NTNU

Norwegian University of Science and Technology

# TMA4125 Matematikk 4N

## Numerics for PDEs II

Ronny Bergmann

Department of Mathematical Sciences, NTNU.

March 10, 2023

# Numerical Methods for PDEs – Overview

**Goal.** Solve a Partial Differential Equation (PDE) numerically.

**Approach.** We will use finite difference methods.

Roughly speaking these consist of

1. Discretize the domain on which the equation is defined.
2. On each grid point:
   Approximate the involved derivatives by finite differences, using the values in neighbouring grid points.
3. Replace the exact solutions by their approximations.
4. Solve the resulting system of equations.

# Numerical Methods for PDEs – Roadmap

1. Numerical Differentiation – How to discretize derivatives?
2. Boundary Value Problems – How to tackle boundary conditions?
3. **Example.** The Heat Equation for some $c$

$$\frac{\partial}{\partial t}u(x,t) = c^2\frac{\partial^2}{\partial x^2}u(x,t), \qquad 0 \leq x \leq L$$

$$u(0,t) = g_0(t), \quad u(1,t) = g_1(t), \qquad \text{Boundary conditions}$$

$$u(x,0) = f(x) \qquad \text{Initial conditions}$$

which we aim to solve for some time interval, that is for $t \in [0,T]$.

**This week.** Numerical schemes taking into account boundary conditions and stability.

$\Rightarrow$ we have to figure out how to discretize time and space.

# Recap. (Different) Boundary Conditions

To get a unique solution of a BVP (or a PDE): more information required, usually given on the the boundaries

We already learned about the most common boundary conditions

1. **Dirichlet conditions.** The solution is known at the boundary.
   We know the temperature $u(0,t) = g_0(t)$ and $u(L,t) = g_L(t)$ on the boundary
2. **Neumann conditions.** The derivative is known at the boundary.
   We know the heat flux $\frac{\partial u}{\partial x}(0,t) = g_0(t)$ and $\frac{\partial u}{\partial x}(L,t) = g_L(t)$
3. **Robin (or mixed) conditions.** A combination of those.
   We might for example know $a_x u(x,t) + b_x \frac{\partial u}{\partial x}(x,t)$ at $x = 0$ and $x = L$

Until now we mostly considered.
Numerical Methods with (zero) Dirichlet boundary conditions.

# Recap. Numerical Differentiation

**Goal.** Numerical approximation of $f'$ and $f''$.

**Approach.** Take some step length $h > 0$ and for

**First order.** For the first derivative we considered

**Forward Difference** $f'(x) = \dfrac{f(x+h) - f(x)}{h} + \mathcal{O}(h)$

**Backward Difference** $f'(x) = \dfrac{f(x) - f(x-h)}{h} + \mathcal{O}(h)$

**Cental Difference** $f'(x) = \dfrac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2)$

**Second order.** Similarly we combine a forward and a backward difference to

$$f''(x) = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} + \mathcal{O}(h^2)$$

# A Grid of points

With the discretization of space and time

$$x_i = x_0 + ih, \qquad i = 0, \dots, M, \quad h = \frac{L}{M}$$
$$t_n = t_0 + nk, \qquad n = 0, \dots, N, \quad k = \frac{T}{N}$$

We obtain a grid of points (See sketch).

**Idea.** We approximate the partial derivatives in the Heat equation as

$$\frac{\partial}{\partial t} u(x,t) = \frac{u(x,t+k) - u(x,t)}{k} + \mathcal{O}(k) \quad \text{(temporal 1st deriv., fw.)}$$
$$\frac{\partial^2}{\partial x^2} u(x,t) = \frac{u(x-h,t) - 2u(x,t) - u(x+h,t)}{h^2} + \mathcal{O}(h^2) \quad \text{(spatial 2nd deriv.)}$$

but only at our grid points which we denote by $U_i^n \approx u(x_i, t_n)$.

# The Explicit Scheme – Forward Euler Method

1. From the initial conditions we know $U_i^0 = u(x_i, 0) = f(x_i)$
2. For each time point $n = 0, 1, 2, \ldots$

   2.1 We have to take into account the boundary conditions

   - Dirichlet: We have $u(0,t) = g_0(t)$ and $u(L,t) = g_L(t)$

   $\Rightarrow$ We can just set $U_0^{n+1} = g_0(t_{n+1})$ and $U_N^{n+1} = g_L(t_{n+1})$
   (other BC on the next slide.)

   2.2 we compute

   $$U_i^{n+1} = U_i^n + \alpha\big(U_{i-1}^n - 2U_i^n + U_{i+1}^n\big),$$

   for $i = 1, \ldots, N-1$, where $\alpha = \frac{c^2 k}{h^2}$

# Neumann Boundary Conditions

If we have

$$\frac{\partial u}{\partial x}(0,t) = g_0(t) \qquad \text{and} \quad \frac{\partial u}{\partial x}(L,t) = g_L(t)$$

for two given functions $h_0, h_L$.
What can we do here for the Step 2.1 on the last slide?
Use finite differences!

**Example.** On the left hand side at $x = 0$ we obtain

$$U_0^{n+1} = U_0^n + 2\alpha(U_1^n - U_0^n + hg_0(t_n))$$

and similarly on the right for $x = L$ we obtain

$$U_M^{n+1} = U_M^n + 2\alpha(U_{M-1}^n - U_M^n + hg_L(t_n))$$

NTNU

# Robin Boundary Conditions

Mixing both the Dirichlet case and the Neumann case we have If we have

$$a_0 \frac{\partial u}{\partial x}(0,t) + b_0 u(0,t) = g_0(t) \qquad \text{and } a_L \frac{\partial u}{\partial x}(L,t) + b_L u(L,t) = g_L(t)$$

You can combine the last two cases and obtain

$$U_0^{n+1} = U_0^n + 2\alpha(U_1^n - U_0^n - \frac{hb_0}{a_0}U_0^n + \frac{h}{a_0}g_0(t_n))$$

as well as

$$U_M^{n+1} = U_M^n + 2\alpha(U_{M-1}^n - U_M^n - \frac{hb_L}{a_L}U_M^n + \frac{h}{a_0}g_L(t_n))$$

# Matrix Notation – homogen. Neumann BC example 1/2

If we set

$$\frac{\partial u}{\partial x}(0,t) = g_0(t) = 0 \qquad \text{and} \qquad \frac{\partial u}{\partial x}(L,t) + b_L u(L,t) == g_L(t) = 0$$

we obtain the homogeneous Neumann boundary conditions.

Then our equations look like

$$\begin{aligned}
U_0^{n+1} &= U_0^n + 2\alpha(U_1^n - U_0^n) \\
U_i^{n+1} &= U_i^n + \alpha\big(U_{i-1}^n - 2U_i^n + U_{i+1}^n\big), \qquad \text{for } i = 1, \dots, M-1 \\
U_M^{n+1} &= U_M^n + 2\alpha(U_{M-1}^n - U_M^n)
\end{aligned}$$

We can write these nicer using Matrix-Vector notation. We introduce $\mathbf{U}^n = (U_0^n, \dots, U_M^n)^{\mathrm{T}} \in \mathbb{R}^{M+1}$ and similarly $\mathbf{U}^{n+1}$

# Matrix Notation – homogen. Neumann BC Example 2/2

$$A = \begin{pmatrix} -2 & 2 & 0 & \cdots & & & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & & & \cdots & 0 & 2 & -2 \end{pmatrix} \in \mathbb{R}^{(M+1)\times(M+1)}.$$

and denote by $I_{M+1} \in \mathbb{R}^{(M+1)\times(M+1)}$ be the $(M+1)$-dimensional identity matrix.

$\Rightarrow$ We can also write the updates as

$$\mathbf{U}^{n+1} = (I_{M+1} + \alpha A)\mathbf{U}^{n}.$$

Let's look at this in an example in code

# (In)Stability

**Idea** (informally)**.** We want to avoid, that the solution “explodes”.

Naming $Q \coloneqq (I_{M+1} + \alpha A)$ and using $\mathbf{U}^0 = \big(f(x_0, \dots, f(x_M)\big)^{\mathrm{T}}$ we get

$$\mathbf{U}^{n+1} = Q^{n+1}\mathbf{U}^0$$

$\Rightarrow$ the largest “scaling” the matrix $Q$ introduces has to be less than one. This corresponds to considering the largest Eigenvalue.

We obtain that for Stability we need $\alpha = \frac{c^2 k}{h^2} \leq \frac{1}{2}$
$\Rightarrow$ If we want to take half the step size ($\frac{h}{2}$) in space, we have to take $\frac{k}{4}$ stepsize in time to still get the same $\alpha$.

Or more drastically, if we want to increase the time resolution to $\frac{h}{10}$, we have to take $\frac{k}{100}$ in time!

# Implicit Euler Method

**Idea.** Take a backwards difference in time instead.

We obtain

$$U_i^{n+1} - c^2\frac{k}{h^2}\left(U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}\right) = U_i^n$$

where we do not get an explicit formula for each $U_i^{n+1}$ but an implicit one, where these new values (on the left) depend on each other.

We shorten again $\alpha = c^2\frac{k}{h^2}$.

We again consider first **Dirichlet** boundary conditions. and obtain

$$U_1^{n+1} - \alpha(U_2^{n+1} - 2U_1^{n+1}) = U_1^n - \alpha g_0(t_{n+1})$$
$$U_{M-1}^{n+1} - \alpha(U_{M-2}^{n+1} - 2U_{M-1}^{n+1}) = U_{M-1}^n - \alpha g_L(t_{n+1})$$

# Implicit Euler Method (Dirichlet BC) in Matrix Form

Using the matrix $B \in \mathbb{R}^{M-1 \times M-1}$ and the mathbftor $\mathbf{b}^{n+1} \in \mathbb{R}^{M-1}$ given by

$$B = \begin{pmatrix} -2 & 1 & 0 & \cdots & & & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & & & \cdots & 0 & 1 & -2 \end{pmatrix} \quad \text{and} \quad \mathbf{b}^{n+1} = \begin{pmatrix} g_0(t_{n-1}) \\ 0 \\ 0 \\ \vdots \\ 0 \\ g_L(t_{n-1}) \end{pmatrix}$$

and the identity $I_{M-1}$ as before, we can write the equations for $\mathbf{U}^{n+1} = \left(u_1^{n+1}, \ldots, U_{M-1}^{n+1}\right)^{\mathrm{T}}$ as

$$(I_{M-1} - \alpha B)\mathbf{U}^{n+1} = \mathbf{U}^n + \alpha \mathbf{U}^{n+1}$$

$\Rightarrow$ This is a linear system of equations $\Rightarrow$ Gaussian elimination
– but since it's tridiagonal, even faster methods available ($\mathcal{O}(M)$)!

# Implicit Euler with Neumann BC

For Neumann boundary conditions we (again) have

$$\partial_x u(0,t) = g_0(t), \qquad \text{and} \qquad \partial_x u(L,t) = g_L(t).$$

$\Rightarrow$ Same approach as before (with detour via $U_{-1}^{n-1}$ and $U_{M+1}^{n+1}$) yields

$$\begin{aligned} U_0^{n+1} - 2\alpha\big(U_1^{n+1} - U_0^{n+1}\big) &= U_0^n + 2\alpha h g_0(t_{n+1}),\\ U_i^{n+1} - \alpha\big(U_{i-1}^{n+1} - 2U_i^{n+1} + U_{i+1}^{n+1}\big) &= U_i^n, \qquad \text{for } i = 1,\ldots,M-1,\\ U_M^{n+1} - 2\alpha\big(U_{M-1}^{n+1} - U_M^{n+1}\big) &= U_M^n - 2\alpha h g_L(t_{n+1}). \end{aligned}$$

$\Rightarrow$ To capture the two highlighted terms, we introduce
$\mathbf{a}^{n+1} = \big(g_0(t_{n+1}), 0, 0, \ldots, 0, -g_L(t_{n+1})\big)^{\mathrm{T}}$.
To obtain $\mathbf{U}^{n+1} = (U_0^{n+1}, \ldots, U_M^{n+1})^{\mathrm{T}}$: Using $A$ from before we get

$$(I_{M+1} - \alpha A)\mathbf{U}^{n+1} = \mathbf{U}^n + 2\alpha h \mathbf{a}^{n+1}.$$

$\Rightarrow$ again a tridiagonal system $\Rightarrow$ efficiently solvable,

# Implicit Euler – Remarks

- The computations are only slightly more costly than for Explicit Euler
- The method is unconditionally stable

$\Rightarrow$ in principle we can use arbitrarily large step sizes

- in practice: For accuracy still small step sizes required (just their ratio in $\alpha$ not so important)

**Approximation Errors.** We used were

- $\mathcal{O}(h^2)$ in space (second order difference)
- $\mathcal{O}(k)$ in time (backward difference)

$\Rightarrow$ to reduce the error by a factor $\frac{1}{4}$: we require $\frac{k}{4}$ but only $\frac{h}{2}$.

Or informally: $k$ has to “behave like” $h^2$.

# Crank-Nicolson Method

**Idea.** Combine Explicit and Implicit Euler Methods.

For example for Neumann boundary conditions, both methods read

$$\begin{aligned} \mathbf{U}^{n+1} &= (I_{M+1} + \alpha A)\mathbf{U}^n + 2\alpha h \mathbf{a}^n, \\ (I_{M+1} - \alpha A)\mathbf{U}^{n+1} &= \mathbf{U}^n + 2\alpha h \mathbf{a}^{n+1}, \end{aligned}$$

**Approach.** Take the average of both. We get
(remember that here $\mathbf{U}^{n+1} \in \mathbb{R}^{M+1}$, includes $U_0^{n+1}, U_M^{n+1}$)

$$\Big(I_{M+1} - \frac{\alpha}{2}A\Big)\mathbf{U}^{n+1} = \Big(I_{M+1} + \frac{\alpha}{2}A\Big)\mathbf{U}^n + \alpha h\big(\mathbf{a}^n + \mathbf{a}^{n+1}\big).$$

**Similarly for Dirichlet.** taking the average we get
(remember that here $\mathbf{U}^{n+1} \in \mathbb{R}^{M-1}$ does not include the boundary)

$$\Big(I_{M-1} - \frac{\alpha}{2}B\Big)\mathbf{U}^{n+1} = \Big(I_{M-1} + \frac{\alpha}{2}B\Big)\mathbf{U}^n + \frac{\alpha}{2}\big(\mathbf{b}^n + \mathbf{b}^{n+1}\big).$$

# Crank-Nicolson Method

The overall algorithm includes

- Choosing stepsizes $h = \frac{L}{M}$, $k = \frac{T}{N}$
- Setting up $U_i^0$
- Setting up $A$ (or $B$ depending on the BC)
- iterating the updates from last slide for $n = 0, 1, 2, \dots$

**Numerical Cost.** The cost is only slightly higher than for the implicit Euler Method.
**Numerical Error.** The numeical error is of order $\mathcal{O}(h^2 + k^2)$.

# Modelling very (very) long bars

NTNU
Norwegian University of Science and Technology

# The Heat Equation on the Real line.

**Motivation / Model Problem.** Consider the Heat equation, $L, c > 0$, with zero Dirichlet BC

$$\frac{\partial}{\partial t}u = c^2 \frac{\partial^2}{\partial x^2}u, \qquad -\frac{L}{2} \leq x \leq \frac{L}{2}, t \geq 0$$

$$u(x,0) = f(x) \qquad \text{(Initial Condition)}$$

$$u(-\tfrac{L}{2}, t) = u(\tfrac{L}{2}, t) = 0, \qquad \text{(Boundary Conditions)}$$

What if we let $L$ tend to $\infty$ ?

**The infinite wire.**

$$\frac{\partial}{\partial t}u = c^2 \frac{\partial^2}{\partial x^2}u, \qquad x \in \mathbb{R}, t \geq 0$$

$$u(x,0) = f(x) \qquad \text{(Initial Condition)}$$

$$\lim_{x \to \pm\infty} u(x,t) = 0, \qquad \text{(Boundary Conditions)}$$

$\Rightarrow$ We use the Fourier Transform!

NTNU

# Recap. The Fourier Transform.

For $f \in L_1(\mathbb{R})$ , the Fourier Transform is defined by

$$\hat{f}(\omega) := \mathcal{F}(f)(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)\mathrm{e}^{-\mathrm{i}\omega x}\,\mathsf{d}x, \qquad \omega \in \mathbb{R}.$$

And for a function $g(\omega) \in L_1(\mathbb{R})$ the inverse Fourier transform is defined by

$$\check{g}(x) := \mathcal{F}^{-1}(g)(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} g(\omega)\mathrm{e}^{\mathrm{i}\omega x}\,\mathsf{d}\omega, \qquad x \in \mathbb{R}.$$

**An important Property.** $\mathcal{F}(f') = \mathrm{i}\omega\mathcal{F}(f)$ or shorter $\widehat{(f')} = i\omega\hat{f}$

**An example.** For the Gaussian function, $a > 0$ we have that (p.534, Kreyszig)

$$\mathcal{F}(\mathrm{e}^{-ax^2}) = \frac{1}{\sqrt{2a}}\mathrm{e}^{\frac{-\omega^2}{4a}}$$

# Simplifying the PDE to an ODE “in Fourier Domain”

Using the Fourier transform in space $\mathcal{F}_x(u(x,t)) = \hat{u}(\omega,t)$ we get

$$\frac{\partial}{\partial t}\hat{u}(\omega,t) = -c^2\omega^2\hat{u}(\omega,t)$$

which is an ODE in time $t$.

**Solution of the ODE.** For each $\omega$ the solution is given by

$$\hat{u}(\omega,t) = C(\omega)\mathrm{e}^{-c^2\omega^2 t}$$

How can we find $C(\omega)$? We still have the initial condition $u(x,0) = f(x)$.
$\Rightarrow \hat{u}(\omega,0) = \hat{f}(\omega) = C(\omega)e^{-c^2\omega^2 0} = C(\omega)$
$\Rightarrow$ Fourier transform the initial condition!

**Solution in Fourier Domain.**

$$\hat{u}(\omega,t) = \hat{f}(\omega)\mathrm{e}^{-c^2\omega^2 t}$$

$\Rightarrow$ Use the inverse Fourier transform to obtain $u(x,t) = \mathcal{F}_x^{-1}(\hat{u}(\omega,t))$.

# Summary / Roadmap to solve the Heat Equation on $\mathbb{R}$

To solve

$$\frac{\partial}{\partial t}u = c^2\frac{\partial^2}{\partial x^2}u, \qquad x\in\mathbb{R}, t\geq 0$$

$$u(x,0) = f(x) \qquad \text{(Initial Condition)}$$

$$\lim_{x\to\pm\infty} u(x,t) = 0, \qquad \text{(Boundary Conditions)}$$

1. use the Fourier transform $\hat{u}(\omega,t)$ to “turn” the second derivative (in space $x$) into a multiplication in frequency $\omega$
2. use the initial condition to obtain $\hat{u}(\omega,t) = \hat{f}(\omega)\mathrm{e}^{-c^2\omega^2 t}$
3. use the inverse Fourier transform $\mathcal{F}_x^{-1}$ obtain $u(x,t)$

**Even better!**
In Step 2 we have a multiplication $\hat{f}(\omega)\cdot\mathrm{e}^{-c^2\omega^2 t}$ in frequency!
$\Rightarrow$ We have a convolution in space!

# The overall solution $u(x,t)$

We derived from $\hat{g}(\omega) = \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-c^2\omega^2 t}$ that

$$g(x) = \frac{1}{2c^2t} \cdot \frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{x^2}{4c^2t}}$$

and hence

$$u(x,t) = (f * g)(x) = \frac{1}{2c\sqrt{\pi t}} \int_{-\infty}^{\infty} f(y)\mathrm{e}^{-\frac{(x-y)^2}{4c^2t}}\,\mathrm{d}y$$

**Example.** Let's set $c = 1$ and choose a specific $f$.
That is, we want to solve

$$\frac{\partial}{\partial t}u = \frac{\partial^2}{\partial x^2}u, \qquad x \in \mathbb{R}, t \geq 0$$

$$u(x,0) = f(x) = \begin{cases} 1 & \text{if } |x| \leq 1 \\ 0 & \text{else,} \end{cases} \qquad \text{(Initial Condition)}$$