1D Parabolic PDEs: Finite Difference Methods

M.Sc. in Mathematical Modelling & Scientific Computing,
Practical Numerical Analysis

Michaelmas Term 2025, Lecture 8

1D Heat Equation

First we consider the simplest parabolic PDE in the form of the heat equation:

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

for $t > 0$ and $x \in [a, b]$ with an initial condition

$$u(x, 0) = u_0(x) ,$$

for $x \in [a, b]$. We begin by considering Dirichlet boundary conditions

$$\begin{aligned} u(a, t) &= u_a(t) , \\ u(b, t) &= u_b(t) , \end{aligned}$$

for $t > 0$.

The Mesh

We define a sequence of uniform timesteps by

$$t_m = m\Delta t$$

for $m = 0, 1, 2, \ldots$ where $\Delta t > 0$ is the constant timestep size.

We also define a set of uniform mesh points by

$$x_j = a + j\Delta x \; ,$$

for $j = 0, 1, \ldots, N$ and with the meshsize $\Delta x = (b - a)/N$.

We write $u(x_j, t_m) = u_j^m$ and seek to approximate u_j^m by U_j^m for $j = 0, 1, \ldots, N$ and $m = 0, 1, 2, \ldots$.

Finite Difference Schemes

We may write a central difference

$$\frac{\partial^2 u}{\partial x^2}(x_j, t) \quad = \quad \frac{u(x_{j+1}, t) - 2u(x_j, t) + u(x_{j-1}, t)}{\Delta x^2} + \mathcal{O}(\Delta x^2) \; .$$

Similarly, (as was the case for ODEs) we may write a forward difference

$$\frac{\partial u}{\partial t}(x, t_m) \quad = \quad \frac{u(x, t_{m+1}) - u(x, t_m)}{\Delta t} + \mathcal{O}(\Delta t) \; ,$$

or a backward difference

$$\frac{\partial u}{\partial t}(x, t_{m+1}) \quad = \quad \frac{u(x, t_{m+1}) - u(x, t_m)}{\Delta t} + \mathcal{O}(\Delta t) \; .$$

Finite Difference Schemes

Alternatively we may combine these to get a θ-method of the form

$$(1-\theta)\frac{\partial u}{\partial t}(x, t_m) + \theta\frac{\partial u}{\partial t}(x, t_{m+1}) \quad = \quad \frac{u(x, t_{m+1}) - u(x, t_m)}{\Delta t} + \mathcal{O}(\Delta t)$$

for $\theta \neq 1/2$ or, when $\theta = 1/2$,

$$\frac{1}{2}\frac{\partial u}{\partial t}(x, t_m) + \frac{1}{2}\frac{\partial u}{\partial t}(x, t_{m+1}) \quad = \quad \frac{u(x, t_{m+1}) - u(x, t_m)}{\Delta t} + \mathcal{O}(\Delta t^2)\ .$$

Finite Difference Schemes

Such equalities lead to finite difference schemes of the form

- Forward Euler (or Explicit Euler)

$$\frac{U_j^{m+1} - U_j^m}{\Delta t} = \frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{\Delta x^2}$$

- Backward Euler (or Implicit Euler)

$$\frac{U_j^{m+1} - U_j^m}{\Delta t} = \frac{U_{j+1}^{m+1} - 2U_j^{m+1} + U_{j-1}^{m+1}}{\Delta x^2}$$

- θ-Method (Crank Nicolson when $\theta = 1/2$)

$$\frac{U_j^{m+1} - U_j^m}{\Delta t} = \theta \frac{U_{j+1}^{m+1} - 2U_j^{m+1} + U_{j-1}^{m+1}}{\Delta x^2} + (1-\theta) \frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{\Delta x^2}$$

Finite Difference Schemes

All these finite difference schemes hold for $j = 1, \ldots, N-1$ and $m = 0, 1, \ldots$.

We must also discretise the initial and boundary conditions as

$$\begin{aligned} U_j^0 &= u_0(x_j)\,, \quad j = 0, 1, \ldots, N \\ U_0^m &= u_a(t_m)\,, \quad m = 1, 2, \ldots \\ U_N^m &= u_b(t_m)\,, \quad m = 1, 2, \ldots \end{aligned}$$

Finite Differences — Implementation

We saw for ODEs that the forward Euler scheme was very simple to implement, whereas the θ-method for $\theta > 0$ required a nonlinear solve. Similar ideas hold for the heat equation but the nonlinear solve is replaced by the solution of a linear system.

Forward Euler Scheme

Recall the forward Euler scheme is

$$\frac{U_j^{m+1} - U_j^m}{\Delta t} = \frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{\Delta x^2}$$

for $j = 1, \ldots, N-1$ and $m = 0, 1, \ldots$. Writing $\mu = \Delta t / \Delta x^2$, we may re-arrange the scheme to get

$$U_j^{m+1} = U_j^m + \mu(U_{j+1}^m - 2U_j^m + U_{j-1}^m) \quad (1)$$

for $j = 1, \ldots, N-1$ and $m = 0, 1, \ldots$.

Thus, once we have used the initial and boundary conditions to assign values to U_j^0 for $j = 0, 1, \ldots, N$ and U_0^m and U_N^m for $m = 1, 2, \ldots$, it is simple to set $m = 0$ in Equation (1) and compute all the U_j^1 etc.

Forward Euler Scheme — Example

Consider the heat equation

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

for $0 < t \leq 0.1$ and $x \in [0, 1]$ with an initial condition

$$u(x, 0) = \sin(\pi x) + 2\pi \cos(2\pi x) ,$$

for $x \in [0, 1]$ and Dirichlet boundary conditions

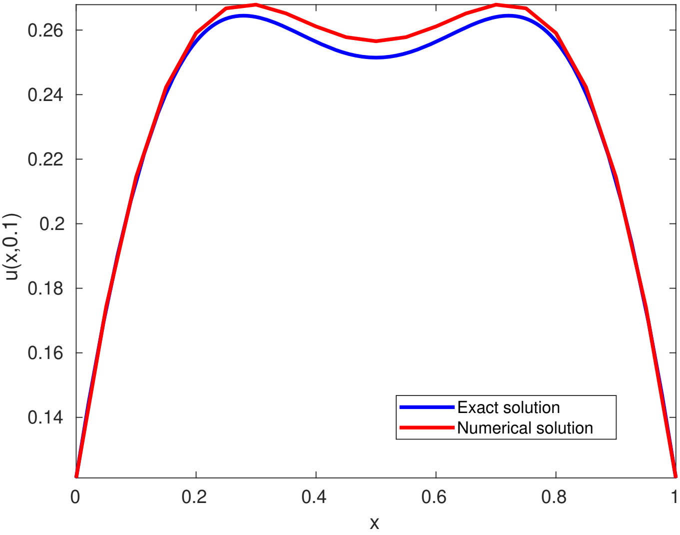
$$u(0, t) = 2\pi \exp(-4\pi^2 t) = u(1, t) ,$$

for $t > 0$. The exact solution is

$$u(x, t) = \exp(-\pi^2 t) \sin(\pi x) + 2\pi \exp(-4\pi^2 t) \cos(2\pi x) .$$

Forward Euler Scheme — Example

Exact solution and numerical solution at $t = 0.1$ with $\Delta x = 1/20$ and $\Delta t = \Delta x^2/4$.



θ-Method

The θ-method is

$$\frac{U_j^{m+1} - U_j^m}{\Delta t} = \theta \frac{U_{j+1}^{m+1} - 2U_j^{m+1} + U_{j-1}^{m+1}}{\Delta x^2} + (1-\theta)\frac{U_{j+1}^{m} - 2U_j^{m} + U_{j-1}^{m}}{\Delta x^2} .$$

(Recall this includes the backward Euler scheme if we take $\theta = 1$.)

Again we may write $\mu = \Delta t/\Delta x^2$ and re-arrange the scheme to get

$$U_j^{m+1} - \mu\theta(U_{j+1}^{m+1} - 2U_j^{m+1} + U_{j-1}^{m+1}) = U_j^m + \mu(1-\theta)(U_{j+1}^m - 2U_j^m + U_{j-1}^m) \quad (2)$$

for $j = 1, \ldots, N-1$ and $m = 0, 1, \ldots$.

This time, once we have used the initial condition to assign values to U_j^0 for $j = 0, 1, \ldots, N$, if we set $m = 0$ in Equation (2) then we have a linear system to solve in order to compute all the U_j^1.

θ-Method — Linear System I

Let $A \in \mathbb{R}^{(N+1)\times(N+1)}$ be the tridiagonal matrix given by

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

Then we may write

$$(I - \mu\theta A)\mathbf{U}^{m+1} = (I' + \mu(1-\theta)A)\mathbf{U}^m + \mathbf{g}^{m+1} .$$

Here, $\mathbf{U}^m = (U_0^m, U_1^m, \ldots, U_N^m)^T$, I is the $(N+1)\times(N+1)$ identity matrix, I' is the $(N+1)\times(N+1)$ identity matrix but with the $(1,1)$ and $(N+1,N+1)$ entries being zero, and $\mathbf{g}^{m+1} = (u_a(t_{m+1}), 0, \ldots, 0, u_b(t_{m+1}))^T$.

θ-Method — Linear System II

Alternatively note that when $j = 1$ Equation (2) becomes

$$\begin{aligned} U_2^{m+1} - \mu\theta(U_2^{m+1} - 2U_1^{m+1}) &= U_1^m + \mu(1-\theta)(U_2^m - 2U_1^m) \\ &\quad + \mu\theta U_0^{m+1} + \mu(1-\theta)U_0^m \,. \end{aligned}$$

Let $A_2 \in \mathbb{R}^{(N-1)\times(N-1)}$ be the tridiagonal matrix given by

$$A_2 = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} .$$

Then we may write

$$(I - \mu\theta A_2)\mathbf{U}^{m+1} = (I + \mu(1-\theta)A_2)\mathbf{U}^m + \mu\theta\mathbf{g_2}^{m+1} + \mu(1-\theta)\mathbf{g_2}^m \,.$$

Here, $\mathbf{U}^m = (U_1^m, U_2^m \ldots, U_{N-1}^m)^T$, I is the $(N-1)\times(N-1)$ identity matrix, and $\mathbf{g_2}^m = (u_a(t_m), 0, \ldots, 0, u_b(t_m))^T$.

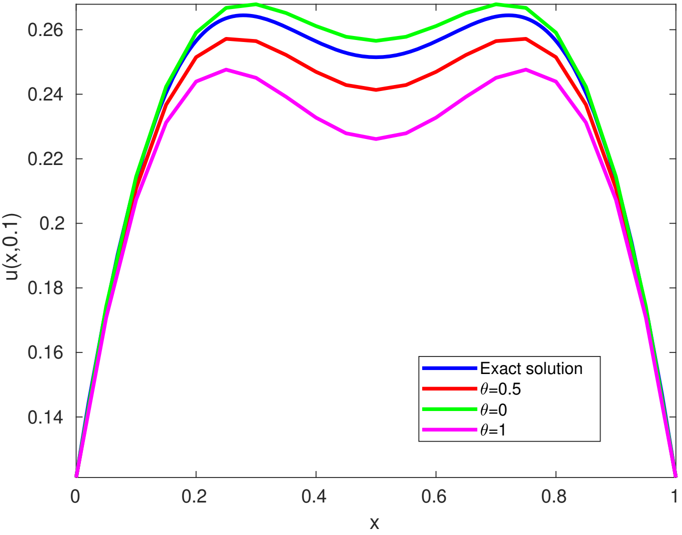
θ-Method — Linear System

Note that the matrix using the approach II is symmetric, but that using approach I is not.

Since both linear systems are tridiagonal either can be solved easily in Python using `numpy.linalg.solve` (or even better by constructing sparse matrices via `scipy.sparse` and using `scipy.sparse.linalg.spsolve`), or the Thomas Algorithm. Either method is fast, but not as fast as using Equation (1) for the forward Euler scheme.

θ-Method — Example

We consider the same problem and the same grid as before.



Truncation Error

The truncation error for the θ-method is given by

$$T_j^m = \frac{u_j^{m+1} - u_j^m}{\Delta t} - \theta D_x^+ D_x^- u_j^{m+1} - (1-\theta) D_x^+ D_x^- u_j^m \,.$$

It is standard to perform Taylor series approximations about the point $(x_j, t_{m+1/2})$. This gives

$$T_j^m = \left(\frac{1}{2} - \theta\right) \Delta t u_{xxt} - \frac{1}{12} \Delta t^2 u_{ttt} - \frac{1}{12} \Delta x^2 u_{xxxx} \,.$$

Thus for θ independent of Δt and Δx:

- in general, the θ-method is first order in Δt and second order in Δx;
- for the particular case $\theta = 1/2$, the Crank Nicolson method is second order in both Δt and Δx.

Stability

You will see in the NSPDE course that issues arise in the stability of the θ-method for parabolic PDEs. The summary is

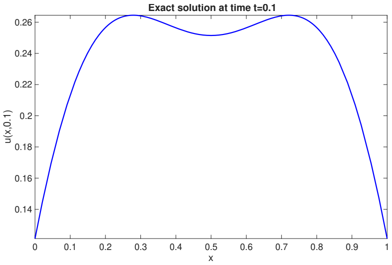
- If $\theta \geq 1/2$ the method is unconditionally stable. In particular this means that the backward Euler and Crank-Nicolson schemes are unconditionally stable.
- If $\theta < 1/2$ the method is only conditionally stable. The values of Δt and Δx must be chosen so that $\mu \leq 1/(2(1-2\theta))$, i.e. so that

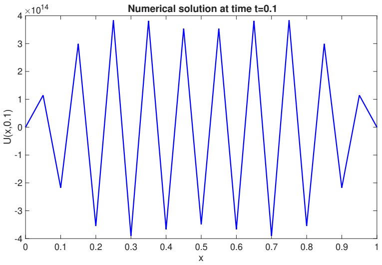
$$\Delta t \quad \leq \quad \frac{\Delta x^2}{2(1-2\theta)} \, .$$

  In particular this means that the forward Euler method is only conditionally stable and the condition for stability is that $\Delta t \leq \Delta x^2/2$.

Example of Instability

Suppose we try to solve the heat equation with Dirichlet boundary conditions and a fixed initial condition with the forward Euler scheme with $\Delta t = \Delta x^2$ (recall we need $\Delta t \leq \Delta x^2/2$ for stability). The solution is disastrous!





More General Boundary Conditions

Instead of applying Dirichlet boundary conditions, we may wish to apply Neumann boundary conditions or mixed boundary conditions. Let us consider a mixed boundary condition

$$\alpha u(a,t) + \beta \frac{\partial u}{\partial x}(a,t) \quad = \quad \gamma(t)$$

for α and β constants with β non-zero. (What follows is easily extended to the case when α and β are functions of time.)

More General Boundary Conditions

Since $x_0 = a$, we may write a forward difference

$$\frac{\partial u}{\partial x}(a,t) \;=\; \frac{\partial u}{\partial x}(x_0,t) \;\;=\;\; \frac{u(x_1,t)-u(x_0,t)}{\Delta x} + \mathcal{O}(\Delta x)\;.$$

This means we may approximate the mixed boundary condition using

$$\alpha U_0^{m+1} + \beta \frac{U_1^{m+1} - U_0^{m+1}}{\Delta x} \;\;=\;\; \gamma(t_{m+1})\;, \qquad (3)$$

for $m = 0, 1, \ldots$.

If we use this with the explicit Euler scheme then we have Equation (1) with $j = 1$,

$$U_1^{m+1} \;\;=\;\; U_1^m + \mu(U_2^m - 2U_1^m + U_0^m)$$

which couples with Equation (3) to give a 2×2 system for the unknowns U_0^{m+1} and U_1^{m+1}.

More General Boundary Conditions

If we use Equation (3) to approximate the mixed boundary condition with the θ-method then we need to adapt the first system we had earlier, namely

$$B\mathbf{U}^{m+1} := (I - \mu\theta A)\mathbf{U}^{m+1} = (I' + \mu(1-\theta)A)\mathbf{U}^{m} + \mathbf{g}^{m+1} .$$

We now replace the first entry of $\mathbf{g}^{m+1}$ with $\Delta x \gamma(t_{m+1})$ and the first row of the matrix B is now $(\alpha\Delta x - \beta, \beta, 0, \ldots, 0)$.

This method applies the boundary condition using an $\mathcal{O}(\Delta x)$ approximation.

More General Boundary Conditions — Fictitious Node

An alternative method for applying the boundary conditions is to use a central difference

$$\frac{\partial u}{\partial x}(a,t) \;=\; \frac{\partial u}{\partial x}(x_0,t) \;=\; \frac{u(x_1,t)-u(x_{-1},t)}{2\Delta x} + \mathcal{O}(\Delta x^2)\;,$$

where $x_{-1} = a - \Delta x$ is a fictitious node to the left of the left-hand end of the interval. This means we may approximate the mixed boundary condition using

$$\alpha U_0^m + \beta \frac{U_1^m - U_{-1}^m}{2\Delta x} \;=\; \gamma(t_m)\;, \tag{4}$$

for $m = 0, 1, \ldots$. To use this with the θ-method we use Equation (2) with $j = 0$, namely

$$\begin{aligned} U_0^{m+1} - \mu\theta(U_1^{m+1} - 2U_0^{m+1} + U_{-1}^{m+1}) \\ = U_0^m + \mu(1-\theta)(U_1^m - 2U_0^m + U_{-1}^m)\;. \end{aligned}$$

More General Boundary Conditions — Fictitious Node

We use Equation (4) to replace U_{-1}^{m+1} and U_{-1}^{m} in this finite difference scheme to get

$$\begin{aligned} U_0^{m+1} - \mu\theta \left(2U_1^{m+1} - 2U_0^{m+1} + \frac{2\alpha\Delta x}{\beta} U_0^{m+1} \right) \\ = \ U_0^m + \mu(1-\theta) \left(2U_1^m - 2U_0^m + \frac{2\alpha\Delta x}{\beta} U_0^m \right) \\ -2\mu\theta\Delta x \frac{\gamma(t_{m+1})}{\beta} - 2\mu(1-\theta)\Delta x \frac{\gamma(t_m)}{\beta} \ . \end{aligned}$$

Again we can use this to replace the first line of the linear system.

This method applies the boundary condition using an $\mathcal{O}(\Delta x^2)$ approximation.

More General Boundary Conditions — Comparison

We solve the problem

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

for $t > 0$ and $x \in (0, 1)$ with an initial condition

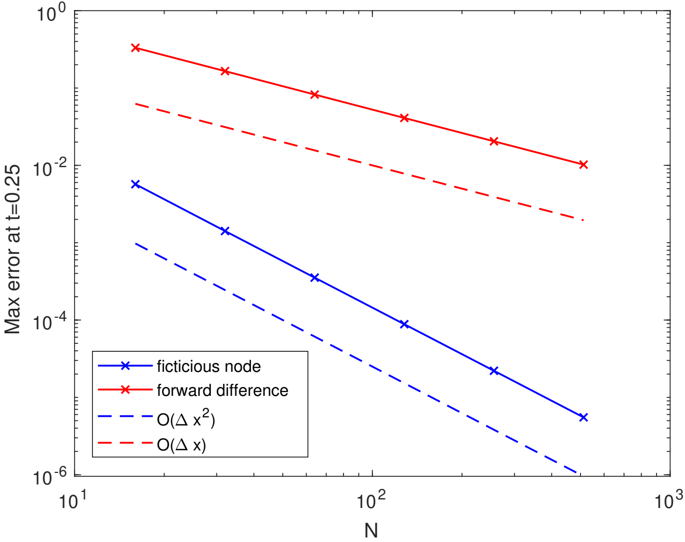
$$u(x,0) = \sin\left(\frac{3\pi x}{2}\right) - \left(\frac{3\pi}{2}\right)\cos\left(\frac{3\pi x}{2}\right),$$

for $x \in [0, 1]$. We use boundary conditions

$$\begin{aligned} 2u(0,t) + 3\frac{\partial u}{\partial x}(0,t) &= \frac{3\pi}{2}\mathrm{e}^{-(3\pi/2)^2 t}, \\ u(1,t) &= \mathrm{e}^{-(3\pi/2)^2 t}, \end{aligned}$$

for $t > 0$.

More General Boundary Conditions — Comparison



Method of Lines

What we have done above is to use the method of lines where we first discretise in space to get a system of ODEs and then use a numerical method to solve the ODEs.

So for the 1D heat equation, with homogeneous Dirichlet boundary conditions we can discretise in space using the standard finite difference scheme to get

$$\frac{\mathrm{d}U_j(t)}{\mathrm{d}t} = \frac{U_{j+1}(t) - 2U_j(t) + U_{j-1}(t)}{\Delta x^2}$$

for $j = 1, \ldots, N-1$ and with $U_0(t) = U_N(t) = 0$. We can re-write this as a system of ODEs of the form

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = A\mathbf{U}$$

with initial condition $\mathbf{U}(0) = u_0(\mathbf{x})$.

There is no reason why the spatial discretisation should be via a finite difference scheme — this could be replaced by a finite element method or a spectral method or . . .