TECHNISCHE UNIVERSITÄT BERLIN
INSTITUT FÜR MATHEMATIK
Dr. D. Peschka, A. Selahi

WS 2019/20

Numerical Mathematics II for Engineers

Homework assignment 4 : *Solve elliptic PDEs.*

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

Deadline : submit before/during the lecture on November 18, 2019.

1. Exercise : The standard 5–point difference stencil **7 points**

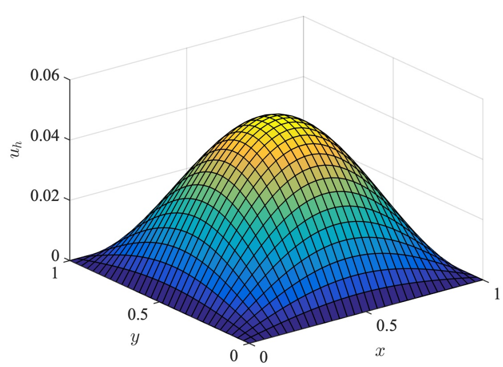
Consider the Laplace operator $Lu = -\Delta u$ in $\Omega = (0, L)^2$ and $u \in C^4(\bar{\Omega})$.

a) Show the 5–point stencil is a 2nd-order consistent approximation of Δu.

b) Download the the code `a04_hint_mesh`, which creates $N{+}2 \times N{+}2$ arrays `xh,yh` (float) and `ix` (integer) with vertices `xh(i,j)=(i-1)h` and `yh(i,j)=(j-1)h` where $h = 1/(N+1)$. The lexicographical indexing is `ix(i,j)=i+(j-1)*(N+2)`. Check that `ixy,ixmy,ixpy,ixym,ixyp` are $\mathbb{N}^{N\times N}$ index arrays of interior points and their respective neighbors in the directions $(-1,0),(+1,0),(0,-1),(0,+1)$ and that `ix_BD` contains boundary indices. Verify that the indices `ii,jj` can be used to build the 5-point stencil including equations for the boundary.

c) Convert `a04_hint_mesh` into a function `[Lh,xh,yh,ix_BD]=a04ex01_Lh5(L,N)`, which for given domain size L returns `xh,yh,ix_BD` and constructs the discrete Laplace operator L_h using the 5-point stencil with $h = L/(N+1)$ on $\bar{\Omega}_h = \{(ih, jh) : 0 \le i, j \le N+1\}$. The only missing piece from b) is the array `aa`!

d) Use `a04ex01_Lh5` to solve $L_h u_h = f$ with $f(x, y) = \sin(\pi x)\sin(\pi y)$ (eigenfunction) and homogeneous Dirichlet boundary conditions. Plot u_h and compare with the exact $u = f/\lambda$ for $N{+}2 = 512$ by computing $\max_{x\in\Omega_h} |u_h(x) - u(x)|$.



Please turn the page!

2. Exercise : Two-point boundary value problem (BVP) **8 points**

Implement a finite difference scheme for the problem

$$\begin{aligned} & -au''(x) + bu'(x) + cu(x) = f(x), \quad x \in (0,1), \\ & u(0) = \alpha, \quad u(1) = \beta, \end{aligned} \tag{1}$$

for a given (possibly non-uniform) mesh $\overline{\Omega}_h$, with constants a, b, c, α, $\beta \in \mathbb{R}$.

a) Consider the reduced linear system $L_h u_h = f_h$ with $L_h \in \mathbb{R}^{N\times N}$ and $N = |\overline{\Omega}_h| - 2$ to problem (1). The matrix L_h is written in compact form as

$$L_h = -a\Big(\tfrac{2}{h_i(h_i+h_{i+1})}, -\tfrac{2}{h_i h_{i+1}}, \tfrac{2}{h_{i+1}(h_i+h_{i+1})}\Big) + b\Big(\tfrac{-1}{h_i+h_{i+1}}, 0, \tfrac{1}{h_i+h_{i+1}}\Big) + c(0,1,0),$$

for $i \in \{1, \ldots, N\}$, where the same notation as in Exercise 2 of Homework 3 is used. Determine the right hand side $f_h \in \mathbb{R}^N$. How does L_h look like if we use the forward or backward differences D^+, D^- instead of D^0 for the approximation of the first order derivative?

b) Write a function `[Lh,fh] = a04ex02getPDE(xh,f,consts,flag)` that determines L_h and f_h, where `xh` is the vector of grid points, the model parameters are contained in `consts=[a b c alpha beta]`, and `f` is a function handle for the right-hand side of (1). The character `flag` selects the approximation for u' with `'-'`, `'+'`, `'0'` for D^-, D^+, D^0, respectively.

3. Exercise : Singularly perturbed BVP (optional) **6 points**

Consider the two-point boundary value problem

$$\begin{cases} -\varepsilon u''(x) + u'(x) = 1, & x \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{2}$$

with small but positive values for ε. Simply setting $\varepsilon = 0$ does not help, because the solution (if it exists) is not close to the one for small values for $\varepsilon > 0$. It also changes the order of the BVP. This behaviour is typical for *singularly perturbed problems.*

The exact solution to (2) is given by

$$u_\varepsilon(x) = x - \frac{\mathrm{e}^{-(1-x)/\varepsilon} - \mathrm{e}^{-1/\varepsilon}}{1 - \mathrm{e}^{-1/\varepsilon}}, \quad x \in (0,1)$$

and shown in the figure below. As we see $u_0(x) = \lim_{\varepsilon\to 0} u_\varepsilon(x) = x$ for $0 \le x < 1$, but $u_0(x)$ does not satisfy the boundary condition at $x = 1$ in a smooth way. Instead, the solution exhibits a thin region near $x = 1$ (also known as boundary layer), where u_ε changes rapidly. The width of the region depends on ε and thereby derivatives of u_ε become large as $x \to 1$ and $\varepsilon \to 0$. That means that the constant $u^{(k)}(\xi)h^{k-2}$ $(k > 2)$ in the remainder of the difference quotient for u'' is large. In order to make the remainder small, we need to make h very small.

See next page!

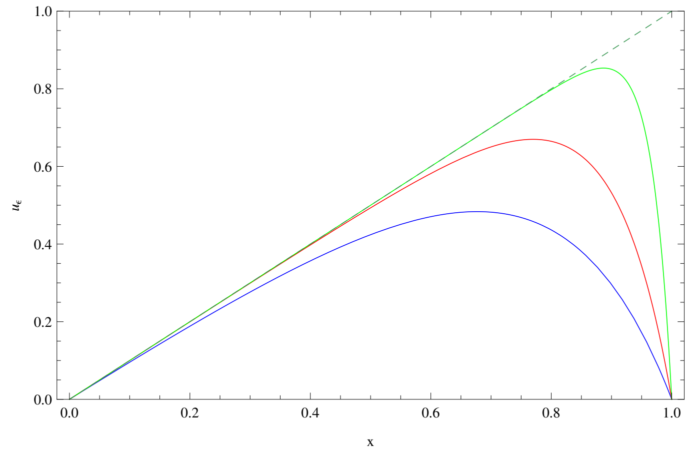


Fig. : u_ε for $\varepsilon \in \{1/5,\ 1/10,\ 1/30\}$ (blue, red, green) and the function $u_0(x) = x$ (dashed)

a) Write a function `uh=a04ex03solve(eps,xh,flag)` which returns the FDM solution of the singularly perturbed problem (2) for given `eps` $= \varepsilon$, grid `xh`, and approximation for the first derivative selected by `flag`. Hint : Use the solution of Exercise 2.

b) Write a function `[err,uex] = a04ex03error(eps,xh,uh)` that returns the error `err` (measured by the norm $\|\cdot\|_\infty$) between `uh` and the restricted exact solution, which is also to be returned as `uex`.

c) Write a function `xh = a04ex03shishkin(N,sigma)` that generates a column vector of size 2N+1 describing a "Shishkin" grid `xh` that is defined by

$$\texttt{xh(i)} = \begin{cases} \texttt{(i-1)*H}, & \text{for} \quad \texttt{i = 1, ..., N}, \\ \texttt{(1-sigma) + (i-N-1)*h}, & \text{for} \quad \texttt{i = N+1, ..., 2N+1}, \end{cases}$$

where `H = (1-sigma)/N` and `h = sigma/N`.

d) Write a script `a04ex03experiment` that plots the exact and approximated solution for `eps` $= 0.001$ and all $N \in \{5,\ 50,\ 500,\ 5000\}$ on a

i) uniform grid with $h = \frac{1}{2N}$ and forward difference operator D^+ (`flag='+'`),

ii) uniform grid with $h = \frac{1}{2N}$ and central difference operator D^0 (`flag='0'`).

iii) uniform grid with $h = \frac{1}{2N}$ and backward difference operator D^- (`flag='-'`).

iv) non-uniform Shishkin grid with `N` and `sigma=4*eps*log(2*N)` (`log` is natural logarithm here) and central difference operator D^0 (`flag='0'`).

Create four different figures for i), ii), iii), and iv). Results with different N but same operator should be plotted into the same figure. Use the figure titles to add information about the used operator, the grid, and the error for different N.

total sum : 15 (+6) points