Technische Universität Berlin
Institut für Mathematik
Dr. R. Kruse, M. Eisenmann, R. Weiske

WiSe 2017/18

Numerical Mathematics for Engineering II

Assignment 3

Theoretical exercises: submit to exercise class on November 15th or 16th, 2017.
Programming exercises: upload to ISIS until Friday 18:00, November 17th, 2017.

1. Exercise: Difference stencils and Taylor expansions **4 points**

Let $u\colon [0,1] \to \mathbb{R}$ be a sufficiently smooth mapping.

a) Consider the difference quotients $D^0u(x)$, $D^+u(x)$, $D^-u(x)$, $D^+D^-u(x)$ with a given step size $h \in (0, \frac{1}{2})$. For each quotient determine all $x \in [0,1]$ such that the quotient is defined.

b) Show that the equality $D^0u(x) = \frac{1}{2}\big(D^+u(x) + D^-u(x)\big)$ holds for all $x \in [0,1]$, for which both sides of the equation are defined.

c) Show that the equality $D^+D^-u(x) = D^-D^+u(x)$ holds for all $x \in [0,1]$, for which both sides of the equation are defined.

d) Apply the Taylor theorem to show that

$$D^0u(x) = u'(x) + h^2 R_0,$$

with $|R_0| \leq \frac{1}{6} \max_{\xi \in [x-h,x+h]} |u'''(\xi)|$, and

$$D^+D^-u(x) = u''(x) + h^2 R_1,$$

with $|R_1| \leq \frac{1}{12} \max_{\xi \in [x-h,x+h]} |u^{(4)}(\xi)|$. Explain why $D^-D^0u(x)$ is not a suitable approximation of the second derivative.

2. Exercise: Difference formulas for non-uniform grids **3 points**

Consider a domain $\bar{\Omega} = [0,1]$ and a grid $\bar{\Omega}_h = \{x_0, \dots, x_{N+1}\}$, where the grid points satisfy $0 = x_0 < x_1 < \cdots < x_{N+1} = 1$ with step sizes $h_i := x_i - x_{i-1}$ and maximal step size $h := \max_{i \in \{1,\dots,N+1\}} h_i$. If $h_i = h_j$ for all $i, j \in \{1, \dots, N+1\}$ we say that $\bar{\Omega}_h$

Please turn the page!

is a *uniform grid*, otherwise a *non-uniform grid*. For $i \in \{1, \ldots, N\}$ and $u \in C^4(\bar{\Omega})$ we define the adapted differences

$$
\begin{aligned}
(D^- u)_i &:= \frac{u(x_i) - u(x_{i-1})}{x_i - x_{i-1}} = \frac{u(x_i) - u(x_{i-1})}{h_i}, \\
(D^+ u)_i &:= \frac{u(x_{i+1}) - u(x_i)}{x_{i+1} - x_i} = \frac{u(x_{i+1}) - u(x_i)}{h_{i+1}}, \\
(D^0 u)_i &:= \frac{u(x_{i+1}) - u(x_{i-1})}{x_{i+1} - x_{i-1}} = \frac{u(x_{i+1}) - u(x_{i-1})}{h_i + h_{i+1}}.
\end{aligned}
$$

a) Determine $\alpha_i, \beta_i, \gamma_i, R_i \in \mathbb{R}$ depending on h_i, h_{i+1} such that

$$u''(x_i) = \alpha_i u(x_{i-1}) + \beta_i u(x_i) + \gamma_i u(x_{i+1})) + R_i,$$

with $\lim_{h \to 0} R_i = 0$. Hint: Use $u'' = au'' + bu''$ with $a + b = 1$ and use suitable Taylor expansions to express the two terms. Choose a and b properly.

b) Determine the order of the remainder R_i for general h_i, h_{i+1}, i.e., find the largest $p > 0$ such that $|R_i| = \mathcal{O}(h^p)$. Is $u \in C^4(\bar{\Omega})$ necessary to obtain this rate?

c) Under which special conditions on h_i, h_{i+1} does the order of R_i improve? Is it necessary that $u \in C^4(\bar{\Omega})$ to obtain the improved rate?

3. Programming exercise: 1D wave equation **6 points**

Consider the wave equation on the real line:

$$
\begin{cases}
u_{tt} - u_{xx} = 0, & t \in (0, \infty),\ x \in \mathbb{R}, \\
u(0, x) = \varphi(x), & x \in \mathbb{R}, \\
u_t(0, x) = \psi(x), & x \in \mathbb{R},
\end{cases}
$$

where $\varphi(x) = \frac{a}{(x-4)^2+1} + \frac{a}{(x+4)^2+1}$ and $\psi = bx \exp(-\frac{1}{2}x^2)$.

a) Write a function `u = a03e03WaveEq(t,x,a,b)` which computes the exact solution for given $t \in (0, \infty)$, $x, a, b \in \mathbb{R}$. Hint: D'Alembert's formula.

b) Write a function `a03e03Surf()` which generates three surface plots of the exact solution u with $t \in [0, 10]$ and $x \in [-15, 15]$. For the plots use the following three sets of parameter values $(a, b) \in \{(0, 1), (1, 0), (1, 1)\}$. Hint: The functions `numpy.meshgrid` and `plot_surface` (see `mplot3d`) might be useful.

c) Learn how to create an animation in Python by studying `FuncAnimation` provided by `matplotlib.animation`. Write a function `a03e03Mov(a,b)` which generates for given parameter values $a, b \in \mathbb{R}$ an animation of the exact solution. Each frame should display a line plot of the exact solution for a fixed t and for $x \in [-15, 15]$. Your animation should show the evolution of the exact solution over $t \in [0, 10]$ with at least 100 frames.

See next page!

4. Programming exercise: 1D FDM **6 points**

Consider the following boundary value problem (BVP):

$$\begin{cases} -u''(x) - 4u'(x) + u(x) = f(x), & \text{for all } x \in \Omega = (0,1), \\ u(0) = 1, \ u(1) = 3. \end{cases}$$

The exact solution is given by $u(x) = 1+3x^2-x^3$. For some integer $p > 1$, discretize this boundary value problem with finite differences on $\bar{\Omega}_h = \{ih \in \mathbb{R} : i = 0, \dots, N+1\}$, mesh size $h = \frac{1}{N+1}$, and $N = 2^p - 1$. Use the standard scheme

$$\begin{aligned} u_h(0) = 1, u_h(1) = 3,& \\ (-D^-D^+ - 4D^0 + I)u_h(ih) = f(ih),& \qquad i = 1, \dots, N, \end{aligned}$$

so that you get a discrete equation $L_h u_h = f_h$ with $L_h \in \mathbb{R}^{N\times N}$.

a) Determine the right hand side of the BVP analytically.

b) Write a function `[xh,Lh,fh] = a03e04getBVP(p)` that sets up the grid `xh`, the sparse matrix `Lh`, and the right hand side `fh` of the corresponding linear system for the refinement level `n=2^p-1`. Hint: Useful functions are `numpy.linspace`, `scipy.sparse.diags`, etc.

c) Write a function `error = a03e04solve()` that solves the discretized problem for every $p \in \{2, \dots, 15\}$ and returns an array `error`. For each p determine the error between the approximation and the restricted exact solution in the maximum norm, i.e. `error(p)` $= \max_i |u_h(ih) - u(ih)|$. Plot the errors versus the grid size using `pyplot.loglog(h,error)`. How fast does `error(p)`$\to 0$ as $h \to 0$?

total sum: 19 points