TMA4130
Mathematics 4N
Fall 2020

Norwegian University of Science and Technology
Department of Mathematical Sciences

Exercise set 13

This exercise should only be handed in, if you have (or expect to have) precisely 7 approved exercises amongst the first 12. The deadline is Sunday, November 15, 23:59. You won't get any detailed feedback about your solutions, and there won't be any guidance for this exercise sheet in the exercise classes, but we will publish solutions.

Exercises supposed to be done by hand are marked with an (H).
Exercises in which you are supposed to use/modify code in the Jupyter notebook are marked with a (J).

1 Assume that $f\colon \mathbb{R} \to \mathbb{R}$ is a smooth function, that $x_0 \in \mathbb{R}$, and h_1, $h_2 > 0$.

a) Find the interpolation polynomial through the function f in the points $x_0 - h_1$, x_0, $x_0 + h_2$, and use the result to find approximations of $f'(x_0)$ and $f''(x_0)$. (H)

b) Assume now that $h_2 = \vartheta h_1$ for some fixed $\vartheta > 0$. Determine the convergence order of the two approximations as $h_1 \to 0$. (H)

2 We consider the two-point boundary value problem

$$u_{xx} - (x+2)u_x + (x+2)u = 0 \qquad \text{for } 0 < x < 1,$$

with boundary conditions

$$u_x(0) = 1 \qquad \text{and} \qquad u(1) = e.$$

The analytic solution of this equation is the function

$$u(x) = xe^x.$$

a) Set up a finite difference scheme for this problem using central differences. For the left hand side boundary, use the idea of a false boundary and central differences. Use equidistant grid points $x_i = i\Delta x$ with a grid size $\Delta x = 1/N$. (H)

b) Set up and solve the system of equations for the case $N = 2$ in order to find approximations of the solution in the points $x = 0$ and $x = 1/2$. (H)

c) Write a python implementation of the method and test it for $N = 10, 20, 40, 80$. You may base your implementation on the code in the note `PDE.ipynb`. Note, however, that this code assumes the coefficients in the PDE to be constant.

Compute for each N the error

$$e(h) = \max_i |u(x_i) - U_i|.$$

Based on these results, what convergence order would you expect for this method? (J)

d) Assume that the boundary condition on the right hand side is replaced by

$$2u(1) - u'(1) = 0.$$

How does the finite difference scheme change in this case? (H)

$\boxed{3}$ We consider the time-dependent PDE

$$u_t = u_{xx} + u_x$$

with initial conditions

$$u(x,0) = x(1-x)^2 \qquad \text{for } 0 < x < 1$$

and boundary conditions

$$u(0,t) = \sin(t) \quad \text{and } u_x(1,t) = 0 \qquad \text{for } t > 0.$$

a) Perform a semi-discretisation of the PDE using central differences for the approximations of the x-derivatives. Use equidistant grid points $x_i = i\Delta x$ with a grid size $\Delta x = 1/M$. (H)

b) We now want to use the trapezoid rule for ODEs in order to compute a numerical solution of the system obtained in part **a)**. Set up the linear system that has to be solved in each step for an arbitrary time step $\Delta t > 0$.

Set up specifically the system for $M = 2$ and $\Delta t = 1/2$, and compute the numerical solution up to time $t_{\text{end}} = 1$. (H)

c) Write a python program that computes the numerical solution of this problem with the approach derived in parts **a)** and **b)**. Test your code with $M = 2$ and $\Delta t = 0.5$, and verify that you obtain the same solution as in part **b)**. You may base your program on the code in the note `PDE.ipynb` for the solution of the heat equation (Numerical example 3).

Solve the problem to $t_{\text{end}} = 1$ using $\Delta x = \Delta t = 0.05$. (J)