Partial Differential Equations
Part 1

Ananda Dasgupta

PH3105, Autumn Semester 2017

The Heat equation in 1D

The general IBVP is given by

- The PDE

$$u_t = ku_{xx} + q(x, t)$$

The Heat equation in 1D

The general IBVP is given by

- The PDE

$$u_t = ku_{xx} + q\left(x, t\right)$$

- The Initial Condition

$$u\left(x, 0\right) = f\left(x\right)$$

The Heat equation in 1D

The general IBVP is given by

- The PDE

$$u_t = ku_{xx} + q(x,t)$$

- The Initial Condition

$$u(x,0) = f(x)$$

- The boundary conditions

$$a_1 u(0,t) + a_2 u_x(0,t) = g_1(t)$$

and

$$b_1 u(c,t) + b_2 u_x(c,t) = g_2(t)$$

The Heat equation in 1D

The general IBVP is given by

- The PDE

$$u_t = ku_{xx} + q(x,t)$$

- The Initial Condition

$$u(x,0) = f(x)$$

- The boundary conditions

$$a_1 u(0,t) + a_2 u_x(0,t) = g_1(t)$$

and

$$b_1 u(c,t) + b_2 u_x(c,t) = g_2(t)$$

In what follows we will solve the simpler case $q(x,t) = 0$

The finite difference scheme

- We begin with a discretization of space-time

$$u_i^n \equiv u\left(x_i, t_n\right) = u\left(x_0 + i\Delta x,\ t_0 + n\Delta t\right)$$

The finite difference scheme

- We begin with a discretization of space-time

$$u_i^n \equiv u\left(x_i, t_n\right) = u\left(x_0 + i\Delta x,\ t_0 + n\Delta t\right)$$

- We use finite difference approximations for the partial derivatives

$$u_t\left(x_i, t_n\right) = \lim_{\delta t \to 0} \frac{u\left(x_i, t_n + \delta t\right) - u\left(x_i, t_n\right)}{\delta t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

The finite difference scheme

- We begin with a discretization of space-time

$$u_i^n \equiv u\left(x_i, t_n\right) = u\left(x_0 + i\Delta x,\ t_0 + n\Delta t\right)$$

- We use finite difference approximations for the partial derivatives

$$u_t\left(x_i, t_n\right) = \lim_{\delta t \to 0} \frac{u\left(x_i, t_n + \delta t\right) - u\left(x_i, t_n\right)}{\delta t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

and

$$u_x\left(x_i, t_n\right) \approx \frac{u_{i+1}^n - u_i^n}{\Delta x}$$

The finite difference scheme

- We begin with a discretization of space-time

$$u_i^n \equiv u\left(x_i, t_n\right) = u\left(x_0 + i\Delta x,\ t_0 + n\Delta t\right)$$

- We use finite difference approximations for the partial derivatives

$$u_t\left(x_i, t_n\right) = \lim_{\delta t \to 0} \frac{u\left(x_i, t_n + \delta t\right) - u\left(x_i, t_n\right)}{\delta t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

and

$$u_x\left(x_i, t_n\right) \approx \frac{u_{i+1}^n - u_i^n}{\Delta x}$$

- We also need approximations for higher derivatives, e.g

$$u_{xx}\left(x_i, t_n\right) \approx \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\left(\Delta x\right)^2}$$

The finite difference scheme for the heat equation

- The PDE $u_t = ku_{xx}$ becomes

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

The finite difference scheme for the heat equation

- The PDE $u_t = ku_{xx}$ becomes

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

- This yields an explicit formula for u_i^{n+1} in terms of the "known" quantities $u_i^n, u_{i\pm 1}^n$:

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right)u_i^n, \quad i = 1, 2, \ldots, N-1$$

The finite difference scheme for the heat equation

- The PDE $u_t = ku_{xx}$ becomes
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = k \left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right]$$
- This yields an explicit formula for u_i^{n+1} in terms of the "known" quantities $u_i^n, u_{i\pm 1}^n$:

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2} \left(u_{i+1}^n + u_{i-1}^n \right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2} \right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- Note that the values u_i^0 can be obtained from the initial conditions $u(x, 0) = f(x)$: $u_i^0 = f(x_i)$

The finite difference scheme for the heat equation

- The PDE $u_t = ku_{xx}$ becomes

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

- This yields an explicit formula for u_i^{n+1} in terms of the "known" quantities $u_i^n, u_{i\pm 1}^n$:

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2} \left(u_{i+1}^n + u_{i-1}^n \right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2} \right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- Note that the values u_i^0 can be obtained from the initial conditions $u(x, 0) = f(x)$: $u_i^0 = f(x_i)$
- The extreme values u_0^{n+1} and u_N^{n+1} can be found using

$$a_1 u_0^{n+1} + a_2 \frac{u_1^{n+1} - u_0^{n+1}}{\Delta t} = g_1(t_{n+1})$$

$$b_1 u_N^{n+1} + b_2 \frac{u_N^{n+1} - u_{N-1}^{n+1}}{\Delta t} = g_2(t_{n+1})$$

The finite difference scheme for the heat equation

- The PDE $u_t = ku_{xx}$ becomes
$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = k \left[\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} \right]$$

- This yields an explicit formula for u_i^{n+1} in terms of the "known" quantities $u_i^n, u_{i\pm 1}^n$:

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2} \left(u_{i+1}^n + u_{i-1}^n \right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2} \right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- Note that the values u_i^0 can be obtained from the initial conditions $u(x,0) = f(x)$: $u_i^0 = f(x_i)$
- The extreme values u_0^{n+1} and u_N^{n+1} can be found using

$$a_1 u_0^{n+1} + a_2 \frac{u_1^{n+1} - u_0^{n+1}}{\Delta t} = g_1(t_{n+1})$$
$$b_1 u_N^{n+1} + b_2 \frac{u_N^{n+1} - u_{N-1}^{n+1}}{\Delta t} = g_2(t_{n+1})$$

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

- This leads to a set of simultaneous linear equations

$$-\mu u_{i+1}^{n+1} + (1 + 2\mu)\, u_i^{n+1} - \mu u_{i-1}^{n+1} = u_i^n, \quad i = 1, 2, \ldots N - 1$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

- This leads to a set of simultaneous linear equations

$$-\mu u_{i+1}^{n+1} + (1+2\mu)\, u_i^{n+1} - \mu u_{i-1}^{n+1} = u_i^n, \quad i = 1, 2, \ldots N-1$$

  where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$
- The extreme values follow from the BCs once again.

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

- This leads to a set of simultaneous linear equations

$$-\mu u_{i+1}^{n+1} + (1 + 2\mu)\, u_i^{n+1} - \mu u_{i-1}^{n+1} = u_i^n, \quad i = 1, 2, \ldots N - 1$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$

- The extreme values follow from the BCs once again.
- These have to be solved for all the u_is at $t = t_{n+1}$.

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = k u_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

- This leads to a set of simultaneous linear equations

$$-\mu u_{i+1}^{n+1} + (1+2\mu)\, u_i^{n+1} - \mu u_{i-1}^{n+1} = u_i^n, \quad i = 1, 2, \ldots N-1$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$

- The extreme values follow from the BCs once again.
- These have to be solved for all the u_is at $t = t_{n+1}$.
- The equations are sparse and so they can be solved efficiently.

The finite difference scheme for the heat equation

The implicit version

- We can get an implicit equation for u_i^{n+1} by replacing the RHS of $u_t = ku_{xx}$ by the finite difference approximation at t_{n+1}
- This gives

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

- This leads to a set of simultaneous linear equations

$$-\mu u_{i+1}^{n+1} + (1 + 2\mu)\, u_i^{n+1} - \mu u_{i-1}^{n+1} = u_i^n, \quad i = 1, 2, \ldots N - 1$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$

- The extreme values follow from the BCs once again.
- These have to be solved for all the u_is at $t = t_{n+1}$.
- The equations are sparse and so they can be solved efficiently.

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

- and

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

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

- and

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

- we took

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

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

- and

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

- we took

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

- This is the Crank-Nicolson algorithm.

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

- and

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

- we took

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

- This is the Crank-Nicolson algorithm.
- This, of course, is also an implicit algorithm.

The finite difference scheme for the heat equation

Crank-Nicolson algorithm

- What if, instead of

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

- and

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

- we took

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

- This is the Crank-Nicolson algorithm.
- This, of course, is also an implicit algorithm.

Stability

- Consider the IBVP

$$u_t = ku_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

Stability

- Consider the IBVP

$$u_t = ku_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.

Stability

- Consider the IBVP

$$u_t = ku_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.
- The explicit finite difference algorithm

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right)u_i^n, \quad i = 1, 2, \ldots, N-1$$

Stability

- Consider the IBVP

$$u_t = ku_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.
- The explicit finite difference algorithm

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- can be written in a matrix form $\mathbf{u}^{(n+1)} = A\mathbf{u}^{(n)}$ where $\mathbf{u}^{(n)} = \begin{pmatrix} u_1^n, & u_2^n, & \ldots, & u_{N-1}^n \end{pmatrix}^T$

Stability

- Consider the IBVP

$$u_t = k u_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.
- The explicit finite difference algorithm

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- can be written in a matrix form $\mathbf{u}^{(n+1)} = A\mathbf{u}^{(n)}$ where $\mathbf{u}^{(n)} = \begin{pmatrix} u_1^n, & u_2^n, & \ldots, & u_{N-1}^n \end{pmatrix}^T$
- and thus $\mathbf{u}^{(n)} = A^n \mathbf{u}^{(0)}$

Stability

- Consider the IBVP

$$u_t = k u_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.
- The explicit finite difference algorithm

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- can be written in a matrix form $\mathbf{u}^{(n+1)} = A\mathbf{u}^{(n)}$ where $\mathbf{u}^{(n)} = \begin{pmatrix} u_1^n, & u_2^n, & \ldots, & u_{N-1}^n \end{pmatrix}^T$
- and thus $\mathbf{u}^{(n)} = A^n \mathbf{u}^{(0)}$
- For $\mathbf{u}^{(n)}$ to decay to zero, all the eigenvalues of A must be smaller than 1 in magnitude.

Stability

- Consider the IBVP

$$u_t = k u_{xx}, \qquad u(x,0) = f(x), \qquad u(0,t) = u(L,t) = 0$$

- On physical grounds, we expect any solution to decrease to 0 over time.
- The explicit finite difference algorithm

$$u_i^{n+1} = \frac{k\Delta t}{(\Delta x)^2}\left(u_{i+1}^n + u_{i-1}^n\right) + \left(1 - 2\frac{k\Delta t}{(\Delta x)^2}\right) u_i^n, \quad i = 1, 2, \ldots, N-1$$

- can be written in a matrix form $\mathbf{u}^{(n+1)} = A\mathbf{u}^{(n)}$ where $\mathbf{u}^{(n)} = \begin{pmatrix} u_1^n, & u_2^n, & \ldots, & u_{N-1}^n \end{pmatrix}^T$
- and thus $\mathbf{u}^{(n)} = A^n \mathbf{u}^{(0)}$
- For $\mathbf{u}^{(n)}$ to decay to zero, all the eigenvalues of A must be smaller than 1 in magnitude.

Stability

- The $(N-1)\times(N-1)$ dimensional matrix A is

$$A = \begin{pmatrix} 1-2\mu & \mu & 0 & & & \\ \mu & 1-2\mu & \mu & & & \\ 0 & \mu & 1-2\mu & \ddots & & \\ \ddots & \ddots & \ddots & \ddots & & \\ & & & \mu & 1-2\mu & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$.

Stability

- The $(N-1) \times (N-1)$ dimensional matrix A is

$$A = \begin{pmatrix} 1-2\mu & \mu & 0 & & & \\ \mu & 1-2\mu & \mu & & & \\ 0 & \mu & 1-2\mu & \ddots & & \\ \ddots & \ddots & \ddots & \ddots & & \\ & & & \mu & 1-2\mu & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$.

- To find its eigenvalues we must solve $D_{N-1}(\lambda) = 0$ where $D_n(\lambda)$ is the determinant of the $n \times n$ tridigonal matrix whose diagonal elements are $2\alpha = 1 - 2\mu - \lambda$ and all the sub-diagonal elements are μ.

Stability

- The $(N-1) \times (N-1)$ dimensional matrix A is

$$A = \begin{pmatrix} 1-2\mu & \mu & 0 & & & \\ \mu & 1-2\mu & \mu & & & \\ 0 & \mu & 1-2\mu & \ddots & & \\ \ddots & \ddots & \ddots & \ddots & & \\ & & & \mu & 1-2\mu & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$.

- To find its eigenvalues we must solve $D_{N-1}(\lambda) = 0$ where $D_n(\lambda)$ is the determinant of the $n \times n$ tridigonal matrix whose diagonal elements are $2\alpha = 1 - 2\mu - \lambda$ and all the sub-diagonal elements are μ.
- The determinant obeys the recursion relation

$$D_n = 2\alpha D_{n-1} - \mu^2 D_{n-2}$$

Stability

- The $(N-1) \times (N-1)$ dimensional matrix A is

$$A = \begin{pmatrix} 1-2\mu & \mu & 0 & & & \\ \mu & 1-2\mu & \mu & & & \\ 0 & \mu & 1-2\mu & \ddots & & \\ \ddots & \ddots & \ddots & \ddots & & \\ & & & \mu & 1-2\mu & \mu \\ & & & & \mu & 1-2\mu \end{pmatrix}$$

where $\mu = \dfrac{k\Delta t}{(\Delta x)^2}$.

- To find its eigenvalues we must solve $D_{N-1}(\lambda) = 0$ where $D_n(\lambda)$ is the determinant of the $n \times n$ tridigonal matrix whose diagonal elements are $2\alpha = 1 - 2\mu - \lambda$ and all the sub-diagonal elements are μ.
- The determinant obeys the recursion relation

$$D_n = 2\alpha D_{n-1} - \mu^2 D_{n-2}$$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = Ar_1^n + Br_2^n$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = A r_1^n + B r_2^n$
- The constants A and B can be determined from
  - $D_0 = 1 \implies A + B = 1$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = Ar_1^n + Br_2^n$
- The constants A and B can be determined from
  - $D_0 = 1 \implies A + B = 1$
  - $D_1 = 2\alpha \implies Ar_1 + Br_2 = 2\alpha = r_1 + r_2$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = Ar_1^n + Br_2^n$
- The constants A and B can be determined from
  - $D_0 = 1 \implies A + B = 1$
  - $D_1 = 2\alpha \implies Ar_1 + Br_2 = 2\alpha = r_1 + r_2$
- leading to $A = \dfrac{r_1}{r_1 - r_2}, B = -\dfrac{r_2}{r_1 - r_2}$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = Ar_1^n + Br_2^n$
- The constants A and B can be determined from
  - $D_0 = 1 \implies A + B = 1$
  - $D_1 = 2\alpha \implies Ar_1 + Br_2 = 2\alpha = r_1 + r_2$
- leading to $A = \dfrac{r_1}{r_1 - r_2}, B = -\dfrac{r_2}{r_1 - r_2}$
- Thus $D_n = \dfrac{r_1^{n-1} - r_2^{n-1}}{r_1 - r_2}$

Stability

- To solve the recursion relation, we may try the Ansatz $D_n = r^n$
- This leads to $r^2 - 2\alpha r + \mu^2 = 0 \implies r_{1,2} = \alpha \pm \sqrt{\alpha^2 - \mu^2}$
- The most general solution, then, is $D_n = A r_1^n + B r_2^n$
- The constants A and B can be determined from
  - $D_0 = 1 \implies A + B = 1$
  - $D_1 = 2\alpha \implies A r_1 + B r_2 = 2\alpha = r_1 + r_2$
- leading to $A = \dfrac{r_1}{r_1 - r_2}, B = -\dfrac{r_2}{r_1 - r_2}$
- Thus $D_n = \dfrac{r_1^{n-1} - r_2^{n-1}}{r_1 - r_2}$
- $D_{N+1} = 0 \implies \dfrac{r_1}{r_2} = \exp\left(i\frac{2\pi l}{N}\right), \quad l = 0, 1, \ldots, N-1$

Stability

- $$\frac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i\frac{2\pi l}{N}\right)$$

Stability

- $$\frac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i\frac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2 - \mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i\sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2 - \mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)}$$

Stability

- $$\frac{\alpha+\sqrt{\alpha^2-\mu^2}}{\alpha-\sqrt{\alpha^2-\mu^2}} = \exp\left(i\frac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2-\mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i\sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2-\mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)} \implies \frac{\alpha}{\mu} = \cos\left(\frac{l\pi}{N}\right) \implies$$
  $$1-2\mu-\lambda = 2\mu\cos\left(\frac{l\pi}{N}\right)$$

Stability

- $$\frac{\alpha+\sqrt{\alpha^2-\mu^2}}{\alpha-\sqrt{\alpha^2-\mu^2}} = \exp\left(i\frac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2-\mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i\sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2-\mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)} \implies \frac{\alpha}{\mu} = \cos\left(\frac{l\pi}{N}\right) \implies$$
  $$1-2\mu-\lambda = 2\mu\cos\left(\frac{l\pi}{N}\right)$$
- $$\lambda = 1-2\mu\left(1+\cos\left(\frac{l\pi}{N}\right)\right)$$

Stability

- $$\frac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i \tfrac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2 - \mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i \sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2 - \mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)} \implies \frac{\alpha}{\mu} = \cos\left(\frac{l\pi}{N}\right) \implies$$
  $$1 - 2\mu - \lambda = 2\mu \cos\left(\frac{l\pi}{N}\right)$$
- $$\lambda = 1 - 2\mu\left(1 + \cos\left(\frac{l\pi}{N}\right)\right)$$
- For stability, we need $-1 < \lambda < 1$ for each $l = 0, 1, \ldots, N-1$

Stability

- $$\frac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i \frac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2 - \mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i \sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2 - \mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)} \implies \frac{\alpha}{\mu} = \cos\left(\frac{l\pi}{N}\right) \implies$$
  $$1 - 2\mu - \lambda = 2\mu \cos\left(\frac{l\pi}{N}\right)$$
- $$\lambda = 1 - 2\mu\left(1 + \cos\left(\frac{l\pi}{N}\right)\right)$$
- For stability, we need $-1 < \lambda < 1$ for each $l = 0, 1, \ldots, N-1$
- We need $-1 < 1 - 4\mu$ or $\mu < \dfrac{1}{2}$

Stability

- $\dfrac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i \frac{2\pi l}{N}\right) \implies \dfrac{\alpha}{\sqrt{\alpha^2 - \mu^2}} = \dfrac{\cos\left(\dfrac{l\pi}{N}\right)}{i \sin\left(\dfrac{l\pi}{N}\right)}$
- $\dfrac{\alpha^2}{\alpha^2 - \mu^2} = \dfrac{\cos^2\left(\dfrac{l\pi}{N}\right)}{-\sin^2\left(\dfrac{l\pi}{N}\right)} \implies \dfrac{\alpha}{\mu} = \cos\left(\dfrac{l\pi}{N}\right) \implies$

  $1 - 2\mu - \lambda = 2\mu \cos\left(\dfrac{l\pi}{N}\right)$
- $\lambda = 1 - 2\mu\left(1 + \cos\left(\dfrac{l\pi}{N}\right)\right)$
- For stability, we need $-1 < \lambda < 1$ for each $l = 0, 1, \ldots, N-1$
- We need $-1 < 1 - 4\mu$ or $\mu < \dfrac{1}{2}$
- Thus, the condition for the explicit algorithm to be stable is

$$\frac{k\Delta t}{(\Delta x)^2} < \frac{1}{2} \qquad \implies \qquad \Delta t < \frac{(\Delta x)^2}{2k}$$

Stability

- $$\frac{\alpha + \sqrt{\alpha^2 - \mu^2}}{\alpha - \sqrt{\alpha^2 - \mu^2}} = \exp\left(i\frac{2\pi l}{N}\right) \implies \frac{\alpha}{\sqrt{\alpha^2 - \mu^2}} = \frac{\cos\left(\frac{l\pi}{N}\right)}{i\sin\left(\frac{l\pi}{N}\right)}$$
- $$\frac{\alpha^2}{\alpha^2 - \mu^2} = \frac{\cos^2\left(\frac{l\pi}{N}\right)}{-\sin^2\left(\frac{l\pi}{N}\right)} \implies \frac{\alpha}{\mu} = \cos\left(\frac{l\pi}{N}\right) \implies$$
  $$1 - 2\mu - \lambda = 2\mu\cos\left(\frac{l\pi}{N}\right)$$
- $$\lambda = 1 - 2\mu\left(1 + \cos\left(\frac{l\pi}{N}\right)\right)$$
- For stability, we need $-1 < \lambda < 1$ for each $l = 0, 1, \ldots, N-1$
- We need $-1 < 1 - 4\mu$ or $\mu < \frac{1}{2}$
- Thus, the condition for the explicit algorithm to be stable is
  $$\frac{k\Delta t}{(\Delta x)^2} < \frac{1}{2} \qquad \implies \qquad \Delta t < \frac{(\Delta x)^2}{2k}$$

Stability

The implicit algorithm

- The algorithm

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

Stability

The implicit algorithm

- The algorithm

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

- Can be rewritten in the matrix form $A\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{b}$ where

$$= \begin{pmatrix} 1+2\mu & -\mu & 0 & & & \\ -\mu & 1+2\mu & -\mu & & & \\ 0 & -\mu & 1+2\mu & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ & & & -\mu & 1+2\mu & -\mu \\ & & & 0 & -\mu & 1+2\mu \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \beta \end{pmatrix}$$

Stability

The implicit algorithm

- The algorithm

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

- Can be rewritten in the matrix form $A\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{b}$ where

$$= \begin{pmatrix} 1+2\mu & -\mu & 0 & & & \\ -\mu & 1+2\mu & -\mu & & & \\ 0 & -\mu & 1+2\mu & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ & & & -\mu & 1+2\mu & -\mu \\ & & & 0 & -\mu & 1+2\mu \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \beta \end{pmatrix}$$

- To analyze stability, consider the case $\alpha = \beta = 0$, so that $\mathbf{u}^{(k)} = A^{-k}\mathbf{u}^{(0)}$

Stability

The implicit algorithm

- The algorithm

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

- Can be rewritten in the matrix form $A\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{b}$ where

$$= \begin{pmatrix} 1+2\mu & -\mu & 0 & & & \\ -\mu & 1+2\mu & -\mu & & & \\ 0 & -\mu & 1+2\mu & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ & & & -\mu & 1+2\mu & -\mu \\ & & & 0 & -\mu & 1+2\mu \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \beta \end{pmatrix}$$

- To analyze stability, consider the case $\alpha = \beta = 0$, so that $\mathbf{u}^{(k)} = A^{-k}\mathbf{u}^{(0)}$
- For stability, we need $\rho\left(A^{-1}\right) < 1$

Stability

The implicit algorithm

- The algorithm

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

- Can be rewritten in the matrix form $A\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{b}$ where

$$= \begin{pmatrix} 1+2\mu & -\mu & 0 & & & \\ -\mu & 1+2\mu & -\mu & & & \\ 0 & -\mu & 1+2\mu & \ddots & & \\ & \ddots & \ddots & \ddots & & \\ & & & -\mu & 1+2\mu & -\mu \\ & & & 0 & -\mu & 1+2\mu \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \beta \end{pmatrix}$$

- To analyze stability, consider the case $\alpha = \beta = 0$, so that $\mathbf{u}^{(k)} = A^{-k}\mathbf{u}^{(0)}$
- For stability, we need $\rho\left(A^{-1}\right) < 1$

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$
- Thus, those of $A^{-1}$are

$$\frac{1}{1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)}$$

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$
- Thus, those of $A^{-1}$are

$$\frac{1}{1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)}$$

- It is obvious that the denominator is always greater than 1

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$
- Thus, those of $A^{-1}$are

$$\frac{1}{1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)}$$

- It is obvious that the denominator is always greater than 1
- The algorithm is unconditionally stable.

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$
- Thus, those of $A^{-1}$are

$$\frac{1}{1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)}$$

- It is obvious that the denominator is always greater than 1
- The algorithm is unconditionally stable.
- So, by the way, is the Crank-Nicolson algorithm!

Stability

The implicit algorithm

- The eigenvalues of A are $1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)$
- Thus, those of $A^{-1}$are

$$\frac{1}{1 + 2\mu\left(1 - \cos\left(\frac{l\pi}{N}\right)\right)}$$

- It is obvious that the denominator is always greater than 1
- The algorithm is unconditionally stable.
- So, by the way, is the Crank-Nicolson algorithm!