_1 & \gamma_1 & 0 & \cdots & 0 \\ \alpha_1 & \beta_2 & \gamma_2 & \cdots & 0 \\ 0 & \alpha_2 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_N \end{bmatrix}$$

We can perform the following operations

$$\beta_i' = \beta_i - \frac{\alpha_{i-1}}{\beta_{i-1}'}\gamma_{i-1} \quad i = 2,\ldots,N \qquad \beta_1' = \beta_1$$

$$b_i' = b_i - \frac{\alpha_{i-1}}{\beta_{i-1}'}b_{i-1}' \quad i = 2,\ldots,N \qquad b_1' = b_1$$

Tridiagonal Matrix - 2

We end up with the following Upper Diagonal Matrix

$$\begin{bmatrix} \beta_1' & \gamma_1 & 0 & \cdots & 0 \\ 0 & \beta_2' & \gamma_2 & \cdots & 0 \\ 0 & 0 & \beta_2' & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_N' \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} b_1' \\ b_2' \\ b_3' \\ \vdots \\ b_N' \end{bmatrix}$$

The solution is then found by back substitution

$$x_N = \frac{b_N'}{\beta_N'} \qquad x_i = \frac{b_i' - \gamma_i x_{i+1}}{\beta_i'} \quad i = N-1,\ldots,1$$

This is referred to as the 'Thomas Algorithm'