Technical University of Munich
School of Computation, Information and Technology
Dr. Michael Obersteiner
Dr. Daniel Lehmberg
Chinmay Datar
Hayden Liu Weng

WT 2022/2023

Lab Course
Scientific Computing

Worksheet 5

distributed: Thu., 19.01.2023
due: Sun., 29.01.2023, 23:59 (submission on the Moodle page)
oral examination: Tue., 31.01.2023 (exact time slots announced on the Moodle page)

In the last worksheet, we examined the stationary heat equation, that is the temperature T was depending on space only. Now, we generalize to the instationary heat equation

$$T_t = T_{xx} + T_{yy} \tag{1}$$

on the unit square $\Omega =]0,1[^2$ for $t \geq 0$ with the temperature $T(x,y,t)$, the two-dimensional coordinates x and y, the time t, homogeneous Dirichlet boundary conditions

$$T(x,y,t) = 0 \quad \forall (x,y) \in \partial\Omega, t \text{ in } [0;\infty[\tag{2}$$

and

$$T(x,y,0) = 1.0 \quad \forall (x,y) \in]0;1[^2 \tag{3}$$

as initial condition.
For the spatial discretization, we again use the grid points $\{(x_i, y_j) = (i \cdot h_x, j \cdot h_y), i = 0,1,\dots,N_x,N_x+1, j = 0,1,\dots,N_y,N_y+1\}$ and the finite difference approximation of the second derivatives

$$T_{xx}|_{i,j} \approx \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{h_x^2},$$

$$T_{yy}\big|_{i,j} \approx \frac{T_{i,j-1} - 2T_{i,j} + T_{i,j+1}}{h_y^2}$$

for $i = 1, \ldots, N_x, j = 1, \ldots, N_y$ with $h_x = \frac{1}{N_x+1}$ and $h_y = \frac{1}{N_y+1}$.

a) To have a rough insight into the exact solution of (1), (2), and (3), determine

$$\lim_{t\to\infty} T(x, y, t).$$

b) Implement an explicit Euler step for (1) and (2) as a function of the grid sizes N_x and N_y, the time step δt, and the computed solution at the current time.

c) **i)** Solve (1), (2), and (3) with the help of this Euler method for the values of δt, N_x, and N_y listed in the table below.

ii) Plot the solutions at times $t = \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}$ as surface plots. Organize your plots into four windows (one per time t). Each window should contain the 28 plots from the table below. Besides this, store all 112 plots as images (JPEG or PNG) with meaningful names.

Note: Do **not** send the picture files with your submission. Instead, the code should generate the files.

iii) Mark cases with stable solutions in the table.

$N_x = N_y$	$\delta t = \frac{1}{64}$	$\delta t = \frac{1}{128}$	$\delta t = \frac{1}{256}$	$\delta t = \frac{1}{512}$	$\delta t = \frac{1}{1024}$	$\delta t = \frac{1}{2048}$	$\delta t = \frac{1}{4096}$
3							
7							
15							
31							

Remark 1: If your simulation is taking too long (more than a couple of minutes on an average laptop) there might be something poorly implemented.

Remark 2: In terms of performance, consider whether solving each problem and plotting it take a similar amount of time.

d) Implement an implicit Euler step for (1) and (2) as a function of the grid sizes N_x and N_y, the time step δt, and the computed solution at the current time.

Hint: Use a modification of the Gauss-Seidel solver implemented in worksheet 4 for the solution of the system of linear equations in each timestep. Iterate until the norm of the residual is below 10^{-6}.

e) **i)** Solve (1), (2), and (3) with the help of the implicit Euler method and time step $\delta t = \frac{1}{64}$ for all spatial resolutions. Do you get stable solutions?

ii) Plot the solutions at times $t = \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}$.

Questions:

Q1 What relation between maximal time step size (to achieve a stable discretization) and spatial step $h_x = h_y$ can you derive from the tabular in **b)**?

Q2 Would you consider it reasonable to use a higher order explicit time discretization in **b)**? (e.g., Heun or RK4 instead of Explicit Euler)

Hint: Assume that all explicit discretizations have similar restrictions with respect to time step size as seen above for the explicit Euler method. Then think about the effort you have to make in order to get a stable solution where temporal and spatial accuracy are **balanced**. That is, such that neither error dominates over the other.

Q3 Is it reasonable to use an implicit Euler method in our example if we want to compute solutions with balanced accuracy in time and space for each N_x, N_y?

Q4 Would you consider it reasonable to use a higher order implicit time discretization (again from the point of view of balanced accuracy in time and space)?

Hint: Assume that all implicit discretizations have similar stability properties as the implicit Euler method.