MATH521: Numerical Analysis of Partial Differential Equations
Winter 2018/19, Term 2

Due Date: Thursday, 17 January 2019
Timm Treskatis

Homework Assignment 2: Model Answers

Please submit the following files as indicated below: source code PDF file image file video file

Question 1 | 3 marks | Let $\Omega \subset \mathbb{R}^d$ be a bounded domain in one, two or three dimensions ($d \in \{1, 2, 3\}$), $a : \Omega \to \mathbb{R}^d$ a continuously differentiable vector field, $D : \Omega \to [D_{\min}, D_{\max}]$ a continuously differentiable and $g : \partial\Omega \to [0, 1]$ a continuous function. $D_{\min} > 0$ and $D_{\max} > 0$ are given constants with $D_{\min} \leq D_{\max}$.

The steady advection-diffusion problem

$$\begin{aligned} \operatorname{div}(ua) - \operatorname{div}(D\nabla u) &= 0 \qquad \text{in } \Omega \\ u &= g \qquad \text{on } \partial\Omega \end{aligned}$$

describes how a certain density u is transported through the domain Ω.

(a) Show that any strong solution $u \in C^2(\Omega) \cap C(\bar{\Omega})$ of this problem satisfies

$$0 \leq u(x) \leq 1 \qquad \text{for all } x \in \bar{\Omega}$$

provided that $\operatorname{div} a = 0$ ("incompressibility condition"). Show all working and clearly indicate what results, e.g. Theorem X.Y.Z or Equation (X.Y) from the notes, you use in each step. Also make sure you verify all assumptions of theorems that you apply.

This equation is given in divergence form. Let's apply the product rule to write it in non-divergence form:

$$\begin{aligned} \operatorname{div}(ua) - \operatorname{div}(D\nabla u) &= a \cdot \nabla u + (\operatorname{div} a)u - D\Delta u - \nabla D \cdot \nabla u \\ &= -D\Delta u + (a - \nabla D) \cdot \nabla u + (\operatorname{div} a)u. \end{aligned}$$

The coefficient matrix of the principal part of this operator (according to Def 2.1.1) is a positive multiple of the identity, $DI_{d\times d}$, so clearly this operator is elliptic. Under the incompressibility condition, this operator has no terms of order zero, hence this PDE satisfies the assumptions of the elliptic maximum and minimum principle (Thm 2.1.4) and we conclude for all $x \in \bar{\Omega}$

$$0 \leq \min_{y\in\partial\Omega} g(y) = \min_{y\in\partial\Omega} u(y) \leq \min_{y\in\bar{\Omega}} u(y) \leq u(x) \leq \max_{y\in\bar{\Omega}} u(y) \leq \max_{y\in\partial\Omega} u(y) = \max_{y\in\partial\Omega} g(y) \leq 1.$$

(b) Why do we have to assume $\operatorname{div} a = 0$ to derive these bounds?

If $\operatorname{div} a \neq 0$, the PDE contains an order-zero reaction term. The elliptic maximum principle from Thm 2.1.4 is no longer applicable.

FYI: If $\operatorname{div} a \geq 0$ everywhere, you could apply a more general maximum principle that implies that the upper bound is still valid and that the solution is nonnegative (but it may have a local minimum in the interior now). If $\operatorname{div} a < 0$ somewhere in the domain, the problem may not be well-posed.

(c) Derive a weak formulation of the steady advection-diffusion problem with homogeneous Dirichlet boundary condition. (This is not a very interesting problem because its unique solution is $u \equiv 0$, but the purpose of this question is to familiarise you with weak formulations!)

Since we're going to apply the *divergence theorem*, the given *divergence form* of the PDE is a lot more convenient to work with than the non-divergence form :)

We multiply with a sufficiently regular test function v that vanishes on the boundary and integrate by parts:

$$\int_\Omega (\operatorname{div}(ua) - \operatorname{div}(D\nabla u))\, v \,\mathrm{d}x = \int_{\partial\Omega} v(ua - D\nabla u)\cdot n \,\mathrm{d}x + \int_\Omega (D\nabla u \cdot \nabla v - ua \cdot \nabla v)\,\mathrm{d}x$$

The boundary integral is zero and the remaining area integral is well-defined for $u, v \in H_0^1(\Omega)$. This yields the weak form:

Find $u \in H_0^1(\Omega)$ such that for all $v \in H_0^1(\Omega)$

$$\int_\Omega (D\nabla u \cdot \nabla v - ua \cdot \nabla v)\,\mathrm{d}x = 0.$$

(Other answers are possible. E.g. you don't have to apply integration by parts to the advection term. This gives the equivalent weak forms

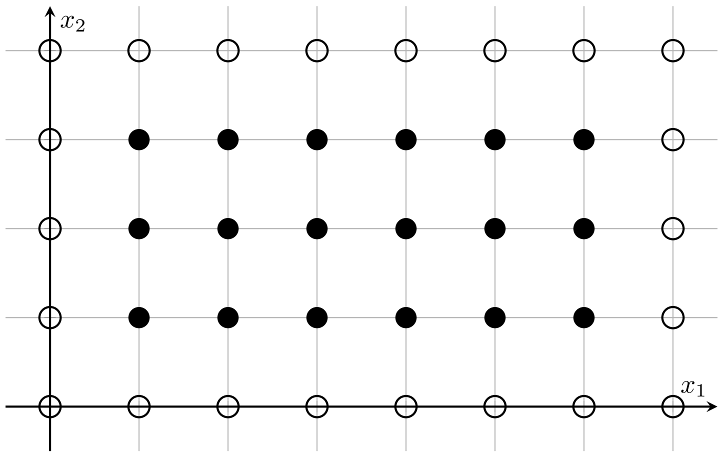
$$\int_\Omega (D\nabla u \cdot \nabla v + \operatorname{div}(ua)v)\,\mathrm{d}x = 0.$$

or

$$\int_\Omega (D\nabla u \cdot \nabla v + (a\cdot\nabla u)v + \operatorname{div}(a)uv)\,\mathrm{d}x = 0.$$

which contain a derivative of u instead of a derivative of v.)

Question 2 | 2 marks | Today we will implement two more auxiliary functions for the finite difference method on rectangular domains. Initially, `X1`, `X2`, a discretisation `F` of the source term and an initial guess `U` for the solution of the PDE are usually given as rectangular arrays. For our computations, we have to re-arrange some of these arrays to column vectors. For plotting and post-processing, we have to re-arrange some column vectors back to rectangular arrays.



If the PDE is equipped with Dirichlet boundary conditions, then the solution values on the boundary (empty circles) are already known. Only the interior solution values (filled circles) are the actual unknowns.

(a) Write a function `msh2vec` that applies so-called *lexicographical ordering* to re-arrange a rectangular array to a column vector. This function takes the two input variables

`U`: an array of size (`msh.N(2)` $- 1) \times ($`msh.N(1)` $- 1)$ corresponding to function values on the *interior* nodes of the grid only, not the boundary nodes

`msh`: the output of `meshRectangle`

and returns

`u`: an array of size (`msh.N(1)` $- 1)($`msh.N(2)` $- 1) \times 1$

The first component of `u` should contain the value of `U` that corresponds to the bottom left interior grid point (filled circle) of the rectangular domain, the second component should correspond to the next grid point to the right etc. Moving from left to right row by row, the last component of `u` will be equal to the value of `U` that corresponds to the interior grid point (filled circle) in the top right corner of the domain.

(b) Implement a function `U = vec2msh(u,msh)` that undoes the action of `msh2vec`. Write a script `hw2.m` that tests whether for the sample function from `hw1.m` the arrays

```
U = u(msh.X1(2:end-1,2:end-1),msh.X2(2:end-1,2:end-1))
```

and

```
V = vec2msh(msh2vec(U,msh),msh)
```

are indeed the same.

Hint: In GNU Octave / MATLAB, the command `reshape` may be helpful.

Please submit all files that are needed to run your code, even if you have submitted them previously.

Your Course Project It's time to start thinking about a small research project. Please consult the course outline for details, and if you have an idea or need one, please come and talk to me!

Your Learning Progress | What is the one most important thing that you have learnt from this assignment?

What is the most substantial new insight that you have gained from this course this week? Any *aha moment*?