Fachbereich Mathematik & Informatik
Freie Universität Berlin
Prof. Dr. Carsten Gräser, Tobias Kies

8th exercise for the lecture

Numerics IV

Winter Term 2016/2017

`http://numerik.mi.fu-berlin.de/wiki/WS_2016/NumericsIV.php`

Due: Tuesday, Jan 3rd, 2017 for theoretical exercises and
Tuesday, Jan 10th, 2017 for programming exercises

Exercise 1 (8 TP)
Let $\Omega \subseteq \mathbb{R}^n$ bounded and $1 \leq p < \infty$. Suppose that $\Phi\colon \mathbb{R} \to \mathbb{R}$ is continuous and linearly bounded, i. e. there exist $a, b \in \mathbb{R}$ such that

$$\forall x \in \mathbb{R}\colon \quad |\Phi(x)| \leq |ax + b| \,.$$

Define $\psi(v) := \Phi \circ v$ for all $v \in L^p(\Omega)$

a) Show that $\psi(v) \in L^p(\Omega)$ for all $v \in L^p(\Omega)$.

b) From now on suppose $u \in L^p(\Omega)$ and $(u_k)_{k\in\mathbb{N}} \subseteq L^p(\Omega)$ such that $u_k \to u$ in $L^p(\Omega)$ for $k \to \infty$. Show that there exists a subsequence of $(u_{k_n})_{n\in\mathbb{N}}$ of $(u_k)_{k\in\mathbb{N}}$ such that $\lim_{n\to\infty} \psi(u_{k_n})(x) = \psi(u)(x)$ for almost-all $x \in \Omega$.

c) Find an unbounded subset $N \subseteq \mathbb{N}$ such that

$$k(x) := \arg\max_{k\in N} |\psi(u_k)(x) - \psi(u)(x)| < \infty.$$

for almost all $x \in \Omega$ and such that

$$\overline{u}(x) := u_{k(x)}(x)$$

is well-defined almost-everywhere on Ω with $\overline{u} \in L^p(\Omega)$.

d) Conclude that

$$\lim_{k\to\infty} \int_\Omega |\psi(u_k)(x) - \psi(u)(x)|^p \,\mathrm{d}x = 0$$

and that in particular

$$\lim_{k\to\infty} \int_\Omega \Phi(u_k(x)) \,\mathrm{d}x = \int_\Omega \Phi(u(x)) \,\mathrm{d}x.$$

Hint: You may use without proof that $f_k \to f$ in $L^p(\Omega)$ for $k \to \infty$ implies that there exists a subsequence of $(f_k)_{k\in\mathbb{N}}$ that converges pointwise to f almost everywhere.

Please turn over...

Programming exercises.
The following exercises will allow you to collect programming points (PP) instead of the usual theory points (TP).

In the following we want to implement a simple Newton method for solving the minimal surface equation. Our goal is to get the script `run_minimal_surface.m` to work by implementing the routines from the following exercises.
Before you run the script, make sure that your version of `pdeutils` is up to date by either re-downloading it from the website or, if you are using `git`, by issuing the command `git pull` from the directory where the library is installed.

Exercise 2 (4 PP)
Implement the Newton method (without damping). Use the template file

`pdeutils_newton.m`

and extend it in such a way that your Newton method conforms with the documentation given therein.

Exercise 3 (4 PP)
Implement Dirichlet boundary conditions as they are explained in the template file

`setDirichletBoundaryValues.m`

Extend the template file in such a way that it conforms with the documentation that is given therein.

Exercise 4 (8 PP)
Implement the operator and functional assembler for the minimal surface equation by completing the template files

`localMinimalSurfaceAssembler.m`

and

`localMinimalSurfaceFunctionalAssembler.m`

Do so in such a way that your implementation is compatible with the code file

`setMinimalSurfaceSystem.m`

Hint: Those assemblers are very similar to the ones for the Poisson problem, which you can find in the files

`assemblers/localLaplaceAssembler.m`

and

`assemblers/localL2FunctionalAssembler.m`

Have fun!