Technische Universität Berlin
Fakultät II – Institut f. Mathematik
Instructor: Peschka
Tutor: Marschall

WS 2013/2014
Handed out: 29.10.
Return during lecture: 6.11.

2. Assignment „Numerische Mathematik für Ingenieure II“

http://www.moses.tu-berlin.de/Mathematik/

Finite differences on $(0, L)^n$

---

1. Exercise: A classical solution for the Poisson equation on a square **6 points**

Consider the following boundary value problem

$$(1) \qquad \begin{aligned} -\Delta u &= 0 && \text{in} \quad \Omega = (0,\pi) \times (0,\pi) \subset \mathbb{R}^2, \\ u &= (\pi - x)x && \text{on} \quad \Gamma = \partial\Omega, \end{aligned}$$

and suppose that after separation of variables a solution can be written as a Fourier series

$$u^f(x,y) = a_0(y) + \sum_{n\in\mathbb{N}} a_n(y)\cos(nx) + b_n(y)\sin(nx).$$

(a) Use the boundary conditions to determine $a_n(0), b_n(0), a_n(\pi), b_n(\pi)$ and explain which series of $g = x(\pi - x)$ one should use

$$g = \tfrac{\pi^2}{6} - (\cos 2x + 2^{-2}\cos 4x + 3^{-2}\cos 6x + ..) \quad \text{or} \quad g = \tfrac{8}{\pi}(\sin x + 3^{-3}\sin 3x + 5^{-3}\sin 5x + ..)?$$

(b) Write down the equations for the y dependence of a_n, b_n and solve them.

(c) We want to check the regularity of u. Compare $u_{xx}(x,0)$ with $g_{xx}(x,0)$ along $0 \le x \le \pi$ using the truncated Fourier series

$$u^{f,\text{trunc}}(x,y) = a_0(y) + \sum_{n=1}^{m} a_k(y)\cos(nx) + b_k(y)\sin(nx)$$

by plotting both for increasing m. What do you observe?

(d) How can you rewrite the problem (1) into a problem of the form

$$\begin{aligned} -\Delta\bar{u} &= f && \text{in} \quad \Omega = (0,\pi)^2 \subset \mathbb{R}^2, \\ \bar{u} &= 0 && \text{on} \quad \Gamma = \partial\Omega, \end{aligned}$$

and how are $\bar{u}$ and u related? In which class of functions $C^k(\bar{\Omega})$ is f?

2. Exercise: Difference stencils **2 points**

Let $u : [0,1] \to \mathbb{R}$. Show the following properties:

(a) $D^0u(x) = \frac{1}{2}\big(D^+u(x) + D^-u(x)\big)$,

(b) $D^+D^-u(x) = D^-D^+u(x)$.

For which x are D^0, D^+, D^-, D^+D^- defined?

3. Exercise: Taylor expansions **3 points**

Let $I \in \mathbb{R}$ an open interval and let $x \in I$ and $h > 0$ with $x \pm h \in \bar{I}$. Show that

(a) $D^0u(x) = u'(x) + h^2R_0$, with $|R_0| \le \frac{1}{6}\max_{\xi\in[x-h,x+h]}|u'''(\xi)|$, if $u \in C^3(\bar{I})$,

(b) $D^+D^-u(x) = u''(x) + h^2R_1$, with $|R_1| \le \frac{1}{12}\max_{\xi\in[x-h,x+h]}|u^{(4)}(\xi)|$, if $u \in C^4(\bar{I})$,

(c) Try to derive a formula for $D^-D^0u(x)$ similar to the one in (b) and explain on the basis of your computation why this difference scheme is unsuitable for the approximation of the second derivative.

4. Programming exercise: First 1D finite differences **6 points**

Consider the following boundary value problem:

$$\begin{aligned} -u''(x) - 3u'(x) + u(x) &= -1 + 10x^2 - x^3, && \forall\, x \in (0,1), \\ u(0) &= u(1) = 1. \end{aligned}$$

The exact solution is $u(x) = 1 + x^2 - x^3$. For a given p discretize this PDE with finite differences with $\bar{\Omega}^h = \{ih \in \mathbb{R} : i = 0, .., N\}$, grid size $h = 1/N$, and $N = 2^p - 1$, $p \geq 1$. Use the standard scheme

$$\begin{aligned} u_i^h &= 1, && i = 0, N \\ \sum_{j=0}^{N} (-D^- D^+ - 3D^0 + I)_{ij}\, u_j^h &= f_i^h, && i = 1, ..., N-1 \end{aligned}$$

so that you get a discrete equation $A^h u^h = f^h$.

(a) Write a `function [xh,Ah,fh] = a02ex04getPDE(p)` that sets up the grid `xh`, the sparse matrix `Ah` and right hand side `fh` of the corresponding linear system for the refinement level `N=2^p-1`. Useful commands are `speye, sparse, linspace` etc.

(b) Write a `function errors = a02ex04solve()` that solves the discretized problem for `p` $\in \{1, \ldots, 15\}$. For each `p` determine the error between approximation and 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,errors)`. How fast does `errors`$\to 0$ as $h \to 0$?

(c) Write a script `a02ex04reduce.m` that transforms $A^h \in \mathbb{R}^{N+1 \times N+1}$ into the reduced form $\tilde{A}^h \in \mathbb{R}^{N-1 \times N-1}$, where boundary conditions are eliminated from the solution and enter the modified $\tilde{f}^h$. Compare the previously computed solution $u^h \in U^h$ with the reduced one $v^h \in V^h$ of $\tilde{A}^h v^h = \tilde{f}^h$ for $p = 7$. The spaces U^h and V^h are as in the lecture.

5. Programming exercise: Now 2D finite differences **6 points**

Consider the following boundary value problem:

$$\begin{aligned} -\Delta u_i &= f_i && \text{in} \quad \Omega = (0,1)^2 \subset \mathbb{R}^2, \\ u_i &= 0 && \text{on} \quad \Gamma = \partial\Omega \end{aligned}$$

with solutions

$$\begin{aligned} u_1(x,y) &= (xy - xy^2 - x^2 y + x^2 y^2), \\ u_2(x,y) &= \sin(\pi x)\sin(2\pi y). \end{aligned}$$

(a) Devise a $f_i(x,y)$ so that $u_i(x,y)$ for $i = 1, 2$ is the solution of the problem.

(b) Write a `function [xh,yh,Ah,fh] = a02ex05getPDE5(L,p,i)` that sets up the sparse matrix `Ah` of the linear system for the refinement level `p` on the domain $[0, L]^2$. Use the standard five-point stencil on a uniform mesh with lexicographical order

$$x_{1+i+j(N+1)} = (ih, jh)^T \in \mathbb{R}^2.$$

where $0 \leq i, j \leq N$ and $h = L/N$.

(c) Write a `function errors = a02ex05solve(i)` that solves the discretized problem for f_1 for $i = 1$,f_2 for $i = 2$ from (a) for `p` $\in \{1, \ldots, 9\}$. Determine for each `p` the error between the computed approximation and the restricted exact solution in the maximum norm and store it in `errors(p)`. Finally plot the errors as before with `loglog` and determine *experimental order of convergence* as before.

(d) (2 extra points): Compute discrete solution for the problem of exercise 1(d) and the corresponding errors using the truncated solution $u^{f,\text{trunc}}$ for sufficiently large m.

Note: Use sparse matrices! The command `meshgrid` might be useful to create `xh,yh`, but check for the lexicographical order. To convert an $\mathbb{R}^{N \times N}$ matrix "`A`" into a $\mathbb{R}^{N^2}$ vector "`a`" use the command `a=reshape(A,N*N)` or simply the command `a=A(:)`. The reverse operation is `A=reshape(A,[N N])`. If you are working with the reduced system, then you might need to set `error(1)=0` because $u^h(0,0) = u^h(1,0) = u^h(0,1) = u^h(1,1) = 0$.

total sum: 23 points