- The weakness of the Backward-Difference method results from the fact that the local truncation error has one of $\operatorname{order} O((\Delta x)^2)$, and another of order $O(\Delta t)$. This requires that time intervals be made much smaller than the x-axis intervals.
- It would clearly desirable to have a procedure with local truncation error of order $O((\Delta x)^2 + (\Delta t)^2)$.
- The first step in this direction is to use central difference equation for time derivative which has a truncation error of $\operatorname{order} O((\Delta t)^2)$. But, unfortunately this method has serious stability problems.

Crank-Nicolson Method

This method is derived by averaging the Forward-Difference method at the *n*th step in *t*,

$$\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} - \alpha^{2} \frac{u_{i+1}^{n} - 2u_{i}^{n} + u_{i-1}^{n}}{\left(\Delta x\right)^{2}} = 0$$

which has the local truncation error,

$$\tau_{\rm F} = \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + O\left(\left(\Delta t\right)^2\right) + O\left(\left(\Delta x\right)^2\right)$$

and the Backward-Difference method at the (n+1)th step in *t*,

$$\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} - \alpha^{2} \frac{u_{i+1}^{n+1} - 2u_{i}^{n+1} + u_{i-1}^{n+1}}{\left(\Delta x\right)^{2}} = 0$$

which has the local truncation error,

$$\tau_{\rm B} = -\frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2} + O\left(\left(\Delta t\right)^2\right) + O\left(\left(\Delta x\right)^2\right)$$

Note that the averaged-difference method,

$$\frac{u_{i}^{n+1} - u_{i}^{n}}{\Delta t} - \frac{\alpha^{2}}{2} \left[\frac{u_{i+1}^{n} - 2u_{i}^{n} + u_{i-1}^{n}}{\left(\Delta x\right)^{2}} + \frac{u_{i+1}^{n+1} - 2u_{i}^{n+1} + u_{i-1}^{n+1}}{\left(\Delta x\right)^{2}} \right] = 0$$

has local truncation error of order $O((\Delta x)^2 + (\Delta t)^2)$. This is known as

the Crank-Nicolson method. Recalling that,

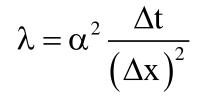
u(0,t) = u(l,t) = 0, t > 0, and u(x,0) = f(x), $0 \le x \le l$

this method can be written in the matrix form as,

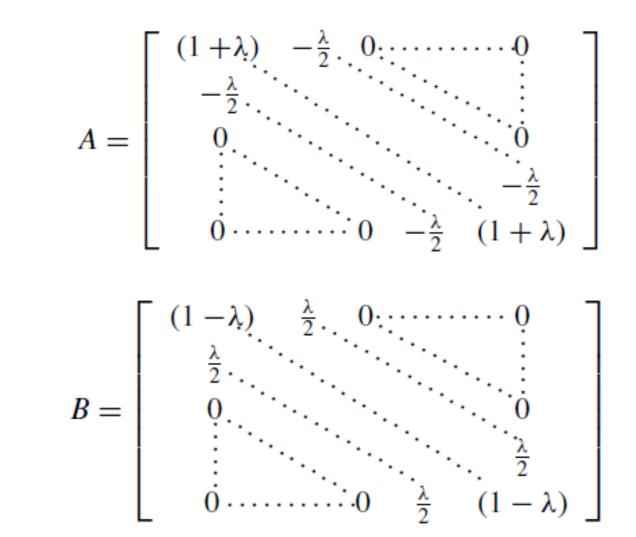
 $Au^{n+1} = Bu^n$

where,

$$u^{n} = \begin{pmatrix} u_{1}^{n} & u_{2}^{n} & \dots & u_{m-1}^{n} \end{pmatrix}^{t}$$



and the matrices A and B are given by,



and,

- The matrix A is **positive definite**, **strictly diagonally dominant** and **tridiagonal**. So, **Crout factorization** (for rather small systems) or **SOR** method (for large systems) can be used to solve the system.
- The Crank-Nicolson method is **unconditionally stable** and has local truncation error $O((\Delta x)^2 + (\Delta t)^2)$

Example

Use the Crank-Nicolson method with $\Delta x = 0.1$ and $\Delta t = 0.01$ to approximate the solution to the heat equation

$$\frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < 1, \quad 0 < t$$

subject to the constraints

u(0,t) = u(1,t) = 0, 0 < t, $u(x,0) = \sin \pi x$, $0 \le x \le 1$

Solution

The following table represents the results for the Crank-Nicolson method.

<i>exact</i>			
x_i	u_i^{50}	$u(x_i, 0.5)$	$ u_i^{50} - u(x_i, 0.5) $
0.0	0	0	
0.1	0.00230512	0.00222241	8.271×10^{-5}
0.2	0.00438461	0.00422728	1.573×10^{-4}
0.3	0.00603489	0.00581836	2.165×10^{-4}
0.4	0.00709444	0.00683989	2.546×10^{-4}
0.5	0.00745954	0.00719188	2.677×10^{-4}
0.6	0.00709444	0.00683989	2.546×10^{-4}
0.7	0.00603489	0.00581836	2.165×10^{-4}
0.8	0.00438461	0.00422728	1.573×10^{-4}
0.9	0.00230512	0.00222241	8.271×10^{-5}
1.0	0	0	

- Recall that the Forward-Difference method gave dramatically poor results for this choice of Δt and Δx , but the Backward-Difference method gave results that were accurate to about 2×10^{-3} for entries in the middle of the table.
- The results in the table indicate the increase in accuracy of the Crank-Nicolson method over the Backward-Difference method.

Solving Hyperbolic Partial Differential Equations

$$\frac{\partial^2 u}{\partial t^2}(x,t) - \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < l, \quad t > 0$$
$$u(0,t) = u(l,t) = 0, \quad \text{for} \quad t > 0$$
$$u(x,0) = f(x), \quad \text{and} \quad \frac{\partial u}{\partial t}(x,0) = g(x), \quad \text{for} \quad 0 \le x \le l$$

Using central approximation for time and spatial derivatives, we have,

$$\frac{u_{i}^{n+1} - 2u_{i}^{n} + u_{i}^{n-1}}{\left(\Delta t\right)^{2}} - \alpha^{2} \frac{u_{i+1}^{n} - 2u_{i}^{n} + u_{i-1}^{n}}{\left(\Delta x\right)^{2}} = 0 \qquad i = 1, 2, \dots, m-1$$

This equation can be rewritten as,

$$\mathbf{u}_{i}^{n+1} = 2\left(1-\lambda^{2}\right)\mathbf{u}_{i}^{n} + \lambda^{2}\left(\mathbf{u}_{i+1}^{n} + \mathbf{u}_{i-1}^{n}\right) - \mathbf{u}_{i}^{n-1} \qquad i = 1, 2, \dots, m-1$$

where, $\lambda = (\alpha \Delta t) / \Delta x$. Recalling that,

u(0,t) = u(l,t) = 0, t > 0, and u(x,0) = f(x), $0 \le x \le l$

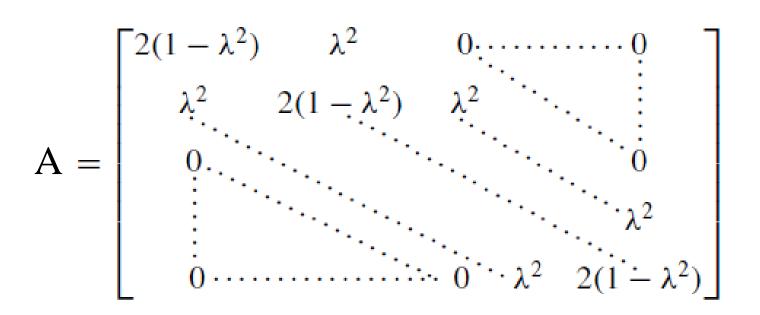
The discretized equation can be written in the matrix form as,

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

where,

$$\mathbf{u}^{\mathbf{n}} = \begin{pmatrix} \mathbf{u}_{1}^{\mathbf{n}} & \mathbf{u}_{2}^{\mathbf{n}} & \dots & \mathbf{u}_{\mathbf{m}-1}^{\mathbf{n}} \end{pmatrix}^{\mathsf{t}}$$

and,



The discretized form of the wave equation imply that the (n+1)th time step requires values from the *n*th and (n-1)th time steps. At the start

point, values for n=0 are given by,

 $u(x,0) = f(x), \quad 0 \le x \le l$

But, values for n=1 which are needed to compute u_i^2 , must be obtained from the initial-velocity condition,

$$\frac{\partial u}{\partial t}(x,0) = g(x), \quad 0 \le x \le l$$

One approach is to replace $\partial u/\partial t$ by a forward-difference approximation,

$$\frac{\partial u}{\partial t}(x_i, 0) = \frac{u_i^1 - u_i^0}{\Delta t}$$

The above equation can be rewrite as,

$$u_{i}^{1} = u_{i}^{0} + \Delta t \frac{\partial u}{\partial x} (x_{i}, 0)$$

Or,
$$u_{i}^{1} = u_{i}^{0} + (\Delta t)g(x_{i})$$

However, this approximation has truncation error of order $O(\Delta t)$, whereas truncation error in the main equation is of order $O((\Delta t)^2)$.

 ~ 2

Improving the Initial Approximation

$$u(x,t) = u(x,t^{n}) + \frac{\partial u}{\partial t} (x,t^{n}) (t-t^{n}) + \frac{\partial^{2} u}{\partial t^{2}} (x,t^{n}) \frac{(t-t^{n})^{2}}{2} + \frac{\partial^{3} u}{\partial t^{3}} (x,t^{n}) \frac{(t-t^{n})^{3}}{6} + \dots$$

The previous equation represents the Taylor expansion of u(x,t) about

$$t^{n}$$
. Setting $t^{n} = 0$, $x = x_{i}$ and $t = t^{1}$, we have,

$$\mathbf{u}(\mathbf{x}_{i},t^{1}) = \mathbf{u}(\mathbf{x}_{i},0) + \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}_{i},0)\Delta t + \frac{\partial^{2}\mathbf{u}}{\partial t^{2}}(\mathbf{x}_{i},0)\frac{(\Delta t)^{2}}{2} + O((\Delta t)^{3})$$

If f'' exists, then

$$\frac{\partial^2 u}{\partial t^2}(x_i, 0) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x_i, 0) = \alpha^2 \frac{d^2 f}{dx^2}(x_i) = \alpha^2 f''(x_i)$$

and

$$u(x_{i},t^{1}) = u(x_{i},0) + g(x_{i})\Delta t + f^{(2)}(x_{i})\frac{\alpha^{2}(\Delta t)^{2}}{2} + O((\Delta t)^{3})$$

This produces an approximation with error $O((\Delta t)^3)$,

$$u_i^1 = f(x_i) + g(x_i)\Delta t + f^{(2)}(x_i)\frac{\alpha^2(\Delta t)^2}{2}$$

If the second derivative of f exists but is not readily available, we can use the equation,

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{(\Delta x)^2}$$

This implies that,

$$u_{i}^{1} = f(x_{i}) + g(x_{i})\Delta t + \left[f(x_{i+1}) - f(x_{i}) + f(x_{i-1})\right] \frac{\alpha^{2}(\Delta t)^{2}}{2(\Delta x)^{2}}$$

Because $\lambda = (\alpha \Delta t) / \Delta x$, we ca write this as,

$$u_{i}^{1} = (1 - \lambda^{2})f(x_{i}) + \frac{\lambda^{2}}{2}f(x_{i+1}) + \frac{\lambda^{2}}{2}f(x_{i-1}) + g(x_{i})\Delta t$$

The method used for solving wave equation is **stable** only if $\alpha \frac{\Delta t}{\Delta x} \leq 1$.