Technische Universität Chemnitz
Dr. M. Winkler

Chemnitz, 12th April 2019
hand in on 24th April 2019

Numerical Methods for Partial Differential Equations

Sheet 3

Exercise 6: A 4th order finite-difference method

When using additional points in a difference stencil, one can achieve a higher approximation order. Show that the **9-point stencil**

$$\frac{1}{6h^2}\begin{bmatrix} -1 & -4 & -1 \\ -4 & 20 & -4 \\ -1 & -4 & -1 \end{bmatrix},$$

i. e. the scheme

$$[-\Delta_h^{(9)} u](x) = \frac{1}{6h^2}\big[20u(x) - 4u(x_1 \pm h, x_2) - 4u(x_1, x_2 \pm h) - u(x_1 \pm h, x_2 \pm h)\big],$$

in combination with

$$f_h^{(5)}(x) = f(x) + \frac{h^2}{12}[\Delta_h^{(5)} f](x),$$

possesses the consistency order 4 w. t. t. the L^∞-Norm, provided that $u \in C^{5,1}(\overline{\Omega})$. Moreover, show that the order is only 2 if $f_h^{(1)}(x) = f(x)$ is used.

Homework 3: Experimental computation of the convergence rate

Frequently, there is a relation between the error $E_h := \|e_h\|$ and the discretization parameter h of the form

$$E_h \leq C\, h^p.$$

To compute the predicted convergence order p experimentally, one can use the ansatz

$$E_h \approx C\, h^p,$$

where C and p are unknown.

(a) Assume that the error functionals E_{h_1} and E_{h_2} have been computed for two different grids with grid sizes h_1 and h_2. How can one estimate the convergence rate p?

(b) The exponent p can be interpreted as the slope of a linear function which interpolates the points (h_1, E_{h_1}) and (h_2, E_{h_2}) in a plot with logarithmic axes. If multiple computations were done, i. e. data points $(h_1, E_{h_1}), \ldots, (h_n, E_{h_n})$ $(n \geq 2)$ are given, one can generalize the previous idea. Now, we can seek C and p of a function $h \mapsto C\, h^p$ which approximates the data points in a least-square sense. Derive the corresponding normal equations.

(c) Implement the solution of (b) in MATLAB. The function header should be:

```
function [p,q] = eoc(h, e)
```

The vectors `h` and `e` contain the grid sizes and the computed errors. The return values `p` and `q` are the least-squares approximations of p and $\ln(C)$.

(d) Estimate the constant C and the convergence order p for the following measurements:

h	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
E_h	6.91e-01	1.78e-01	4.33e-02	1.78e-02	3.07e-03

Draw the measurements and the function $C\, h^p$ into a diagram with logarithmic axes (MATLAB: `loglog`).

Homework 4: Implementation of different norms

For functions $v_h, w_h \in V_h$, $u_h \in U_h$ we define

- the **discrete L^∞-norm** by

$$\|v_h\|_{L^\infty,h} = \max_{x \in \Omega_h} |v_h(x)|,$$

- the **discrete L^2-scalar product** by

$$(v_h, w_h)_h = h^d \sum_{x \in \Omega_h} v_h(x)\, w_h(x)$$

  and the corresponding **discrete L^2-norm** by

$$\|v_h\|^2_{L^2,h} = h^d \sum_{x \in \Omega_h} |v_h(x)|^2,$$

- the **discrete H^1-seminorm** by

$$|u_h|^2_{H^1,h} = h^d \sum_{j=1}^{d} \sum_{\substack{x \in \overline{\Omega}_h \\ x + h e_j \in \overline{\Omega}_h}} \left|[D_j^+ u_h](x)\right|^2.$$

Here, Ω_h denotes the interior grid points and h the grid size. Implement methods realizing these expressions in MATLAB for the case that Ω is the 2D unit square (cf. § 3). The function for the H^1-seminorm requires a vector of the form

$$\vec{u}_h = (\underbrace{u_{0,0}, u_{1,0}, \ldots, u_{n,0}}_{0.\text{ row in }\bar{\Omega}_h}, \underbrace{u_{0,1}, u_{1,1}, \ldots, u_{n,1}}_{1.\text{ row in }\bar{\Omega}_h}, \ldots, \underbrace{u_{0,n}, u_{1,n}, \ldots, u_{n,n}}_{n.\text{ row in }\bar{\Omega}_h})^\top.$$

The remaining functions are applied to vectors of the form

$$\vec{v}_h = (v_{1,1}, \ldots, v_{n-1,1}, \ldots, v_{1,n-1}, \ldots, v_{n-1,n-1})^\top.$$

Implement also functions `Restrict_to_Inner`, which truncate the boundary components of $\vec{u}_h$ and `Extend_to_Boundary`, which fills the missing boundary points by 0. The function headers should read:

```
function norm = Linfty_Norm_Unitsquare(v)
function ip   = L2_InnerProduct_Unitsquare(v,w)
function norm = L2_Norm_Unitsquare(v)
function norm = H1_Seminorm_Unitsquare(u)
function v = Restrict_to_Inner(u)
function u = Extend_to_Boundary(v)
```

Extend your program from Sheet 2, Exercise 4 and meeasure the consistency order in the L^2-norm and the convergence order in the H^1-norm.