TMA4125 Matematikk 4N

Numerical Methods for PDEs

Ronny Bergmann

Department of Mathematical Sciences, NTNU.

April 20, 2022

NTNU

Norwegian University of Science and Technology

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, replace the derivatives with an approximation, 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

$$
\begin{aligned}
\frac{\partial}{\partial t} u(x,t) &= \frac{\partial^2}{\partial x^2} u(x,t), && 0 \leq x \leq 1 \\
u(0,t) &= g_0(t), \quad u(1,t) = g_1(t), && \text{Boundary conditions} \\
u(x,0) &= f(x) && \text{Initial conditions}
\end{aligned}
$$

which we aim to solve for some time interval $[0, T]$.
$\Rightarrow$ we have to figure out how to discretize time and space.

Numerical Differentiation

Question. How to approximate derivatives, for us $f'(x)$ and $f''(x)$, at some point x using only point evaluations of f?

Idea. Since the derivative is defined as

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h}$$

we take a small h on the right as an approximation.
We obtain obtain for $h > 0$

- $f'(x) \approx \partial^+ f(x) := \dfrac{f(x+h) - f(x)}{h}$ forward difference.
- $f'(x) \approx \partial^- f(x) := \dfrac{f(x) - f(x-h)}{h}$ backward difference.
- $f'(x) \approx \partial^\circ f(x) := \dfrac{f(x+h) - f(x-h)}{2h}$ central difference.

where the last one is the mean of the first two.

Numerical Differentiation II

Idea. For the second derivative – Do the approximation twice.

We obtain a common approximation of the second derivative as the central difference operator for some $h > 0$

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

Example. For $f(x) = \sin(x)$ and $x = \frac{\pi}{4}$ we can check how well these approximations work.

Error Analysis

Goal. Investigate the error $e(x;h)$ w. r. t. h using a Taylor expansion.
Main question. How fast does the error decrease when $h \to 0$?

Errors. We summarize

$$f'(x) = \begin{cases} \dfrac{f(x+h)-f(x)}{h} - \dfrac{h}{2} f''(\xi), & \text{Forward difference} \\ \dfrac{f(x)-f(x-h)}{h} + \dfrac{h}{2} f''(\xi), & \text{Backward difference} \\ \dfrac{f(x+h)-f(x-h)}{2h} - \dfrac{h^2}{6} f'''(\xi). & \text{Central difference} \end{cases}$$

$$f''(x) = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi), \qquad \text{Central difference}$$

Or in other words: We obtain the approximation orders

- forward and backward differences: 1
- first and second order central differences: 2

The 1D Poisson Problem

We start with a few terms only: Let $[a, b]$ be an interval and $f\colon [a, b] \to \mathbb{R}$ a function.

We consider the two point boundary problem to find $u\colon [a, b] \to \mathbb{R}$ such that

$$\begin{aligned} -u''(x) &= f(x) \qquad \text{for } a \leq x \leq b \\ u(a) &= u_a \\ u(b) &= u_b \end{aligned}$$

where $u_a, u_b \in \mathbb{R}$ are given values.

Simplification. Let's consider for this example $a = 0$ and $b = 1$.

Comparison. This is a second order ODE.
Instead of two inital values we have two boundary values,
since we are in space domain x.

The 1D Poisson Problem – Discretization

Step 1. Discretize

Numerically we can not handle a function u.
$\Rightarrow$ We discretize $[a, b] = [0, 1]$ for a given $N \in \mathbb{N}$ and $h = \frac{b-a}{N} = \frac{1}{N}$:

$$x_i := a + ih \quad \text{for } i = 0, 1, \dots, N,$$

i. e. we use equally spaced points with $x_0 = a = 0$ and $x_N = b = 1$.

Step 2. We approximate the derivative u'' using central differences

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

on the internal grid points $x_1, \dots, x_{N-1}$

The 1D Poisson Problem – Approximation

Step 3. We do two “tricks”

(a) we ignore the error term $\mathcal{O}(h^2)$
$\Rightarrow$ we only obtain an approximate solution $u_h(x)$

(b) We represent u_h by its discrete values

$$U_i := u_h(x_i), \qquad \text{for } i = 0, \ldots, N$$

(we of course hope/have the goal that $U_i \approx u(x_i)$)

Plugging this into the central differences from Step 2
we obtain at an internal point $U_i \approx u(x_i)$ that

$$-\partial^+\partial^- U_i = -\frac{U_{i+1} - 2U_i + U_{i+1}}{h^2} = f(x_i), \qquad \text{for } i = 1, \ldots, N.$$

Boundary. What about the boundary U_0, U_N?
Problem. $\partial^+\partial^- U_i$ is not well defined there.

The 1D Poission Problem – Linear System

Step 4. the $N-1$ equations from the last slide
with their $N+1$ unknowns U_i lead to the linear system

$$\frac{1}{h^2}\begin{pmatrix} -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 & -1 \end{pmatrix}\begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_N \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{N-1}) \end{pmatrix}$$

Question. (Uniquely) solvable?
Any vector $U_0 = U_1 = \ldots = U_N = c \in \mathbb{R}$ is in the kernel of the matrix.
$\Rightarrow$ solution not unique.

The 1D Poission Problem – Linear System

Step 4. the $N-1$ equations from the last slide
with their $N+1$ unknowns U_i lead to the linear system

$$\frac{1}{h^2}\begin{pmatrix} h^2 & & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & -1 & 2 & -1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -1 & 2 & -1 \\ & & & & & & h^2 \end{pmatrix}\begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ \vdots \\ U_N \end{pmatrix} = \begin{pmatrix} u_a \\ f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{N-1}) \\ u_b \end{pmatrix}$$

Question. (Uniquely) solvable? Yes.

Boundary conditions. We can easily include $U_0 = u_a$ and $U_N = u_b$.

To conclude. Knowing $f(x)$, u_a, u_b and N
$\Rightarrow$ setup matrix A and right hand side $\boldsymbol{F}$ and solve $A\boldsymbol{U} = \boldsymbol{F}$.

The 1D Poisson Problem – Numerical Example

Let‘s consider the right hand side

$$f(x) = (2\pi)^2 \sin(2\pi x), \qquad x \in [0, 1]$$

and $u_a = u(0) = u_b = u(1) = 0$ then we know the solution, since

Now we can directly see/derive the solution here (note again our focus is the numerics)

$$u(x) = \sin(2\pi x)$$

since $-u'' = f$.

Two Point Boundary Problems

Let‘s continue slightly more general: the numerical solution for

$$u'' + p(x)u' + q(x)u = r(x), \qquad a \le x \le b, \qquad u(a) = u_a, \quad u(b) = u_b,$$

with given functions $p(x), q(x)$ and given boundary values $u_a, u_b \in \mathbb{R}$.

Step 1. Discretize $[a, b]$: Choose some $N \in \mathbb{N}$, set $h = \frac{b-a}{N}$ and

$$x_i = a + ih, \qquad i = 0, \dots, N$$

Step 2. For each $i = 1, \dots, N-1$:
replace $u'(x_i)$ and $u''(x_i)$ by their approximations

$$\frac{u(x_i+h) - 2u(x_i) + u(x_i-h)}{h^2} + p(x_i)\frac{u(x_i+h) - u(x_i-h)}{2h} + q(x_i)u(x_i) + \mathcal{O}(h^2) = r(x_i)$$

where $\mathcal{O}(h^2)$ represents the common error of approximation.

Two Point Boundary Problems II

Step 3. We again do the two "tricks":

(a) ignore the error term

(b) replace the exact (unknown) solution $u(x)$ at the x_i by their numerical (still unknown) approximation $U_i \approx u(x_i)$

We obtain

$$\frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + p(x_i)\frac{U_{i+1} - U_{i-1}}{2h} + q(x_i)U_i = r(x_i), \qquad i = 1, \dots, N-1.$$

Boundary. At the boundary we have $U_0 = u_a$ and $U_N = u_b$, which we can (again) include, to obtain $N+1$ equations for $N+1$ unknowns.

Two Point Boundary Problems III – A Linear System

This time we multiply all equations by h^2 and obtain

$$A = \begin{pmatrix} 1 & 0 & & & & & \\ v_1 & d_1 & w_1 & & & & \\ & v_2 & d_2 & w_2 & & & \\ & & v_3 & \ddots & \ddots & & \\ & & & \ddots & \ddots & w_{N-2} & \\ & & & & v_{N-1} & d_{N-1} & w_{N-1} \\ & & & & & 0 & 1 \end{pmatrix} \quad \text{with} \quad \begin{aligned} v_i &= 1 - \frac{h}{2} p(x_i) \\ d_i &= -2 + h^2 q(x_i) \\ w_i &= 1 + \frac{h}{2} p(x_i) \end{aligned}$$

and

$$\boldsymbol{F} = \begin{pmatrix} u_a \\ h^2 r(x_1) \\ \vdots \\ h^2 r(x_{N-1}) \\ u_b \end{pmatrix}$$

A Two Point Boundary Problem Example

We consider the equation

$$u'' + 2u' - 3u = 9x, \qquad u(0) = u_a = 1, \quad u(1) = u_b = e^{-3} + 2e - 5 \approx 0.486351,$$

where we previously learned how to compute the exact solution

$$u(x) = e^{-3x} + 2e^x - 3x - 2$$

Choose N, set $h = 1/N$ and we get for $i = 1, \ldots, N$ that

$$\frac{u(x_i + h) - 2u(x_i) + u(x_i - h)}{h^2} + 2\frac{u(x_i + h) - u(x_i - h)}{2h} - 3u(x_i) + \mathcal{O}(h^2) = 9x_i.$$

Discretizing u (to U_i), multiplying by h^2 and including the boundary conditions, we get

$$\begin{aligned} U_0 &= 1, \\ (1-h)U_{i-1} + (-2 - 3h^2)U_i + (1+h)U_{i+1} &= 9x_i h^2, \qquad i = 1, \ldots, N-1, \\ U_N &\approx 0.486351. \end{aligned}$$

A Two Point Boundary Problem Example (cont.)

To be even more concrete, for $N = 4$, we get $h = 0.25$ and the linear system of equations becomes

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0.75 & -2.1875 & 1.25 & 0 & 0 \\ 0 & 0.75 & -2.1875 & 1.25 & 0 \\ 0 & 0 & 0.75 & -2.1875 & 1.25 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} U_0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} 1. \\ 0.140625 \\ 0.28125 \\ 0.421875 \\ 0.486351 \end{pmatrix}$$

We can include U_0, U_4 also directly to obtain

$$\begin{pmatrix} -2.1875 & 1.25 & 0 \\ 0.75 & -2.1875 & 1.25 \\ 0 & 0.75 & -2.1875 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} 0.140625 - 0.75 \cdot 1 \\ 0.28125 \\ 0.421875 - 1.25 \cdot 0.48635073 \end{pmatrix},$$

Numerically. $U_1 \approx 0.293176, \quad U_2 \approx 0.025557, \quad U_3 \approx 0.093820.$
exact. $u(0.25) \approx 0.290417, \quad u(0.5) \approx 0.020573, \quad u(0.75) \approx 0.089400$

(More about) 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 condition The solution is known at the boundary.
2. Neumann condition The derivative is known at the boundary.
3. Robin (or mixed) condition A combination of those.

Unti now. Numerical Methods with Dirichlet boundary conditions.

So how can we model Neumann conditions? (Mixed are then similarly)

Our Model BVP – now with Neumann boundary

We modify our BVP

$$u'' + p(x)u' + q(x)u = r(x), \qquad a \leq x \leq b, \qquad u'(a) = u'_a, \quad u(b) = u_b,$$

where u_b is as before and $u'_a \in \mathbb{R}$ is a value for the derivative of u at a.

Idea. Employ an approximation!
For example, as the simplest idea: forward difference

$$u'_a = u'(a) = \frac{u(x_1) - u(x_0)}{h} + \mathcal{O}(h) \qquad \text{resulting in} \qquad \frac{U_1 - U_0}{h} = u'_a.$$

Problem. Only a first order approximation.

That is, with all other approximations being central differences
(higher accuracy!)

$\Rightarrow$ We loose accuracy in our system.
A central difference using x_0, x_1, x_2 would
destroy our tridiagonal structure.

The Idea: Introduce a False Boundary

Idea. Assume we can extend u beyond the boundary $x = a$
$\Rightarrow$ at $x_{-1} = a - h$ we obtain a fictitious approximation $U_{-1} = u(x_{-1})$
$\Rightarrow$ we can introduce two new equations:

$$\frac{U_1 - 2U_0 + U_{-1}}{h^2} + p(x_0)\frac{U_1 - U_{-1}}{2h} + q(x_0)U_0 = r(x_0),$$
$$\frac{U_1 - U_{-1}}{2h} = u'_a.$$

Solving the second for U_{-1} yields

$$U_{-1} = U_1 - 2hu'_a.$$

We plug this into the first and obtain one equation

$$\frac{2U_1 - 2U_0 - 2hu'_a}{h^2} + p(x_0)u'_a + q(x_0)U_0 = r(x_0).$$

Note. This changes the first line, but we keep tridiagonality!

(finally, again!) The Heat Equation

Let's (re-)introduce the time dependency of our PDE and consider strategies to solve the PDE numerically. As a concrete example we consider the heat equation.

We are given the equation, well known from a few weeks ago

$$\frac{\partial}{\partial t}u(x,t) = \frac{\partial^2}{\partial x^2}u(x,t), \qquad 0 \leq x \leq 1$$

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

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

where we are looking for a solution for the time interval $[0, T]$.

Semi-Discretization

Idea. Combine the discretisation for BVP (in space) with techniques for solving ODEs (in time).

Step 1. Discretize the x-direction: Choose $M \in \mathbb{N}$ and let $h = \frac{1}{M}$ (last slide $a = 0$, $b = 1$) and define the grid points

$$x_i = ih, \qquad i = 0, \dots, M$$

Note. For each grid point x_i the function $u(x_i, t)$ is a function of time t alone.

Step 2. Fix some arbitrary $t \in [0, T]$ and discretize the right hand side of the PDE ($\frac{\partial^2}{\partial x^2} u$) with a central difference :

$$\frac{\partial u}{\partial t}(x_i, t) = \frac{u(x_{i+1}, t) - 2u(x_i, t) + u(x_{i-1}, t)}{h^2} + \mathcal{O}(h^2).$$

NTNU

Semi-Discretisation II

Step 3. Ignore $\mathcal{O}(h^2)$ amd introduce $U_i(t) \approx u(x_i, t)$.
Plugging this into our PDE, we get

$$U_i'(t) = \frac{U_{i+1}(t) - 2U_i(t) + U_{i-1}(t)}{h^2}, \qquad i = 1, 2, \dots, M-1,$$

where on the left hand side we have $U_i'(t) = \dfrac{\mathrm{d}}{\mathrm{d}t} U_i(t)$.

This time we plug in the boundary conditions (here already in their "new form")

$$U_0(t) = g_0(t) \quad \text{and } U_M(t) = g_1(t)$$

Into the first and last equation above. They read

$$U_1'(t) = \frac{U_2(t) - 2U_1(t) + U_0(t)}{h^2} = \frac{U_2(t) - 2U_1(t) + g_0(t)}{h^2}$$
$$U_{M-1}'(t) = \frac{U_M(t) - 2U_{M-1}(t) + U_{M-2}(t)}{h^2} = \frac{g_1(t) - 2U_{M-1}(t) + U_{M-2}(t)}{h^2}$$

Semi-Discretication III: A System of ODEs

We further have the Initial conditions $U_i(0) = f(x_i)$, $i = 0, \ldots, M$.

We obtain system of ordinary differential equations

$$
\begin{aligned}
U_i'(t) &= \frac{U_{i+1}(t) - 2U_i(t) + U_{i-1}(t)}{h^2}, \quad i = 1, 2, \ldots, M-1, \\
U_0(t) &= g_0(t), \\
U_M(t) &= g_1(t),
\end{aligned}
$$

which is called a **semi-discretisation (in space)** of the PDE.

This is also called the Method of Lines (MoL).

The System of ODEs in Matrix form

We can write the system shorter in matrix form

$$\boldsymbol{U}' = \frac{1}{h^2}\left(A\boldsymbol{U} + \boldsymbol{g}(t)\right),$$

where

$$\boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{M-1} \end{pmatrix} \in \mathbb{R}^{M-1}, \qquad A = \begin{pmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{M-1,M-1}$$

and $$\boldsymbol{g}(t) = \begin{pmatrix} g_0(t) \\ 0 \\ \vdots \\ 0 \\ g_1(t) \end{pmatrix} \in \mathbb{R}^{M-1}.$$

NTNU

Semi-Discretization IV - Solving the ODEs

Step 4. Solve the system of ODEs by the method of your preference (Numerical Methods for ODEs, lectures 13-16).

Example. We use Explicit Euler with a step size k:

$$U_i^{n+1} = U_i^n + r\left(U_{i+1}^n - 2U_i^n + U_{i-1}^n \right), \quad i = 1, 2, \ldots, M-1, \qquad \text{where } r := \frac{k}{h^2}.$$

Thus $U_i^n \approx u(x_i, t_n)$ with $t_n = nk$.

Note. We have space (i) and time (n) indices here.
$\Rightarrow$ we denote time indices by superscripts, space indices by subscripts.

A Numerical Example

Example 1. Solve the heat equation $\frac{\partial}{\partial t}u = \frac{\partial^2}{\partial x^2}u$ on the interval $0 < t < 1$ with the following initial and boundary conditions

$$u(x,0) = \sin(\pi x), \qquad \text{Initial vaconditionslue,}$$
$$g_0(t) = g_1(t) = 0. \qquad \text{Boundary conditios.}$$

Use step sizes $k = \frac{1}{N}$ and $h = \frac{1}{M}$ for $N \in \{20, 40, 80\}$, $M \in \{4, 8, 16\}$.

The analytic solution of this problem is given by

$$u(x,t) = e^{-\pi^2 t}\sin(\pi x).$$

Example 2. Perform the same example with initial condition

$$u(x,0) = \begin{cases} 2x & \text{if } 0 \leq x \leq 0.5, \\ 2(1-x), & \text{if } 0.5 < x \leq 1. \end{cases}$$

Stability Analysis

The semi-discretized system $U_0(t) = g_0(t)$, $U_M(t) = g_1(t)$ and

$$U_i'(t) = \frac{U_{i+1}(t) - 2U_i(t) + U_{i-1}(t)}{h^2}, \quad i = 1, 2, \ldots, M-1, \quad ,$$

is a linear ordinary differential equation:

$$\boldsymbol{U}' = \frac{1}{h^2}\big(A\boldsymbol{U} + \boldsymbol{g}(t)\big),$$

where

- $\boldsymbol{U} = \begin{pmatrix} U_1 & U_2 & \cdots & U_{M-1} \end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{M-1}$
- $A = \begin{pmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{M-1,M-1}$
- $\boldsymbol{g}(t) = \begin{pmatrix} g_0(t) & 0 & \cdots & 0 & g_1(t) \end{pmatrix}^{\mathrm{T}} \in \mathbb{R}^{M-1}$

NTNU

Stability Analysis and Eigenvalues

For our matrix $\frac{1}{h^2}A$ we already learned in the numerics for ODEs lectures that our step size k has to be chosen such that

$$|k\lambda_j + 1| \leq 1, \qquad j = 1, \dots, M-1.$$

Since A is symmetric all its eigenvalues are real and this reduces to the two inequalities

$$\pm(k\lambda_j + 1) \leq 1 \Leftrightarrow -2 \leq k\lambda_j \leq 0.$$

We can even derive the Eigenvalues of A. They are

$$\lambda_j = -4\sin^2\left(\frac{j\pi}{M}\right), \qquad j = 1, \dots, M-1,$$

such that the Eigenvalues $\dfrac{1}{h^2}A$ satisfy $-\dfrac{4}{h^2} < \lambda_j < 0$.

Courant-Friedrich-Lax (CFL) number

Summary. The numerical solution is stable if $k < -\frac{2}{\lambda_j}$ for all j, which means that we obtain the condition

$$r = \frac{k}{h^2} \leq \frac{1}{2}.$$

This also known as **Courant-Friedrich-Lax (CFL) number**, and the stability condition number above is also know as **(parabolic) CFL-condition** (since the heat equation is the prototype example of a so-called parabolic PDE.)

Implicit Methods

A remedy are methods that handle so-called stiff ODEs well, where the semi-discrete system is an example of.

These are the $A(0)$-stable methods like implicit Euler or the trapezoidal rule, which is also called Crank-Nicolson.

Both start with the discretized system

$$\boldsymbol{U}' = \frac{1}{h^2}\big(A\boldsymbol{U} + \boldsymbol{g}(t)\big),$$

Implicit Euler

The **implicit Euler method** is given by

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + r\,A\boldsymbol{U}^{n+1} + r\boldsymbol{g}(t_{n+1}), \qquad \text{with} \qquad r = \frac{k}{h^2}.$$

where $\boldsymbol{U}^n = \begin{pmatrix} U_1^n & U_2^n & \cdots & U_{M-1}^n \end{pmatrix}^{\mathrm{T}}$ and $U_i^n \approx u(x_i, t_n)$.

For each time step, we have to solve the system of linear equations

$$(I_{M-1} - r\,A)\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + r\,\boldsymbol{g}(t_{n+1}),$$

where I_{M-1} is the identity matrix of dimension $(M-1) \times (M-1)$.

Error Estimate. The error in the grid points is of order $\mathcal{O}(k + h^2)$.

Crank-Nicolson (trapezoidal rule)

The trapezoidal rule applied to the semi-discretized system is often referred to as the **Crank-Nicolson method**. The method is $A(0)$-stable and of order 2 in time $\Rightarrow$ better accuracy.

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \frac{k}{2h^2} A\left(\boldsymbol{U}^{n+1} + \boldsymbol{U}^n\right) + \frac{k}{2h^2}\left(\boldsymbol{g}(t_n) + \boldsymbol{g}(t_{n+1})\right).$$

For each time step, we have to solve the system of linear equations

$$(I_{M-1} - \frac{r}{2}A)\boldsymbol{U}^{n+1} = (I_{M-1} + \frac{r}{2}A)\boldsymbol{U}^n + \frac{r}{2}\left(\boldsymbol{g}(t_n) + \boldsymbol{g}(t_{n+1})\right), \qquad r = \frac{k}{h^2}.$$

Error Estimate. The error in the grid points is of order $\mathcal{O}(k^2 + h^2)$.

NTNU

Numerical Example (II)

Example 3. Solve the equation

$$u_t = u_{xx}, \qquad u(0,t) = e^{-\pi^2 t}, \quad u(1,t) = -e^{-\pi^2 t}, \quad u(x,0) = \cos(\pi x).$$

up to $t_{\text{end}} = 0.2$ by implicit Euler and Crank-Nicolson.

Plot the solution and the error. The exact solution is $u(x,t) = e^{-\pi^2 t}\cos(\pi x)$.

Use $N = M$, and $M = 10$ and $M = 100$ (for example).

Note. There are no stability issues, even for large values of r. Also notice the difference in accuracy for the two methods.