Technische Universität Berlin
Institut für Mathematik
Dr. D. Peschka, A. Selahi

WS 2019/20

Numerical Mathematics II for Engineers

Homework assignment 3 : *Rise of the stencil.*

Programming assignments can be solved in MATLAB/Python/Julia.
Use sparse matrices where appropriate!

Deadline : submit before/during the lecture on November 11, 2019.

1. Exercise : Implementation of higher-order difference stencils **7 points**

For given $m \in \mathbb{N}_0$, let $\bar{\Omega} = [0, 1]$ and $\bar{\Omega}_h = \{0, h, 2h, \ldots, (N+1)h\}$ with $h = 1/(N+1)$ the corresponding grid. Consider $x_n = nh \in \bar{\Omega}_h$ with sufficiently many neighbors to either side, i.e., for any j with $n - m \le j \le n + m$ we require $x_j = jh \in \bar{\Omega}_h$. We abbreviate $u_n = u(x_n)$ with $u : \bar{\Omega} \to \mathbb{R}$ being r-times continously differentiable.

a) Perform Taylor expansions for u_j for $n - m \le j \le n + m$ in terms of $u(x_n)$ and derivatives $u^{(k)}(x_n)$ for $k \ge 0$. Expand and, by neglecting the remainders, invert this relationship to find symmetric difference stencils for arbitrary derivatives and arbitrary desired consistency level. What are k, r and the remainders R?

b) Based on a) write a `function S=a03ex01getstencil(m)` which returns the $2m+1$ stencils for $u^{(k)}(x_n)$ for $k = 0 \ldots 2m$ and compare the symmetric stencils here :

`https://en.wikipedia.org/wiki/Finite_difference_coefficient`

c) Write a `function [xh,Lh]=a03ex01getlaplace(m,N)` which returns the reduced sparse matrix for the Laplace operator $L_h \in \mathbb{R}^{(N+1)\times(N+1)}$ based on the stencil `S` from b) with periodic boundary conditions and the corresponding mesh $x_h = (x_0, .., x_N)$ for given $N, m \in \mathbb{N}$.

Hint : Assume $u : \mathbb{R} \to \mathbb{R}$ is 1-periodic $u(x + 1) = u(x)$. Discretize u on $[0, 1)$ or u_h on $\{0, h, 2h, \ldots, Nh\}$ with $h = 1/(N + 1)$ and identify $u_{N+1+i} = u_i$ for $i \in \mathbb{Z}$.

d) For $u(x) = \sin(2\pi x)$ plot the error $\max_j |(L_h u_h)(x_j) - u''(x_j)|$ as a function of h doubly logarithmically for $m = 1, .., 4$ and $N = 2^p - 1$ for $p = 2, .., 15$ and discuss.

Please turn the page!

2. Exercise : Difference formulas for non-uniform grids **7 points**

Consider a domain $\Omega = (0,1)$, its closure $\overline{\Omega} = [0,1]$, and a grid $\overline{\Omega}_h = \{x_0, \ldots, x_{N+1}\}$, 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 $h := \max_{i \in \{1,\ldots,N+1\}} h_i$. If $h_i = h_j$ for all $i, j \in \{1, \ldots, N+1\}$ we say that $\overline{\Omega}_h$ is a *uniform grid*, otherwise a *non-uniform grid*. For $i \in \{1, \ldots, N\}$ and $u \in C^4(\overline{\Omega})$ 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 $u \in C^4(\overline{\Omega})$ 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 $u \in C^4(\overline{\Omega})$ to obtain the improved rate ?

3. Programming exercise : 1D FDM **7 points**

Consider the following boundary value problem (BVP) :

$$\begin{cases} -u''(x) - 4u'(x) + u(x) = f(x), & \text{for all } x \in \Omega = (0,1), \\ u(0) = 1, \ u(1) = 2. \end{cases}$$

The exact solution is given by $u(x) = 1 + 4x^2 - 3x^3$. Discretize this BVP with finite differences on $\overline{\Omega}_h = \{ih \in \mathbb{R} : i = 0, \ldots, N+1\}$ with mesh size $h = \frac{1}{N+1}$, where $N = 2^p - 1$ for some integer $p > 1$. Use the standard scheme

$$\begin{aligned}
&u_h(0) = 1, u_h(1) = 2, \\
&(-D^- D^+ - 4D^0 + I) u_h(ih) = f(ih), \qquad i = 1, \ldots, N,
\end{aligned}$$

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.

See next page !

b) Write a function `[xh,Lh,fh] = a03ex03getBVP(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 MATLAB commands are `speye`, `spdiags`, `linspace`, etc. (useful Python functions are `numpy.linspace`, `scipy.sparse.diags`, etc.).

c) Write a function `error = a03ex03solve()` that solves the discretized problem for $p \in \{1, \dots, 15\}$. 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 `loglog(h,error)` (in Python `matplotlib.pyplot.loglog(h,error)`). How fast does `error(p)`$\to 0$ as $h \to 0$?

total sum : 21 points