Numerical Methods for PDEs - Theory and Implementation

- Estimated time: 8 h/week
- Language of instruction: English

1 What is the seminar about?

Partial differential equations (PDE) appear in many situations in mathematics, physics, biology, engineering and computer science – just to mention a few. PDE theory gives us analytical tools to assess the qualitative bahvior of solutions. However, in most practically relevant applications it is important to also quantify solutions to very complicated PDEs. Numerical methods are used to find approximate solutions to such equations but their choice highly depends on the problem at hand.

In this seminar the students will get aquainted with

(i) Examples for PDE and related problems from different disciplines in sciences

(ii) A set of numerical methods to solve PDE problems

(iii) Tools to implement numerical methods, i.e., programming languages

(iv) A basic workflow for a structured implementation (IDEs, version control, unit testing, documentation)

2 Organization and Goals

Each (group of) student(s) will work on a project from the list. Suggestions for alternative problems are welcome. The outcome of the project should be as follows

(i) An implemtation that solves the problem

(ii) A writtem report containing

  a) a brief description of the problem

  b) a description of the numerical method used

  c) a documentation of the implementation and

d) one or two numerical examples

(iii) a short talk about the project of max. 15 minutes plus 5 minutes of discussion in the end of the semester

(iv) each group's presentation will be evaluated by another group or student in a mini report. This way the students shall learn how to critically (and diplomatically) judge other people's work

Since understanding the problem and implementing a solution takes some time I will (in the first few sessions) introduce a set of tools – that the students can (but do not have to) use – that are supposed to facilitate the workflow:

- An brief overview of the projects and numerical methods
- An overview of suitable programming languages
- Maybe a little introduction to a relatively new programming language for scientific computing – Julia – and a good C++ library.
- Version control with git
- A few words about documentation, unit testing, debugging (verification and validation) and visualization

3 Possible Topics

In addition to this list participants are welcome to suggest topics.

3.1 Finite Difference Methods for the Heat Equation

The student s will implement a set of different finite difference methods to solve a simple diffusion equation

$$\begin{aligned} \partial_t u - \nabla \cdot A \nabla u &= f \quad \text{in } B \\ +\text{boundary and initial conditions,} & \end{aligned} \tag{1}$$

compare the results and discuss them. $A = A(x) \in \mathbb{R}^{2,2}$ is a uniformly positive definite matrix to ensure ellipticity of the spatial operator and f is a source term.

3.2 Discontinuous Galerkin Methods for Transport Problems

Discontinuous Galerkin methods gained a lot of attention in the recent years due to their superiority over classical finite element methods and finite volume methods for advection

dominated problems. In some sense this method combines the advantages of FVMs (stability, simlicity, conservation etc.) and FEMs (good accessibility to analysis, possibly high order etc.). The students will implement and test a DG method on a simple 2D problem and report about their findings in a short talk together with a short introduction to the idea of DG.

3.3 Finite Elements with Adaptive Mesh Refinement

In many scenarios the initial mesh used in FEM applications is coarse and does not resolve interesting features of the solution sufficiently well. One option would be to refine the entire mesh until all desired features are well resolved. This is often too expensive for practical applications. The idea of adaptive mesh refinement is to use a local a posteriori error estimator error to find regions of the mesh where the error of the solution is higher than a chosen threshold. Then one can refine only these regions and get a good solution without refining globally.

The students will implement and test an AMR method on a simple 2D diffusion problem and report about their findings in a short talk together with a short introduction to the idea of a posteriori estimators.

3.4 Stabilized Finite Element Methods for Advection-Diffusion Problems

Classical Finite Element methods do not have very good stability properties when it comes to flow dominated problems for stationary flows. Stability can be improved by adding diffusion along streamlines. This idea is implemented in so-called streamline upwind Petrov-Galerkin methods (SUPG). The sudents will implement and test the SUPG method on a stationary 2D advection-diffusion equation and report about their findings in a short talk together with a short introduction to the idea.

3.5 Multigrid Methods for a Poisson Type Equation

Multigrid methods are based on the idea that errors can be smoothed relative to the comutational grid. High frequency errors can be eliminated by an iteration process on a fine grid while low frequency errors are resiliant to damping (they are already smooth). The idea is now to transfer the solution to coarser grids and do the iteration there since low frequency errors on fine grids are high frequency errors on coarser grids.

The students should implement a multigrid solver (V-cycle) for a Poisson Type equation

$$\begin{aligned} -\nabla \cdot (A \nabla u) &= f \quad \text{in } D \\ u &= g \quad \text{on } \partial D\,, \end{aligned} \tag{2}$$

where $A = A(x) \in \mathbb{R}^{2,2}$ is a uniformly positive definite matrix.

3.6 Stationary Flows at Low Reynolds Number

The Stokes equations for an incompressible fluid are a good model for for flows that have small advective inertial forces, with small velocities or at small length scales. It is not exactly an elliptic problem as the Poisson equation since it has a linear constraint. Therefore it is something like a Poisson equation on a subspace:

$$\begin{aligned} -\Delta u + \nabla p &= f \ , \\ \nabla \cdot u &= 0 \quad \text{in } B \\ +&\text{boundary conditions.} \end{aligned} \tag{3}$$

The student students will learn, implement and test a mixed finite element method to solve this type of a saddle point problem.

3.7 Multiscale Finite Elements for Subsurface Flows

In oil reservoir modeling or, more generally, in the modelling of subsurface flows the engineer is often interested in the large scale (average) behavior of a concentration or similar. In this context Darcy's law arises, describing an incompressible single phase flow in a porous medium:

$$\begin{aligned} u &= -a(x/\varepsilon)\nabla p \quad \text{in } D \quad \text{(Darcy's law)} \\ \nabla \cdot u &= f \quad \text{in } B \quad \text{(conservation of mass)} \\ u &= g \quad \text{on } \partial D \, . \end{aligned} \tag{4}$$

The coefficient $a > 0$ is supposed to be sufficiently regular and ε is a small number. Standard finite elements are problematic in this case because they either need a very high resolution or do not get the large scale behavior correctly.

The student will implement a multiscale finite element method and compare the solution to the solution that a standard FEM.

3.8 Mixed Finite Elements for Subsurface Flows

The above mentioned Darcy law can be solved a system of PDEs for the velocities u and the pressure p simultaneously. This requires the use of suitable pairs of finite elements for both variables since an arbitrary choice is not guaranteed to be stable. The student(s) will implement and test a combination of the lowest order Raviart-Thomas element $\mathcal{RT}_0$ and the discontinuous DP_0 or DQ_0 element. The report should contain a description of the elements and at least mention what the stability condition is.

3.9 Finite Elements for Linearized Elasticity

For small deformations the equations of so-called linearized elasticity can be simplified to

$$-\nabla \cdot \sigma = f \quad \text{in } B \subset \mathbb{R}^2 \,. \tag{5}$$

Finite elements are a common tool in computational mechnics to tackle such problems. The students will decide on implementing a certain method (based on displacements or the Hellinger-Reissner principle) in two dimensions, test it and discuss advantages and disadvantages.

3.10 A Shape Descriptor in Computer Graphics: The Heat Kernel Signature

Comparing shapes, e.g., given as triangular surface meshes is an important task in computer vision, medical imaging and graphics. Deforming a shape produces many instances of the same object, called articulations. Ideally one would like to point out differences in articulations but also allow a comparison among objects (shapes) that have a different meaning (semantic).

A way to allow a comparison different objects that can appear in different articulations is to give every object a signature that is invariant under certain class of deformations. Modeling heat flow on an embedded surface can give a signature that is invariant under isometric deformations of the shape. The key role is played here by eigenvalues of the so-called Laplace-Beltrami operator.

The students will implement a method that computes the heat kernel signature (HKS) of a set of triangular surface models, show results on a few example meshes and introduce applications of the HKS (verbally using for example results fom papers).