Numerical Methods for Partial Differential Equations

Thomas Wick

AG Wissenschaftliches Rechnen (GWR)

Institut für Angewandte Mathematik (IfAM)

Gottfried Wilhelm Leibniz Universität Hannover (LUH)

Welfengarten 1, 30167 Hannover, Germany

https://www.ifam.uni-hannover.de/wick.html

thomas.wick@ifam.uni-hannover.de

Last update:
Tuesday, July 10, 2018

by Elisa Wick

Foreword

These notes accompany the lecture to the class

Numerik partieller Differentialgleichungen 1+2

at the Leibniz Universität Hannover (LUH). This class is a classical $4 + 2$ lecture in the German university system. This class was followed in the semester after by a 2 hours

Finite-Elemente-Programmierpraktikum (C++)

in which algorithmic techniques have been implemented by the participants.

The **topics** are:

- Recapitulation of characteristic concepts of numerical mathematics
- Brief review of mathematical modeling
- Finite differences (FD) for elliptic boundary value problems
- Finite elements (FEM) for elliptic boundary value problems
- A posteriori error estimation of goal functionals using duality arguments
- Numerical solution of the discretized problems
- A brief introduction to vector-valued problems (elasticity/Stokes) (if time permits)
- Methods for parabolic and hyperbolic problems
- A brief introduction to numerical methods for nonlinear problems (if time permits)

The prerequisites are lectures in calculus, linear algebra and an introduction to numerics or as well classes on the theory of ordinary (ODE) or partial (PDE) differential equations. Classes in continuum mechanics may also be helpful, but are not mandatory to understand these notes.

On the other hand, for the convenience of all of us, many results from linear algebra, analysis and functional analysis are included (often without the proofs though) in order to achieve a more or less self-contained booklet.

To fully enjoy these notes, the reader should have a taste for practical aspects and algorithms as well as he/she should be ready to dive into theoretical aspects of variational formulations and numerical analysis. In most sections, **numerical tests complement algorithms and theory**.

At this place I already want to thank my students from the WS 2017/2018 class and my PhD students for letting me know about mistakes and typos. Furthermore thanks to Katrin Mang for some tikz pictures and latex improvements.

I would be more than grateful if you let me know about errors inside these lecture notes via

`thomas.wick@ifam.uni-hannover.de`

Thomas Wick

(Hannover, July 2018)

Contents

1 Main literature **9**

2 Motivation **11**
2.1 Scientific computing (english) / Wissenschaftliches Rechnen (german) . . . 11
2.2 Differential equations and guiding questions . . . 11
2.3 Errors . . . 12
2.4 Concepts in numerical mathematics . . . 12
2.5 Well-posedness (in the sense of Hadamard 1923) . . . 13
2.6 Numerical methods for solving PDEs . . . 14
2.7 Chapter summary and outlook . . . 14

3 Notation **15**
3.1 Domains . . . 15
3.2 Independent variables . . . 15
3.3 Function, vector and tensor notation . . . 15
3.4 Partial derivatives . . . 15
3.5 Multiindex notation . . . 16
3.6 Gradient, divergence, trace, Laplace, rotation . . . 16
3.7 Invariants of a matrix . . . 17
3.8 Vector spaces . . . 17
3.9 Normed spaces . . . 18
3.10 Linear mappings . . . 18
3.11 Little o and big O - the Landau symbols . . . 19
3.12 Taylor expansion . . . 20
3.13 Transformation of integrals: substitution rule / change of variables . . . 20
3.14 Gauss-Green theorem / Divergence theorem . . . 21
3.15 Integration by parts and Green's formulae . . . 22
3.16 Chapter summary and outlook . . . 22

4 An extended introduction **23**
4.1 Examples of differential equations . . . 23
4.1.1 Laplace equation / Poisson problem . . . 23
4.1.2 Mass conservation / First-order hyperbolic problem . . . 24
4.1.3 Elasticity (Lamé-Navier) . . . 26
4.1.4 The incompressible, isothermal Navier-Stokes equations (fluid mechanics) . . . 27
4.2 Three important PDEs . . . 28
4.2.1 Elliptic equations and Poisson problem/Laplace equation . . . 28
4.2.2 Parabolic equations and heat equation . . . 30
4.2.3 Hyperbolic equations and the wave equation . . . 31
4.3 Boundary conditions and initial data . . . 32
4.4 The general definition of a differential equation . . . 32
4.5 Further remarks on the classification into three important types . . . 35
4.6 Chapter summary and outlook . . . 37

5 Finite differences (FD) for elliptic problems **39**
5.1 A 1D model problem: Poisson . . . 39
5.2 Well-posedness of the continuous problem . . . 39
5.2.1 Green's function and well-posedness . . . 40
5.2.2 Maximum principle on the continuous level . . . 42
5.2.3 Regularity of the solution . . . 42
5.3 Spatial discretization . . . 42
5.4 Solving the linear equation system . . . 43

5.5 Well-posedness of the discrete problem . . . 43
5.5.1 Existence and uniqueness of the discrete problem . . . 44
5.5.2 Maximum principle on the discrete level . . . 44
5.6 Numerical analysis: consistency, stability, and convergence . . . 46
5.6.1 Consistency . . . 46
5.6.2 Stability in L^∞ . . . 47
5.6.3 Convergence L^∞ . . . 47
5.6.4 Convergence L^2 . . . 48
5.7 Numerical test: 1D Poisson . . . 49
5.7.1 How can we improve the numerical solution? . . . 51
5.8 Finite differences in 2D . . . 51
5.8.1 Discretization . . . 51
5.9 Chapter summary and outlook . . . 52

6 Theory and finite elements (FEM) for elliptic problems 53
6.1 Preliminaries . . . 53
6.1.1 Construction of an exact solution (only possible for simple cases!) . . . 53
6.1.2 Equivalent formulations . . . 54
6.2 The weak form: defining a bilinear form and a linear form . . . 57
6.3 Finite elements in 1D . . . 59
6.3.1 The mesh . . . 59
6.3.2 Linear finite elements . . . 59
6.3.3 The process to construct the specific form of the shape functions . . . 61
6.3.4 The discrete weak form . . . 62
6.3.5 Evaluation of the integrals . . . 63
6.3.6 Definition of a finite element . . . 64
6.3.7 Properties of the system matrix A . . . 65
6.3.8 Numerical test: 1D Poisson . . . 65
6.4 Algorithmic details . . . 67
6.4.1 Assembling integrals on each cell K . . . 67
6.4.2 Example in $\mathbb{R}^5$ . . . 67
6.4.3 Numerical quadrature . . . 69
6.4.4 Details on the evaluation on the master element . . . 70
6.4.5 Generalization to s elements . . . 72
6.4.6 Example: Section 6.4.2 continued using Sections 6.4.4 and 6.4.5 . . . 72
6.5 Quadratic finite elements: P_2 elements . . . 74
6.5.1 Algorithmic aspects . . . 74
6.5.2 Numerical test: 1D Poisson using quadratic FEM . . . 75
6.6 Galerkin orthogonality, a geometric interpretation of the FEM, and a first error estimate . . . 76
6.6.1 Excursus: Approximation theory emphasizing on the projection theorem . . . 76
6.6.2 Application to our FEM setting . . . 83
6.7 Neumann & Robin problems and varying coefficients . . . 85
6.7.1 Robin boundary conditions . . . 85
6.7.2 Neumann boundary conditions . . . 87
6.7.3 Coupled problems / Transmission problems . . . 88
6.8 Variational formulations of elliptic problems in higher dimensions . . . 91
6.8.1 Comments on the domain Ω and its boundaries $\partial\Omega$ . . . 91
6.8.2 Integration by parts and Green's formulae . . . 91
6.8.3 Variational formulations . . . 92
6.8.4 A short excursus to analysis, linear algebra, functional analysis and Sobolev spaces . . . 93
6.8.5 The Lax-Milgram lemma . . . 100
6.8.6 The energy norm . . . 103
6.8.7 The Poincaré inequality . . . 103
6.8.8 Trace theorems . . . 104

6.8.9 A compactness result . . . 105
6.9 Theory of elliptic problems . . . 105
6.9.1 The formal procedure . . . 105
6.9.2 Poisson's problem: homogeneous Dirichlet conditions . . . 105
6.9.3 Nonhomogeneous Dirichlet conditions . . . 107
6.9.4 Neumann conditions . . . 108
6.9.5 Robin conditions . . . 109
6.10 Finite element spaces . . . 110
6.10.1 Example: a triangle in 2D . . . 111
6.10.2 Example: a quadrilateral in 2D . . . 113
6.10.3 Well-posedness of the discrete problem . . . 113
6.11 Numerical analysis: error estimates . . . 114
6.11.1 The Céa lemma . . . 114
6.11.2 H^1 and L^2 estimates in 1D . . . 115
6.11.3 An improved L^2 estimate - the Aubin-Nitsche trick . . . 119
6.11.4 Analogy between finite differences and finite elements . . . 120
6.11.5 2D and higher dimensions . . . 120
6.12 Numerical analysis: influence of numerical quadrature, first Strang lemma . . . 121
6.12.1 Approximate solution of A and b . . . 121
6.12.2 Interpolation with exact polynomial integration . . . 123
6.12.3 Integration with numerical quadrature . . . 125
6.12.4 Numerical tests . . . 132
6.13 Numerical tests and computational convergence analysis . . . 135
6.13.1 2D Poisson . . . 135
6.13.2 Numerical test: 3D . . . 136
6.13.3 Checking programming code and convergence analysis for linear and quadratic FEM . . . 136
6.13.4 Convergence analysis for 1D Poisson using linear FEM . . . 138
6.13.5 Computing the error norms $\|u - u_h\|$ . . . 139
6.14 Chapter summary and outlook . . . 140

7 A posteriori error estimation and mesh adaptivity 141
7.1 Principles of error estimation . . . 141
7.2 Preliminaries . . . 141
7.3 Goal-oriented error estimation using duality arguments: dual-weighted residuals (DWR) . . . 142
7.3.1 Motivation . . . 142
7.3.2 Excursus: Variational principles and Lagrange multipliers in mechanics . . . 143
7.3.3 First-order optimality system . . . 146
7.3.4 Excursus: Differentiation in Banach spaces . . . 147
7.3.5 Linear problems and linear goal functionals (Poisson) . . . 149
7.3.6 Nonlinear problems and nonlinear goal functionals . . . 150
7.4 Approximation of the adjoint solution for the primal estimator $\rho(u_h)(\cdot)$ . . . 152
7.4.1 Summary and alternatives for computing the adjoint . . . 152
7.5 Approximation of the primal solution for the adjoint estimator $\rho^*(z_h)(\cdot)$ . . . 153
7.6 Measuring the quality of the error estimator η . . . 153
7.7 Localization techniques . . . 153
7.7.1 The classical way of error localization of the primal estimator for linear problems . . . 154
7.7.2 The classical way for the combined estimator . . . 155
7.7.3 A variational primal-based error estimator with PU localization . . . 156
7.7.4 PU localization for the combined estimator . . . 156
7.8 Comments to adjoint-based error estimation . . . 157
7.9 Residual-based error estimation . . . 157
7.10 Mesh refinement strategies . . . 157
7.10.1 How to refine marked cells . . . 158
7.10.2 Convergence of adaptive algorithms . . . 158

7.11 Numerical test: Poisson with mean value goal functional . . . 159
7.12 Numerical test: L-shaped domain with Dirac rhs and Dirac goal functional . . . 162
7.13 Final comments to error estimation in numerical simulations . . . 164
7.14 Example: Stationary Navier-Stokes 2D-1 benchmark . . . 164
7.14.1 Equations . . . 164
7.14.2 Functionals of interest . . . 166
7.14.3 A duality-based a posteriori error estimator . . . 166
7.14.4 2D-1 configuration . . . 167
7.14.5 Boundary conditions . . . 167
7.14.6 Parameters and right hand side data . . . 167
7.14.7 Step 1: Verification of benchmark values . . . 167
7.14.8 Step 2 and Step 3: Computing $J(u)$ on a fine mesh, $J(u) - J(u_h)$ and I_{eff} . . . 167
7.14.9 More findings - graphical solutions . . . 168
7.15 Chapter summary and outlook . . . 169

8 Numerical solution of the discretized problems 171
8.1 On the condition number of the system matrix . . . 171
8.2 Fixed-point schemes: Richardson, Jacobi, Gauss-Seidel . . . 171
8.3 Gradient descent . . . 173
8.4 Conjugate gradients (a Krylov space method) . . . 175
8.4.1 Formulation of the CG scheme . . . 175
8.4.2 Convergence analysis of the CG scheme . . . 177
8.5 Preconditioning . . . 180
8.6 Comments on other Krylov space methods such as GMRES and BiCGStab . . . 181
8.7 Numerical tests . . . 182

9 Applications in solid mechanics: linearized elasticity and briefly Stokes 183
9.1 Modeling . . . 183
9.2 Well-posedness . . . 184
9.3 Finite element discretization . . . 187
9.4 Numerical tests . . . 189
9.4.1 3D . . . 189
9.4.2 2D test with focus on the maximum principle . . . 190
9.5 Stokes - FEM discretization . . . 191
9.6 Numerical test - 2D-1 benchmark without convection term . . . 192
9.7 Chapter summary and outlook . . . 192

10 Methods for parabolic and hyperbolic problems 193
10.1 Principle procedures for discretizing time and space . . . 193
10.2 Bochner spaces - space-time functions . . . 193
10.3 Methods for parabolic problems . . . 194
10.3.1 Problem statement . . . 194
10.3.2 Temporal discretization . . . 195
10.3.3 Spatial discretization . . . 196
10.3.4 Evaluation of the integrals (1D in space) . . . 197
10.3.5 Final algorithms . . . 197
10.3.6 Recapitulation of ODE stability analysis using finite differences . . . 198
10.3.7 Stiff problems . . . 204
10.3.8 Numerical analysis: stability analysis . . . 205
10.3.9 Numerical tests . . . 207
10.4 Methods for second-order-in-time hyperbolic problems . . . 210
10.4.1 Problem statement . . . 210
10.4.2 Variational formulations . . . 210
10.4.3 A space-time formulation . . . 211
10.4.4 Energy conservation and consequences for good time-stepping schemes . . . 212

10.4.5 One-step-theta times-stepping for the wave equation . . . 214
10.4.6 Stability analysis / energy conservation on the time-discrete level . . . 215
10.4.7 Summary of stability, convergence and energy conservation . . . 216
10.5 Numerical tests: scalar wave equation . . . 217
10.6 Numerical tests: elastic wave equation . . . 219
10.6.1 Constant force - stationary limit . . . 219
10.6.2 Time-dependent force - Crank-Nicolson scheme . . . 219
10.6.3 Time-dependent force - backward Euler scheme . . . 221
10.7 Chapter summary and outlook . . . 221

11 Nonlinear problems **223**
11.1 Differentiation in Banach spaces . . . 223
11.2 Linearization techniques . . . 223
11.2.1 Fixed-point iteration . . . 223
11.2.2 Newton's method in $\mathbb{R}$ - the Newton-Raphson method . . . 224
11.2.3 Newton's method: overview. Going from $\mathbb{R}$ to Banach spaces . . . 226
11.2.4 A basic algorithm for a residual-based Newton method . . . 227
11.3 Newton's method for variational formulations . . . 227
11.3.1 Pseudo C++ code of a Newton implementation with line search . . . 229
11.4 An academic example of finite-difference-in-time, Galerkin-FEM-in-space-discretization and linearization in a Newton setting . . . 230
11.5 Navier-Stokes - FEM discretization . . . 234
11.6 Chapter summary and outlook . . . 234

12 Computational convergence analysis **235**
12.1 Discretization error . . . 235
12.1.1 Relationship between h and N (DoFs) . . . 235
12.1.2 Discretization error . . . 236
12.1.3 Computationally-obtained convergence order . . . 237
12.2 Iteration error . . . 238
12.3 Chapter summary and outlook . . . 238

13 Wrap-up **239**
13.1 Quiz . . . 239
13.2 The end . . . 241

Bibliography **243**

Index **247**

1 Main literature

1. C. Johnson 1987; Numerical solution of partial differential equations by the finite element method [43] (in English; original version in Swedish)
2. P. Ciarlet 1987: The finite element method for elliptic problems [20] (in English)
3. R. Rannacher 2017 (in German); Numerik partieller Differentialgleichungen [56]
4. Ch. Grossmann, H.-G. Roos, M. Stynes; Numerical treatment of partial differential equations [33] (in English; original version in German).
5. T. Hughes 2000; The finite element method [40] (in English)
6. G. F. Carey and J. T. Oden 1984; Finite Elements. Volume III. Compuational Aspects [16] (in English)
7. D. Braess 2007; Finite Elemente [12] (in German)
8. G. Allaire, F. Alouges: Analyse variationnelle des équations aux dérivées partielles [2] (in French)
9. S. Brenner and L.R. Scott 2008; The mathematical theory of finite element methods [13] (in English)
10. W. Hackbusch 1986: Theorie und Numerik elliptischer Differentialgleichungen [34] (in German; also available in English)
11. H.R. Schwarz: Methode der finiten Elemente [63] (in German)
12. G. Allaire: Introduction to mathematical modelling, numerical simulation, and optimization [1] (in English; original version in French).

2 Motivation

This lecture is devoted to **the numerical solution of partial differential equations (PDEs)**. PDEs arise in many fields and are extremely important in modeling of technical processes with applications in physics, biology, chemisty, economics, mechanical engineering, and so forth.

2.1 Scientific computing (english) / Wissenschaftliches Rechnen (german)

Developing and analyzing algorithms for solving PDEs with a computer is a part of **numerical methods**, which itself is a part of **scientific computing**. Scientific computing comprises three main fields:

- **Mathematical modeling** and analysis of physical, biological, chemical, economical, financial processes, and so forth;
- Development of reliable and efficient **numerical methods** and algorithms and their analysis;
- Implementation of these algorithms into a **software**.

All these steps work in a feed-back manner and the different subtasks interact with each other. It is in fact the third above aspect, namely *software and computers*, who helped to establish this third category of science. Thus, a new branch of mathematics, *numerical mathematics/scientific computing*, has been established. This kind of mathematics is experimental like experiments in physics/chemistry/biology.

Therefore, numerical modeling offers to investigate research fields that have partially not been addressable. Why? On the one hand experiments are often too expensive, too far away (Mars, Moon, astronomy in general), the scales are too small (nano-scale for example); or experiments are simply too dangerous. On the other hand, mathematical theory or the explicit solution of an (ambitious) engineering problem in an analytical manner is often impossible!

2.2 Differential equations and guiding questions

Let us first roughly define the meaning of a differential equation:

Definition 2.1. *A differential equation is a mathematical equation that relates the function with its derivatives.*

Differential equations can be split into two classes:

Definition 2.2 (Ordinary differential equation (ODE))**.** *An ordinary differential equation (ODE) is an equation (or equation system) involving an unknown function of one independent variable and certain of its derivatives.*

Definition 2.3 (Partial differential equation (PDE))**.** *A partial differential equation (PDE) is an equation (or equation system) involving an unknown function of two or more variables and certain of its partial derivatives.*

A solution u of a differential equation is in most cases (except for simple academic test cases in which a manufactured solution can be constructed) computed with the help of discretization schemes generating a sequence of approximate solutions $\{u_h\}_{h\to 0}$. Here h is the so-called **discretization parameter**. For $h \to 0$ we (hopefully) approach the continuous (unknown) solution u.

Important questions are:

- What kind of discretization scheme shall we use?
- How do we design algorithms to compute u_h?
- Can we proof that these algorithms really work?
- Are they robust and, ideally, efficient?
- How far is u_h away from u in a certain (error) norm?
- The discretized systems (to obtain u_h) are often large with a huge number of unknowns: how do we solve these linear equation systems?
- What is the computational cost?

2.3 Errors

In order to realize these algorithmic questions, we go ahead and implement them in a software (for instance Matlab/octave, python, fortran, C++) using a computer or cluster. Here, we face three major error sources that need to be taken into account:

- The set of numbers is finite and a calculation is limited by machine precision, which results in **round-off errors**.
- The memory of a computer (or cluster) is finite and thus functions and equations can only be represented through approximations. Thus, continuous information has to be represented through discrete information, which results into the investigation of so-called **discretization errors**.
- All further simplifications of a numerical algorithm (in order to solve the discrete problem), with the final goal to reduce the computational time, are so-called **systematic errors**. One example is the stopping criterion after how many steps an iterative method is stopped.

Numerical simulations are also subject to **model errors** that additionally the influence the interpretation of numerical results:

- In order to make a 'quick guess' of a possible solution and to start the development of an algorithm to address at a later stage a difficult problem, often complicated (nonlinear) differential equations are reduced to simple (in most cases linear) versions, which results in the so-called **model error**.
- **Data errors**: the data (e.g., input data, boundary conditions, parameters) are finally obtained from experimental data and may be inaccurate themselves.

It very important to understand that we **never can avoid** all these errors. The important aspect is to **control** these errors and to provide answers if these errors are sufficiently big to influence the interpretation of numerical simulations or if they can be assumed to be small. A big branch of numerical mathematics is to derive error estimates that allow to predict about the size of arising errors.

2.4 Concepts in numerical mathematics

In introductory classes to numerical methods, we deal with concepts that are very characteristic for numerical modeling. In [61], we summarized them into seven points, which will frequently encounter us in the forthcoming chapters:

1. **Approximation**: since analytical solutions are not possible to achieve as we just learned in the previous section, solutions are obtained by **numerical approximations**.
2. **Convergence**: is a qualitative expression that tells us when members a_n of a sequence $(a_n)_{n\in\mathbb{N}}$ are sufficiently close to a limit a. In numerical mathematics this limit is often the solution that we are looking for.

3. **Order of convergence**: While in analysis, we are often interested in the convergence itself, in numerical mathematics we must pay attention how long it takes until a numerical solution has sufficient accuracy. The longer a simulation takes, the more time and more energy (electricity to run the computer, air conditioning of servers, etc.) are consumed. Therefore, we are heavily interested in developing fast algorithms. In order to judge whether a algorithm is fast or not we have to determine the order of convergence.

4. **Errors**: Numerical mathematics can be considered as the branch 'mathematics of errors'. What does this mean? Numerical modeling is not wrong, inexact or non-precise! Since we cut sequences after a final number of steps or accept sufficiently accurate solutions obtained from our software, we need to say *how well the (unknown) exact solution by this numerical solution is approximated.* In other words, we need to determine the error, which can arise in various forms as we discussed in the previous section.

5. **Error estimation**: This is one of the biggest branches in numerical mathematics. We need to derive error formulae to judge the outcome of our numerical simulations and to measure the difference of the numerical solution and the (unknown) exact solution in a certain norm.

6. **Efficiency**: In general we can say, the higher the convergence order of an algorithm is, the more efficient our algorithm is. Therefore, we obtain faster the numerical solution to a given problem. But numerical efficiency is not automatically related to resource-effective computing. For instance, developing a parallel code using MPI (message passing interface) will definitely yield in less CPU (central processing unit) time a numerical solution. However, whether a parallel machine does need less electricity (and thus less money) than a sequential desktop machine/code is a priori unclear.

7. **Stability**: Despite being the last concept, in most developments, this is the very first step to check. How robust is our algorithm against different model and physical parameters? Is the algorithm stable with respect to different input data? This condition relates in the broadest sense to the third condition of Hadamard defined in Section 2.5.

2.5 Well-posedness (in the sense of Hadamard 1923)

The concept of well-posedness is very general and in fact very simple.

Definition 2.4. *Let $A : X \to Y$ be a mapping and X, Y topological spaces. Then, the problem*

$$Ax = y \tag{1}$$

is well posed if

1. *for each $y \in Y$, Problem (1) has a solution x,*
2. *the solution x is unique,*
3. *the solution x depends continuously on the problem data.*

The first condition is immediately clear. The second condition is also obvious but often difficult to meet - and in fact many physical processes do not have unique solutions. The last condition says if a variation of the input data (right hand side, boundary values, initial conditions) vary only a little bit, then also the (unique) solution should only vary a bit.

Remark 2.5. *Problems in which one of the three conditions is violated are **ill-posed**.*

Example 2.6 (Numerical differentiation)**.** *In view of the later chapters and that we shall deal with derivatives throughout these notes, we consider the difference quotient $D_h g$ as approximation of the derivative g' of a function g. We know from introduction to numerics:*

$$\begin{aligned}
D_h g(x) &= \frac{g(x+h) - g(x)}{h}, \quad 0 \le x \le h/2, \\
D_h g(x) &= \frac{g(x+h/2) - g(x-h/2)}{h}, \quad h/2 < x < 1 - h/2, \\
D_h g(x) &= \frac{g(x) - g(x-h)}{h}, \quad 1 - h/2 \le x \le 1,
\end{aligned}$$

with a step size h. We know that

$$|g'(x) - D_h g(x)| = O(h^2) \quad \textit{for the central difference quotient}$$
$$|g'(x) - D_h g(x)| = O(h) \quad \textit{for the forward/backward difference quotients},$$

when g is three-times continuously-differentiable. When g is only of class $C^2[0,1]$. If we only have disturbed data g_ε (e.g., data or model or round-off errors) with $\|g - g_\varepsilon\| \leq \varepsilon$, then we have for the data error

$$|D_h(g_\varepsilon - g)(x)| \leq \frac{2\varepsilon}{h}, \quad 0 \leq x \leq 1.$$

The total error is composed by the data error $D_h(g_\varepsilon - g)$ and the discretization error $g' - D_h g$:

$$D_h g_\varepsilon - g' = D_h(g_\varepsilon - g) + D_h g - g'$$

yielding the estimate

$$\|D_h g_\varepsilon - g'\| \leq \frac{2\varepsilon}{h} + \frac{1}{2}\|g''\|_{C^\infty} h.$$

This means that for noisy data, i.e., $\varepsilon \neq 0$, the total error cannot become arbitrarily small as we normally hope to wish. ***This problem is ill-posed - specifically for $h \ll \varepsilon$.*** *The optimal step size is*

$$h_\varepsilon = \frac{2}{\|g''\|_{C^\infty}^{1/2}} \varepsilon^{1/2},$$

if $g'' \neq 0$. In conclusion the total error is of order $O(\varepsilon^{1/2})$.

2.6 Numerical methods for solving PDEs

There exist various methods for solving numerically PDEs:

- Difference methods: the derivatives in the differential equation will be replaced by difference quotients. As they belong to the most simple methods and are still used frequently in engineering and industry, we will provide a short introduction on them as well.
- Finite volume methods: they are based on local conservation properties of the underlying PDE.
- Variational methods such as Galerkin finite elements (FEM).
- Boundary element methods (BEM).
- Isogeometric analysis (IGA) [41].

2.7 Chapter summary and outlook

In this first chapter, a motivation was given why PDEs are important. Moreover, we recapitulated the definition of a partial differential equation and also recapitulated the characteristic concepts of numerical mathematics. In the next chapter, we shall introduce the notation used throughout these lecture notes.

3 Notation

In this chapter, we collect most parts of the notation and some important results from linear algebra, analysis, vector and tensor analysis, necessary for numerical mathematics.

3.1 Domains

We consider open, bounded domains $\Omega \subset \mathbb{R}^d$ where $d = 1, 2, 3$ is the dimension. The boundary is denoted by $\partial\Omega$. The outer **normal vector** with respect to (w.r.t.) to $\partial\Omega$ is n. We assume that Ω is sufficiently smooth (i.e., a Lipschitz domain or domain with Lipschitz boundary) such that the normal n can be defined. What also works for most of our theory are convex, polyhedral domains with finite corners. For specific definitions of nonsmooth domains, we refer to the literature, e.g., [31].

3.2 Independent variables

A point in $\mathbb{R}^d$ is denoted by

$$x = (x_1, \dots, x_d).$$

The variable for 'time' is denoted by t. The euclidian scalar product is denoted by $(x, y) = x \cdot y = \sum_{i=1}^{d} x_i y_i$.

3.3 Function, vector and tensor notation

Functions are denoted by

$$u := u(x)$$

if they only depend on the spatial variable $x = (x_1, \dots, x_d)$. If they depend on time and space, they are denoted by

$$u = u(t, x).$$

Usually in physics or engineering vector-valued and tensor-valued quantities are denoted in bold font size or with the help of arrows. Unfortunately in mathematics, this notation is only sometimes adopted. We continue this crime and do not distinguish scaler, vector, and tensor-valued functions. Thus for points in $\mathbb{R}^3$ we write:

$$x := (x, y, z) = \mathbf{x} = \vec{x}.$$

Similar for functions from a space $u : \mathbb{R}^3 \supseteq U \to \mathbb{R}^3$:

$$u := (u_x, u_y, u_z) = \mathbf{u} = \vec{u}.$$

And also similar for tensor-valued functions (which often have a bar or two bars under the tensor quantity) as for example the Cauchy stress tensor $\sigma_f \in \mathbb{R}^{3\times 3}$ of a fluid:

$$\sigma_f := \underline{\sigma}_f = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix}.$$

3.4 Partial derivatives

We frequently use:

$$\frac{\partial u}{\partial x} = \partial_x u$$

and

$$\frac{\partial u}{\partial t} = \partial_t u$$

and

$$\frac{\partial^2 u}{\partial t \partial t} = \partial_t^2 u$$

and

$$\frac{\partial^2 u}{\partial x \partial y} = \partial_{xy} u$$

3.5 Multiindex notation

For a general description of ODEs and PDEs the multiindex notation is commonly used.

- A multiindex is a vector $\alpha = (\alpha_1, \ldots, \alpha_n)$, where each component $\alpha_i \in \mathbb{N}_0$. The order is
$$|\alpha| = \alpha_1 + \ldots + \alpha_n.$$
- For a given multiindex we define the partial derivative:
$$D^\alpha u(x) := \partial_{x_1}^{\alpha_1} \cdots \partial_{x_n}^{\alpha_n} u$$
- If $k \in \mathbb{N}_0$, we define the set of all partial derivatives of order k:
$$D^k u(x) := \{D^\alpha u(x) : |\alpha| = k\}.$$

Example 3.1. *Let the problem dimension $n = 3$. Then, $\alpha = (\alpha_1, \alpha_2, \alpha_3)$. For instance, let $\alpha = (2, 0, 1)$. Then $|\alpha| = 3$ and $D^\alpha u(x) = \partial_x^2 \partial_z^1 u(x)$.*

3.6 Gradient, divergence, trace, Laplace, rotation

Well-known in physics, it is convenient to work with the **nabla-operator** to define derivative expressions. The gradient of a single-valued function $v : \mathbb{R}^n \to \mathbb{R}$ reads:
$$\nabla v = \begin{pmatrix} \partial_1 v \\ \vdots \\ \partial_n v \end{pmatrix}.$$

The gradient of a vector-valued function $v : \mathbb{R}^n \to \mathbb{R}^m$ is called **Jacobian matrix** and reads:
$$\nabla v = \begin{pmatrix} \partial_1 v_1 & \ldots & \partial_n v_1 \\ \vdots & & \vdots \\ \partial_1 v_m & \ldots & \partial_n v_m \end{pmatrix}.$$

The divergence is defined for vector-valued functions $v : \mathbb{R}^n \to \mathbb{R}^n$:
$$\text{div } v := \nabla \cdot v := \nabla \cdot \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \sum_{k=1}^{n} \partial_k v_k.$$

The divergence for a tensor $\sigma \in \mathbb{R}^{n \times n}$ is defined as:
$$\nabla \cdot \sigma = \Big(\sum_{j=1}^{n} \frac{\partial \sigma_{ij}}{\partial x_j}\Big)_{1 \le i \le n}.$$

The trace of a matrix $A \in \mathbb{R}^{n \times n}$ is defined as
$$tr(A) = \sum_{i=1}^{n} a_{ii}.$$

Definition 3.2 (Laplace operator)**.** *The Laplace operator of a two-times continuously differentiable scalar-valued function $u : \mathbb{R}^n \to \mathbb{R}$ is defined as*
$$\Delta u = \sum_{k=1}^{n} \partial_{kk} u.$$

Definition 3.3. *For a vector-valued function $u : \mathbb{R}^n \to \mathbb{R}^m$, we define the Laplace operator component-wise as*

$$\Delta u = \Delta \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^n \partial_{kk} u_1 \\ \vdots \\ \sum_{k=1}^n \partial_{kk} u_m \end{pmatrix}.$$

Let us also introduce the **cross product** of two vectors $u, v \in \mathbb{R}^3$:

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}.$$

With the help of the cross product, we can define the **rotation**:

$$\mathrm{rot} v = \nabla \times v = \begin{pmatrix} \partial_x \\ \partial_y \\ \partial_z \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \partial_y v_3 - \partial_z v_2 \\ \partial_z v_1 - \partial_x v_3 \\ \partial_x v_2 - \partial_y v_1 \end{pmatrix}.$$

3.7 Invariants of a matrix

The principal invariants of a matrix A are the coefficients of the characteristic polynomial $\det(A - \lambda I)$. A matrix $A \in \mathbb{R}^{3\times 3}$ has three principal invariants; namely $i_A = (i_1(A), i_2(A), i_3(A))$ with

$$\det\ (\lambda I - A) = \lambda^3 - i_1(A)\lambda^2 + i_2(A)\lambda - i_3(A).$$

Let $\lambda_1, \lambda_2, \lambda_3$ be the eigenvalues of A. Then we have

$$\begin{aligned} i_1(A) &= \mathrm{tr}\ (A) = \lambda_1 + \lambda_2 + \lambda_3, \\ i_2(A) &= \frac{1}{2}(\mathrm{tr}\ A)^2 - \mathrm{tr}\ (A)^2 = \lambda_1\lambda_1 + \lambda_1\lambda_3 + \lambda_2\lambda_3, \\ i_3(A) &= \det\ (A) = \lambda_1\lambda_2\lambda_3. \end{aligned}$$

Remark 3.4. *If two different matrices have the same principal invariants, they also have the same eigenvalues.*

Remark 3.5 (Cayley-Hamilton)**.** *Every second-order tensor (i.e., a matrix) satisfies its own characteristic equation:*

$$A^3 - i_1 A^2 + i_2 A - i_3 I = 0.$$

3.8 Vector spaces

Let $\mathbb{K} = \mathbb{R}$. In fact, $\mathbb{K} = \mathbb{C}$ would work as well and any general field. But we restrict our attention in the entire lecture notes to real numbers $\mathbb{R}$.

Definition 3.6 (Vector space)**.** *A **vector space** or **linear space** over a field $\mathbb{K}$ is a nonempty set X (later often denotes by V, U or also W). The space X contains elements $x_1, x_2, \ldots$ which are the so-called **vectors**. We define two algebraic operations:*

- *Vector addition: $x + y$ for $x, y \in X$.*
- *Multiplication of vectors with scalars: αx for $x \in X$ and $\alpha \in \mathbb{K}$.*

These operations satisfy the usual laws that they are commutative, associative, and satisfy distributive laws.

3.9 Normed spaces

Let X be a linear space. The mapping $||\cdot|| : X \to \mathbb{R}$ is a **norm** if

i)	$\|\|x\|\| \geq 0 \quad \forall x \in X$	(Positivity)
ii)	$\|\|x\|\| = 0 \Leftrightarrow x = 0$	(Definiteness)
iii)	$\|\|\alpha x\|\| = \|\alpha\| \, \|\|x\|\|, \quad \alpha \in \mathbb{K}$	(Homogeneity)
iv)	$\|\|x + y\|\| \leq \|\|x\|\| + \|\|y\|\|$	(Triangle inequality)

A space X is a normed space when the norm properties are satisfied. If condition ii) is not satisfied, the mapping is called a **semi-norm** and denoted by $|x|_X$ for $x \in X$.

Definition 3.7. *Let $\|\cdot\|$ be a norm on X. Then $\{X, \|\cdot\|\}$ is called a (real) normed space.*

Example 3.8. *We provide some examples:*

1. *$\mathbb{R}^n$ with the euclidian norm $\|x\| = (\sum_{i=1}^n x_i^2)^{1/2}$ is a normed space.*
2. *Let $\Omega := [a, b]$. The space of continuous functions $C(\Omega)$ endowed with the **maximum norm***

$$\|u\|_{C(\Omega)} = \max_{x \in \Omega} \|u(x)\|$$

   is a normed space.

3. *The space $\{C(\Omega), \|\cdot\|_{L^2}\}$ with*

$$\|u\|_{L^2} = \Big(\int_\Omega u(x)^2 \, dx \Big)^{1/2}$$

   is a normed space.

Definition 3.9. *Two norms are equivalent if converging sequences have the same limits.*

Proposition 3.10. *Two norms $||\cdot||_A, ||\cdot||_B$ on X are equivalent if and only if there exist two constants $C_1, C_2 > 0$ such that*

$$C_1||x||_A \leq ||x||_B \leq C_2||x||_A \quad \forall x \in X$$

The limits are the same.

Proof. See e.g., [69]. □

Remark 3.11. *This statements has indeed some immediate consequences. For instance, often convergence of an iterative scheme is proven with the help of the Banach fixed point scheme in which the contraction constant q must be smaller than 1; see for instance Section 8. It is important that not all norms may satisfy $q < 1$, but when different norms are equivalent and we pick one that satisfy $q < 1$, we can proof convergence.*

3.10 Linear mappings

Let $\{U, \|\cdot\|_U\}$ and $\{V, \|\cdot\|_V\}$ be normed spaces over $\mathbb{R}$.

Definition 3.12 (Linear mappings)**.** *A mapping $T : U \to V$ is called **linear** or **linear operator** when*

$$T(u) = T(au_1 + bu_2) = aT(u_1) + bT(u_2),$$

for $u = au_1 + bu_2$ and for $a, b \in \mathbb{R}$.

Example 3.13. *We discuss two examples:*

1. *Let $T(u) = \Delta u$. Then:*

$$T(au_1 + bu_2) = \Delta(au_1 + bu_2) = a\Delta u_1 + b\Delta u_2 = aT(u_1) + bT(u_2).$$

   Thus, T is linear.

2. Let $T(u) = (u \cdot \nabla)u$. Then:

$$T(au_1 + bu_2) = ((au_1 + bu_2) \cdot \nabla)(au_1 + bu_2) \neq a(u_1 \cdot \nabla)u_1 + b(u_2 \cdot \nabla)u_2 = aT(u_1) + bT(u_2).$$

Here, T is nonlinear.

Definition 3.14 (Linear functional)**.** *A mapping $T : U \to V$ with $V = \mathbb{R}$ is called **linear functional**.*

Definition 3.15. *A mapping $T : U \to V$ is called **continuous** when*

$$\lim_{n\to\infty} u_n = u \quad \Rightarrow \quad \lim_{n\to\infty} Tu_n = Tu.$$

Definition 3.16. *A linear operator $T : U \to V$ is called **bounded**, when the following estimate holds true:*

$$\|Tu\|_V \leq c\|u\|_U.$$

Theorem 3.17. *A linear operator is bounded if and only if it is continuous.*

Proof. See [69], Satz II.1.2. □

Definition 3.18. *Let $T : U \to V$ be a linear and continuous operator. The norm of T is defined as*

$$\|T\| = \sup_{\|u\|_U=1} \|Tu\|_V.$$

Since a linear T is bounded (when continuous) there exists $\|Tu\|_V \leq c\|u\|_U$ for all $u \in U$. The smallest number for c is $c = \|T\|$.

Definition 3.19. *The linear space of all linear and bounded operators from U to V is denoted by*

$$L(U, V).$$

The norm of $L(U, V)$ is the operator norm $\|T\|$.

Definition 3.20 (Dual space)**.** *The linear space of all linear, bounded functionals on U (see Def. 3.14) is the dual space, denoted by U^*, i.e.,*

$$U^* = L(U, \mathbb{R}).$$

For $f \in U^*$, the norm is given by:

$$\|f\|_{U^*} = \sup_{\|u\|_U=1} |f(u)|.$$

The dual space is always a Banach space; see again [69]. For more details on Banach spaces, we also refer to Section 6.8.4 of these lecture notes.

3.11 Little o and big O - the Landau symbols

Definition 3.21 (Landau symbols)**.** (i) *Let $g(n)$ a function with $g \to \infty$ for $n \to \infty$. Then $f \in O(g)$ if and only if when*

$$\limsup_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| < \infty$$

and $f \in o(g)$ if and only if

$$\lim_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| = 0.$$

(ii) *Let $g(h)$ a function with $g(h) \to 0$ for $h \to 0$. As before, we define $f \in O(g)$ and $f \in o(g)$:*

$$\limsup_{h\to 0} \left|\frac{f(h)}{g(h)}\right| < \infty \quad \Leftrightarrow \quad f \in O(g),$$

and

$$\lim_{h\to 0}\left|\frac{f(h)}{g(h)}\right| = 0 \quad \Leftrightarrow \quad f \in o(g).$$

(iii) *Specifically:*

$$\limsup_{h\to 0}\big|f(h)\big| < \infty \quad \Leftrightarrow \quad f \in O(1),$$

and

$$\lim_{h\to 0}\big|f(h)\big| = 0 \quad \Leftrightarrow \quad f \in o(1).$$

Often, the notation $f = O(g)$ is used rather than $f \in O(g)$ and similarely $f = o(g)$ rather than $f \in o(g)$.

Example 3.22. *Five examples:*

1. $\frac{1}{x} = \mathrm{o}(\frac{1}{x^2}) \quad (x \to 0)$,
2. $\frac{1}{x^2} = \mathrm{o}(\frac{1}{x}) \quad (|x| \to \infty)$,
3. $e^{-x} = \mathrm{o}(x^{-26}) \quad (x \to \infty)$.
4. *Let $\varepsilon \to 0$ and $h \to 0$. We write*

   $$h = o(\varepsilon)$$

   when

   $$\frac{h}{\varepsilon} \to 0 \quad \text{for } h \to 0, \quad \varepsilon \to 0,$$

   which means that h tends faster to 0 than ε.
5. *Let us assume that we have the error estimate (see sections on the numerical analysis)*

   $$\|y(t_n) - y_n\|_2 = O(k).$$

   Here the O notation means nothing else than

   $$\frac{\|y(t_n) - y_n\|_2}{k} \to C \quad \text{for } k \to 0.$$

Here the fraction converges to a constant C (and not necessarily 0!), which illustrates that $O(\cdot)$ convergence is weaker than $o(\cdot)$ convergence. On the other hand, this should not yield the wrong conclusion that $\|y(t_n) - y_n\|_2$ may not tend to zero. Since $k \to 0$, also $\|y(t_n) - y_n\|_2 \to 0$ must hold necessarily.

3.12 Taylor expansion

The Taylor series of a function $f \in C^\infty(\mathbb{R})$ developed at a point $a \neq x$ reads:

$$T(f(x)) = \sum_{j=0}^{\infty} \frac{f^{(j)}(a)}{j!}(x-a)^j.$$

3.13 Transformation of integrals: substitution rule / change of variables

One of the most important formulas in continuum mechanics and variational formulations is the substitution rule that allows to transform integrals from one domain to another.

In 1D it holds:

Proposition 3.23 (Substituion rule in 1D)**.** *Let $I = [a, b]$ be given. To transform this interval to a new interval, we use a mapping $T(I) = [\alpha, \beta]$ with $T(a) = \alpha$ and $T(b) = \beta$. If $T \in C^1$ (a continuously differentiable mapping) and monotonically increasing (i.e., $T' > 0$), we have the transformation rule:*

$$\int_\alpha^\beta f(y)\,dy = \int_{T(a)}^{T(b)} f(y)\,dy = \int_a^b f(T(x))\,T'(x)\,dx.$$

Proof. Any analysis book. Here Analysis 2, Rolf Rannacher, Heidelberg University [55]. □

Remark 3.24. *In case that $T' < 0$ the previous proposition still holds true, but with a negative sign:*

$$\int_\alpha^\beta f(y)\,dy = \int_{T(b)}^{T(a)} f(y)\,dy = -\int_{T(a)}^{T(b)} f(y)\,dy = \int_a^b f(T(x))\,(-T'(x))\,dx.$$

For both cases with $T' \neq 0$ the formula works and finally yields:

Theorem 3.25. *Let $I = [a,b]$ be given. To transform this interval to a new interval $[\alpha,\beta]$, we employ a mapping T. If $T \in C^1$ (a continuously differentiable mapping) and $T' \neq 0$, it holds:*

$$\int_{T(I)} f(y)\,dy := \int_\alpha^\beta f(y)\,dy = \int_a^b f(T(x))\,|T'(x)|\,dx =: \int_I f(T(x))\,|T'(x)|\,dx.$$

Proof. Any analysis book. Here Analysis 2, Rolf Rannacher, Heidelberg University [55]. □

Remark 3.26. *We observe the relation between the integration increments:*

$$dy = |T'(x)|\,dx.$$

Example 3.27. *Let T be a affin-linear transformation defined as*

$$T(x) = ax + b.$$

Then,

$$dy = |a|dx.$$

In higher dimensions, we have the following result of the substitution rule (also known as **change of variables** under the integral):

Theorem 3.28. *Let $\Omega \subset \mathbb{R}^n$ be an open, measurable, domain. Let the function $T : \Omega \to \mathbb{R}$ be of class C^1, one-to-one (injective) and Lipschitz-continuous. Then:*

- *The domain $\hat{\Omega} := T(\Omega)$ is measurable.*
- *The function $f(T(\cdot))|detT'(\cdot)| : \Omega \to \mathbb{R}$ is (Riemann)-integrable.*
- *For all measurable subdomains $M \subset \Omega$ it holds the substitution rule:*

$$\int_{T(M)} f(y)\,dy = \int_M f(T(x))|detT'(x)|\,dx,$$

and in particular as well for $M = \Omega$.

Proof. Any analysis book. See e.g., [55] or [45][Chapter 9]. □

Remark 3.29. *In continuum mechanics, T' is the so-called **deformation gradient** and $J := det(T')$, the **volume ratio**.*

3.14 Gauss-Green theorem / Divergence theorem

The Gauss-Green theorem or often known as **divergence theorem**, is one of the most useful formulas in continuum mechanics and numerical analysis.

Let $\Omega \subset \mathbb{R}^n$ an bounded, open domain and $\partial\Omega$ of class C^1.

Theorem 3.30 (Gauss-Green theorem / Divergence theorem)**.** *Suppose that $u := u(x) \in C^1(\bar{\Omega})$ with $x = (x_1,\ldots,x_n)$. Then:*

$$\int_\Omega u_{x_i}\,dx = \int_{\partial\Omega} u n_i\,ds, \quad \text{for } i = 1,\ldots,n.$$

In compact notation, we have

$$\int_\Omega div\,u\,dx = \int_{\partial\Omega} u \cdot n\,ds$$

for each vector field $u \in C^1(\bar{\Omega};\mathbb{R}^n)$.

Proof. The proof is nontrivial. See for example [45]. □

3.15 Integration by parts and Green's formulae

One of the most important formulae in applied mathematics, physics and continuum mechanics is **integration by parts**.

From the divergence Theorem 3.30, we obtain immediately:

Proposition 3.31 (Integration by parts)**.** *Let $u, v \in C^1(\bar{\Omega})$. Then:*

$$\int_\Omega u_{x_i} v \, dx = -\int_\Omega u v_{x_i} \, dx + \int_{\partial\Omega} uvn_i \, ds, \quad \text{for } i = 1, \ldots, n.$$

In compact notation:

$$\int_\Omega \nabla u v \, dx = -\int_\Omega u \nabla v \, dx + \int_{\partial\Omega} uvn \, ds.$$

Proof. Apply the divergence theorem to uv. Exercise. □

We obtain now some further results, which are very useful, but all are based directly on the integration by parts. For this reason, it is more important to know the divergence theorem and integration by parts formula.

Proposition 3.32 (Green's formulas)**.** *Let $u, v \in C^2(\bar{\Omega})$. Then:*

$$\int_\Omega \Delta u \, dx = \int_{\partial\Omega} \partial_n u \, ds,$$
$$\int_\Omega \nabla u \cdot \nabla v \, dx = -\int_\Omega \Delta u \, v \, dx + \int_{\partial\Omega} v \, \partial_n u \, ds.$$

Proof. Apply integration parts. □

Proposition 3.33 (Green's formula in 1D)**.** *Let $\Omega = (a, b)$. Let $u, v \in C^2(\bar{\Omega})$. Then:*

$$\int_\Omega u'(x) \cdot v'(x) \, dx = -\int_\Omega u''(x) \, v(x) \, dx + [u'(x)v(x)]_{x=a}^{x=b}$$

Proof. Apply integration parts. □

3.16 Chapter summary and outlook

Having the notation setup, we shall now begin by the specific topics promised in the title of these lecture notes.

4 An extended introduction

In this introduction (despite being already in Chapter 4), we first start with modeling on how the Laplace equation can be derived from physical fundamental principles.

4.1 Examples of differential equations

We start with some examples:

4.1.1 Laplace equation / Poisson problem

Laplace's equation (boundary value problem) . This equation has several applications, e.g., Fick's law of diffusion or heat conduction (temporal diffusion) or the deflection of a solid membrane:

Formulation 4.1 (Laplace problem / Poisson problem)**.** *Let Ω be an open set. The **Laplace equation** reads:*

$$-\Delta u = 0 \quad \text{in } \Omega.$$

*The **Poisson problem** reads:*

$$-\Delta u = f \quad \text{in } \Omega.$$

Definition 4.2. *A C^2 function (C^2 means two times continuously differentiable) that satisfies the Laplace equation is called **harmonic** function.*

The physical interpretation is as follows. Let u denote the density of some quantity, for instance concentration or temperature, in equilibrium. If G is any smooth region $G \subset \Omega$, the flux F of the quantity u through the boundary ∂G is zero:

$$\int_{\partial G} F \cdot n \, dx = 0. \tag{2}$$

Here F denotes the flux density and n the outer normal vector. Gauss' divergence theorem yields:

$$\int_{\partial G} F \cdot n \, dx = \int_G \nabla \cdot F \, dx = 0.$$

Since this integral relation holds for arbitrary G, we obtain

$$\nabla \cdot F = 0 \quad \text{in } \Omega. \tag{3}$$

Now we need a second assumption (or better a relation) between the flux and the quantity u. Such relations do often come from material properties. In many situations it is reasonable to assume that the flux F is proportional to the negative gradient $-\nabla u$ of the quantity u. This means that flow goes from regions with a higher concentration to lower concentration regions. For instance, the rate at which energy 'flows' (or diffuses) as heat from a warm body to a colder body is a function of the temperature difference. The larger the temperature difference, the larger the diffusion. We consequently obtain as further relation:

$$F = -\nabla u.$$

Plugging into the Equation (3) yields:

$$\nabla \cdot F = \nabla \cdot (-\nabla u) = -\nabla \cdot (\nabla u) = -\Delta u = 0.$$

This is the simplest derivation one can make. Adding more knowledge on the underlying material of the body, a material parameter $a > 0$ can be added:

$$\nabla \cdot F = \nabla \cdot (-a\nabla u) = -\nabla \cdot (a\nabla u) = -a\Delta u = 0.$$

And adding a nonconstant and spatially dependent material further yields:

$$\nabla \cdot F = \nabla \cdot (-a(x)\nabla u) = -\nabla \cdot (a(x)\nabla u) = 0.$$

In this last equation, we do not obtain any more the classical Laplace equation but a diffusion equation in divergence form.

4.1.2 Mass conservation / First-order hyperbolic problem

In this next example let us again consider some concentration, but now a time dependent situation, which brings us to a first-order hyperbolic equation. The application might be transport of a species/concentration in some fluid flow (e.g. water) or nutrient transport in blood flow. Let $\Omega \subset \mathbb{R}^n$ be an open domain, $x \in \Omega$ the spatial variable and t the time. Let $\rho(x,t)$ be the density of some quantity (e.g., concentration) and let $v(x,t)$ its velocity. Then the vector field

$$F = \rho v \quad \text{in } \mathbb{R}^n$$

denotes the flux of this quantity. Let G be a subset of Ω. Then we have as in the previous case (Equation (2)) the definition:

$$\int_{\partial G} F \cdot n \, dx.$$

But this time we do not assume the 'equilibrium state' but 'flow'. That is to say that the outward flow through the boundary ∂G must coincide with the temporal decrease of the quantity:

$$\int_{\partial G} F \cdot n \, ds = -\frac{d}{dt} \int_G \rho \, dx.$$

We then apply again Gauss' divergence theorem to the left hand side and bring the resulting term to the right hand side. We now encounter a principle difficulty that the domain G may depend on time and consequently integration and differentiation do not commute. Therefore, in a first step we need to transform the integrand of

$$\frac{d}{dt} \int_G \rho \, dx$$

onto a fixed referene configuration $\hat{G}$ in which we can insert the time derivative under the integral sign. Then we perform the calculation and transform lastly everything back to the physical domain G. Let the mapping between $\hat{G}$ and G be denoted by T. Then, it holds:

$$x \in G : x = T(\hat{x}, t), \quad \hat{x} \in \hat{G}.$$

Moreover, $dx = J(\hat{x},t)d\hat{x}$, where $J := \det(\nabla T)$. Using the substitution rule (change of variables) in higher dimensions (see Section 3.13) yields:

$$\frac{d}{dt} \int_G \rho(x,t) \, dx = \frac{d}{dt} \int_{\hat{G}} \rho(T(\hat{x},t),t) J(\hat{x},t) d\hat{x}.$$

We eliminated time dependence on the right hand side integral and thus differentiation and integration commute now:

$$\int_{\hat{G}} \frac{d}{dt} \Big(\rho(T(\hat{x},t),t) J(\hat{x},t) \Big) d\hat{x} = \int_{\hat{G}} \Big(\frac{d}{dt} \rho(T(\hat{x},t),t) \cdot J(\hat{x},t) + \rho(T(\hat{x},t),t) \frac{d}{dt} J(\hat{x},t) \Big) d\hat{x}$$

Here, $\frac{d}{dt}\rho(T(\hat{x},t),t)$ is the material time derivative of a spatial field (see e.g., [39]), which is <u>not</u> the same as the partial time derivative! In the last step, we need the Eulerian expansion formula

$$\frac{d}{dt} J = \nabla \cdot v \, J.$$

Then:

$$\begin{aligned}
&\int_{\hat{G}} \Big(\frac{d}{dt}\rho(T(\hat{x},t),t)\cdot J(\hat{x},t) + \rho(T(\hat{x},t),t)\frac{d}{dt}J(\hat{x},t)\Big) d\hat{x} \\
&= \int_{\hat{G}} \Big(\frac{d}{dt}\rho(T(\hat{x},t),t)\cdot J(\hat{x},t) + \rho(T(\hat{x},t),t)\nabla\cdot v\, J\Big) d\hat{x} \\
&= \int_{\hat{G}} \Big(\frac{d}{dt}\rho(T(\hat{x},t),t) + \rho(T(\hat{x},t),t)\nabla\cdot v\Big) J(\hat{x},t) d\hat{x} \\
&= \int_{G} \Big(\frac{d}{dt}\rho(x,t) + \rho(x,t)\nabla\cdot v\Big) dx \\
&= \int_{G} \Big(\partial_t\rho(x,t) + \nabla\rho(x,t)\, v + \rho(x,t)\nabla\cdot v\Big) dx \\
&= \int_{G} \Big(\partial_t\rho(x,t) + \nabla\cdot(\rho v)\Big) dx.
\end{aligned}$$

Collecting the previous calculations brings us to:

$$\int_G (\partial_t\rho + \nabla\cdot F)\, dx = \int_G (\partial_t\rho + \nabla\cdot(\rho v)\, dx, \quad \text{where } F = \rho v \text{ as introduce before.}$$

This is the so-called continuity equation (or mass conservation). Since G was arbitrary, we are allowed to write the strong form:

$$\partial_t\rho + \nabla\cdot(\rho v) = 0 \quad \text{in } \Omega. \tag{4}$$

If there are sources or sinks, denoted by f, inside Ω we obtain the more general formulation:

$$\partial_t\rho + \nabla\cdot(\rho v) = f \quad \text{in } \Omega.$$

On the other hand, various simplifications of Equation (4) can be made when certain requirements are fulfilled. For instance, if ρ is not spatially varying, we obtain:

$$\partial_t\rho + \rho\nabla\cdot v = 0 \quad \text{in } \Omega.$$

If furthermore ρ is constant in time, we obtain:

$$\nabla\cdot v = 0 \quad \text{in } \Omega.$$

This is now the mass conservation law that appears for instance in the incompressible Navier-Stokes equations, which are discussed a little bit later below.

In terms of the density ρ, we have shown in this section:

Theorem 4.3 (Reynolds' transport theorem)**.** *Let $\Phi := \Phi(x,t)$ be a smooth scalar field and Ω a (moving) domain. It holds:*

$$\frac{d}{dt}\int_\Omega \Phi\, dx = \int_{\partial\Omega} \Phi v\cdot n\, ds + \int_\Omega \frac{\partial\Phi}{\partial t}\, dx$$

*The first term on the right hand side represents the **rate of transport** (also known as the **outward normal flux**) of the quantity Φv across the boundary surface $\partial\Omega$. This contribution originates from the moving domain Ω. The second contribution is the **local time rate of change** of the spatial scalar field Φ. If the domain Ω does not move, the first term on the right hand side will of course vanish. Using Gauss' divergence theorem it holds furthermore:*

$$\frac{d}{dt}\int_\Omega \Phi\, dx = \int_\Omega \nabla\cdot(\Phi\, v)\, dx + \int_\Omega \frac{\partial\Phi}{\partial t}\, dx.$$

4.1.3 Elasticity (Lamé-Navier)

This example is already difficult because a system of nonlinear equations is considered:

Formulation 4.4. *Let $\widehat{\Omega}_s \subset \mathbb{R}^n, n = 3$ with the boundary $\partial\widehat{\Omega} := \widehat{\Gamma}_D \cup \widehat{\Gamma}_N$. Furthermore, let $I := (0, T]$ where $T > 0$ is the end time value. The equations for geometrically non-linear elastodynamics in the reference configuration $\widehat{\Omega}$ are given as follows: Find vector-valued displacements $\hat{u}_s := (\hat{u}_s^{(x)}, \hat{u}_s^{(y)}, \hat{u}_s^{(z)}) : \widehat{\Omega}_s \times I \to \mathbb{R}^n$ such that*

$$\begin{aligned}
\hat{\rho}_s \partial_t^2 \hat{u}_s - \widehat{\nabla} \cdot (\widehat{F}\widehat{\Sigma}) &= 0 \quad && \text{in } \widehat{\Omega}_s \times I, \\
\hat{u}_s &= 0 && \text{on } \widehat{\Gamma}_D \times I, \\
\widehat{F}\widehat{\Sigma} \cdot \hat{n}_s &= \hat{h}_s && \text{on } \widehat{\Gamma}_N \times I, \\
\hat{u}_s(0) &= \hat{u}_0 && \text{in } \widehat{\Omega}_s \times \{0\}, \\
\hat{v}_s(0) &= \hat{v}_0 && \text{in } \widehat{\Omega}_s \times \{0\}.
\end{aligned}$$

We deal with two types of boundary conditions: Dirichlet and Neumann conditions. Furthermore, two initial conditions on the displacements and the velocity are required. The constitutive law is given by the geometrically nonlinear tensors (see e.g., Ciarlet [19]):

$$\widehat{\Sigma} = \widehat{\Sigma}_s(\hat{u}_s) = 2\mu\widehat{E} + \lambda tr(\widehat{E})I, \quad \widehat{E} = \frac{1}{2}(\widehat{F}^T\widehat{F} - I). \tag{5}$$

Here, μ and λ are the Lamé coefficients for the solid. The solid density is denoted by $\hat{\rho}_s$ and the solid deformation gradient is $\widehat{F} = \hat{I} + \widehat{\nabla}\hat{u}_s$ where $\hat{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. Furthermore, $\hat{n}_s$ denotes the normal vector.

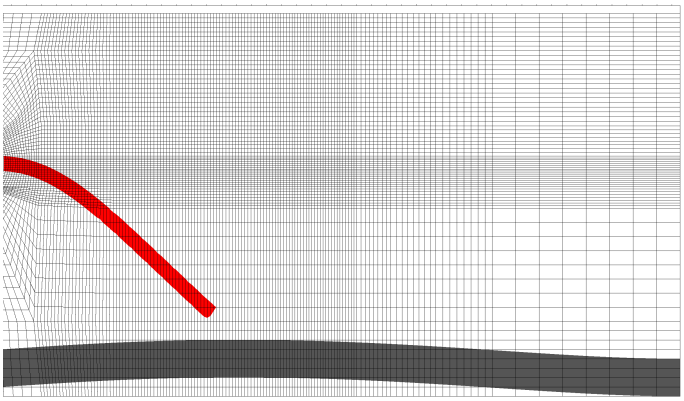

Figure 1: Prototype example of two deforming solids (elasticity). The underlying code is based on deal.II [6] www.dealii.org.

4.1.4 The incompressible, isothermal Navier-Stokes equations (fluid mechanics)

Flow equations in general are extremely important and have an incredible amount of possible applications such as for example water (fluids), blood flow, wind, weather forecast, aerodynamics:

Formulation 4.5. *Let $\Omega_f \subset \mathbb{R}^n, n = 3$. Furthermore, let the boundary be split into $\partial\Omega_f := \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_D \cup \Gamma_i$. The isothermal, incompressible (non-linear) Navier-Stokes equations read: Find vector-valued velocities $v_f : \Omega_f \times I \to \mathbb{R}^n$ and a scalar-valued pressure $p_f : \Omega_f \times I \to \mathbb{R}$ such that*

$$\begin{aligned} \rho_f \partial_t v_f + \rho_f v_f \cdot \nabla v_f - \nabla \cdot \sigma_f(v_f, p_f) &= 0 \quad && \text{in } \Omega_f \times I, \\ \nabla \cdot v_f &= 0 && \text{in } \Omega_f \times I, \\ v_f^D &= v_{in} && \text{on } \Gamma_{in} \times I, \\ v_f &= 0 && \text{on } \Gamma_D \times I, \\ -p_f n_f + \rho_f \nu_f \nabla v_f \cdot n_f &= 0 && \text{on } \Gamma_{out} \times I, \\ v_f &= h_f && \text{on } \Gamma_i \times I, \\ v_f(0) &= v_0 && \text{in } \Omega_f \times \{t = 0\}, \end{aligned} \tag{6}$$

where the (symmetric) Cauchy stress is given by

$$\sigma_f(v_f, p_f) := -p_f I + \rho_f \nu_f (\nabla v_f + \nabla v_f^T),$$

with the density ρ_f and the kinematic viscosity ν_f. The normal vector is denoted by n_f.

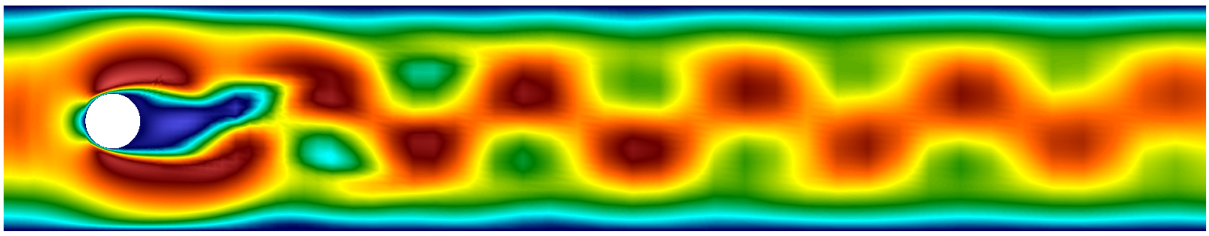

Figure 2: Prototype example of a fluid mechanics problem (isothermal, incompressible Navier-Stokes equations): the famous Karman vortex street. The setting is based on the benchmark setting [62] and the code can be found in NonStat Example 1 in [30] www.dopelib.net.

Remark 4.6. *The two Formulations 4.4 and 4.5 are very important in many applications and their coupling results in fluid-structure interaction. Here we notice that fluid flows are usually modeled in Eulerian coordinates and solid deformations in Lagrangian coordinates. In the case of small displacements, the two coordinate systems can be identified, i.e., $\widehat{\Omega} \simeq \Omega$. This is the reason why in many basic books - in particular basics of PDE theory or basics of numerical algorithms - the 'hat' notation (or similar notation to distinguish coordinate systems) is not used.*

Exercise 1. *Recapitulate (in case you have had classes on continuum mechanics) the differences between Lagrangian and Eulerian coordinates.*

Remark 4.7. *In Section 7.14, we study in more detail the stationary Navier-Stokes equations (NSE) in terms of a well-acknowledged benchmark problem.*

4.2 Three important PDEs

From the previous considerations, we can extract three important types of PDEs. But we refrain from giving the impression that all differential equations can be classified and then a recipe for solving them applies. This is definitely not true. However, in particular for PDEs, we are often faced with three outstanding types of equations and often 'new' equations can be related or simplified to these three types.

They read:

- Poisson problem: $-\Delta u = f$ is elliptic: second order in space and no time dependence.
- Heat equation: $\partial_t u - \Delta u = f$ is parabolic: second order in space and first order in time.
- Wave equation: $\partial_t^2 u - \Delta u = f$ is hyperbolic: second order in space and second order in time.

All these three equations have in common that their spatial order is two (the highest derivative with respect to spatial variables that occurs in the equation). With regard to time, there are differences: the heat equation is of first order whereas the wave equation is second order in time.

Even so that we define separate types of PDEs, in many processes there is a mixture of these classes in one single PDE - depending on the size of certain parameters. For instance, the Navier-Stokes equations for modeling fluid flow, vary between parabolic and hyperbolic type depending on the Reynold's number $Re \sim \nu_f^{-1}$ (respectively a characteristic velocity and a characteristic length). In this case the coefficients a_{ij} of the operator L are non-constant and change with respect to space and time.

Let us now consider these three types in a bit more detail.

4.2.1 Elliptic equations and Poisson problem/Laplace equation

The Poisson equation is a boundary-value problem and will be derived from the general definition (very similar to the form before). Elliptic problems are second-order in space and have no time dependence.

Formulation 4.8. *Let $f : \Omega \to \mathbb{R}$ be given. Furthermore, Ω is an open, bounded set of $\mathbb{R}^n$. We seek the unknown function $u : \bar{\Omega} \to \mathbb{R}$ such that*

$$Lu = f \quad \text{in } \Omega, \tag{7}$$

$$u = 0 \quad \text{on } \partial\Omega. \tag{8}$$

Here, the linear second-order differential operator is defined by:

$$Lu := -\sum_{i,j=1}^{n} \partial_{x_j}(a_{ij}(x)\partial_{x_i} u) + \sum_{i=1}^{n} b_i(x) u \partial_{x_i} + c(x)u, \quad u = u(x), \tag{9}$$

with the symmetry assumption $a_{ij} = a_{ji}$ and given coefficient functions a_{ij}, b_i, c. Alternatively we often use the compact notation with derivatives defined in terms of the nabla-operator:

$$Lu := -\nabla \cdot (a\nabla u) + b\nabla u + cu.$$

Finally we notice that the boundary condition (8) *is called* ***homogeneous Dirichlet condition****.*

Remark 4.9. *Other possible boundary conditions are introduced in Section 4.3.*

Remark 4.10. *If the coefficient functions a, b, c also depend on the solution u we obtain a nonlinear PDE.*

The operator L generates a quadratic form Σ, which is defined as

$$\Sigma(\xi) := \sum_{i,j=1}^{n} a_{ij}(x)\xi_i\xi_j.$$

The properties of the form Σ (and consequently the classification of the underlying PDE) depends on the eigenvalues of the matrix A:

$$A = (a_{ij})_{ij=1}^{n}.$$

At the a given point $x \in \Omega$, the differential operator L is said to be elliptic if all eigenvalues of A are non-zero (the matrix A is positive definite) and have the same sign: Equivalently one can say:

Definition 4.11. *A PDE operator L is (uniformly) elliptic if there exists a constant $\theta > 0$ such that*

$$\sum_{i,j=1}^{n} a_{ij}(x)\xi_i\xi_j \geq \theta|\xi|^2,$$

for a.e. $x \in \Omega$ and all $\xi \in \mathbb{R}^n$.

Remark 4.12. *For parabolic PDEs one eigenvalue of A is zero whereas the others have the same sign. Finally, a hyperbolic equation has one eigenvalue that has a different sign (negative) than all the others (positive). We come back to some examples in Section 4.5.*

Formulation 4.13 (Poisson problem)**.** *Setting in Formulation 4.8, $a_{ij} = \delta_{ij}$ and $b_i = 0$ and $c = 0$, we obtain the Laplace operator. Let $f : \Omega \to \mathbb{R}$ be given. Furthermore, Ω is an open, bounded set of $\mathbb{R}^n$. We seek the unknown function $u : \bar{\Omega} \to \mathbb{R}$ such that*

$$Lu = f \quad \text{in } \Omega, \tag{10}$$
$$u = 0 \quad \text{on } \partial\Omega. \tag{11}$$

Here, the linear second-order differential operator is defined by:

$$Lu := -\nabla \cdot (\nabla u) = -\Delta u.$$

In the following we discuss some characteristics of the solution of the Laplace problem, which can be generalized to general elliptic problems. Solutions to Poisson's equation attain their maximum on the boundary and not in the interior of the domain. This property will also carry over to the discrete problem and can be used to check if the numerical solution is correct.

Theorem 4.14 (Strong maximum principle for the Laplace problem)**.** *Suppose $u \in C^2(\Omega) \cap C(\bar{\Omega})$ is a solution of Laplace equation. Then*

$$\max_{\bar{\Omega}} u = \max_{\partial\Omega} u.$$

Moreover, if Ω is connected and there exists a point $y \in \Omega$ in which

$$u(y) = \max_{\bar{\Omega}} u,$$

then u is constant within Ω. The same holds for $-u$, but then for minima.

Proof. See Evans [25]. □

Remark 4.15 (Maximum principle in practice)**.** *An illustration of the maximum principle in practice for a 1D setting is provided in Figure 4, where the maximum values are clearly obtained on the two boundary points of the domain.*

Corollary 4.16. *For finding $u \in C^2(\Omega) \cap C(\bar{\Omega})$, such that*

$$\Delta u = 0 \quad \text{in } \Omega, u = g \quad \text{on } \partial\Omega,$$

with $g \geq 0$, it holds that

$$u > 0 \quad \text{everywhere in } \Omega \quad \text{if} \quad g > 0 \text{ on some part of } \partial\Omega. \tag{12}$$

From the maximum principle we obtain immediately uniqueness of a solution:

Theorem 4.17 (Uniqueness of the Laplace problem)**.** *Let $g \in C(\partial\Omega)$ and $f \in C(\Omega)$. Then there exists at most one solution $u \in C^2(\Omega) \cap C(\bar{\Omega})$ of the boundary-value problem:*

$$-\Delta u = f \quad \text{in } \Omega, \tag{13}$$
$$u = g \quad \text{on } \partial\Omega. \tag{14}$$

4.2.2 Parabolic equations and heat equation

We now study parabolic problems, which contain as the most famous example the heat equation. Parabolic problems are second order in space and first order in time.

In this section, we assume $\Omega \subset \mathbb{R}^n$ to be an open and bounded set. The time interval is given by $I := (0, T]$ for some fixed end time value $T > 0$.

We consider the initial/boundary-value problem (IBVP):

Formulation 4.18. *Let $f : \Omega \times I \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ be given. We seek the unknown function $u : \bar{\Omega} \times I \to \mathbb{R}$ such that*[1]

$$\partial_t u + Lu = f \quad \text{in } \Omega \times I, \tag{15}$$
$$u = 0 \quad \text{on } \partial\Omega \times [0, T], \tag{16}$$
$$u = g \quad \text{on } \Omega \times \{t = 0\}. \tag{17}$$

Here, the linear second-order differential operator is defined by:

$$Lu := -\sum_{i,j=1}^{n} \partial_{x_j}(a_{ij}(x,t)\partial_{x_i} u) + \sum_{i=1}^{n} b_i(x,t)\partial_{x_i} u + c(x,t)u, \quad u = u(x,t)$$

for given (possibly spatial and time-dependent) coefficient functions a_{ij}, b_i, c.

We have the following:

Definition 4.19. *The PDE operator $\partial_t + L$ is (uniformly) parabolic if there exists a constant $\theta > 0$ such that*

$$\sum_{i,j=1}^{n} a_{ij}(x,t)\xi_i\xi_j \geq \theta|\xi|^2,$$

for all $(x,t) \in \Omega \times I$ and all $\xi \in \mathbb{R}^n$. In particular this operator is elliptic in the spatial variable x for each fixed time $0 \leq t \leq T$.

Formulation 4.20 (Heat equation)**.** *Setting in Formulation 4.18, $a_{ij} = \delta_{ij}$ and $b_i = 0$ and $c = 0$, we obtain the Laplace operator. Let $f : \Omega \to \mathbb{R}$ be given. Furthermore, Ω is an open, bounded set of $\mathbb{R}^n$. We seek the unknown function $u : \bar{\Omega} \to \mathbb{R}$ such that*

$$\partial_t u + Lu = f \quad \text{in } \Omega \times I, \tag{18}$$
$$u = 0 \quad \text{on } \partial\Omega \times [0, T], \tag{19}$$
$$u = g \quad \text{on } \Omega \times \{t = 0\}. \tag{20}$$

Here, the linear second-order differential operator is defined by:

$$Lu := -\nabla \cdot (\nabla u) = -\Delta u.$$

The heat equation has an infinite propagation speed for disturbances. If the initial temperature is non-negative and is positive somewhere, the temperature at any later time is everywhere positive. For the heat equation, a very similar maximum principle as for elliptic problems holds true. This principle is extended to the parabolic boundary. We deal with a diffusions problem and flow points into the direction of the negative gradient. For $f = 0$ the maximum temperature u is obtained on the parabolic boundary:

$$\Gamma = \Big([0, T] \times \partial\Omega\Big) \cup \Big(\{0\} \times \Omega\Big)$$

in the space-time cylinder $Q = (0, T) \times \Omega$. In other words, the maximum is taken at the initial time step or on the boundary of the spatial domain.

[1] For the heat equation, we should have better used the notation T, the temperature, for the variable rather than u. But to stay consistant with the entire notation, we still use u.

Theorem 4.21 (Strong maximum principle for the heat equation)**.** *Suppose $u \in C_1^2(\Omega \times I) \cap C(\bar{\Omega} \times I)$ is a solution of the heat equation:*

$$\partial_t u = \Delta u \quad \text{in } \Omega \times I = (0, T).$$

Then

$$\max_{\bar{\Omega} \times I} u = \max_{\partial\Omega} u.$$

In other words: the maximum u (or the minimum) is obtained either at $t = 0$ or on $[0, T] \times \partial\Omega$. A numerical test demonstrating the maximum principle is provided in Section 10.3.9.

Remark 4.22. *As for elliptic problems, the maximum principle can be used to proof uniqueness for the parabolic case.*

4.2.3 Hyperbolic equations and the wave equation

In this section, we shall study hyperbolic problems, which are natural generalizations of the wave equation. As for parabolic problems, let $\Omega \subset \mathbb{R}^n$ to be an open and bounded set. The time interval is given by $I := (0, T]$ for some fixed end time value $T > 0$.

We consider the initial/boundary-value problem:

Formulation 4.23. *Let $f : \Omega \times I \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ be given. We seek the unknown function $u : \bar{\Omega} \times I \to \mathbb{R}$ such that*

$$\partial_t^2 u + Lu = f \quad \text{in } \Omega \times I, \tag{21}$$
$$u = 0 \quad \text{on } \partial\Omega \times [0, T], \tag{22}$$
$$u = g \quad \text{on } \Omega \times \{t = 0\}, \tag{23}$$
$$\partial_t u = h \quad \text{on } \Omega \times \{t = 0\}. \tag{24}$$

In the last line, $\partial_t u = v$ can be identified as the velocity. Furthermore, the linear second-order differential operator is defined by:

$$Lu := -\sum_{i,j=1}^{n} \partial_{x_j}(a_{ij}(x,t)\partial_{x_i} u) + \sum_{i=1}^{n} b_i(x,t)\partial_{x_i} u + c(x,t)u, \quad u = u(x,t)$$

for given (possibly spatial and time-dependent) coefficient functions a_{ij}, b_i, c.

Remark 4.24. *The wave equation is often written in terms of a first-order system in which the velocity is introduced and a second-order time derivative is avoided. Then the previous equation reads: Find $u : \bar{\Omega} \times I \to \mathbb{R}$ and $v : \bar{\Omega} \times I \to \mathbb{R}$ such that*

$$\partial_t v + Lu = f \quad \text{in } \Omega \times I, \tag{25}$$
$$\partial_t u = v \quad \text{in } \Omega \times I, \tag{26}$$
$$u = 0 \quad \text{on } \partial\Omega \times [0, T], \tag{27}$$
$$u = g \quad \text{on } \Omega \times \{t = 0\}, \tag{28}$$
$$v = h \quad \text{on } \Omega \times \{t = 0\}. \tag{29}$$

We have the following:

Definition 4.25. *The PDE operator $\partial_t^2 + L$ is (uniformly) hyperbolic if there exists a constant $\theta > 0$ such that*

$$\sum_{i,j=1}^{n} a_{ij}(x,t)\xi_i\xi_j \geq \theta|\xi|^2,$$

for all $(x, t) \in \Omega \times I$ and all $\xi \in \mathbb{R}^n$. In particular this operator is elliptic in the spatial variable x for each fixed time $0 \leq t \leq T$.

And as before, setting the coefficient functions to trivial values, we obtain the original wave equation. In contrast to parabolic problems, a strong maximum principle does not hold for hyperbolic equations. And consequently, the propagation speed is finite. This means that a disturbance at some point $x \in \Omega$ at some time $t \in I$ is not immediately transported to any point $x \in \Omega$ at any time $t \in I$.

4.3 Boundary conditions and initial data

As seen in the previous sections, all PDEs are complemented with boundary conditions and in time-dependent cases, also with initial conditions. Actually, these are crucial ingredients for solving differential equations. Often, one has the (wrong) impression that only the PDE itself is of importance. But what happens on the boundaries finally yields the 'result' of the computation. And this holds true for analytical (classical) as well as computational solutions.

In Section 4.2, for simplicity, we have mainly dealt with homogeneous Dirichlet conditions of the form $u = 0$ on the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_R$. In general three types of conditions are of importance:

- Dirichlet (or essential) boundary conditions: $u = g_D$ on $\partial\Omega_D$; when $g_D = 0$ we say 'homogeneous' boundary condition.
- Neumann (or natural) boundary conditions: $\partial_n u = g_N$ on $\partial\Omega_N$; when $g_N = 0$ we say 'homogeneous' boundary condition.
- Robin (third type) boundary condition: $au + b\partial_n u = g_R$ on $\partial\Omega_R$; when $g_R = 0$ we say 'homogeneous' boundary condition.

On each boundary section, only one of the three conditions can be described. In particular the natural conditions generalize when dealing with more general operators L. Here in this section, we have assumed $L := -\Delta u$.

Remark 4.26. *Going from differential equations to variational formulations, Dirichlet conditions are explicitely built into the function space, for this reason they are called* ***essential boundary conditions****. Neumann conditions appear explicitly through integration by parts; for this reason they are called* ***natural boundary conditions****.*

Example 4.27. *Let us compute the heat distribution in our room b302. The room volume is Ω. The window wall is a Dirichlet boundary $\partial_D\Omega$ and the remaining walls are Neumann boundaries $\partial_N\Omega$. Let K be the air viscosity.*

We consider the heat equation: Find $T : \Omega \times I \to \mathbb{R}$ such that

$$\begin{aligned} \partial_t T + (v \cdot \nabla)T - \nabla \cdot (K\nabla T) &= f && \text{in } \Omega \times I, \\ T &= 18°C && \text{on } \partial_D\Omega \times I, \\ K\nabla T \cdot n &= 0 && \text{on } \partial_N\Omega \times I, \\ T(0) &= 15°C && \text{in } \Omega \times \{0\}. \end{aligned}$$

The homogeneous Neumann condition means that there is no heat exchange on the respective walls (thus neighboring rooms will have the same room temperature on the respective walls). The nonhomogeneous Dirichlet condition states that there is a given temperature of $18°C$, which is constant in time and space (but this condition may be also non-constant in time and space). Possible heaters in the room can be modeled via the right hand side f. The vector $v : \Omega \to \mathbb{R}^3$ denotes a given flow field yielding a convection of the heat, for instance wind. We can assume $v \approx 0$. Then the above equation is reduced to the original heat equation: $\partial_t T - \nabla \cdot (K\nabla T) = f$.

4.4 The general definition of a differential equation

After the previous examples, let us precise the definition of a differential equation. In order to allow for largest possible generality we first need to introduce some notation. Common is to use the multiindex notation as introduced in Section 3.5.

Definition 4.28 (Evans [25])**.** *Let $\Omega \subset \mathbb{R}^n$ be open. Here, n denotes the total dimension including time. Furthermore, let $k \geq 1$ an integer that denotes the order of the differential equation. Then, a differential equation can be expressed as: Find $u : \Omega \to \mathbb{R}$ such that*

$$F(D^k u, D^{k-1}u, \ldots, D^2 u, Du, u, x) = 0 \quad x \in \Omega,$$

where

$$F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \ldots \times \mathbb{R}^{n^2} \times \mathbb{R}^n \times \mathbb{R} \times \Omega \to \mathbb{R}.$$

Example 4.29. *We provide some examples. Let us assume the spatial dimension to be* 2 *and the temporal dimension is* 1*. That is for a time-dependent ODE (ordinary differential equation) problem* $n = 1$ *and for time-dependent PDE cases,* $n = 2 + 1 = 3$*, and for stationary PDE examples* $n = 2$*.*

1. *ODE model problem:* $F(Du, u) := u' - au = 0$ *where* $F : \mathbb{R}^1 \times \mathbb{R} \to \mathbb{R}$*. Here* $k = 1$*.*

2. *Laplace operator:* $F(D^2u) := -\Delta u = 0$ *where* $F : \mathbb{R}^4 \to \mathbb{R}$*. That is* $k = 2$ *and lower derivatives of order* 1 *and* 0 *do not exist. We notice that in the general form it holds*

$$D^2u = \begin{pmatrix} \partial_{xx}u & \partial_{yx}u \\ \partial_{xy}u & \partial_{yy}u \end{pmatrix}$$

   and for the Laplacian:

$$D^2u = \begin{pmatrix} \partial_{xx}u & 0 \\ 0 & \partial_{yy}u \end{pmatrix}$$

3. *Heat equation:* $F(D^2u, Du) = \partial_t u - \Delta u = 0$ *where* $F : \mathbb{R}^9 \times \mathbb{R}^3 \to \mathbb{R}$*. Here* $k = 2$ *is the same as before, but a lower-order derivative of order* 1 *in form of the time derivative does exist. We provide again details on* D^2u*:*

$$D^2u = \begin{pmatrix} \partial_{tt}u & \partial_{xt}u & \partial_{yt}u \\ \partial_{tx}u & \partial_{xx}u & \partial_{yx}u \\ \partial_{ty}u & \partial_{xy}u & \partial_{yy}u \end{pmatrix}, \qquad Du = \begin{pmatrix} \partial_t u \\ \partial_x u \\ \partial_y u \end{pmatrix}$$

   and for the heat equation:

$$D^2u = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \partial_{xx}u & 0 \\ 0 & 0 & \partial_{yy}u \end{pmatrix}, \qquad Du = \begin{pmatrix} \partial_t u \\ 0 \\ 0 \end{pmatrix}$$

4. *Wave equation:* $F(D^2u) = \partial_t^2 u - \Delta u = 0$ *where* $F : \mathbb{R}^9 \to \mathbb{R}$*. Here it holds specifically*

$$D^2u = \begin{pmatrix} \partial_{tt}u & 0 & 0 \\ 0 & \partial_{xx}u & 0 \\ 0 & 0 & \partial_{yy}u \end{pmatrix}.$$

We finally, provide classifications of **linear** and **nonlinear** differential equations. Simply speaking: each differential equation, which is not linear is called nonlinear. However, in the nonlinear case a further refined classification can be undertaken.

Definition 4.30 (Evans[25])**.** *Differential equations are divided into linear and nonlinear classes as follows:*

1. *A differential equation is called* ***linear*** *if it is of the form:*

$$\sum_{|\alpha| \le k} a_\alpha(x) D^\alpha u - f(x) = 0.$$

2. *A differential equation is called* ***semi-linear*** *if it is of the form:*

$$\sum_{|\alpha| = k} a_\alpha(x) D^\alpha u + a_0(D^{k-1}u, \dots, D^2u, Du, u, x) = 0.$$

   Here, nonlinearities may appear in all terms of order $|\alpha| < k$*, but the highest order* $|\alpha| = k$ *is fully linear.*

3. *A differential equation is called* ***quasi-linear*** *if it is of the form:*

$$\sum_{|\alpha| = k} a_\alpha(D^{k-1}u, \dots, D^2u, Du, u, x) D^\alpha u + a_0(D^{k-1}u, \dots, D^2u, Du, u, x) = 0.$$

   Here, full nonlinearities may appear in all terms of order $|\alpha| < k$*, in the highest order* $|\alpha| = k$*, nonlinear terms appear up to order* $|\alpha| < k$*.*

4. *If none of the previous cases applies, a differential equation is called (fully)* ***nonlinear****.*

How can we now proof in practice, whether a PDE is linear or nonlinear? The simplest way is to go term by term and check the linearity condition from Definition 3.12.

Example 4.31. *We provide again some examples:*

1. *All differential equations from Example 4.29 are* ***linear****. Check term by term Definition 3.12!*

2. *Euler equations (fluid dynamics, special case of Navier-Stokes with zero viscosity). Here,* $n = 2+1+1 = 4$ *(in two spatial dimensions) in which we have* 2 *velocity components,* 1 *pressure variable, and* 1 *temporal variable. Let us consider the momentum part of the Euler equations:*

$$\partial_t v_f + v_f \cdot \nabla v_f + \nabla p_f = f.$$

   Here the highest order is $k = 1$ *(in the temporal variable as well as the spatial variable). But in front of the spatial derivative, we multiply with the zero-order term* v_f*. Consequently, the Euler equations are* ***quasi-linear*** *because a lower-order term of the solution variable is multiplied with the highest derivative.*

3. *Navier-Stokes momentum equation:*

$$\partial_t v_f - \rho_f \nu_f \Delta v_f + v_f \cdot \nabla v_f + \nabla p_f = f.$$

   Here $k = 2$*. But the coefficients in front of the highest order term, the Laplacian, do not depend on* v_f*. Consequently, the Navier-Stokes equations are neither fully nonlinear nor quasi-linear. Well, we have again the nonlinearity of the first order convection term* $v_f \cdot \nabla v_f$*. Thus the Navier-Stokes equations are* ***semi-linear****.*

4. *A fully* ***nonlinear*** *situation would be:*

$$\partial_t v_f - \rho_f \nu_f (\Delta v_f)^2 + v_f \cdot \nabla v_f + \nabla p_f = f.$$

Example 4.32 (Development of numerical methods for nonlinear equations)**.** *In case you are given a nonlinear IBVP (initial-boundary value problem) and want to start developing numerical methods for this specific PDE, it is often much easier to start with appropriate simplifications in order to build and analyze step-by-step your final method. Let us say you are given the nonlinear time-dependent PDE*

$$\nabla u \partial_t^2 u + u \cdot \nabla u - (\Delta u)^2 = f$$

Then, you could tackle the problem as follows:

1. *Consider the linear equation:*

$$\partial_t^2 u - \Delta u = f$$

   which is nothing else than the wave equation.

2. *Add a slight nonlinearity to make the problem semi-linear:*

$$\partial_t^2 u + u \cdot \nabla u - \Delta u = f$$

3. *Add* ∇u *such that the problem becomes quasi-linear:*

$$\nabla u \partial_t^2 u + u \cdot \nabla u - \Delta u = f$$

4. *Make the problem fully nonlinear by considering* $(\Delta u)^2$*:*

$$\nabla u \partial_t^2 u + u \cdot \nabla u - (\Delta u)^2 = f.$$

In each step, make sure that the corresponding numerical solution makes sense and that your developments so far are correct. If yes, proceed to the next step.

Remark 4.33. *The contrary to the previous example works as well and should be kept in mind! If you have implemented the full nonlinear PDE and recognize a difficulty, you can reduce the PDE term by term and make it step by step simpler. The same procedure holds true for the mathematical analysis.*

Exercise 2. *Classify the following PDEs into linear and nonlinear PDEs (including a short justification):*

$$\begin{aligned} -\nabla \cdot (|\nabla u|^{p-2}\nabla u) &= f \\ det(\nabla^2 u) &= f \\ \partial_t^2 u + 2a\partial_t u - \partial_{xx} u &= 0, \quad a > 0 \\ \partial_t u + u\partial_x u &= 0 \\ \partial_{tt} - \Delta u + m^2 u &= 0, \quad m > 0. \end{aligned}$$

How can we linearize nonlinear PDEs? Which terms need to be simplified such that we obtain a linear PDE?

4.5 Further remarks on the classification into three important types

The previous general definitions into *elliptic, parabolic, and hyperbolic* types can be made clearer when we simplify the situation of (9) in $\mathbb{R}^2$:

$$Lu := a_{11}\partial_x^2 u + 2a_{12}\partial_x\partial_y u + a_{22}\partial_y^2 u + a_1\partial_x u + a_2\partial_y u + au = f$$

with, for the moment, constant coefficients a_{ij}. Moreover, $_{11}, a_{12}, a_{22}$ should not be zero simultaneously. The main part of L can be represented by the quadratic form:

$$q(x, y) = a_{11}x^2 + 2a_{12}xy + a_{22}y^2.$$

Solving $q(x, y) = 0$ yields conic sections in the xy plane. Specifically, we have a relation to geometry:

- Circle, ellipse: $x^2 + y^2 = 1$;
- Parabola: $y = x^2$;
- Hyperbola: $x^2 - y^2 = 1$.

In order to apply this idea to L we write the main part of L in matrix-vector form:

$$L_0 := a_{11}\partial_x^2 + 2a_{12}\partial_x\partial_y + a_{22}\partial_y^2 = \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}^T \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix}.$$

The type of the PDE can now be dermined by investigating the properties of the coefficient matrix:

Proposition 4.34. *The differential operator L can be brought through a linear transformation to one of the following forms:*

1. Elliptic, *if* $a_{12}^2 < a_{11}a_{22}$.
   $\to \partial_x^2 u + \partial_y^2 u + \ldots = 0 \quad \overset{example}{\to} \Delta u = 0$
2. Hyperbolic, *if* $a_{12}^2 > a_{11}a_{22}$.
   $\to \partial_x^2 u - \partial_y^2 u + \ldots = 0 \quad \overset{example}{\to} \partial_t^2 u - \Delta u = 0$
3. Parabolic, *if* $a_{12}^2 = a_{11}a_{22}$.
   $\to \partial_x^2 u + \ldots = 0 \quad \overset{example}{\to} \partial_t u - \Delta u = 0$

The ... represent terms of lower order.

Remark 4.35. *Recall that this corresponds exactly to our previous definitions of elliptic, parabolic, and hyperbolic cases.*

Proof. We proof the result for the elliptic case. The other cases will follow in a similar manner. Without loss of generality, we assume $a_{11} = 1, a_1 = a_2 = a_0 = 0$. Completing the square yields

$$\begin{aligned} 0 &= (\partial_x + a_{12}\partial_y)^2 u + (a_{22} - a_{12}^2)\partial_y^2 u \\ &= (\partial_x^2 + 2a_{12}\partial_x\partial_y + a_{12}^2\partial_y^2)u + (a_{22} - a_{12}^2)\partial_y^2 u \end{aligned} \tag{30}$$

under the assumption that $a_{12}^2 < a_{11} \cdot a_{22}$. Set $b := \sqrt{a_{22} - a_{12}^2} > 0$.

We now do a coordinate transformation

$$x = \xi, \quad y = a_{12} \cdot \xi + b \cdot \eta \quad \text{(linear in } \xi \text{ and } \eta\text{)}$$

Recall how derivatives transform under coordinate transformations. We define

$$v(\xi, \eta) = u(x, y)$$

and interprete x and y as functions:

$$x(\xi, \eta), y(\xi, \eta) \to v(\xi, \eta) = u(x(\xi, \eta), y(\xi, \eta)).$$

We compute $(\partial_\xi v, \partial_\eta v)$:

$$\begin{aligned} (\partial_\xi v, \partial_\eta v) &= (\partial_x u, \partial_y u)\begin{pmatrix} \partial_\xi x & \partial_\eta x \\ \partial_\xi y & \partial_\eta y \end{pmatrix} \\ &= (\partial_x u, \partial_y u)\begin{pmatrix} 1 & 0 \\ a_{12} & b \end{pmatrix} \\ &= (\partial_x u + a_{12}\partial_y u, 0 \cdot \partial_x u + b \cdot \partial_y u). \end{aligned}$$

This yields: $\partial_\xi = \partial_x + a_{12}\partial_y$ and $\partial_\eta = b \cdot \partial_y$.

We plug the result into (30). The resulting equation is:

$$\partial_\xi^2 + \partial_\eta^2 = 0,$$

i.e., the Laplace equation.

□

Remark 4.36. *This definition also works for L that depends on space and time. Just replace the variable y through the time t.*

Example 4.37. *Classify the following equations:*

1. $\partial_{xx} u - 5\partial_{xy} u = 0$
2. $4\ \partial_{xx} u - 12\partial_{xy} u + 9\partial_{yy} u + \partial_y u = 0$
3. $4\ \partial_{xx} u + 6\partial_{xy} u + 9\partial_{yy} u = 0$

As previously discussed we compute the discriminant $D = a_{12}^2 - a_{11}a_{22}$ in order to see which case we have on hand:

1. $D = (-\frac{5}{2})^2 - 1 \cdot 0 = \frac{25}{4} > 0 \quad \to$ *hyperbolic*
2. $D = (-6)^2 - 4 \cdot 9 = 36 - 36 = 0 \quad \to$ *parabolic*
3. $D = 3^2 - 4 \cdot 9 = 9 - 36 = -25 < 0 \quad \to$ *elliptic.*

Example 4.38. *Determine subdomains of the xy plane in which*

$$y\partial_{xx} u - 2\partial_{xy} u + x\partial_{yy} u = 0$$

is elliptic, hyperbolic or parabolic. First, we have $D = (-1)^2 - y \cdot x = 1 - yx$. The equation is parabolic for $xy = 1$ because $1 - 1 = 0$. The elliptic parts are convex $xy > 1$ because $D = 1 - yx < 0$. In the connected part $xy < 1$, the equation is hyperbolic.

4.6 Chapter summary and outlook

In this extended introduction, we formulated several important PDEs and discussed prototype applications and equations. Furthermore, we classified PDEs and introduced their level of linearity. In the next section, we introduce finite differences as a first class of numerical methods for elliptic equations. In Section 6, we then discuss finite elements as another class of discretization methods.

5 Finite differences (FD) for elliptic problems

In this section, we address finite differences (FD) for boundary value problems (BVP). The key idea using FD is to approximate the derivatives of the PDEs with the help of difference quotients. Despite that finite differences are very old and other methods are better (in terms of domain shapes, general material coefficients) in state-of-the art approximations of PDEs, it is still worth to discuss them:

- They are very simple to understand and implement and yield a fast numerical solution and idea how a solution could look like;
- They are still used in (big) research codes in industry;
- Parts, such as difference quotients, serve as basis in all kind of software implementations of numerical methods for computing ODEs and PDEs in which derivatives need to be approximated.

First, we complete the missing properties of the continuous problem and show well-posedness for the one-dimensional Poisson problem with non-homogeneous Dirichlet boundary data. Then, we provide few words on the maximum principle and come afterwards to the discretization and finish by the numerical analysis.

5.1 A 1D model problem: Poisson

We study in this section a finite difference discretization for the one-dimensional Poisson problem:

$$\begin{cases} \text{Find} \quad u \in \mathcal{C}^2([0,L]) \quad \text{such that} \\ -u''(x) = f \quad ,\forall x \in (0,L), \\ u(0) = a, \\ u(L) = b, \end{cases} \tag{31}$$

where f is a given right hand side function $C^0([0,L])$ and $a, b \in R$.

For $f = -1$ and $a = 0$ and $L = 1$ we obtain a situation as sketched in Figure 3.

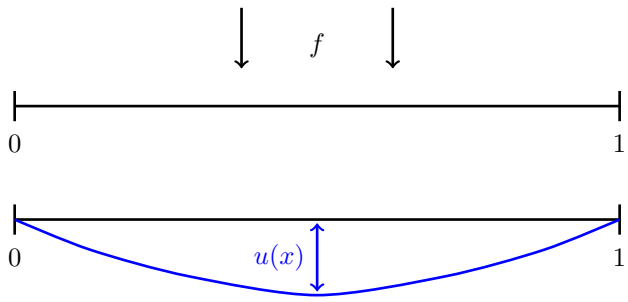

Figure 3: The clothesline problem: a uniform force $f = -1$ acts on a 1D line yielding a displacement $u(x)$.

Remark 5.1. *A derivation based on first principles in physics is provided in Section 7.3.2.*

5.2 Well-posedness of the continuous problem

We investigate in the following well-posedness, the maximum principle and the regularity of the continuous problem.

5.2.1 Green's function and well-posedness

We start with Green's function that represents the integral kernel of the Laplace operator (here the second-order derivative in 1D). Using Green's function, we obtain an explicit representation formula for the solution u. We define Green's function as:

$$G(x,s) = \begin{cases} \dfrac{s(L-x)}{L} & \text{if } 0 \le s \le x \le L, \\ \dfrac{x(L-s)}{L} & \text{if } 0 \le x \le s \le L. \end{cases} \tag{32}$$

We now integrate two times the differential equation (31). For all $\forall x \in [0, L]$, we obtain

$$u(x) = -\int_0^x \left(\int_0^t f(s)ds \right) dt + C_1 x + C_2, \tag{33}$$

where C_1 and C_2 are integration constants. Using Fubini's theorem (see e.g., [45]), we obtain

$$\int_0^x \left(\int_0^t f(s)ds \right) dt = \int_0^x \left(\int_s^x f(s)dt \right) ds = \int_0^x (x-s)\, f(s)ds$$

and therefore

$$u(x) = -\int_0^x (x-s)\, f(s)ds + C_1 x + C_2.$$

Using the boundary conditions from (31), we can determine C_1 and C_2:

$$\begin{aligned} u(x) &= -\int_0^x (x-s)f(s)ds + \frac{x}{L}\int_0^L (L-s)f(s)ds + a\frac{L-x}{L} + b\frac{x}{L} \\ &= \int_0^x \left(-\frac{L(x-s)}{L} + \frac{x}{L}(L-s) \right) f(s)ds + \frac{x}{L}\int_x^L (L-s)f(s)ds + a\frac{L-x}{L} + b\frac{x}{L} \\ &= \int_0^L G(x,s)f(s)ds \; + \; a\,\frac{(L-x)}{L} \; + b\,\frac{x}{L} \end{aligned} \tag{34}$$

in which we employed Green's function G defined in (32).

Now, we are able to investigate the well-posedness (see Section 2.5 for the meaning of well-posedness):

Proposition 5.2. *Problem* (31) *has a unique solution and depends continuously on the problem data via the following stability estimate:*

$$\|u\|_{C^2} = \max(\|u\|_{C^0}, L\|u'\|_{C^0}, L^2\|u''\|_{C^0}) \le C_1\|f\|_{C^0} + C_2|a| + C_3|b|,$$

where the constants may be different than before. If $a = b = 0$ (homogeneous Dirichlet conditions), the estimate simplifies to:

$$\|u\|_{C^2} \le C_1\|f\|_{C^0}.$$

An analoguous result for more general spaces is stated in Proposition 6.144.

Proof. Existence and uniquness follow immediately from the representation of the solution as

$$u(x) = \int_0^L G(x,s)f(s)\,ds \; + \; a\,\frac{(L-x)}{L} \; + b\,\frac{x}{L}. \tag{35}$$

The stability estimate follows via the following arguments. Green's function $G(x,s)$ is positive in $(0, L)$ and we can estimate as follows:

$$|u(x)| \le \|f\|_{C^0} \int_0^L G(x,s)\,ds + \; |a| \; + \; |b|$$

and $\forall x \in [0, L]$, we have

$$\int_0^L G(x,s)\,ds = \frac{(L-x)}{L}\int_0^x s ds + \frac{x}{L}\int_x^L (L-s)ds \tag{36}$$

$$\tag{37}$$

$$= \frac{(L-x)x^2}{2L} + \frac{x(L-x)^2}{2L} = \frac{(L-x)x}{2} \leq \frac{L^2}{8}, \quad \text{in } x = L/2, \tag{38}$$

yielding

$$\|u\|_{\mathcal{C}^0} = \max_{x\in[0,L]} |u(x)| \leq \frac{L^2}{8}\|f\|_{\mathcal{C}^0} + |a| + |b|.$$

All values on the right hand side are given by the problem data and are therefore known. However, the estimate is not yet optimal in the regularity of u. In the following we will see that we can even bound the second-order derivative of u by the right hand side f.

Indeed, by differentiating the solution (35) (actually better to see for the $G(x,s)$ function in (36)), we obtain $\forall x \in [0,L]$:

$$u'(x) = -\frac{1}{L}\int_0^x sf(s)ds + \frac{1}{L}\int_x^L (L-s)f(s)ds - \frac{a}{L} + \frac{b}{L}.$$

Since the integrand is still positive, we can again estimate:

$$|u'(x)| \leq \|f\|_{C^0}\frac{1}{L}\int_0^x s\,ds + \frac{1}{L}\int_x^L (L-s)ds + \frac{|a|}{L} + \frac{|b|}{L}$$

and $\forall x \in [0,L]$, the Green's function part yields:

$$\frac{1}{L}\int_0^x s\,ds + \frac{1}{L}\int_x^L (L-s)ds = \frac{x^2}{2L} + \frac{(L-x)^2}{2L} \leq \frac{L}{2}$$

thus

$$\|u'\|_{\mathcal{C}^0} = \max_{x\in[0,L]} |u'(x)| \leq \frac{L}{2}\|f\|_{\mathcal{C}^0} + \frac{|a|}{L} + \frac{|b|}{L}.$$

Differentiating a second time, we obtain (of course) $\forall x \in [0,L]$:

$$u''(x) = -f(x)$$

i.e., our original PDE. Therefore, we obtain the third estimate as:

$$\|u''\|_{\mathcal{C}^0} = \max_{x\in[0,L]} |u''(x)| \leq \|f\|_{\mathcal{C}^0}.$$

Consequently, the solution depends continuously on the problem data and therefore the problem is well-posed. □

Exercise 3. *Recapitulate and complete the above steps:*

- *Derive step-by-step the explicit form* (33)*;*
- *Determine* C_1 *and* C_2 *and convince yourself that* (34) *holds true;*
- *Convince yourself by an explicit computation that* $u''(x) = -f(x)$ *follows from* (35)*.*

5.2.2 Maximum principle on the continuous level

Since $G(x,s)$ is positive in $(0,L)$, the representation (35) yields immediately the maximum principle. That is to say:

$$f \geq 0,\, a \geq 0,\, b \geq 0 \quad \Rightarrow \quad u \geq 0.$$

Remark 5.3. *Intuitively this is immediately clear. Just consider a parabola function* $u(x) = \frac{1}{2}(-x^2 + x)$*, which is* $-u''(x) = 1$*, where* $f = 1$*. The function is positive with maximum in* 0.5 *and has its minima on the boundaries* 0 *and* 1*.*

Moreover, the maximum principle yields also uniqueness of the solution. Assume that u_1 and u_2 are two different solutions. Their difference is $w := u_1 - u_2$. Of course the original PDE, namely $-u''(x) = f$, also holds for w. Thus: $-w''(x) = f$. We apply f, a, b and $-f, -a, -b$ and obtain $-\Delta w = f$ from which we get $w \geq 0$ and $-\Delta(-w) = -f$ from which we get $w \leq 0$, respectively. Consequently, $w = 0$ and therefore $u_1 = u_2$.

5.2.3 Regularity of the solution

The regularity in the case of Poisson's problem carries over from the right hand side. If $f \in \mathcal{C}^0([0,L])$ we have shown that $u \in \mathcal{C}^2([0,L])$. For higher regularity $k \geq 1$, if $f \in \mathcal{C}^k([0,L])$, therefore for $1 \leq k' \leq k$, $u^{(k'+2)} = -f^{(k')} \in \mathcal{C}^0([0,L])$ and consequently $u \in \mathcal{C}^{k+2}([0,L])$. The solution is always two orders more regular than the right hand side data f.

5.3 Spatial discretization

After the previous theoretical investigations we address now a finite difference discretization for Poisson's problem. As previously mentioned, using finite differences, derivatives are approximated via difference quotients. For simplicity we consider homogeneous Dirichlet boundary values. We recall the corresponding problem:

Formulation 5.4.

$$\begin{cases} \textit{Find} \quad u \in \mathcal{C}^2([0,L]) \quad \textit{such that} \\ -u''(x) = f \quad \forall x \in (0,L), \\ u(0) = 0, \\ u(L) = 0. \end{cases} \tag{39}$$

In the first step, we need to discretize the domain by introducing discrete support points. Let $n \in \mathbb{N}$ and the step size (spatial discretization parameter) h (the distance between two support points) and the support points x_j be defined as

$$h = \frac{L}{n+1}, \quad x_j = jh \quad \text{for all } 0 \leq j \leq n+1.$$

By this construction, we have

- n inner points;
- $n+2$ total points (including the two boundary points);
- $n+1$ intervals (so-called mesh or grid elements).

The difference quotient for approximating a second order derivative is given by (recall classes to the introduction of numerical methods, e.g., [61]):

$$u''(x_j) \approx \frac{U_{j+1} - 2U_j + U_{j-1}}{h^2},$$

where we used a capitel letter U to distinguish the discrete solution from the exact solution u. In the following, be careful with the minus sign since we approximate the negative second order derivative of u:

$$-u''(x_j) \approx \frac{-U_{j+1} + 2U_j - U_{j-1}}{h^2}.$$

Consequently, the approximated, discrete, PDE reads:

$$\frac{-U_{j+1}+2U_j-U_{j-1}}{h^2}=f(x_j),\quad 1\le j\le n,$$

with $U_0=U_{n+1}=0$. Going from $j=0,\dots n$, we obtain a linear equation system

$$\underbrace{\frac{1}{h^2}\begin{pmatrix}2&-1&0&\cdots&0\\-1&2&-1&\ddots&\vdots\\0&\ddots&\ddots&\ddots&0\\\vdots&\ddots&-1&2&-1\\0&\cdots&0&-1&2\end{pmatrix}}_{=A\in\mathbb{R}^{n\times n}}\underbrace{\begin{pmatrix}U_1\\U_2\\\vdots\\\vdots\\U_{n-1}\\U_n\end{pmatrix}}_{=U\in R^n}=\underbrace{\begin{pmatrix}f(x_1)\\f(x_2)\\\vdots\\\vdots\\f(x_{n-1})\\f(x_n)\end{pmatrix}}_{=F\in\mathbb{R}^n}$$

In compact form:

$$AU=F.$$

The 'only' remaining task consists now in solving this linear equation system. The solution U vector will then contain the discrete solution values on the corresponding support (inner) points.

Remark 5.5. *We could have formulated the above system for all points (including the two boundary points):*

$$\underbrace{\frac{1}{h^2}\begin{pmatrix}1&0&&&\cdots&\cdots&0\\0&2&-1&0&\cdots&\cdots&0\\\vdots&-1&2&-1&\ddots&&\vdots\\&0&\ddots&\ddots&\ddots&&0\\&\vdots&\ddots&-1&2&-1&0\\&0&\cdots&0&-1&2&0\\0&0&\cdots&0&0&0&1\end{pmatrix}}_{=A\in\mathbb{R}^{(n+2)\times(n+2)}}\underbrace{\begin{pmatrix}U_0\\U_1\\U_2\\\vdots\\\vdots\\U_{n-1}\\U_n\\U_{n+1}\end{pmatrix}}_{=U\in R^{n+2}}=\underbrace{\begin{pmatrix}0\\f(x_1)\\f(x_2)\\\vdots\\\vdots\\f(x_{n-1})\\f(x_n)\\0\end{pmatrix}}_{=F\in\mathbb{R}^{n+2}}$$

But we see immediately that we can save this slightly bigger matrix since the bounddary values are already known and can be computed immediately.

5.4 Solving the linear equation system

Depending on the fineness of the discretization (the smaller h, the smaller the discretization error as we will see later), we deal with large matrices $A\in\mathbb{R}^{n\times n}$ for $n\sim 10^6$ and larger. For moderate sizes, a **direct** solution à la LU or Cholesky is competitive. For large n, due to

- computational cost,
- and memory restrictions,

we have to approximate U with the help of iterative solution schemes such as Jacobi, Gauss-Seidel, conjugate gradient, multigrid, and so forth. We come back to this topic in Section 8 in which we provide a basic introduction.

5.5 Well-posedness of the discrete problem

We now investigate well-posedness of the discrete problem.

5.5.1 Existence and uniqueness of the discrete problem

We first show that the matrix A is symmetric ($A = A^T$, where A^T is the transpose), positive definite. A direct calculation for two vectors $X, Y \in \mathbb{R}^n$ gives:

$$\begin{aligned}
< AX, Y >_n &= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij} X_j Y_i \\
&= \frac{1}{h^2}\sum_{i=1}^{n}(-X_{i-1} + 2X_i - X_{i+1})Y_i \\
&= \frac{1}{h^2}\sum_{i=1}^{n}(X_i - X_{i+1})Y_i + \frac{1}{h^2}\sum_{i=1}^{n}(X_i - X_{i-1})Y_i \\
&= \frac{1}{h^2}\sum_{i=0}^{n}(X_i - X_{i+1})Y_i + \frac{1}{h^2}\sum_{i=1}^{n+1}(X_i - X_{i-1})Y_i \\
&= \frac{1}{h^2}\sum_{i=1}^{n+1}(X_{i-1} - X_i)Y_{i-1} + \frac{1}{h^2}\sum_{i=1}^{n+1}(X_i - X_{i-1})Y_i \\
&= \frac{1}{h^2}\sum_{i=1}^{n+1}(X_i - X_{i-1})(Y_i - Y_{i-1}) \\
&= \sum_{i=1}^{n+1}\frac{X_i - X_{i-1}}{h}\frac{Y_i - Y_{i-1}}{h} \\
&= < X, AY >_n,
\end{aligned}$$

with $X_0 = Y_0 = X_{n+1} = Y_{n+1} = 0$. This first calculation shows the symmetry. It remains to show that

$$< AX, Y >_n$$

is positive definite. We now consider the term

$$< AX, Y >= \ldots = \frac{1}{h^2}\sum_{i=1}^{n+1}(X_i - X_{i-1})(Y_i - Y_{i-1})$$

from the previous calculation. Setting $Y = X$ yields:

$$< AX, X >_n = \sum_{i=1}^{n+1}\left(\frac{X_i - X_{i-1}}{h}\right)^2 \geq 0.$$

This shows the positivity. It remains to show the definiteness: if $< AX, X >_n = 0$ for all $1 \leq i \leq n+1$,

$$X_0 = X_1 = \cdots = X_{i-1} = X_i = \cdots = X_n = X_{n+1},$$

the zero boundary conditions yield finally $X = 0$. Consequently, the matrix A is symmetric positive definite. The eigenvalues are in $\mathbb{R}$ and strictly positive. Therefore, A is regular (recall the results from linear algebra!) and thus invertible, which finally yields:

Proposition 5.6 (Existence and uniqueness of the discrete problem)**.** *The discrete problem*

$$\frac{-U_{j+1} + 2U_j - U_{j-1}}{h^2} = f(x_j) \quad 1 \leq j \leq n, \tag{40}$$

with $U_0 = U_{n+1} = 0$, has one and only one solution for all $F \in \mathbb{R}^n$.

5.5.2 Maximum principle on the discrete level

As previously stated, the maximum principle on the continuous level has a discrete counterpart. Let F be given such that $F_j \geq 0$ for $1 \leq j \leq n$. We perform an indirecte proof and show that $\mu := \min_{1 \leq j \leq n} U_j \geq 0$. We

suppose $\mu < 0$ and take $i \in \{1, \cdots, n\}$ as one index (it exists at least one!) such that the minimum is taken: $U_i = \min_{1 \leq j \leq n} U_j = \mu$. If $1 < i < n$, and U is solution of $AU = F$, the ith row can be written as:

$$2U_i - U_{i+1} - U_{i-1} = h^2 F_i \geq 0$$

under the assumption on F being positive. Therefore, we split

$$0 \leq (U_i - U_{i+1}) + (U_i - U_{i-1}) = (\mu - U_{i+1}) + (\mu - U_{i-1}) \leq 0$$

because $U_i = \mu$ is the minimum value, which we assumed to be negative. To satisfy this inequality chain it must hold:

$$U_{i-1} = U_i = U_{i+1} = \mu,$$

which shows that also the neighboring indices $i - 1$ and $i + 1$ satisfy the minimum. Extending this idea to all indices yields finally, $U_1 = U_n = \mu$. Therefore, we can investigate the first or the last row of the matrix A:

$$0 \leq F_1 = 2U_1 + U_2 = U_1 + (U_1 - U_2) \leq U_1 = \mu < 0.$$

But

$$0 \leq U_1 = \mu < 0$$

is a contradiction. Consequently, $\mu > 0$. Respecting the boundary conditions, we notice that for $U_0 = U_{n+1} = 0$ and $F_j \geq 0$ for all $1 \leq j \leq n$, we have the desired result

$$\mu = \min_{0 \leq j \leq n+1} U_j \geq 0$$

showing that we have a discrete maximum principle.

Proposition 5.7 (Monotone inverse)**.** *The matrix A corresponding to the finite difference discretization is a so-called M matrix, i.e., the inverse A^{-1} is element-wise non-negative, i.e.,*

$$A^{-1} \geq 0.$$

Proof. For $1 \leq j \leq n$, we fix $F^j = (0 \cdots 1 \cdots 0) = (\delta_{ij})_{1 \leq i \leq n}$, where δ_{ij} is the Kronecker symbol and X^j is the solution of $AX^j = F^j$. By construction, $X^j = A^{-1}F^j$ and for $1 \leq i \leq n$, the ith row can be written as

$$X_i^j = \sum_{k=1}^{n} A_{ik}^{-1} F_k^j = \sum_{k=1}^{n} A_{ik}^{-1} \delta_{kj} = A_{ij}^{-1}.$$

Using the maximum principle, we know that for $1 \leq i, j \leq n$ and $F_i^j = \delta_{ij} \geq 0$, the solution is positive, i.e.,

$$X_i^j \geq 0$$

(j is the jth solution and i is the component of that solution). Thanks to $X_i^j = A_{ij}^{-1}$, we obtain

$$A_{ij}^{-1} \geq 0,$$

which means that the inverse of the matrix A is an M matrix. □

5.6 Numerical analysis: consistency, stability, and convergence

We finally investigate the convergence properties of the finite difference discretization. As we know from ODE theory, for linear schemes, consistency plus stability will yield convergence.

We first derive an expression for the **truncation error** η (the local discretization error), which is obtained by plugging in the exact (unknown) solution into the numerical scheme.

Let

$$e_j = U_j - \bar{U}_j \quad \text{for } 1 \le j \le n,$$

where $\bar{U}_j = (\bar{U}_i)_{i \le i \le n} = (u(x_i))_{1 \le i \le n} \in \mathbb{R}^n$. Then, we have for $1 \le i \le n$:

$$\begin{aligned}(Ae)_i &= \big[A(U-\bar{U})\big]_i = (F - A\bar{U})_i = f(x_i) - (A\bar{U})_i \\ &= -u''(x_i) - \frac{-\bar{U}_{i+1} + 2\bar{U}_i - \bar{U}_{i-1}}{h^2} \\ &= \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} - u''(x_i)\end{aligned}$$

and thus

$$Ae = \eta$$

where $\eta \in \mathbb{R}^n$ is the local truncation error that arises because the exact solution u does not satisfy the numerical scheme (it is only the discrete solution U that is a solution of the discrete scheme!). Consequently, for all $1 \le i \le n$, we have

$$\eta_i := \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} - u''(x_i).$$

5.6.1 Consistency

The consistency is based on estimates of the local truncation error. The usual method to work with is a Taylor expansion (see Section 3.12). We have previously established that for $f \in \mathcal{C}^2([0,L])$, we have $u \in \mathcal{C}^4([0,L])$. Consequently, we can employ a Taylor expansion up to order 4 at the point $x_i, 1 \le i \le n$. This brings us to:

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\tau_i^+)$$

$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\tau_i^-)$$

with $\tau_i^+ \in [x_i, x_{i+1}]$ and $\tau_i^- \in [x_{i-1}, x_i]$. It follows that for all $1 \le i \le n$

$$|\eta_i| = \left|\frac{h^2}{24}u^{(4)}(\tau_i^+) + \frac{h^2}{24}u^{(4)}(\tau_i^-)\right| \le \frac{h^2}{12}\|u^{(4)}\|_{\mathcal{C}^0} = \frac{h^2}{12}\|f''\|_{\mathcal{C}^0}$$

in which we used that $u^{(4)} = -f''$. As final result, we obtain:

Proposition 5.8. *The local truncation error of the finite difference approximation of Poisson's problem can be estimated as*

$$\|\eta\|_\infty := \max_{1 \le i \le n} |\eta_i| \le C_f\, h^2, \qquad \text{with } C_f = \frac{1}{12}\|f''\|_{\mathcal{C}^0} \tag{41}$$

Therefore, the scheme (40) *is consistent with order* 2.

Remark 5.9. *We notice that the additional order of the scheme (namely order two and not one) has been obtained since we work with a central difference quotient to approximate the second order derivative. Of course, here it was necessary to assume sufficient regularity of the right hand side f and the solution u.*

Remark 5.10. *The requirements on the regularity of the solution u to show the above estimates is much higher than the corresponding estimates for finite elements (see Section 6.11). In practice, such a high regularity can only be ensured in a few cases, which is a drawback of finite differences in comparison to finite elements. But still, we can use the method in practice, but one has to be careful whether the results are still robust and reliable.*

5.6.2 Stability in L^∞

We now investigate the stability of the finite difference scheme. We recapitulate the matrix norm $\|.\|_\infty$ for $M \in \mathbb{R}^{n,n}$ defined by

$$\|M\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |M_{ij}|$$

Moreover, we denote by $w(x)$ the exact solution for the problem in which $f = 1$ on $[0, L]$. The choice of $f = 1$ is only for simplicity. More general f work as well, but will require more work in the proof below.

Proposition 5.11. *It holds:*

$$\|A^{-1}\|_\infty = \|\bar{W}\|_\infty,$$

and

$$\|A^{-1}\|_\infty \leq \frac{L^2}{8}.$$

Proof. For $f = 1$, the second derivative vanishes: $f'' = 0$. From Section 5.6.1, we know that in this case

$$\|\eta\|_\infty = 0$$

and therefore

$$0 = Ae = A(W - \bar{W})$$

where $W \in \mathbb{R}^n$ denotes the discrete solution corresponding to $f = 1$ and the exact solutio w. Moreover, we denote $\bar{W} = (w(x_i))_{1\leq i\leq n} \in \mathbb{R}^n$. Consequently, $W = \bar{W} = A^{-1}F_1$. For $1 \leq i \leq n$, we obtain

$$\bar{W}_i = \sum_{j=1}^{n} A^{-1}_{ij}(F_1)_i = \sum_{j=1}^{n} A^{-1}_{ij} = \sum_{j=1}^{n} |A^{-1}_{ij}|$$

because A is an M matrix. We finally obtain:

$$\|A^{-1}\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |A^{-1}_{ij}| = \max_{1\leq i\leq n} |\bar{W}_i| = \|\bar{W}\|_\infty$$

The exact solution to the problem $-w''(x) = 1$ on $(0, L)$ with homogeneous Dirichlet conditions is $w(x) = \frac{x(L-x)}{2}$, which atteins its maximum at $x = L/2$. This yields $\max(w) = L^2/8$ and therefore,

$$\|A^{-1}\|_\infty \leq \frac{L^2}{8},$$

which shows the assertion. □

5.6.3 Convergence L^∞

With the previous two results, we can now proof convergence of the finite difference scheme. We assume as before $f \in \mathcal{C}^2([0, L])$ and obtain:

Proposition 5.12. *For the convergence of the finite difference scheme, it holds:*

$$\|e\|_\infty = \|A^{-1}\eta\|_\infty \leq \|A^{-1}\|_\infty \|\eta\|_\infty \leq \frac{L^2}{8} C_f\, h^2 = \frac{L^2}{96} \|f''\|_{\mathcal{C}^0} h^2 = O(h^2),$$

where we employed the stability and consistency estimates and (41)*. The scheme* (40) *converges in the L^∞ norm with order* 2 *as h tends to zero.*

5.6.4 Convergence L^2

In this final section, we proof a second convergence result, but now in the L^2 norm. We define two norms:

$$\|X\|_A = \left(\sum_{j=1}^{n+1} \frac{1}{h}|X_j - X_{j-1}|^2\right)^{\frac{1}{2}} \qquad \|X\|_2 = \left(\sum_{j=1}^{n} h|X_j|^2\right)^{\frac{1}{2}}$$

and notice that $< . >_A$ and $< . >_2$ are the corresponding scalar products.

Proposition 5.13. *It holds for each $X \in R^n$:*

$$\|X\|_2 \leq L^{1/2}\|X\|_\infty \leq L\|X\|_A.$$

Proof. Let $X \in \mathbb{R}^n$ for all $1 \leq i \leq n$ and we set as usually $X_0 = X_{n+1} = 0$. Then:

$$\begin{aligned} |X_i| &= \left|X_0 + \sum_{j=1}^{i}(X_j - X_{j-1})\right| = \sum_{j=1}^{i} h\frac{X_j - X_{j-1}}{h} * 1 =< \frac{X_j - X_{j-1}}{h}, 1 >_i \\ &\leq \left(\sum_{j=1}^{i} h\left|\frac{X_j - X_{j-1}}{h}\right|^2\right)^{1/2}\left(\sum_{j=1}^{i} h|1|^2\right)^{1/2} \\ &\leq \|X\|_A L^{1/2} \end{aligned}$$

where we used the Cauchy-Schwarz inequality. It follows that

$$\|X\|_\infty \leq L^{1/2}\|X\|_A.$$

Moreover,

$$\|X\|_2^2 = \left(\sum_{j=1}^{n} h|X_j|^2\right) \leq \|X\|_\infty^2 \left(\sum_{j=1}^{n} h\right) \leq \|X\|_\infty^2 L.$$

We finally obtain

$$\|X\|_2 \leq \|X\|_\infty L^{1/2} \leq L\|X\|_A.$$

□

Remark 5.14. *We notice that $\|X\|_2 \leq L\|X\|_A$ holds because $X_0 = 0$ and $\sum_{j=1}^{i} h|1|^2$ is bounded. Otherwise the previous result would be not correct.*

Proposition 5.15. *Let $X, Y \in \mathbb{R}^n$. For a symmetric, positive, definite matrix A, we can define the so-called A-norm; see also Section 8.4. It holds*

$$< X, Y >_A =< AX, Y >_2$$

and

$$\|e\|_A \leq L\|\eta\|_2.$$

Proof. We already have shown that for all $X, Y \in \mathbb{R}^n$, it holds

$$< AX, Y >_n = \sum_{i=1}^{n+1} \frac{X_i - X_{i-1}}{h}\frac{Y_i - Y_{i-1}}{h}.$$

Therefore,

$$< AX, Y >_2 = h < AX, Y >_n = h\sum_{i=1}^{n+1} \frac{X_i - X_{i-1}}{h}\frac{Y_i - Y_{i-1}}{h} =< X, Y >_A$$

from which follows

$$\|e\|_A^2 =< e, e >_A =< Ae, e >_2 =< \eta, e >_2 \leq \|\eta\|_2\|e\|_2 \leq L\|\eta\|_2\|e\|_A$$

in which we used again the Cauchy-Schwarz inequality. This shows the assertion. □

Proposition 5.16. *It holds:*

$$\|e\|_A \leq L\|\eta\|_2 \leq L^{3/2}\|\eta\|_\infty \leq L^{3/2}C_f\, h^2$$

and

$$\|e\|_2 \leq L\|e\|_A \leq L^{5/2}C_f\, h^2.$$

In other words: the scheme (40) *converges in the* L^2 *norm and the* A *norm with order* 2 *when* h *tends to zero.*

5.7 Numerical test: 1D Poisson

We finish this 1D section with a numerical test implemented in octave [42].

Formulation 5.17. *Let* $\Omega = (0,1)$

$$\begin{cases} \text{Find} \quad u \in \mathcal{C}^2(\Omega) \quad \text{such that} \\ -u''(x) = f \quad \text{in } \Omega \\ u(0) = 0, \\ u(1) = 0. \end{cases} \tag{42}$$

where $f = -1$. *For the discretization we choose* $n = 4$. *Thus we have five support points*

$$x_0, x_1, x_2, x_3, x_4$$

and $h = 1/4 = 0.25$.

With these settings we obtain for the matrix A (see Section (5.3)):

```
16    0    0    0    0
0    32  -16    0    0
0   -16   32  -16    0
0     0  -16   32    0
0     0    0    0   16
```

and for the right hand side vector b:

```
0
-1
-1
-1
0
```

We use the famous backslash solution operator (in fact an LU decomposition):

```
U = A\b
```

and obtain as solution $U = (U_0, U_1, U_2, U_3, U_4)$:

```
 0.00000
-0.09375
-0.12500
-0.09375
 0.00000
```

A plot of the discrete solution is provided in Figure 4.

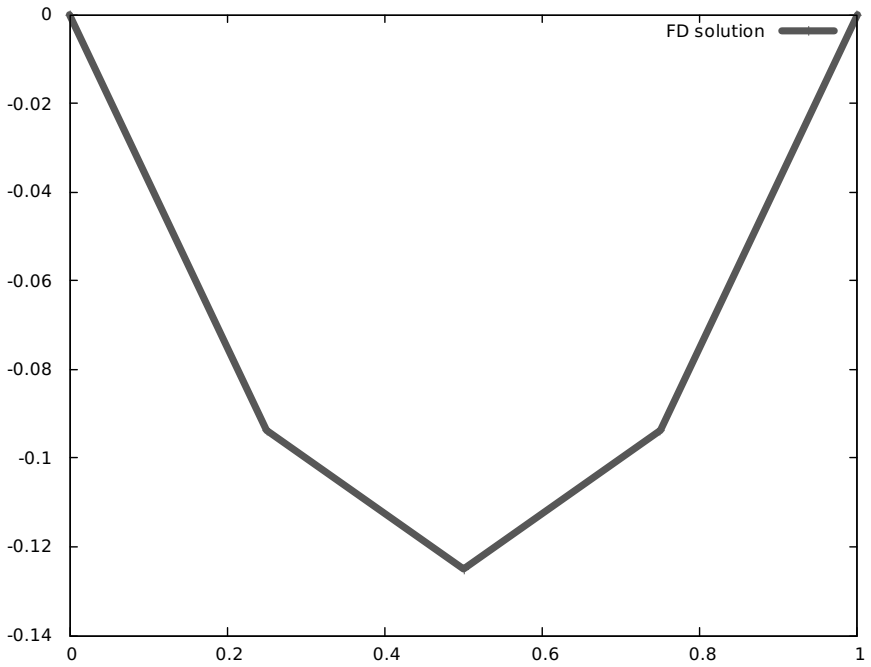

Figure 4: Solution of the 1D Poisson problem with $f = -1$ using five support points with $h = 0.25$. We observe that the approximation is rather coarse. A smaller h would yield more support points and more solution values and therefore a more accurate solution.

Remark 5.18 (Matrix A in FD vs. FEM)**.** *In the above matrix A and RHS b, we have implicitly $1/h^2$ and 1 as factors, respectively. Indeed*

$$A_{FD} = \frac{1}{h^2}\begin{pmatrix} 1 & 0 & & & \cdots & \cdots & 0 \\ 0 & 2 & -1 & 0 & \cdots & \cdots & 0 \\ \vdots & -1 & 2 & -1 & \ddots & & \vdots \\ & 0 & \ddots & \ddots & \ddots & & 0 \\ & \vdots & \ddots & -1 & 2 & -1 & 0 \\ & 0 & \cdots & 0 & -1 & 2 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad b_{FD} = (1, \ldots, 1)^T, \qquad \frac{1}{h^2} = \frac{1}{16}.$$

The corresponding matrix using an FEM scheme (Section 6) looks slightly different, but is in fact the same. Here, we integrate only u', but multiplied with a test function φ'. On the right hand side we also have φ. Here, we obtain:

$$A_{FEM} = \frac{1}{h}\begin{pmatrix} 1 & 0 & & & \cdots & \cdots & 0 \\ 0 & 2 & -1 & 0 & \cdots & \cdots & 0 \\ \vdots & -1 & 2 & -1 & \ddots & & \vdots \\ & 0 & \ddots & \ddots & \ddots & & 0 \\ & \vdots & \ddots & -1 & 2 & -1 & 0 \\ & 0 & \cdots & 0 & -1 & 2 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad b_{FEM} = h * (1, \ldots, 1)^T, \qquad \frac{1}{h} = \frac{1}{4}.$$

Dividing over h in A_{FEM} and b_{FEM} yields directly A_{FD} and b_{FD}.

5.7.1 How can we improve the numerical solution?

An important question is how we can improve the numerical solution? A five-point approximation as employed in Figure 4 is obviously not very accurate. After all previous subsections in this chapter, the answer is clear: increasing the number of support points x_n, i.e., $n \to \infty$ in the given domain Ω will yield $h \to 0$ (recall $h = x_{j+1} - x_j$). According to our findings in the numerical analysis section, the discretization error will become very small and we have therefore an accurate approximation of the given problem. On the other hand, large n will engross a lot of computational resources. We have finally to find a compromise between accuracy of a numerical solution and computational cost (recall the concepts outlined in Section 2.4).

5.8 Finite differences in 2D

We extend the idea of finite differences now to two-dimensional problems. The goal is the construction of the method. For a complete numerical analysis we refer to the literature, e.g., [33, 56].

In this section we follow (more or less) [58]. We consider

$$-\Delta u = f \quad \text{in } (0,1)^2, \tag{43}$$

$$u = u_D \quad \text{on } \partial\Omega_D. \tag{44}$$

5.8.1 Discretization

In extension to the previous section, we now construct a grid in two dimensions. The local grid size is h. We define

$$\Omega_h := \{(x_{ij}), i,j = 0,\dots,N |\ x_{ij} = (ih, jh), h = N^{-1}\}$$

We seek a solution in all inner grid points

$$\Omega_h^0 := \{(x_{ij}), i,j = 1,\dots,N-1 |\ x_{ij} = (ih, jh), h = N^{-1}\}$$

For the following, we introduce a grid function $u_h := (u_{ij})_{i,j=1}^{N-1}$ and seek unknown function as the solution of

$$-\Delta_h u_h = F_h, \quad x_{ij} \in \Omega_h^0.$$

Using a central finite difference operator of second order we obtain

$$-(\Delta_h u_h)_{ij} = \frac{1}{h^2}(4u_{ij} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}), \quad (F_h)_{ij} := f(x_{ij})$$

On the boundary, we set

$$u_{0,j} = u_{N,j} = u_{i,0} = u_{i,N}$$

This difference quotient is known as a five-point stencil:

$$S_h = \frac{1}{h^2}\begin{pmatrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{pmatrix}$$

In total we obtain the following linear equation system:

$$A_h u_h = f_h \tag{45}$$

with

$$A_h := \begin{pmatrix} B_m & -I_m & & \\ -I_m & B_m & -I_m & \\ & -I_m & B_m & \ddots \\ & & \ddots & \end{pmatrix}, \quad B := \begin{pmatrix} 4 & -1 & & \\ -1 & 4-1 & & \\ & -1 & 4 & \ddots \\ & & \ddots & \end{pmatrix} \tag{46}$$

The matrix has size $A_h \in \mathbb{R}^{N^2 \times N^2}$ and bandwidth $2N+1$. Further properties are:

Proposition 5.19. *The matrix A_h has the following properties:*

1. *A_h has five entries per row;*
2. *it is symmetric;*
3. *weakly diagonal-dominant;*
4. *positive definite;*
5. *an M matrix.*

It holds:

Proposition 5.20. *The five-point stencil approximation of the Poisson problem in two dimensions satisfies the following a priori error estimate:*

$$\max_{ij} |u(x_{ij}) - u_{ij}| \leq \frac{1}{96} h^2 \max_{\Omega} (|\partial_x^4 u| + |\partial_y^4 u|).$$

Proof. Later. □

The resulting linear equation system $A_h u_h = F_h$ is sparse in the matrix A_h, but maybe large, for large N (i.e., very small h and consequently high accuracy).

To determine the condition number (which is important for the numerical solution), we recall the eigenvalues and eigenvectors:

$$\lambda_{kl} = h^{-2}(4 - 2(cos(kh\pi) + cos(lh\pi))),$$
$$\omega_{kl} = (\sin(ikh\pi)\sin(jlh\pi))_{ij=1}^{N-1}.$$

The largest and smallest eigenvalues are, respectively:

$$\lambda_{max} = h^{-2}(4 - 4\cos(1-h)\pi) = 8h^{-2} + O(1),$$
$$\lambda_{min} = h^{-2}(4 - 4\cos(h\pi)) = h^{-2}(4 - 4(1 - 0.5(\pi^2 h^2)) + O(h^2) = 2\pi^2 + O(h^2).$$

Therefore, the condition number is

$$cond_2(A_h) = \frac{\lambda_{max}}{\lambda_{min}} = \frac{2}{\pi^2 h^2},$$

which is similar to the heat equation.

Remark 5.21 (on the condition number)**.** *The order $O(h^{-2})$ of the matrix A_h is determined by the differential equation and* ***neither*** *by the dimension of the problem (here 2)* ***nor*** *the quadratic order of the consistency error. This statement holds true for finite differences as well as finite elements; see also Section 8.1.*

5.9 Chapter summary and outlook

In this chapter, we considered finite differences for elliptic problems. Despite that finite elements or isogeometric analysis have become in general more important, finite differences still serve their purposes because they are simple to implement and do still exist in large industrial codes. FD are nevertheless further developed (mimetic finite differences, etc.), and are also often used in combination with other methods such as finite elements. For instance, a widely used discretization method for initial boundary value problems is to discretize in time with a finite difference scheme (e.g., One-Step-Theta) and to discretize in space with a finite element method, which is the topic in Section 10. But let us first investigate FEM in the next chapter.

6 Theory and finite elements (FEM) for elliptic problems

In this section, we discuss the finite element method (FEM). Our strategy is to introduce the method first for 1D problems in which deeper mathematical results with respect to functional analysis and Sobolev theory are not required for understanding. Later we also provide deeper mathematical results. The discretization parts are again complemented with the corresponding numerical analysis as well as algorithmic aspects and prototype numerical tests. As before, we augment from time to time with nonstandard formulations, such as for instance formulating a prototype coupled problem.

6.1 Preliminaries

6.1.1 Construction of an exact solution (only possible for simple cases!)

In this section, we construct again an exact solution first. In principle this section is the same as Section 5.2.1 except that we derive a very explicit representation of the solution (but we could have done this in Section 5.2.1 as well). We consider again the model problem (D)[2]:

$$-u'' = f \quad \text{in } \Omega = (0,1), \tag{47}$$

$$u(0) = u(1) = 0. \tag{48}$$

Please make yourself clear that this is nothing else than expressing Formulation 4.13 in 1D. Thus, we deal with a second-order elliptic problem.

In 1D, we can derive the explicit solution of (47). First we have

$$u'(\tilde{x}) = -\int_0^{\tilde{x}} f(s)ds + C_1$$

where C_1 is a positive constant. Further integration yields

$$u(x) = \int_0^x u'(\tilde{x})\, d\tilde{x} = -\int_0^x \Big(\int_0^{\tilde{x}} f(s)\, ds\Big) d\tilde{x} + C_1 x + C_2$$

with another integration constant C_2. We have two unknown constants C_1, C_2 but also two given boundary conditions which allows us to determine C_1 and C_2.

$$0 = u(0) = -\int_0^x \Big(\int_0^{\tilde{x}} f(s)\, ds\Big) d\tilde{x} + C_1 \cdot 0 + C_2$$

yields $C_2 = 0$. For C_1, using $u(1) = 0$, we calculate:

$$C_1 = \int_0^1 \Big(\int_0^{\tilde{x}} f(s)\, ds\Big)\, d\tilde{x}.$$

Thus we obtain as final solution:

$$u(x) = -\int_0^x \Big(\int_0^{\tilde{x}} f(s)\, ds\Big) d\tilde{x} + x \left(\int_0^1 \Big(\int_0^{\tilde{x}} f(s)\, ds\Big)\, d\tilde{x}\right).$$

Thus, for a given right hand side f we obtain an explicit expression. For instance $f = 1$ yields:

$$u(x) = \frac{1}{2}(-x^2 + x).$$

It is trivial to double-check that this solution also satisfies the boundary conditions: $u(0) = u(1) = 0$. Furthermore, we see by differentiation that the original equation is obtained:

$$u'(x) = -x + \frac{1}{2} \quad \Rightarrow \quad -u''(x) = 1.$$

[2] (D) stands for differential problem. (M) stands for minimization problem. (V) stands for variational problem.

Exercise 4. *Let $f = -1$.*

- *Compute C_1;*
- *Compute $u(x)$;*
- *Check that $u(x)$ satisfies the boundary conditions;*
- *Check that $u(x)$ satisfies the PDE.*

6.1.2 Equivalent formulations

We first discuss that the solution of (47) (D) is also the solution of a minimization problem (M) and a variational problem (V). To formulate these (equivalent) problems, we first introduce the scalar product

$$(v, w) = \int_0^1 v(x)w(x)\,dx.$$

Furthermore we introduce the linear space

$$V := \{v|\ v \in C[0,1], v' \text{ is piecewise continuous and bounded on } [0,1], v(0) = v(1) = 0\}. \tag{49}$$

We also introduce the linear functional $F : V \to \mathbb{R}$ such that

$$F(v) = \frac{1}{2}(v', v') - (f, v).$$

We state

Definition 6.1. *We deal with three (equivalent) problems:*

(D) Find $u \in C^2$ such that $-u'' = f$ with $u(0) = u(1) = 0$;

(M) Find $u \in V$ such that $F(u) \leq F(v)$ for all $v \in V$;

(V) Find $u \in V$ such that $(u', v') = (f, v)$ for all $v \in V$.

*In physics, the quantity $F(v)$ stands for the **total potential energy** of the underlying model; see also Section 7.3.2. Moreover, the first term in $F(v)$ denotes the internal elastic energy and (f, v) the load potential. Therefore, formulation (M) corresponds to the fundamental **principle of minimal potential energy** and the variational problem (V) to the **principle of virtual work** (e.g., [19]). The proofs of their equivalence will be provided in the following.*

Remark 6.2 (Euler-Lagrange equations)**.** *When (V) is derived from (M), we also say that (V) is the **Euler-Lagrange equation** related to (M).*

In the following we show that the three problems $(D), (M), (V)$ are equivalent.

Proposition 6.3. *It holds*

$$(D) \to (V).$$

Proof. We multiply $u'' = f$ with an arbitrary function ϕ (a so-called **test function**) from the space V defined in (49). Then we integrate over the interval $\Omega = (0, 1)$ yielding

$$-u'' = f \tag{50}$$

$$\Rightarrow -\int_\Omega u''\phi\,dx = \int_\Omega f\phi\,dx \tag{51}$$

$$\Rightarrow \int_\Omega u'\phi'\,dx - u'(1)\phi(1) + u'(0)\phi(0) = \int_\Omega f\phi\,dx \tag{52}$$

$$\Rightarrow \int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx \quad \forall \phi \in V. \tag{53}$$

In the second last term, we used integration by parts.

The boundary terms vanish because $\phi \in V$. This shows that

$$\int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx$$

is a solution of (V). □

Proposition 6.4. *It holds*

$$(V) \leftrightarrow (M).$$

Proof. We first assume that u is a solution to (V). Let $\phi \in V$ and set $w = \phi - u$ such that $\phi = u + w$ and $w \in V$. We obtain

$$\begin{aligned} F(\phi) = F(u+w) &= \frac{1}{2}(u'+w', u'+w') - (f, u+w) \\ &= \frac{1}{2}(u',u') - (f,u) + (u',w') - (f,w) + \frac{1}{2}(w',w') \geq F(u) \end{aligned}$$

We use now the fact that (V) holds true, namely

$$(u', w') - (f, w) = 0.$$

and also that $(w', w') \geq 0$. Thus, we have shown that u is a solution to (M). We show now that $(M) \to (V)$ holds true as well. For any $\phi \in V$ and $\varepsilon \in \mathbb{R}$ we have

$$F(u) \leq F(u + \varepsilon\phi),$$

because $u + \varepsilon\phi \in V$. We differentiate with respect to ε and show that (V) is a first order necessary condition to (M) with a minimum at $\varepsilon = 0$. To do so, we define

$$g(\varepsilon) := F(u+\varepsilon\phi) = \frac{1}{2}(u',u') + \varepsilon(u',\phi') + \frac{\varepsilon^2}{2}(\phi',\phi') - (f,u) - \varepsilon(f,\phi).$$

Thus

$$g'(\varepsilon) = (u',\phi') + \varepsilon(\phi',\phi') - (f,\phi).$$

A minimum is obtained for $\varepsilon = 0$. Consequently,

$$g'(0) = 0.$$

In detail:

$$(u',\phi') - (f,\phi) = 0,$$

which is nothing else than the solution of (V). □

Proposition 6.5. *The solution u to (V) is unique.*

Proof. Assume that u_1 and u_2 are solutions to (V) with

$$\begin{aligned} (u_1', \phi') &= (f,\phi) \quad \forall \phi \in V, \\ (u_2', \phi') &= (f,\phi) \quad \forall \phi \in V. \end{aligned}$$

We subtract these solutions and obtain:

$$(u_1' - u_2', \phi') = 0.$$

We choose as test function $\phi' = u_1' - u_2'$ and obtain

$$(u_1' - u_2', u_1' - u_2') = 0.$$

Thus:

$$u_1'(x) - u_2'(x) = (u_1 - u_2)'(x) = 0.$$

Since the difference of the derivatives is zero, this means that the difference of the functions themselves is constant:

$$(u_1 - u_2)(x) = const$$

Using the boundary conditions $u(0) = u(1) = 0$, yields

$$(u_1 - u_2)(x) = 0.$$

Thus $u_1 = u_2$. □

Remark 6.6. *The last statements are related to the definiteness of the norm. It holds:*

$$\|u\|_{L^2}^2 = \int_\Omega u^2 \, dx.$$

Thus the results follow directly because

$$\|u\| = 0 \quad \Leftrightarrow \quad u = 0.$$

Proposition 6.7. *It holds*

$$(V) \to (D).$$

Proof. We assume that u is a solution to (V), i.e.,

$$(u', \phi') = (f, \phi) \quad \forall \phi \in V.$$

If we assume sufficient regularity of u (in particular $u \in C^2$), then u'' exists and we can integrate backwards. Moreover, we use that $\phi(0) = \phi(1) = 0$ since $\phi \in V$. Then:

$$(-u'' - f, \phi) = 0 \quad \forall \phi \in V.$$

Since we assumed sufficient regularity for u'' and f the difference is continuous. We can now apply the fundamental principle (see Proposition 6.9):

$$w \in C(\Omega) \quad \Rightarrow \quad \int_\Omega w\phi \, dx = 0 \quad \Rightarrow \quad w \equiv 0.$$

We proof this result later. Before, we obtain

$$(-u'' - f, \phi) = 0 \quad \Rightarrow \quad -u'' - f = 0,$$

which yields the desired expression. Since we know that $(D) \to (V)$ holds true, u has the assumed regularity properties and we have shown the equivalence. □

It remains to state and proof the fundamental lemma of calculus of variations. To this end, we introduce

Definition 6.8 (Continuous functions with compact support)**.** *Let $\Omega \subset \mathbb{R}^n$ be an open domain.*

- *The set of continuous functions from Ω to $\mathbb{R}$ with **compact support** is denoted by $C_c(\Omega)$. Such functions vanish on the boundary.*
- *The set of **smooth functions** (infinitely continuously differentiable) with compact support is denoted by $C_c^\infty(\Omega)$.*

Proposition 6.9 (Fundamental lemma of calculus of variations)**.** *Let $\Omega = [a, b]$ be a compact interval and let $w \in C(\Omega)$. Let $\phi \in C^\infty$ with $\phi(a) = \phi(b) = 0$, i.e., $\phi \in C_c^\infty(\Omega)$. If for all ϕ it holds*

$$\int_\Omega w(x)\phi(x) \, dx = 0,$$

then, $w \equiv 0$ in Ω.

Proof. We perform an indirect proof. We suppose that there exist a point $x_0 \in \Omega$ with $w(x_0) \neq 0$. Without loss of generality, we can assume $w(x_0) > 0$. Since w is continuous, there exists a small (open) neighborhood $\omega \subset \Omega$ with $w(x) > 0$ for all $x \in \omega$; otherwise $w \equiv 0$ in $\Omega \setminus \omega$. Let ϕ now be a positive test function (recall that ϕ can be arbitrary, specifically positive if we wish) in Ω and thus also in ω. Then:

$$\int_\Omega w(x)\phi(x)\,dx = \int_\omega w(x)\phi(x)\,dx.$$

But this is a contradiction to the hypothesis on w. Thus $w(x) = 0$ for all in $x \in \omega$. Extending this result to all open neighborhoods in Ω we arrive at the final result. □

6.2 The weak form: defining a bilinear form and a linear form

We continue to consider the variational form in this section and provide more definitions and notation.

We recall the key idea:

- multiply the strong form (D) with a test function,
- integrate,
- apply integration by parts on second-order terms.

The last operation 'weakens' the derivative information because rather evaluating 2nd order derivatives, we only need to evaluate a 1st order derivative on the trial function and another 1st order derivative on the test function.

We recall the procedure how to obtain a weak form:

$$-u'' = f \tag{54}$$

$$\Rightarrow -\int_\Omega u''\phi\,dx = \int_\Omega f\phi\,dx \tag{55}$$

$$\Rightarrow \int_\Omega u'\phi'\,dx - \int_{\partial\Omega} \partial_n u\phi\,ds = \int_\Omega f\phi\,dx \tag{56}$$

$$\Rightarrow \int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx. \tag{57}$$

To summarize we have:

$$\int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx \tag{58}$$

A common short-hand notation in mathematics is to use parentheses for L^2 scalar products: $\int_\Omega ab\,dx =: (a,b)$:

$$(u',\phi') = (f,\phi) \tag{59}$$

A mathematically-correct statement is:

Formulation 6.10. *Find $u \in V$ such that*

$$(u',\phi') = (f,\phi) \quad \forall\phi \in V. \tag{60}$$

In the following, a very common notation is introduced. First, we recall concepts known from linear algebra:

Definition 6.11 (Linear form and bilinear form)**.** *If V is a linear space, l is a **linear form** on V if $l : V \to \mathbb{R}$ or in other words $l(v) \in \mathbb{R}$ for $v \in V$. Moreover, l is linear:*

$$l(av + bw) = al(v) + bl(w) \quad \forall a,b \in R, \quad \forall v,w \in V.$$

*A problem is a **bilinear form** [3] on $V \times V$ if $a : V \times V \to \mathbb{R}$ and $a(v, w) \in \mathbb{R}$ for $v, w \in V$ and $a(v, w)$ is linear in each argument, i.e., for $\alpha, \beta \in R$ and $u, v, w \in V$, it holds*

$$a(u, \alpha v + \beta w) = \alpha a(u, v) + \beta a(u, w),$$
$$a(\alpha u + \beta v, w) = \alpha a(u, w) + \beta a(v, w).$$

*A bilinear form is said to be **symmetric** if*

$$a(u, v) = a(v, u).$$

*A symmetric bilinear form $a(\cdot, \cdot)$ on $V \times V$ defines a **scalar product** on V if*

$$a(v, v) > 0 \quad \forall v \in V, v \neq 0.$$

*The associated **norm** is denoted by $\| \cdot \|_a$ and defined by*

$$\|v\|_a := \sqrt{a(v, v)} \quad \text{for } v \in V.$$

Definition 6.12 (Frobenius scalar product)**.** *For vector-valued functions in second-order problems, we need to work with a scalar product defined for matrices because their gradients are matrices. Here the natural form is the Frobenius scalar product, which is defined as:*

$$< A, B >_F = A : B = \sum_i \sum_j a_{ij} b_{ij}$$

The Frobenius scalar product induces the Frobenius norm:

$$\|A\|_F = \sqrt{< A, B >}.$$

Remark 6.13. *For vector-valued PDEs (such as elasticity, Stokes or Navier-Stokes), we formulate in compact form:*

$$a(u, \phi) = l(\phi)$$

with some $l(\phi)$ and

$$a(u, \phi) = \int_\Omega \nabla u : \nabla \phi \, dx$$

where $\nabla u : \nabla \phi$ is then evaluated in terms of the Frobenius scalar product.

Definition 6.14 (Bilinear and linear forms of the continuous Poisson problem)**.** *For the Poisson problem, we can define:*

$$a(u, \phi) = (u', \phi'),$$
$$l(\phi) = (f, \phi).$$

We shall state the variational form of Poisson's problem in terms of $a(\cdot, \cdot)$ and $l(\cdot)$:

Formulation 6.15 (Variational Poisson problem on the continuous level)**.** *Find $u \in V$ such that*

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V, \tag{61}$$

*where $a(\cdot, \cdot)$ and $l(\cdot)$ are defined in Definition 6.14. The unknown function u is called the **trial** or **(ansatz)** function whereas ϕ is the so-called **test function**.*

[3] For nonlinear problems we deal with a semi-linear form, which is only linear in one argument.

6.3 Finite elements in 1D

In the following we want to concentrate how to compute a discrete solution. As in the previous chapter, this, in principle, allows us to address even more complicated situations and also higher spatial dimensions. The principle of the FEM is rather simple:

- Introduce a mesh $\mathcal{T}_h := \bigcup K_i$ (where K_i denote the single mesh elements) of the given domain $\Omega = (0,1)$ with mesh size (diameter/length) parameter h
- Define on each mesh element $K_i := [x_i, x_{i+1}], i = 0, \ldots, n$ polynomials for trial and test functions. These polynomials must form a basis in a space V_h and they should reflect certain conditions on the mesh edges;
- Use the variational (or weak) form of the given problem and derive a discrete version;
- Evaluate the arising integrals;
- Collect all contributions on all K_i leading to a linear equation system $AU = B$;
- Solve this linear equation system; the solution vector $U = (u_0, \ldots, u_n)^T$ contains the discrete solution at the nodal points $x_0, \ldots, x_n$;
- Verify the correctness of the solution U.

6.3.1 The mesh

Let us start with the mesh. We introduce nodal points and divide $\Omega = (0,1)$ into

$$x_0 = 0 < x_1 < x_2 < \ldots < x_n < x_{n+1} = 1.$$

In particular, we can work with a uniform mesh in which all nodal points have equidistant distance:

$$x_j = jh, \quad h = \frac{1}{n+1}, \quad 0 \leq j \leq n+1, \quad h = x_{j+1} - x_j.$$

Remark 6.16. *An important research topic is to organize the points x_j in certain non-uniform ways in order to reduce the discrete error (see Section 6.11).*

6.3.2 Linear finite elements

In the following we denote P_k the space that contains all polynomials up to order k with coefficients in $\mathbb{R}$:

Definition 6.17.

$$P_k := \{\sum_{i=0}^{k} a_i x^i |\ a_i \in \mathbb{R}\}.$$

In particular we will work with the space of linear polynomials

$$P_1 := \{a_0 + a_1 x |\ a_0, a_1 \in \mathbb{R}\}.$$

A finite element is now a function localized to an element $K_i \in \mathcal{T}_h$ and uniquely defined by the values in the nodal points x_i, x_{i+1}.

We then define the space:

$$V_h^{(1)} = V_h := \{v \in C[0,1] |\ v|_{K_i} \in P_1, K_i := [x_i, x_{i+1}], 0 \leq i \leq n, v(0) = v(1) = 0\}.$$

The boundary conditions are build into the space through $v(0) = v(1) = 0$. This is an important concept that Dirichlet boundary conditions will not appear explicitly later, but are contained in the function spaces.

All functions inside V_h are so called **shape functions** and can be represented by so-called **hat functions**. Hat functions are specifically linear functions on each element K_i. Attaching them yields a hat in the geometrical sense.

For $j = 1, \ldots, n$ we define:

$$\phi_j(x) = \begin{cases} 0 & \text{if } x \notin [x_{j-1}, x_{j+1}] \\ \frac{x - x_{j-1}}{x_j - x_{j-1}} & \text{if } x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{x_{j+1} - x_j} & \text{if } x \in [x_j, x_{j+1}] \end{cases} \tag{62}$$

with the property

$$\phi_j(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}. \tag{63}$$

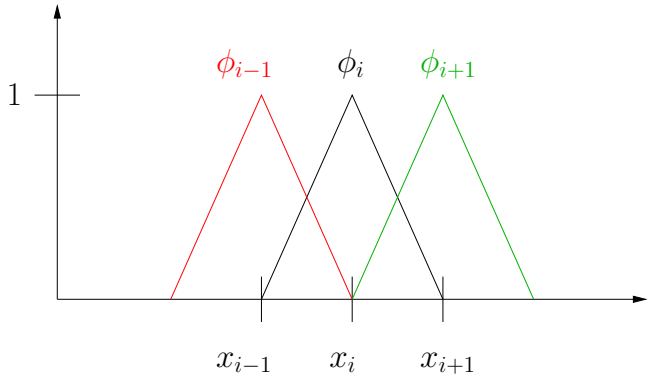

Figure 5: Hat functions.

For a uniform step size $h = x_j - x_{j-1} = x_{j+1} - x_j$ we obtain

$$\phi_j(x) = \begin{cases} 0 & \text{if } x \notin [x_{j-1}, x_{j+1}] \\ \frac{x - x_{j-1}}{h} & \text{if } x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{h} & \text{if } x \in [x_j, x_{j+1}] \end{cases}$$

and for its derivative:

$$\phi_j'(x) = \begin{cases} 0 & \text{if } x \notin [x_{j-1}, x_{j+1}] \\ +\frac{1}{h} & \text{if } x \in [x_{j-1}, x_j] \\ -\frac{1}{h} & \text{if } x \in [x_j, x_{j+1}] \end{cases}$$

Lemma 6.18. *The space V_h is a subspace of $V := C[0,1]$ and has dimension n (because we deal with n basis functions). Thus the such constructed finite element method is a **conforming** method. Furthermore, for each function $v_h \in V_h$ we have a unique representation:*

$$v_h(x) = \sum_{j=1}^{n} v_{h,j} \phi_j(x) \quad \forall x \in [0,1], \quad v_{h,j} \in \mathbb{R}.$$

Proof. Sketch: The unique representation is clear, because in the nodal points it holds $\phi_j(x_i) = \delta_{ij}$, where δ_{ij} is the Kronecker symbol with $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. □

The function $v_h(x)$ connects the discrete values $v_{h,j} \in \mathbb{R}$ and in particular the values between two support points x_j and x_{j+1} can be evaluated.

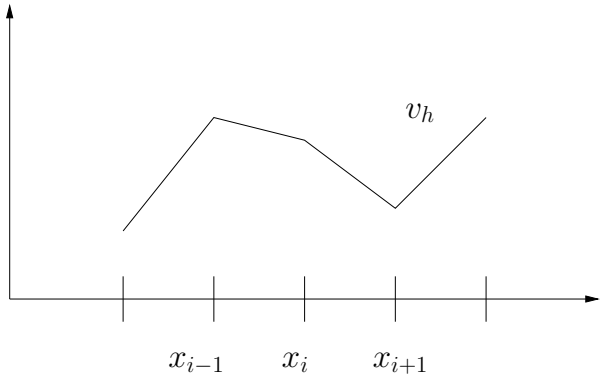

Figure 6: The function $v_h \in V_h$.

Remark 6.19. *The finite element method introduced above is a Lagrange method, since the basis functions ϕ_j are defined only through its values at the nodal points without using derivative information (which would result in Hermite polynomials).*

6.3.3 The process to construct the specific form of the shape functions

In the previous construction, we have hidden the process how to find the specific form of $\phi_j(x)$. For 1D it is more or less clear and we would accept the $\phi_j(x)$ really has the form as it has in (62) . In $\mathbb{R}^n$ this task is a bit of work. To understand this procedure, we explain the process in detail. Here we first address the defining properties of a finite element (see also Section 6.3.6):

- Intervals $[x_i, x_{i+1}]$;
- A linear polynomial $\phi(x) = a_0 + a_1 x$;
- Nodal values at x_i and x_{i+1} (the so-called degrees of freedom).

Later only these properties are stated (see also the general literature in Section 1) and one has to construct the specific $\phi(x)$ such as for example in (62). Thus, the main task consists in finding the unknown coefficients a_0 and a_1 of the shape function. The key property is (63) (also valid in $\mathbb{R}^n$ in order to have a small support) and therefore we obtain:

$$\begin{aligned} \phi_j(x_j) &= a_0 + a_1 x_j = 1, \\ \phi_j(x_i) &= a_0 + a_1 x_i = 0. \end{aligned}$$

To determine a_0 and a_1 we have to solve a small linear equation system:

$$\begin{pmatrix} 1 & x_j \\ 1 & x_i \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

We obtain

$$a_1 = -\frac{1}{x_i - x_j}$$

and

$$a_0 = \frac{x_i}{x_i - x_j}.$$

Then:

$$\phi_j(x) = a_0 + a_1 x = \frac{x_i - x}{x_i - x_j}.$$

At this stage we have now to distinguish whether $x_j := x_{i-1}$ or $x_j := x_{i+1}$ or $|i - j| > 1$ yielding the three cases in (62).

Remark 6.20. *Of course, for higher-order polynomials and higher-order problems in $\mathbb{R}^n$, the matrix system to determining the coefficients becomes larger. However, in all these state-of-the-art FEM software packages, the shape functions are already implemented.*

Remark 6.21. *A very practical and detailed derivation of finite elements in different dimensions can be found in [63].*

6.3.4 The discrete weak form

We have set-up a mesh and local polynomial functions with a unique representation, all these developments result in a 'finite element'. Now, we use the variational formulation and derive the discrete counterpart:

Formulation 6.22. *Find $u_h \in V_h$ such that*

$$(u_h', \phi_h') = (f, \phi_h) \quad \forall \phi_h \in V_h. \tag{64}$$

Or in the previously introduced compact form:

Formulation 6.23 (Variational Poisson problem on the discrete level)**.** *Find $u_h \in V_h$ such that*

$$a(u_h, \phi_h) = l(\phi_h) \quad \forall \phi_h \in V_h, \tag{65}$$

where $a(\cdot,\cdot)$ and $l(\cdot)$ are defined in Definition 6.14 by adding h as subindex for u and ϕ.

Remark 6.24 (Galerkin method)**.** *The process going from V to V_h using the variational formulation is called **Galerkin method**. Here, it is not necessary that the bilinear form is symmetric. As further information: not only is Galerkin's method a numerical procedure, but it is also used in analysis when establishing existence of the continuous problem. Here, one starts with a finite dimensional subspace and constructs a sequence of finite dimensional subspaces $V_h \subset V$ (namely passing with $h \to 0$; that is to say: we add more and more basis functions such that $dim(V_h) \to \infty$). The idea of numerics is the same: finally we are interested in small h such that we obtain a discrete solution with sufficient accuracy.*

Remark 6.25 (Ritz method)**.** *If we discretize the minimization problem (M), the above process is called **Ritz method**. In particular, the bilinear form of the variational problem is symmetric.*

Remark 6.26 (Ritz-Galerkin method)**.** *For general bilinear forms (i.e., not necessarily symmetric) the discretization procedure is called **Ritz-Galerkin method**.*

Remark 6.27 (Petrov-Galerkin method)**.** *In a **Petrov-Galerkin method** the trial and test spaces can be different.*

In order to proceed we can express $u_h \in V_h$ with the help of the basis functions ϕ_j in $V_h := \{\phi_1, \ldots, \phi_n\}$, thus:

$$u_h = \sum_{j=1}^{n} u_j \phi_j(x), \quad u_j \in \mathbb{R}.$$

Since (64) holds for all $\phi_i \in V_h$ for $1 \leq i \leq n$, it holds in particular for each i:

$$(u_h', \phi_i') = (f, \phi_i) \quad \text{for } 1 \leq i \leq n. \tag{66}$$

We now insert the representation for u_h in (66), yielding the **Galerkin equations**:

$$\sum_{j=1}^{n} u_j (\phi_j', \phi_i') = (f, \phi_i) \quad \text{for } 1 \leq i \leq n. \tag{67}$$

We have now extracted the coefficient vector of u_h and only the **shape functions** ϕ_j and ϕ_i (i.e., their derivatives of course) remain in the integral.

This yields a linear equation system of the form

$$AU = B$$

where

$$U = (u_j)_{1\le j\le n} \in \mathbb{R}^n, \tag{68}$$

$$B = \big((f,\phi_i)\big)_{1\le i\le n} \in \mathbb{R}^n, \tag{69}$$

$$A = \Big((\phi_j',\phi_i')\Big)_{1\le j,i\le n} \in \mathbb{R}^{n\times n}. \tag{70}$$

Thus the final solution vector is U which contains the values u_j at the nodal points x_j of the mesh. Here we remark that x_0 and x_{n+1} are not solved in the above system and are determined by the boundary conditions $u(x_0) = u(0) = 0$ and $u(x_{n+1}) = u(1) = 0$.

Remark 6.28 (Regularity of A)**.** *It remains the question whether A is regular such that A^{-1} exists. We show this rigorously using Assumption No. 3 in Definition 6.129 in Section 6.10.3.*

6.3.5 Evaluation of the integrals

It remains to determine the specific entries of the system matrix (also called stiffness matrix) A and the right hand side vector B. Since the basis functions have only little support on two neighboring elements (in fact that is one of the key features of the FEM) the resulting matrix A is sparse, i.e., it contains only a few number of entries that are not equal to zero.

Let us now evaluate the integrals that form the entries a_{ij} of the matrix $A = (a_{ij})_{1\le i,j\le n}$. For the diagonal elements we calculate:

$$a_{ii} = \int_\Omega \varphi_i'(x)\varphi_i'(x)\,dx = \int_{x_{i-1}}^{x_{i+1}} \varphi_i'(x)\varphi_i'(x)\,dx \tag{71}$$

$$= \int_{x_{i-1}}^{x_i} \frac{1}{h^2}\,dx + \int_{x_i}^{x_{i+1}} \Big(-\frac{1}{h}\Big)^2\,dx \tag{72}$$

$$= \frac{h}{h^2} + \frac{h}{h^2} \tag{73}$$

$$= \frac{2}{h}. \tag{74}$$

For the right off-diagonal we have:

$$a_{i,i+1} = \int_\Omega \varphi_{i+1}'(x)\varphi_i'(x)\,dx = \int_{x_i}^{x_{i+1}} \frac{1}{h}\cdot\Big(-\frac{1}{h}\Big)\,dx = -\frac{1}{h}. \tag{75}$$

It is trivial to see that $a_{i,i+1} = a_{i-1,i}$.

In compact form we summarize:

$$a_{ij} = \int_\Omega \phi_j'(x)\phi_i'(x)\,dx = \begin{cases} -\frac{1}{x_{j+1}-x_j} & \text{if } j = i-1 \\ \frac{1}{x_j-x_{j-1}} + \frac{1}{x_{j+1}-x_j} & \text{if } j = i \\ -\frac{1}{x_j-x_{j-1}} & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases}. \tag{76}$$

For a uniform mesh (as we assume in this section) we can simplify the previous calculation since we know that $h = h_j = x_{j+1} - x_j$:

$$a_{ij} = \int_\Omega \phi_j'(x)\phi_i'(x)\,dx = \begin{cases} -\frac{1}{h} & \text{if } j = i-1 \\ \frac{2}{h} & \text{if } j = i \\ -\frac{1}{h} & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases}. \tag{77}$$

The resulting matrix A for the 'inner' points $x_1, \ldots, x_n$ reads then:

$$A = h^{-1} \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Remark 6.29. *Since the boundary values at $x_0 = 0$ and $x_{n+1} = 1$ are known to be $u(0) = u(1) = 0$, they are not assembled in the matrix A. We could have considered all support points $x_0, x_1, \ldots, x_n, x_{n+1}$ we would have obtained:*

$$A = h^{-1} \begin{pmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{pmatrix} \in \mathbb{R}^{(n+2) \times (n+2)}.$$

Here the entries $a_{00} = a_{n+1,n+1} = 1$ (and not 2) because at the boundary only the half of the two corresponding test functions do exist. Furthermore, working with this matrix A in the solution process below, we have to modify the entries $a_{0,1} = a_{1,0} = a_{n,n+1} = a_{n+1,n}$ from the value -1 to 0 such that $u_{h,0} = u_{h,n+1} = 0$ can follow.

It remains to evaluate the integrals of the right hand side vector b. Here the main difficulty is that the given right hand side f may be complicated. Of course it always holds for the entries of b:

$$b_i = \int_\Omega f(x)\phi_i(x)\,dx \quad \text{for } 1 \le i \le n.$$

The right hand side now depends explicitly on the values for f. Let us assume a constant $f = 1$ (thus the original problem would be $-u'' = 1$), then:

$$b_i = \int_\Omega 1 \cdot \phi_i(x)\,dx = 1 \cdot \int_\Omega \phi_i(x)\,dx = 1 \cdot h \quad \text{for } 1 \le i \le n.$$

Namely

$$B = (h, \ldots, h)^T.$$

For non-uniform step sizes we obtain:

$$\frac{1}{h_i} \int_{K_i} \phi_i(x)\,dx, \quad \text{and} \quad \frac{1}{h_{i+1}} \int_{K_{i+i}} \phi_i(x)\,dx$$

with $h_i = x_i - x_{i-1}, h_{i+1} = x_{i+1} - x_i$ and $K_i = [x_{i-1}, x_i]$ and $K_{i+1} = [x_i, x_{i+1}]$.

Exercise 5. *Derive the system matrix A by assuming non-uniform step sizes $h_j = x_j - x_{j+1}$.*

Exercise 6. *Derive the system matrix A for a P_2 discretization, namely working with quadratic basis functions. Sketch how the basis functions look like.*

6.3.6 Definition of a finite element

We briefly summarize the key ingredients that define a **finite element**. A finite element is a triple (K, P_K, Σ) where

- K is an element, i.e., a geometric object (in 1D an interval);
- $P_k(K)$ is a finite dimensional linear space of polynomials defined on K;
- Σ, not introduced so far, is a set of degrees of freedom (DoF), e.g., the values of the polynomial at the vertices of K.

These three ingredients yield a uniquely determined polynomial on an element K.

6.3.7 Properties of the system matrix A

We first notice:

Remark 6.30. *The system matrix A has a factor $\frac{1}{h}$ whereas in the corresponding matrix A, using a finite difference scheme, we have a factor $\frac{1}{h^2}$. The reason is that we integrate the weak form using finite elements and one h is hidden in the right hand side vector b. Division by h we yield the same system for finite elements as for finite differences.*

Next, we show that A is symmetric and positive definite. It holds

$$(\phi_i', \phi_j') = (\phi_j', \phi_i').$$

Using the representation

$$u(x) = \sum_{j=1}^{n} u_{j,h}\phi_{j,h}(x),$$

we obtain

$$\sum_{i,j=1}^{n} u_i(\phi_i', \phi_j')u_j = (\sum_{i,j=1}^{n} u_i\phi_i', \sum_{i,j=1}^{n} u_j\phi_j') = (u', u') \geq 0.$$

We have

$$(u', u') = 0$$

only if $u' \equiv 0$ since $u(0) = 0$ only for $u \equiv 0$ or $u_{j,h} = 0$ for $j = 1, \ldots, n$. We recall that a symmetric matrix $B \in \mathbb{R}^{n\times n}$ is said to be positive definite if

$$\xi \cdot B\xi = \sum_{i,j=1}^{n} \xi_i b_{ij} \xi_j > 0 \quad \forall \xi \in \mathbb{R}^n, \xi \neq 0.$$

Finally, it holds:

Proposition 6.31. *A symmetric positive matrix B is positive definite if and only if the eigenvalues of B are strictly positive. Furthermore, a positive definite matrix is regular and consequently, the linear system to which B is associated with has a unique solution.*

Proof. Results from linear algebra. □

6.3.8 Numerical test: 1D Poisson

We compute the same test as in Section 5.7, but now with finite elements. Since we can explicitly derive all entries of the system matrix A as in the finite difference method, everything would result in the same operations as in Section 5.7. Rather we adopt deal.II [4, 6] an open source C++ finite element code to carry out this computation. Even that in such packages many things are hidden and performed in the background, deal.II leaves enough room to **see** the important points:

- Creating a domain Ω and decomposition of this domain into elements;
- Setting up the vectors u, b and the matrix A with their correct length;
- Assembling (not yet solving!) the system $Au = b$. We compute locally on each mesh element the matrix entries and right hand side entries on a master cell and insert the respective values at their global places in the matrix A;
- In the assembly, the integrals are evaluated using numerical quadrature in order to allow for more general expressions when for instance the material parameter a is not constant to 1 or when higher-order polynomials need to be evaluated;
- Solution of the linear system $Au = b$ (the implementation of the specific method is hidden though;

- Postprocessing of the solution: here writing the solution data $U_j, 1 \leq j \leq n$ into a file that can be read from a graphic visualization program such as gnuplot, paraview, visit.

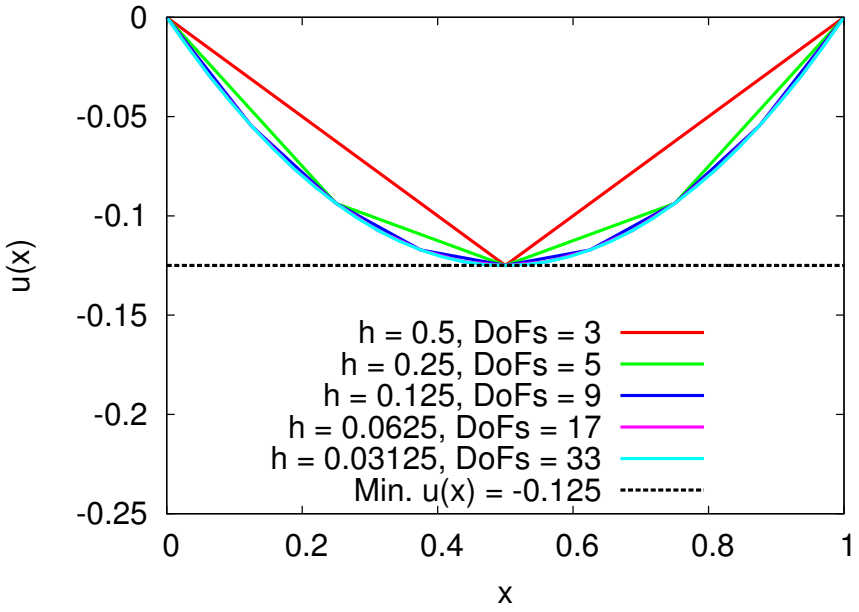

Figure 7: Solution of the 1D Poisson problem with $f = -1$ using finite elements with various mesh sizes h. DoFs is the abbreviation for degrees of freedom; here the number of support points x_j. The dimension of the discrete space is $DoFs$. For instance for $h = 0.5$, we have 3 DoFs and two basis functions, thus $dim(V_h) = 3$. The numerical solutions are computed with an adaptation of step-3 in deal.II [4, 6]. Please notice that the picture norm is not a proof in the strict mathematical sense: to show that the purple, and blue lines come closer and closer must be confirmed by error estimates as presented in Section 6.11 accompanied by numerical simulations in Section 6.13. Of course, for this 1D Poisson problem, we easily observe a limit case, but for more complicated equations it is often not visible whether the solutions do converge.

Level	Elements	DoFs
1	2	3
2	4	5
3	8	9
4	16	17
5	32	33

6.4 Algorithmic details

We provide some algorithmic details for implementing finite elements in a computer code. Of course, for constant coefficients, a 'nice' domain Ω with uniform step lengths, we can immediately build the matrix, right hand side vector and compute the solution. In practice, we are interested in more general formulations. Again, we explain all details in 1D, and partially follow Suttmeier's lecture [64]. Further remarks and details (in particular for higher dimensions) can be found in the literature provided in Section 1.

6.4.1 Assembling integrals on each cell K

In practice, one does not proceed as in Section 6.3.5 since this only makes sense for very simple problems.

A more general procedure is based on an element-wise consideration, which is motivated by:

$$a(u,\phi) = \int_\Omega u'\,\phi'\,dx = \sum_{K\in\mathcal{T}_h} \int_K u'\,\phi'\,dx.$$

The basic procedure is:

Algorithm 6.32 (Basic assembling - robust, but partly inefficient)**.** *Let $K_s, s = 0,\dots n$ be an element and let i and j be the indices of the degrees of freedom (namely the basis functions). The basic algorithm to compute all entries of the system matrix and right hand side vector is:*

$$\begin{aligned}
&\textit{for all elements } K_s \textit{ with } s = 0,\dots,n\\
&\quad \textit{for all DoFs } i \textit{ with } i = 0,\dots,n+1\\
&\quad\quad \textit{for all DoFs } j \textit{ with } j = 0,\dots,n+1\\
&\quad\quad\quad a_{ij} += \int_{K_s} \phi_i'(x)\,\phi_j'(x)\,dx
\end{aligned}$$

Here $+=$ means that entries with the same indices i, j are summed. For the right hand side, we have

$$\begin{aligned}
&\textit{for all elements } K_s \textit{ with } s = 0,\dots,n\\
&\quad \textit{for all DoFs } i \textit{ with } i = 0,\dots,n+1\\
&\quad\quad b_i += \int_{K_s} f(x)\phi_i(x)\,dx.
\end{aligned}$$

Again $+=$ means that only the b_i with the same i are summed.

Remark 6.33. *This algorithm is a bit inefficient since a lot of zeros are added. Knowing in advance the polynomial degree of the shape functions allows to add an if-condition to assemble only non-zero entries.*

6.4.2 Example in $\mathbb{R}^5$

We illustrate the previous algorithm for a concrete example. Let us compute 1D Poisson on four support points $x_i, i = 0,1,2,3,4$ that are equidistantly distributed yielding a uniform mesh size $h = x_j - x_{j-1}$. The discrete space V_h is given by:

$$V_h = \{\phi_0,\phi_1,\phi_2,\phi_3,\phi_4\}, \quad dim(V_h) = 5.$$

The number of cells is $\#K = 4$.

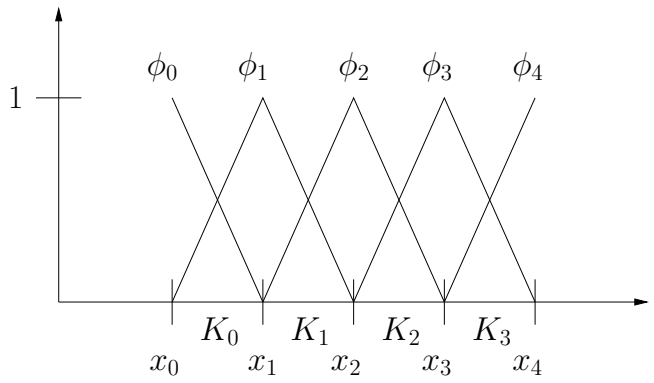

Figure 8: Basis functions ϕ_i, elements K_s and nodes x_s used in Section 6.4.2.

It holds furthermore:

$$U \in \mathbb{R}^5, \quad A \in \mathbb{R}^{5\times 5}, \quad b \in \mathbb{R}^5.$$

We start with $s = 0$, namely K_0:

$$a_{00}^{s=0} = a_{00} = \int_{K_0} \phi_0'\phi_0' = \frac{1}{h}, \quad a_{01}^{s=0} = a_{01} = \int_{K_0} \phi_0'\phi_1' = -\frac{1}{h},$$
$$a_{02}^{s=0} = a_{02} = \int_{K_0} \phi_0'\phi_2' = 0, \quad a_{03}^{s=0} = a_{03} = \int_{K_0} \phi_0'\phi_3' = 0, \quad a_{04}^{s=0} = a_{04} = \int_{K_0} \phi_0'\phi_4' = 0.$$

- Similarly, we evaluate $a_{1j}, a_{2j}, a_{3j}, a_{4j}, j = 0, \ldots 4$.
- Next, we increment $s = 1$ and work on cell K_1. Here we again evaluate all a_{ij} and sum them to the previously obtained values on K_0. Therefore the $+=$ in the above algorithm.
- We also see that we add a lot of zeros when $|i-j| > 1$. For this reason, a good algorithm first design the sparsity pattern and determines the entries of A that are non-zero. This is clear due to the construction of the hat functions and that they only overlap on neighboring elements.

After having assembled the values on all four elements $K_s, s = 1, 2, 3, 4$, we obtain the following system matrix:

$$A = \begin{pmatrix} \sum_s a_{00}^s & \sum_s a_{01}^s & \sum_s a_{02}^s & \sum_s a_{03}^s & \sum_s a_{04}^s \\ \sum_s a_{10}^s & \sum_s a_{11}^s & \sum_s a_{12}^s & \sum_s a_{13}^s & \sum_s a_{14}^s \\ \sum_s a_{20}^s & \sum_s a_{21}^s & \sum_s a_{22}^s & \sum_s a_{23}^s & \sum_s a_{24}^s \\ \sum_s a_{30}^s & \sum_s a_{31}^s & \sum_s a_{32}^s & \sum_s a_{33}^s & \sum_s a_{34}^s \\ \sum_s a_{40}^s & \sum_s a_{41}^s & \sum_s a_{42}^s & \sum_s a_{43}^s & \sum_s a_{44}^s \end{pmatrix} = \frac{1}{h}\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

To fix the homogeneous Dirichlet conditions, we can manipulate directly the matrix A or work with a 'constraint matrix'. We eliminate the entries of the rows and columns of the off-diagonals corresponding to the boundary indices; here $i = 0$ and $i = 4$. Then:

$$A = \frac{1}{h}\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

6.4.3 Numerical quadrature

As previously stated, the arising integrals may easily become difficult such that a direct integration is not possible anymore:

- Non-constant right hand sides $f(x)$ and non-constant coefficients $\alpha(x)$;
- Higher-order shape functions;
- Non-uniform step sizes, more general domains.

In modern FEM programs, Algorithm 6.32 is complemented by an alterative evaluation of the integrals using numerical quadrature. The general formula reads:

$$\int_\Omega g(x) \approx \sum_{l=0}^{n_q} \omega_l g(q_l)$$

with quadrature weights ω_l and quadrature points q_l. The number of quadrature points is $n_q + 1$.

Remark 6.34. *The support points x_i and q_l do not need to be necessarily the same. For Gauss quadrature, they are indeed different. For lowest-order interpolatory quadrature rules (box, Trapez) they correspond though.*

We continue the above example by choosing the trapezoidal rule, which in addition, should integrate the arising integrals exactly:

$$\int_{K_s} g(x) \approx h_s \sum_{l=0}^{n_q} \omega_l g(q_l)$$

where h_s is the length of interval/element K_s, $n_q = 1$ and $\omega_l = 0.5$. This brings us to:

$$\int_{K_s} g(x) \approx h_s \frac{g(q_0) + g(q_1)}{2}.$$

Applied to our matrix entries, we have on an element K_s:

$$a_{ii} = \int_{K_s} \phi_i'(x)\phi_i'(x)\,dx \approx \frac{h_s}{2}\Big(\phi_i'(q_0)\phi_i'(q_0) + \phi_i'(q_1)\phi_i'(q_1)\Big).$$

For the right hand side, we for the case $f = 1$ we can use for instance the mid-point rule:

$$\frac{1}{h_i}\int_{K_i} \phi_i(x)\,dx \approx \frac{1}{h_i} h_i \phi_i\left(\frac{x_i + x_{i-1}}{2}\right) = \phi_i\left(\frac{x_i + x_{i-1}}{2}\right).$$

Remark 6.35. *If $f = f(x)$ with an dependency on x, we should use a quadrature formula that integrates the function $f(x)\phi_i(x)$ as accurate as possible.*

Remark 6.36. *It is important to notice that the order of the quadrature formula must be sufficiently high since otherwise the quadrature error dominates the convergence behavior of the FEM scheme.*

We have now all ingredients to extend Algorithm 6.32:

Algorithm 6.37 (Assembling using the trapezoidal rule)**.** *Let $K_s, s = 0, \dots n$ be an element and let i and j be the indices of the degrees of freedom (namely the basis functions). The basic algorithm to compute all entries of the system matrix A and right hand side vector b is:*

for all elements K_s with $s = 0, \dots, n$
for all DoFs i with $i = 0, \dots, n+1$
for all DoFs j with $j = 0, \dots, n+1$
for all quad points l with $l = 0, \dots, n_q$

$$a_{ij} += \frac{h_s}{2}\phi_i'(q_l)\phi_j'(q_l)$$

where $n_q = 1$. Here $+=$ means that entries with the same indices are summed. This is necessary because on all cells K_s we assemble again a_{ij}.

$$\begin{aligned}&\textit{for all elements } K_s \textit{ with } s = 0, \dots, n\\ &\quad \textit{for all DoFs } i \textit{ with } i = 0, \dots, n+1\\ &\quad\quad \textit{for all quad points } l \textit{ with } l = 0, \dots, n_q\\ &\quad\quad\quad b_i += \frac{h_s}{2} f(q_l)\phi_i(q_l)\end{aligned}$$

Remark 6.38. *In practice it is relatively easy to reduce the computational cost when a lot of zeros are computed. We know due to the choice of the finite element, which DoFs are active on a specific element and can assemble only these shape functions. For linear elements, we could easily add an if condition that only these a_{ij} are assembled when $|i-j| \leq 1$. We finally remark that these loops will definitely work, but there is a second aspect that needs to be considered in practice which is explained in Section 6.4.4.*

6.4.4 Details on the evaluation on the master element

In practice, all integrals are transformed onto a **master element** (or so-called **reference element**) and evaluated there. This has the advantage that

- we only need to evaluate once all basis functions;
- numerical integration formulae are only required on the master element;
- independence of the coordinate system. For instance quadrilateral elements in 2D change their form when the coordinate system is rotated [56].

The price to pay is to compute at each step a deformation gradient and a determinant, which is however easier than evaluating all the integrals.

We consider the (physical) element $K_i^{(h_i)} = [x_i, x_{i+1}], i = 0, \dots n$ and the variable $x \in K_i^{(h_i)}$ with and $h_i = x_{i+1} - x_i$. Without loss of generality, we work in the following with the first element $K_0^{(h_0)} = [x_0, x_1]$ and $h = h_0 = x_1 - x_0$ as shown in Figure 9. The generalization to s elements is briefly discussed in Section 6.4.5.

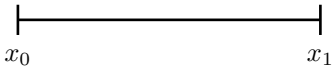

Figure 9: The mesh element $K_0^{(h_0)}$.

The element $K_i^{(h_i)}$ is transformed to the master element (i.e., the unit interval with mesh size $h = 1$) $K^{(1)} := [0, 1]$ with the local variable $\xi \in [0, 1]$ as shown in Figure 10.

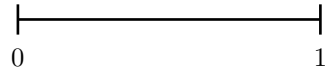

Figure 10: The master element K_1.

For the transformations, we work with the substitution rule (see Section 3.13). Here in 1D, and in higher dimensions with the analogon. We define the mapping

$$\begin{aligned}T_h : K^{(1)} &\to K_0^{(h_0)}\\ \xi &\mapsto T_h(\xi) = x = x_0 + \xi \cdot (x_1 - x_0) = x_0 + \xi h.\end{aligned}$$

While function values can be identified in both coordinate systems, i.e.,

$$f(x) = \hat{f}(\xi), \quad \hat{f} \text{ defined in } K^{(1)},$$

derivatives will be complemented by further terms due to the chain rule that we need to employ. Differentiation in the physical coordinates yields

$$\begin{aligned} \frac{d}{dx}: \quad & 1 = (x_1 - x_0)\,\frac{d\xi}{dx} \\ \Rightarrow \quad & dx = (x_1 - x_0)\cdot d\xi. \end{aligned}$$

The volume (here in 1D: length) change can be represented by the determinant of the Jacobian of the transformation:

$$J := x_1 - x_0 = h.$$

These transformations follow exactly the way as they are known in continuum mechanics. Thus for higher dimensions we refer exemplarily to [39], where transformations between different domains can be constructed.

We now construct the inverse mapping

$$\begin{aligned} T_h^{-1} : K_0^{(h_0)} &\to K^{(1)} \\ x &\mapsto T_h^{-1}(x) = \xi = \frac{x - x_0}{x_1 - x_0} = \frac{x - x_0}{h}, \end{aligned}$$

with the derivative

$$\partial_x T_h^{-1}(x) = \xi_x = \frac{d\xi}{dx} = \frac{1}{x_1 - x_0}.$$

A basis function φ_i^h on $K_0^{(h_0)}$ reads:

$$\varphi_i^h(x) := \varphi_i^1(T_h^{-1}(x)) = \varphi_i^1(\xi)$$

and for the derivative we obtain with the chain rule:

$$\partial_x \varphi_i^h(x) = \partial_\xi \varphi_i^1(\xi)\cdot \partial_x T_h^{-1}(x) = \partial_\xi \varphi_i^1(\xi)\cdot \xi_x$$

with $T_h^{-1}(x) = \xi$.

Example 6.39. *We provide two examples. Firstly:*

$$\int_{K_h} f(x)\,\varphi_i^h(x)\,dx \overset{Sub.}{=} \int_{K^{(1)}} f(T_h(\xi))\cdot \varphi_i^1(\xi)\cdot J\cdot d\xi, \tag{78}$$

and secondly,

$$\int_{K_h} \partial_x \varphi_i^h(x)\cdot \partial_x \varphi_j^h(x)\,dx = \int_{K^{(1)}} \Big(\partial_\xi \varphi_i^1(\xi)\Big)\cdot \xi_x \cdot \Big(\partial_\xi \varphi_j^1(\xi)\Big)\cdot \xi_x \cdot J\,d\xi. \tag{79}$$

We can now apply numerical integration using again the trapezoidal rule and obtain for the two previous integrals:

$$\int_{T_h} f(x)\,\varphi_i^h(x)\,dx \overset{(143)}{\approx} \sum_{k=1}^{q} \omega_k f(F_h(\xi_k))\,\varphi_i^1(\xi_k)\cdot J$$

and for the second example:

$$\int_{T_h} \partial_x \varphi_j^h(x)\partial_x \varphi_i^h(x)\,dx \approx \sum_{k=1}^{q} \omega_k \Big(\partial_\xi \varphi_j^1(\xi_k)\cdot \xi_x\Big)\cdot \Big(\partial_\xi \varphi_i^1(\xi_k)\cdot \xi_x\Big)\cdot J.$$

Remark 6.40. *These final evaluations can again be realized by using Algorithm 6.37, but are now performed on the unit cell $K^{(1)}$.*

6.4.5 Generalization to s elements

We briefly setup the notation to evaluate the integrals for s elements:

- Let n be the index of the end point $x_n = b$ (b is the nodal point of the right boundary). Then, $n-1$ is the number of elements (intervals in 1d), and $n+1$ is the number of the nodal points (degrees of freedom - DoFs) and the number shape functions, respectively (see also Figure 8);
- $K_s^{(h_s)} = [x_s, x_{s+1}], s = 0, \dots n-1.$
- $h_s = x_{s+1} - x_s$;
- $T_s : K^{(1)} \to K_s^{(h_s)} : \xi \mapsto T_s(\xi) = x_s + \xi(x_{s+1} - x_s) = x_s + \xi h_s$;
- $T_s^{-1} : K_s^{(h_s)} \to K^{(1)} : x \mapsto T_s^{-1}(x) = \frac{x - x_s}{h_s}$;
- $\nabla T_s^{-1}(x) = \partial_x T_s^{-1}(x) = \frac{1}{h_s}$ (in 1D);
- $\nabla T_s(\xi) = \partial_x T_s(\xi) = (x_s + \xi h_s)' = h_s$ (in 1D);
- $J_s := \det(\nabla T_s(\xi)) = (x_s + \xi h_s)' = h_s$ (in 1D).

6.4.6 Example: Section 6.4.2 continued using Sections 6.4.4 and 6.4.5

We continue Example 6.4.2 and build the system matrix A by using the master element. In the following, we evaluate the Laplacians in weak form:

$$\begin{aligned} a_{ij} = \int_{K_s} \phi_i'(x) \cdot \phi_j'(x)\, dx &= \int_{K^{(1)}} \phi_i'(\xi)\, \partial_x T_s^{-1}(x) \cdot \phi_j'(\xi)\, \partial_x T_s^{-1}(x)\, J_s\, d\xi \\ &= \int_{K^{(1)}} \phi_i'(\xi)\, \frac{1}{h_s} \cdot \phi_j'(\xi)\, \frac{1}{h_s}\, h_s\, d\xi \\ &= \int_{K^{(1)}} \phi_i'(\xi) \cdot \phi_j'(\xi)\, \frac{1}{h_s}\, d\xi. \end{aligned}$$

We now compute **once!** the required integrals on the unit element (i.e., the master element) $K^{(1)}$. Here, $\xi_0 = 0$ and $\xi_1 = 1$ with $h^{(1)} = \xi_1 - \xi_0 = 1$ resulting in two shape functions:

$$\phi_0(\xi) = \frac{\xi_1 - \xi}{h^{(1)}}, \qquad \phi_1(\xi) = \frac{\xi - \xi_0}{h^{(1)}}.$$

The derivatives are given by:

$$\phi_0'(\xi) = \frac{-1}{h^{(1)}} = -1, \qquad \phi_1'(\xi) = \frac{1}{h^{(1)}} = 1.$$

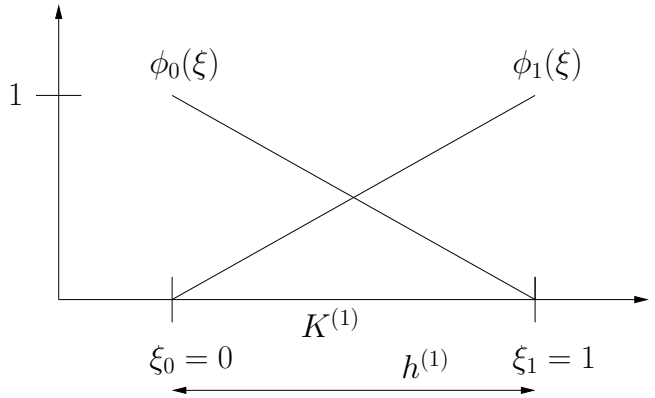

Figure 11: The master element $K^{(1)}$ including nodal points and mesh size parameter $h^{(1)}$.

With these calculations, we compute all combinations on the master element required to evaluate the Laplacian in 1d:

$$\begin{aligned}
a_{00}^{(1)} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_0'(\xi)\, d\xi = \int_{K^{(1)}} (-1)^2\, d\xi = 1,\\
a_{01}^{(1)} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_1'(\xi)\, d\xi = \int_{K^{(1)}} (-1)\cdot 1\, d\xi = -1,\\
a_{10}^{(1)} &= \int_{K^{(1)}} \phi_1'(\xi) \cdot \phi_0'(\xi)\, d\xi = \int_{K^{(1)}} 1\cdot (-1)\, d\xi = -1,\\
a_{11}^{(1)} &= \int_{K^{(1)}} \phi_1'(\xi) \cdot \phi_1'(\xi)\, d\xi = \int_{K^{(1)}} 1\cdot 1\, d\xi = 1.
\end{aligned}$$

It is clear what happens on each element K_s using Algorithm 6.32:

$$\begin{aligned}
a_{00}^{s} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_0'(\xi)\, \frac{1}{h_s}\, d\xi = \frac{1}{h_s}\int_{K^{(1)}} (-1)^2\, d\xi = \frac{1}{h_s},\\
a_{01}^{s} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_1'(\xi)\, \frac{1}{h_s}\, d\xi = \frac{1}{h_s}\int_{K^{(1)}} (-1)\cdot 1\, d\xi = \frac{-1}{h_s},\\
a_{02}^{s} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_2'(\xi)\, \frac{1}{h_s}\, d\xi = \frac{1}{h_s}\int_{K^{(1)}} (-1)\cdot 0\, d\xi = 0,\\
a_{03}^{s} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_3'(\xi)\, \frac{1}{h_s}\, d\xi = \frac{1}{h_s}\int_{K^{(1)}} (-1)\cdot 0\, d\xi = 0,\\
a_{04}^{s} &= \int_{K^{(1)}} \phi_0'(\xi) \cdot \phi_4'(\xi)\, \frac{1}{h_s}\, d\xi = \frac{1}{h_s}\int_{K^{(1)}} (-1)\cdot 0\, d\xi = 0.
\end{aligned}$$

Next, we increase $i \to i+1$ resulting in $i = 1$ and compute:

$$a_{10}^{s}, \quad a_{11}^{s}, \quad a_{12}^{s}, \quad a_{13}^{s}, \quad a_{14}^{s}.$$

We proceed until $i = 4$. This procedure is done for all elements K_s with $s = 0, 1, 2, 3$. As stated in Algorithm 6.32, the procedure is however inefficient since a lot of zeros are assembled. Knowing the structure of the matrix (i.e., the sparsity pattern) allows us upfront only to assemble the entries with non-zero entries.

Anyhow, the system matrix is composed by (the same as in Section 6.4.2):

$$\begin{aligned}
A &= \begin{pmatrix}
\sum_s a_{00}^s & \sum_s a_{01}^s & \sum_s a_{02}^s & \sum_s a_{03}^s & \sum_s a_{04}^s \\
\sum_s a_{10}^s & \sum_s a_{11}^s & \sum_s a_{12}^s & \sum_s a_{13}^s & \sum_s a_{14}^s \\
\sum_s a_{20}^s & \sum_s a_{21}^s & \sum_s a_{22}^s & \sum_s a_{23}^s & \sum_s a_{24}^s \\
\sum_s a_{30}^s & \sum_s a_{31}^s & \sum_s a_{32}^s & \sum_s a_{33}^s & \sum_s a_{34}^s \\
\sum_s a_{40}^s & \sum_s a_{41}^s & \sum_s a_{42}^s & \sum_s a_{43}^s & \sum_s a_{44}^s
\end{pmatrix}\\
&= \begin{pmatrix}
\frac{1}{h_0} & \frac{-1}{h_0} & 0 & 0 & 0 \\
\frac{-1}{h_0} & \frac{1}{h_0}+\frac{1}{h_1} & \frac{-1}{h_1} & 0 & 0 \\
0 & \frac{-1}{h_1} & \frac{1}{h_1}+\frac{1}{h_2} & \frac{-1}{h_2} & 0 \\
0 & 0 & \frac{-1}{h_2} & \frac{1}{h_2}+\frac{1}{h_3} & \frac{-1}{h_3} \\
0 & 0 & 0 & \frac{-1}{h_3} & \frac{1}{h_3}
\end{pmatrix}.
\end{aligned}$$

This matrix is the same as if we had evaluated all shape functions on the physical elements with a possibly non-uniform step size $h_s, s = 0, 1, 2, 3$.

Remark 6.41. *Good practical descriptions for higher dimensions can be found in [43][Chapter 12] and [63].*

6.5 Quadratic finite elements: P_2 elements

6.5.1 Algorithmic aspects

We explain the idea how to construct higher-order FEM in this section. Specifically, we concentrate on P_2 elements, which are quadratic on each element K_s. First we define the discrete space:

$$V_h = \{v \in C[0,1] |\ v|_{K_j} \in P_2\}$$

The space V_h is composed by the basis functions:

$$V_h = \{\phi_0, \ldots, \phi_{n+1}, \phi_{\frac{1}{2}}, \ldots, \phi_{n+\frac{1}{2}}\}.$$

The dimension of this space is $\dim(V_h) = 2n+1$. Here, we followed the strategy that we use the elements K_s as in the linear case and add the mid-points in order to construct unique parabolic functions on each K_s. The mid-points represent degrees of freedom as the two edge points. For instance on each $K_j = [x_j, x_{j+1}]$ we have as well $x_{j+\frac{1}{2}} = x_j + \frac{h}{2}$, where $h = x_{j+1} - x_j$. From this construction it is clear (not proven though!) that quadratic FEM have a higher accuracy than linear FEM.

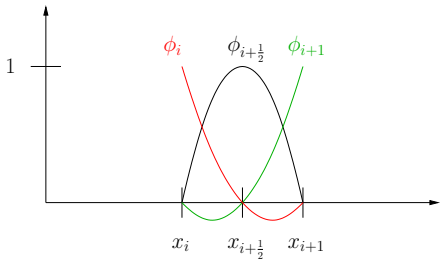

Figure 12: Quadratic basis functions

The specific construction of shape functions can be done as shown in 6.3.3 or using directly Lagrange basis polynomials (see lectures on introduction of numerical methods).

Definition 6.42 (P_2 shape functions)**.** *On the master element $K^{(1)}$, we have*

$$\phi_0(\xi) = 1 - 3\xi + 2\xi^2,$$
$$\phi_{\frac{1}{2}}(\xi) = 4\xi - 4\xi^2,$$
$$\phi_1(\xi) = -\xi + 2\xi^2.$$

These basis functions fulfill the property:

$$\phi_i(\xi_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

for $i, j = 0, \frac{1}{2}, 1$. On the master element, a function has therefore the prepresentation:

$$u(\xi) = \sum_{j=0}^{1} u_j \phi_j(\xi) + u_{\frac{1}{2}} \phi_{\frac{1}{2}}(\xi).$$

This definition allows us to construct a global function from V_h:

Proposition 6.43. *The space V_h is a subspace of V. Each function from V_h has a unique representation and is defined by its nodal points:*

$$u_h(x) = \sum_{j=0}^{n+1} u_j \phi_j(x) + \sum_{j=0}^{n} u_{j+\frac{1}{2}} \phi_{\frac{1}{2}}(x).$$

Remark 6.44. *The assembling of A, u and b is done in a similar fashion as shown in detail for linear finite elements.*

6.5.2 Numerical test: 1D Poisson using quadratic FEM

We continue Section 6.3.8.

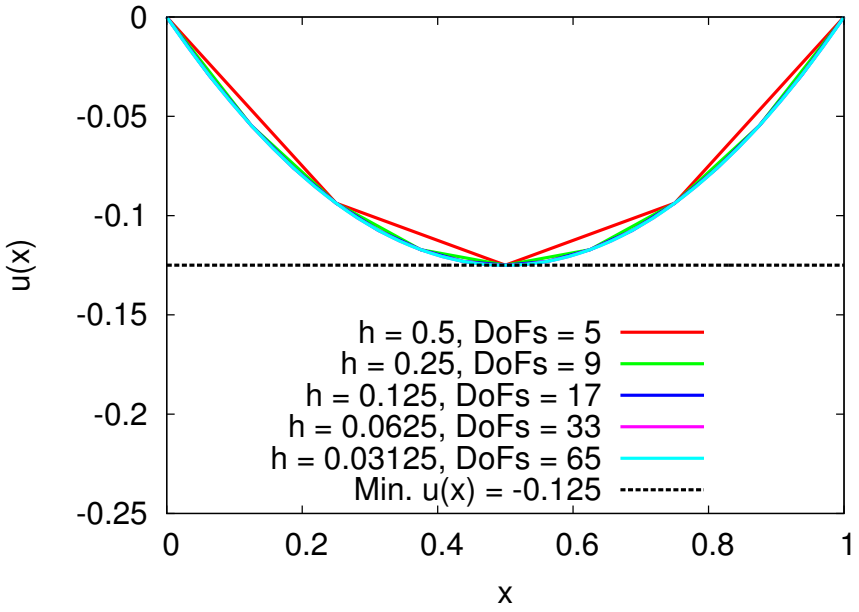

Figure 13: Solution of the 1D Poisson problem with $f = -1$ using quadratic finite elements with various mesh sizes h. DoFs is the abbreviation for degrees of freedom; here the number of support points x_j. The dimension of the discrete space is $DoFs$. For instance for $h = 0.5$, we have 2 mesh elements, we have 5 DoFs and three basis functions, thus $dim(V_h) = 5$. The numerical solutions are computed with an adaptation of step-3 in deal.II [4, 6]. Please notice that the picture norm is not a proof in the strict mathematical sense: to show that the purple, and blue lines come closer and closer must be confirmed by error estimates as presented in Section 6.11 accompanied by numerical simulations in Section 6.13. Of course, for this 1D Poisson problem, we easily observe a limit case, but for more complicated equations it is often not visible whether the solutions do converge.

Remark 6.45. *Be careful when plotting the solution using higher-order FEM. Sometimes, the output data is only written into the nodal values values defining the physical elements. In this case, a quadratic function is visually the same as a linear function. Plotting in all degrees of freedom would on the other hand shows also visually that higher-order FEM are employed.*

Level	Elements	DoFs
1	2	5
2	4	9
3	8	17
4	16	33
5	32	65

6.6 Galerkin orthogonality, a geometric interpretation of the FEM, and a first error estimate

The finite element method has a remarkable geometric interpretation, which is finally the key how to solve and analyze a given PDE in an infinite-dimensional vector space V using the finite-dimensional space V_h as we have previously constructured from a practical point of view. The techniques are based on results from functional analysis.

The goal is to study the accuracy of our FEM scheme in terms of the discretization error $u - u_h$. Here, $u \in V$ is the exact (unknown) solution and $u_h \in V_h$ our finite element solution. The key is to use first **best approximation** results and later (see the later sections) **interpolation estimates**.

The best approximation is based on findings from **approximation theory**, which we briefly recapitulate in the following.

6.6.1 Excursus: Approximation theory emphasizing on the projection theorem

We recall the main ingredients of orthogonal projections as they yield the best approximation properties.

6.6.1.1 Best approximation

Definition 6.46. *Let $U \subset X$ be a subset of a normed space X and $w \in X$. An element $v \in U$ is called best approximation to w in U if*

$$\|w - v\| = \inf_{u \in U} \|w - u\| \ ,$$

i.e., $v \in U$ has the smallest distance to w.

Proposition 6.47. *Let U be a finite-dimensional subspace of a normed space X. Then, there exists for each element in X a best approximation in U.*

Proof. Let $w \in X$. We choose a minimizing sequence (u_n) from U to approximate w. This fulfills:

$$\|w - u_n\| \to d \quad \text{with } n \to \infty$$

where $d := \inf_{u \in U} \|w - u\|$. Since

$$\begin{aligned} \|u_n\| &= \|u_n\| - \|w\| + \|w\| \\ &\leq | \ \|u_n\| - \|w\| \ | + \|w\| \\ &= | \ \|w\| - \|u_n\| \ | + \|w\| \\ &\leq \|w - u_n\| + \|w\| \end{aligned}$$

the sequence (u_n) is bounded. Because U is a finite-dimensional normed space, there exists a converging subsequence $(u_{n(l)})$ with

$$\lim_{l \to \infty} u_{n(l)} \to v \in U.$$

It follows that

$$\|w - v\| = \lim_{l \to \infty} \|w - u_{n(l)}\| = d.$$

□

Definition 6.48. *A pre-Hilbert space is a linear space with a scalar product.*

Proposition 6.49. *Let U be a linear subspace of a pre-Hilbert space X. An element $v \in U$ is a best approximation to $w \in X$ if and only if*

$$(w - v, u) = 0, \quad \text{for all } u \in U,$$

i.e., $w - v \perp U$. For each $w \in X$, we have at most one best approximation w.r.t. U.

Proof. " $\Leftarrow$ ":
Let $\|a\| := \sqrt{(a,a)}$, which holds for all $v, u \in U$:

$$\begin{aligned}
\|w-u\|^2 &= \|w-v+v-u\|^2 \\
&= ((w-v)+(v-u),(w-v)+(v-u)) \\
&= (w-v,w-v)+(w-v,v-u)+(v-u,w-v)+(v-u,v-u) \\
&= \|w-v\|^2+(w-v,v-u)+\overline{(w-v,v-u)}+\|v-u\|^2 \\
&= \|w-v\|^2+2\cdot Re((w-v,v-u))+\|v-u\|^2.
\end{aligned}$$

Since U is a linear space, we have $v-u \in U$ and we require $(w-v,u)=0 \forall u \in U$, it holds:

$$\begin{aligned}
&\|w-u\|^2 = \|w-v\|^2+\|v-u\|^2 \\
\Rightarrow\quad &\|w-v\|^2 = \|w-u\|^2-\|v-u\|^2\ , \quad \forall u \in U \\
\Rightarrow\quad &\|w-v\| < \|w-u\|\ , \quad \forall u \in U, u \neq 0
\end{aligned}$$

Therefore,

$$\|w-v\| = \inf_{u\in U}\|w-u\|$$

and thus v is a best approximation to w in U.
" $\Rightarrow$ ":

Let v be a best approximation to w in U. We assume that $(w-v,u_0) \neq 0$ for a $u_0 \in U$. Let us assume further that $(w-v,u_0) \in \mathbb{R}$ because U is a linear subspace of X. We choose $u = v + \frac{(w-v,u_0)}{\|u_0\|^2}u_0$ and conclude

$$\begin{aligned}
\|w-u\|^2 &= \|w-v\|^2+2Re\left(w-v, v-\left(v+\frac{(w-v,u_0)}{\|u_0\|^2}u_0\right)\right) \\
&\quad +\left\|v-\left(v+\frac{(w-v,u_0)}{\|u_0\|^2}u_0\right)\right\|^2 \\
&= \|w-v\|^2+2Re\left(w-v,-\frac{(w-v,u_0)}{\|u_0\|^2}u_0\right)+\left\|-\frac{(w-v,u_0)}{\|u_0\|^2}u_0\right\|^2 \\
&= \|w-v\|^2+2\left(-\frac{(w-v,u_0)}{\|u_0\|^2}Re(w-v,u_0)\right)+\frac{(w-v,u_0)^2}{\|u_0\|^4}\|u_0\|^2 \\
&= \|w-v\|^2-2\frac{(w-v,u_0)^2}{\|u_0\|^2}+\frac{(w-v,u_0)^2}{\|u_0\|^2} \\
&= \|w-v\|^2-\frac{(w-v,u_0)^2}{\|u_0\|^2} \\
&< \|w-v\|^2.
\end{aligned}$$

This is a contradiction that v is a best approximation to w in U.

Uniqueness

Assume that v_1, v_2 are best approximations. Then,

$$(w-v_1,v_1-v_2) = 0 = (w-v_2,v_1-v_2), \quad \text{da } v_1-v_2 \in U.$$

and

$$\begin{aligned}
\Rightarrow\quad &(w,v_1-v_2)-(v_1,v_1-v_2) = (w,v_1-v_2)-(v_2,v_1-v_2) \\
\Rightarrow\quad &(v_1,v_1-v_2) = (v_2,v_1,-v_2) \\
\Rightarrow\quad &(v_1,v_1-v_2)-(v_2,v_1-v_2) = 0 \\
\Rightarrow\quad &(v_1-v_2,v_1-v_2) = 0 \\
\Rightarrow\quad &v_1-v_2 = 0 \\
\Rightarrow\quad &v_1 = v_2.
\end{aligned}$$

Thus, the best approximation is unique. □

Proposition 6.50 (Orthogonal projection)**.** *Let U be a complete linear subspace of a pre-Hilbert space X. Then, there exists for each $w \in X$ a unique best approximation in U. The operator $P : X \to U$, which maps each $w \in X$ on its best approximation, is a bounded linear operator. The properties of P are:*

$$P^2 = P \quad \text{and} \quad \|P\| = 1$$

This is called an ***orthogonal projection*** *of X on U.*

Proof. We choose a sequence (u_n) with

$$\|w - u_n\|^2 \leq d^2 + \frac{1}{n}, \quad n \in \mathbb{N}$$

with $d := \inf_{u \in U} \|w - u\|$. Then:

$$\begin{aligned}
&\|(w - u_n) + (w - u_m)\|^2 + \|u_n - u_m\|^2 \\
&= ((w - u_n) + (w - u_m), (w - u_n) + (w - u_m)) \\
&\quad + ((u_n - w) + (w - u_m), (u_n - w) + (w - u_m)) \\
&= (w - u_n, w - u_n) + (w - u_n, w - u_m) + (w - u_m, w - u_n) + (w - u_m, w - u_m) \\
&\quad + (u_n - w, u_n - w) + (u_n - w, w - u_m) + (w - u_m, u_n - w) + (w - u_m, w - u_m) \\
&= 2\|w - u_n\|^2 + 2\|w - u_m\|^2 \\
&\leq 4d^2 + \frac{2}{n} + \frac{2}{m} \quad \text{for all } n, m \in \mathbb{N}.
\end{aligned}$$

Since $\frac{1}{2}(u_n + u_m) \in U$, it holds

$$\|w - \frac{1}{2}(u_n + u_m)\|^2 \geq d^2$$

and we can estimate $\|u_n - u_m\|^2$ via

$$\begin{aligned}
\|u_n - u_m\|^2 &\leq 4d^2 + \frac{2}{n} + \frac{2}{m} - \|(w - u_n) + (w - u_m)\|^2 \\
&= 4d^2 + \frac{2}{n} + \frac{2}{m} - \|2(w - \frac{1}{2}(u_n + u_m))\|^2 \\
&= 4d^2 + \frac{2}{n} + \frac{2}{m} - 4\|w - \frac{1}{2}(u_n + u_m)\|^2 \\
&\leq \frac{2}{n} + \frac{2}{m}.
\end{aligned}$$

Therefore, (u_n) is bounded and therefore a Cauchy sequence.

Since U is complete, there exists a $v \in U$ with $u_n \to v,\ n \to \infty$. From

$$\|w - u_n\|^2 \leq d^2 + \frac{1}{n}$$

it follows that $\lim_{n \to \infty} u_n = v$ is a best apprioximation to w in U. The uniqueness follows from Proposition (6.49).

Next, we show $P^2 = P$. Always, it holds $P(u) = u$. It follows

$$P^2(w) = P(P(w)) = P(u) = u = P(w)$$

The linearity of P is shown as follows: let $\alpha \neq 0$, then

$$\begin{aligned} & P(\alpha x) = v \\ \Leftrightarrow \quad & (\alpha x - v, u) = 0 \\ \Leftrightarrow \quad & \alpha(x - \frac{v}{\alpha}, u) = 0 \\ \Leftrightarrow \quad & P(x) = \frac{v}{\alpha} \\ \Leftrightarrow \quad & \alpha P(x) = v. \end{aligned}$$

Consequently, $P(\alpha x) = \alpha P(x)$.

Let v_1, v_2 be best approximations of x, y, i.e., it holds $P(x) = v_1, P(y) = v_2$. Then:

$$\begin{aligned} & P(x) + P(y) = v_1 + v_2 \\ \Leftrightarrow \quad & (x - v_1, u) = 0 \quad \wedge \quad (y - v_2, u) = 0 \\ \Leftrightarrow \quad & (x - v_1, u) + (y - v_2, u) = 0 \\ \Leftrightarrow \quad & (x - v_1 + y - v_2, u) = 0 \\ \Leftrightarrow \quad & ((x + y) - (v_1 + v_2), u) = 0. \end{aligned}$$

Thus, $P(x + y) = v_1 + v_2 = P(x) + P(y)$. It remains to show that $\|P\| = 1$. It holds:

$$\begin{aligned} \|w\|^2 &= (P(w) + w - P(w), P(w) + w - P(w)) \\ &= (P(w), P(w)) + (w - P(w), w - P(w)) \\ &\quad + (P(w), w - P(w)) + (w - P(w), P(w)) \\ &= \|P(w)\|^2 + \|w - P(w)\|^2 \\ &\geq \|P(w)\|^2 \quad \forall w \in X \end{aligned}$$

Thus, P is bounded with $\|P\| \leq 1$. Because $P^2 = P$ and since for two lineaer operators A and B, it holds $\|AB\| \leq \|A\|\|B\|$, it follows finally

$$\begin{aligned} & \|P\| = \|P^2\| = \|PP\| \leq \|P\|\|P\| \\ \Rightarrow \quad & \|P\| \geq 1 \\ \Rightarrow \quad & \|P\| = 1. \end{aligned}$$

□

6.6.1.2 The Fréchet-Riesz representation theorem

An important principle in functional analysis is to gain information of normed spaces with the help of functionals.

6.6.1.2.1 Direct sum

Definition 6.51 (Direct sum)**.** *A vector space X can be written as direct sum of two subspaces Y and Z:*

$$X = Y \oplus Z,$$

if each element $x \in X$ has a unique representation such as

$$x = y + z, \quad y \in Y, z \in Z$$

Remark 6.52. *The space Z is the **algebraic complement** of Y in X.*

In Hilbert space theory we are specifically interested in such representations. A general Hilbert space H can be written as direct sum of a closed subspace Y and its orthogonal complement $Y^\perp$.

Proposition 6.53 (Projection theorem, direct sum)**.** *Let Y be a closed subspace of a Hilbert space H. Then:*

$$H = Y \oplus Z, \quad Z = Y^\perp$$

6.6.1.2.2 The Fréchet-Riesz representation theorem

Definition 6.54. *The space $\mathcal{L} = L(X, \mathbb{K})$ of the linear, bounded functionals on a normed space is called the dual space of X. We often find the notations $X' = \mathcal{L} = L(X, \mathbb{K})$ or if X is a Hilbert space, we write $X^* = \mathcal{L} = L(X, \mathbb{K})$.*

One of the key theorems of Hilbert space theory is:

Proposition 6.55. *Let X be a Hilbert space. For each linear, bounded functional $F \in \mathcal{L}$, there exists a unique element $f \in X$, so that*

$$F(u) = (u, f) \qquad \forall u \in X.$$

We have constructed a bijective, isogeometric conjugate linear mapping with

$$\|f\| = \|F\|.$$

Proof. We divide the proof into several parts:

- **Uniqueness**
  When f maps on $F = 0$, then
  $$F(x) = (x, f) = 0 \qquad \forall x \in X$$
  and also
  $$(f, f) = 0$$
  from which follows
  $$f = 0.$$
  Thus, the mapping $f \to F$ is unique because $f = 0$ is the only element, which induces the zero function $F = 0$.

- **Equality of norms**
  Cauchy-Schwarz yields
  $$\begin{aligned} & ||F|| = |F| = |(x, f)| \leq ||x|| \cdot ||f|| \\ \Rightarrow \qquad & \frac{|F|}{||x||} \leq ||f|| \\ \Leftrightarrow \qquad & \sup_{||x|| \leq 1} \frac{|F(x)|}{||x||} \leq ||f|| \\ \Rightarrow \qquad & ||F|| \leq ||f|| \end{aligned}$$
  with
  $$||F|| = \sup_{||x|| \leq 1} \frac{|F(x)|}{||x||}$$
  We use specifically $x = f$ in $F(x) = (x, f)$ and obtain:
  $$|(f, f)| = ||f||^2 = |F(f)| \leq ||f|| \cdot ||F||.$$
  Therefore:
  $$||f|| \leq ||F||$$
  and therefore, we have on the one hand $||f|| \geq ||F||$. On the other hand, we have $||f|| \leq ||F||$. From this it follows:
  $$||f|| = ||F||$$

- **Construction**
  For $F \in X^* = \mathcal{L}$ we need to construct $f \in X$:
  The kernel
  $$N(F) = \{u \in X : F(u) = 0\}$$
  is a closed, linear subspace of the Hilbert space X. If $N = X$, then $f = 0$ yields the desired result. If $N \neq X$, then it holds
  $$X = N \oplus N^{\perp}$$
  We choose now a $w \in X$ with $F(w) \neq 0$. From the projection theorem it follows that if v is best approximation of w onto the subspace N, then
  $$w - v \perp N(F)$$
  It holds with $g := w - v$,
  $$F(g)u - F(u)g \quad \in N(F) \qquad \forall u \in X$$
  because
  $$F(F(g)u - F(u)g) \;=\; F(g)F(u) - F(u)F(g) \;=\; 0.$$
  Then:
  $$\begin{aligned} & (F(g)u - F(u)g, g) &&= 0 \qquad \forall u \in X \\ \Rightarrow \quad & (F(g)u, g) - (F(u)g, g) &&= 0 \\ \Rightarrow \quad & (F(u)g, g) &&= (F(g)u, g) \\ \Rightarrow \quad & F(u)(g, g) &&= (u, \overline{F(g)}g) \\ \Rightarrow \quad & F(u) &&= \left(\tfrac{u, \overline{F(g)}g}{||g||^2}\right) \end{aligned}$$

□

6.6.1.3 The Pythagorean theorem This section recapitulates and extends our findings from the previous two sections. Let

$$\varphi(a) = (u - av, u - av) \tag{80}$$
$$= (u, u) + a^2 \cdot (v, v) - a \cdot \underbrace{[(u, v) + (v, u)]}_{=2\mathrm{Re}(u,v)} \tag{81}$$

The smallest distance is the minimum of (141). A necessary condition is

$$\varphi'(a) = 2a \cdot (v, v) - 2\mathrm{Re}(u, v) = 0$$

The minimum is

$$a_0 = \frac{\mathrm{Re}(u, v)}{(v, v)}.$$

Hence, we have found a minimal point a_0, which fulfills the condition $0 \leq \varphi(a_0) \leq \varphi(a)$. In detail:

$$(u, u) - a_0 \cdot 2\mathrm{Re}(u, v) + a_0^2 \cdot (v, v) \quad \leq \quad (u, u) + a^2 \cdot (v, v) - a^2 \cdot \mathrm{Re}(u, v).$$

Furthermore

$$\begin{aligned} 0 &\leq (u, u) - \frac{2\mathrm{Re}(u, v)\,\mathrm{Re}(u, v)}{(v, v)} + \frac{|\mathrm{Re}(u, v)|^2}{(v, v)^2}\,(v, v) \\ &= (u, u) - \frac{|\mathrm{Re}(u, v)|^2}{(v, v)}. \end{aligned}$$

It follows:

$$|\mathrm{Re}(u,v)| \leq (u,u)\cdot(v,v)$$

We now define the **projection** of a vector w onto a vector v:

$$P_{<v>}(w) = \frac{(w,v)}{(v,v)}\cdot v.$$

For an illustration, we refer to Figure 14.

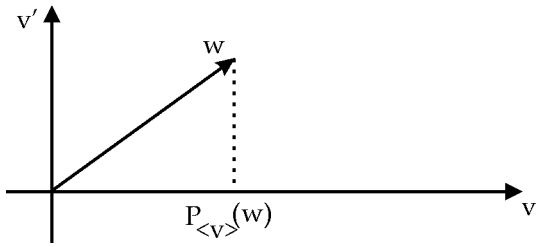

Figure 14: Projection of w on v.

6.6.1.4 Properties of the projection We investigate now the properties of the projection:

i) $P^2_{<v>} = P_{<v>}$ (Idempotenz (germ.))

ii) $P^*_{<v>} = P_{<v>}$ (Self-adjointness)

The second property can be shown easily using the scalar product. It says that the projection operator can be shifted into the second argument. It follows that

$$(P_{<v>}(w), w') = (w, P_{<v>}(w')) \quad \Leftrightarrow \quad P^*_{<v>} = P_{<v>}$$

Proof. To i)
Let $w' = P_{<v>}(w)$. We compute

$$P^2_{<v>}(w) = P_{<v>}(w') = P_{<v>}\left[\frac{(w,v)}{(v,v)}\cdot v\right] = \frac{(w,v)}{(v,v)}\underbrace{P_{<v>}(v)}_{=v} = P_{<v>}(w)$$

To ii)
On the one hand, we have

$$(P_{<v>}(w), w') = \frac{(w,v)}{(v,v)}(v,w').$$

Secondly,

$$(w, P_{<v>}(w')) = \left(w, \frac{(w',v)}{(v,v)}\cdot v\right) = \frac{(v,w')\cdot(w,v)}{(v,v)}$$

Comparing both sides yields the assertion. □

6.6.1.5 The projection theorem (the Pythagorean theorem) Using Figure 14, we can derive the Pythagorean theorem. It holds

$$\begin{aligned}
||w||^2 &= ||P_{<v>}(w) + (w - P_{<v>}(w))||^2 \\
&= ||P_{<v>}(w)||^2 + ||w - P_{<v>}(w)||^2 + (P_{<v>}(w), w - P_{<v>(w)}) + \overline{(\ldots)} \\
&= ||P_{<v>}(w)||^2 + ||w - P_{<v>}(w)||^2 + (P_{<v>(w)}, w) - (P_{<v>}(w), P_{<v>}(w))
\end{aligned}$$

Furthermore, using $P^*_{<v>}P_{<v>} = P^2_{<v>} = P_{<v>}$, we obtain

$$\begin{aligned} ||w||^2 &= ||P_{<v>}(w)||^2 + ||w - P_{<v>}(w)||^2 + (P_{<v>(w)}, w) - (P^*_{<v>}P_{<v>}(w), w) \\ &= ||P_{<v>}(w)||^2 + ||w - P_{<v>}(w)||^2. \end{aligned}$$

Proposition 6.56 (Pythagorean theorem)**.** *We define:*

$$||w||^2 = ||P_{<v>}(w)||^2 + ||w - P_{<v>}(w)||^2$$

Specifically, it holds

$$||w||^2 \geq ||P_{<v>}(w)||^2 \tag{82}$$

The equality is obtained for $w = ||P_{<v>}(w)||$.

Remark 6.57. *We can write in terms of scalar products:*

$$(w, w) \geq \frac{|(w, v)|^2}{(v, v)^2} \cdot (v, v)$$

and can obtain backwards the Cauchy-Schwarz inequality:

$$(w, w)(v, v) \geq |(w, v)|^2.$$

6.6.2 Application to our FEM setting

We have

$$\begin{aligned} (u', \phi') &= (f, \phi) \quad \forall \phi \in V, \\ (u'_h, \phi'_h) &= (f, \phi_h) \quad \forall \phi_h \in V_h. \end{aligned}$$

Taking in particular only discrete test functions from $V_h \subset V$ and subtraction of both equations yields:

Proposition 6.58 (Galerkin orthogonality)**.** *It holds:*

$$((u - u_h)', \phi_h) = 0 \quad \forall \phi_h \in V_h,$$

or in the more general notation:

$$a(u - u_h, \phi_h) = 0 \quad \forall \phi_h \in V_h.$$

Proof. Taking $\phi_h \in V_h$ in both previous equations yields:

$$(u', \phi') - (u'_h, \phi'_h) = ((u - u_h)', \phi_h) = (f - f, \phi_h) = 0.$$

□

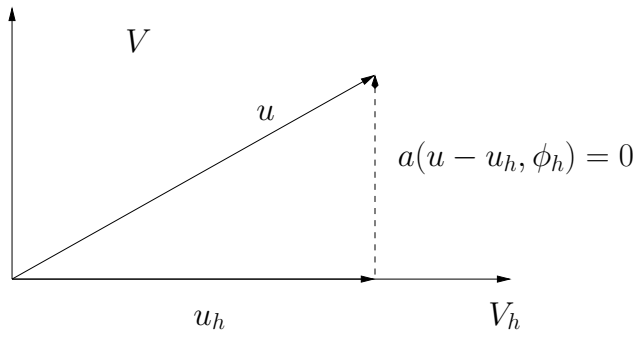

Figure 15: Illustration of the Galerkin orthogonality.

Since $(\cdot,\cdot)$ is a scalar product (for the definition we refer the reader to Definition 6.89); here in the L^2 sense, [4] Galerkin orthogonality yields immediately a geometric interpretation of the finite element method. The error (measured in terms of the first derivative) stands orthogonal to all elements ϕ_h of V_h. Therefore, the solution u_h is the **best approximation** since it has the smallest distance to u.

Proposition 6.59 (Best approximation in finite-dimensional subspaces)**.** *Let U be a finite-dimensional subspace of V (for instance $U := V_h$, our finite element function space). Then, there exists for each element in V a best approximation in U.*

Proof. Let $u \in V$. Choosing a minimizing sequence[5] $(v_n)_{n\in\mathbb{N}}$ from U to approximate u brings us to

$$\|u - v_n\| \to d \quad \text{for } n \to \infty,$$

where $d := \inf_{v\in U} \|u - v\|$. Moreover, we can estimate as follows:

$$\begin{aligned}\|v_n\| &= \|v_n\| - \|u\| + \|u\| \\ &\leq |\|v_n\| - \|u\|| + \|u\| \\ &= |\|u\| - \|v_n\|| + \|u\| \\ &\leq \|u - v_n\| + \|u\|.\end{aligned}$$

Thus, the sequence $(v_n)_{n\in N}$ is bounded. Since U is finite dimensional, there exists a converging subsequence $(v_{n_l})_{l\in N}$ (a result from functional analysis, e.g., [69]) with

$$\lim_{l\to\infty} u_{n_l} \to v \in U.$$

It follows that

$$\|u - v\| = \lim_{l\to\infty} \|u - v_{n_l}\| = \inf_{\phi\in U} \|u - \phi\|$$

which shows the assertion. □

In the following, we use the fact that the L^2 scalar product induces a norm (this is due to the fact that L^2 is a Hilbert space; see Section 6.90)

$$\|w\| = \sqrt{(w,w)} = \sqrt{\int_\Omega w^2\,dx}.$$

We recall Cauchy's inequality, which we need in the following:

$$|(v,w)| \leq \|v\|\|w\|.$$

We show that u_h is a best approximation to u.

Theorem 6.60 (Céa lemma - first version)**.** *For any $v_h \in V_h$ it holds*

$$\|(u - u_h)'\| \leq \|(u - v_h)'\|$$

Proof. We estimate as follows:

$$\begin{aligned}\|(u - u_h)'\|^2 &= (u' - u_h', u' - u_h') \\ &= a(u - u_h, u - u_h) \\ &= a(u - u_h, u - v_h + v_h - u_h) \\ &= a(u - u_h, u - v_h) + \underbrace{a(u - u_h, v_h - u_h)}_{=0,\text{ Galerkin ortho.}} \\ &\leq \|(u - u_h)'\|\,\|(u - v_h)'\|.\end{aligned}$$

[4] We later define more precisely the L^2 space. The reader can think of a square-integrable functions defined on Ω with

$$L^2(\Omega) := \{v|\ \int_\Omega v^2\,dx < \infty\}.$$

[5] Working with minimizing sequences is an important concept in optimization and calculus of variations when minimizers to functionals need to be established. If (v_n) is a minimizing sequence and the functional $F(\cdot)$ is bounded from below, we obtain that $F(v_n) < \infty$ (assuming that F is proper) and $\lim_n F(v_n) = \inf_u F(u)$.

We can now divide by $|(u-u_h)'\|$ and obtain

$$\|(u-u_h)'\| \leq \|(u-v_h)'\|,$$

which shows the assertion. We finally remark that this proof technique (using Galerkin orthogonality at some point) is essential in finite element error analyis. □

Remark 6.61. *This theorem is a first version of Céa's Lemma, which will be later augmented with certain constants yielding a more precise result. However, this result itself is extremely important as it shows the best approximation property: we choose the best approximation u_h from the space V_h. In particular, any interpolation error is based on the best approximation property. Specifically, v_h will be chosen as an appropriate interpolation $i_h v \in V_h$ from which we then can explicitely calculate the discrete functions and obtain a quantitative error estimate in terms of the mesh size h. Secondly, we also see (trivial, but nevertheless worth to be mentioned) that if u is already an element of V_h, the error would be identically zero, which is clear from geometrical considerations.*

The previous result can be extended as follows: not only does the Galerkin orthogonality yield the best approximation property, but it does also hold the contrary, namely that a best approximation property yields

$$((u-u_h)', \phi_h) = 0.$$

Proposition 6.62. *Let U be a linear subspace of a Pre-Hilbert space V. An element $v \in U$ is best approximation to $u \in V$ if and only if*

$$(u-v,\phi) = 0 \quad \forall \phi \in U.$$

For each $u \in V$ there exists at most one best approximation with respect to U.

Proof. The backward part follows from Theorem 6.60. The forward direction follows from Section 6.6.1. □

6.7 Neumann & Robin problems and varying coefficients

6.7.1 Robin boundary conditions

We briefly introduce and discuss well-posedness of Neumann and Robin boundary value problems. Let $\Omega = (0,1)$ and let $f \in C(\Omega)$ and $\beta > 0$. We consider the following problem:

Formulation 6.63 ((D))**.** *Find $u \in C^2(\bar{\Omega})$ such that*

$$\begin{cases} -u'' = f & \text{in } \Omega, \\ \beta u(0) - u'(0) = \alpha_0, \\ \beta u(1) + u'(1) = \alpha_1. \end{cases} \tag{83}$$

Remark 6.64. *For $\beta = 0$ we would obtain a pure Neumann (or so-called traction) problem.*

To derive a variational formulation, a first step is the definition of an appropriate function space V. We recall that boundary conditions on function values (such as Dirichlet or the first terms $u(0)$ and $u(1)$ in the Robin type conditions) are built into V. If we change the problem statement (thus other boundary conditions), also the definition of V will change. Here, we define

$$V = \mathcal{C}^1(\Omega) \cap \mathcal{C}(\bar{\Omega}).$$

In the second step, we can state the variational form of (83):

Formulation 6.65 ((V))**.** *Find $u \in V$ such that*

$$a(u,\phi) = l(\phi) \quad \forall \phi \in V$$

where

$$\begin{aligned} a(u,\phi) &= \int_\Omega u'\phi'\,dx + \beta u(1)\phi(1) + \beta u(0)\phi(0), \\ l(\phi) &= \int_\Omega f\phi\,dx + \alpha_1\phi(1) + \alpha_0\phi(0). \end{aligned} \tag{84}$$

Proposition 6.66. *For $u \in \mathcal{C}^2(\bar{\Omega})$, it holds*

$$(D) \leftrightarrow (V).$$

Proof. First, we show $(D) \rightarrow (V)$. Multiplication with a test function, integration, and applying integration by parts yield

$$\begin{aligned}
-u'' &= f \\
\Rightarrow (-u'', \phi) &= (f, \phi) \\
\Rightarrow (u', \phi') - [u'\phi]_0^1 &= (f, \phi) \\
\Rightarrow (u', \phi') - [u'(1)\phi(1) - u'(0)\phi(0)] &= (f, \phi) \\
\Rightarrow (u', \phi') - (\alpha_1 - \beta u(1))\phi(1) + (\beta u(0) - \alpha_0)\phi(0) &= (f, \phi) \\
\Rightarrow \underbrace{(u', \phi') + \beta u(1)\phi(1) + \beta u(0)\phi(0)}_{=a(u,\phi)} &= \underbrace{(f, \phi) + \alpha_1\phi(1) + \alpha_0\phi(0)}_{=l(\phi)}.
\end{aligned}$$

We show the other direction $(V) \rightarrow (D)$. Let u be solution to the variational problem. After backward integration by parts, we obtain:

$$\begin{aligned}
(-u'' - f, \phi) &= -[u'(x)\phi(x)]_{x=0}^{x=1} + [\alpha_0 - \beta u(0)]\phi(0) + [\alpha_1 - \beta u(1)]\phi(1) \\
&= [\alpha_0 + u'(0) - \beta u(0)]\phi(0) + [\alpha_1 - u'(1) - \beta u(1)]\phi(1)
\end{aligned}$$

We work in a two-step procedure and discuss first the domain integral terms and in a second step the boundary terms. First, we work with the test space C_c^∞ (see Definition 6.8). We note that C_c^∞ is a dense subspace of V (see Definition 6.104) and therefore an admissible choice as test space. Using C_c^∞, the boundary terms vanish and we have

$$(-u'' - f, \phi) = 0 \quad \Rightarrow \quad -u'' - f = 0 \quad \Rightarrow \quad -u'' = f \quad \forall \phi \in C_c^\infty.$$

Since the domain terms fulfill the equation, the boundary terms remain:

$$[\alpha_0 + u'(0) - \beta u(0)]\phi(0) + [\alpha_1 - u'(1) - \beta u(1)]\phi(1) = 0.$$

We choose special test functions $\phi \in V$ with $\phi(0) = 1$ and $\phi(1) = 0$ yielding

$$\alpha_0 + u'(0) - \beta u(0) = 0$$

and thus the first boundary condition. Vice versa, we choose $\phi(1) = 0$ and $\phi(0) = 1$ and obtain

$$\alpha_1 - u'(1) - \beta u(1) = 0,$$

which is the second boundary condition. In summary, we have obtained the strong formulation and everything is shown. □

Proposition 6.67. *For $\beta > 0$ the solutions to (D) and (V) are unique.*

Proof. We assume that there are two solutions u_1 and u_2 and we define the difference function $w = u_1 - u_2$. Thanks to linearity it holds:

$$a(w, \phi) = 0 \quad \forall \phi \in V.$$

We recall that $a(\cdot, \cdot)$ has been defined in (84). As test function, we choose $\phi := w$:

$$a(w, w) = 0.$$

In detail:

$$\int_\Omega (w')^2 \, dx + \beta w(1)^2 + \beta w(0)^2 = 0.$$

Recalling that $\beta > 0$, the previous equation can only be satisfied when $w(1) = w(0) = \int_\Omega (w')^2 = 0$. In particular the integral yields that

$$w' \equiv 0 \quad \Rightarrow \quad w = constant.$$

Since the solutions u_1 and u_2 are continuous (because we work with V - the reader may recall its definition!) the difference function w is continuous as well. Since on the boundary, we have $w(0) = w(1) = 0$, it follows from the continuity of w that necessarily

$$w \equiv 0 \quad \Rightarrow \quad u_1 - u_2 = 0.$$

□

6.7.2 Neumann boundary conditions

Working with Neumann boundary conditions, we have derivative information on the boundaries.

Formulation 6.68 ((D)). *Let $\Omega = (0,1)$. Find $u \in C^2(\bar{\Omega})$ such that*

$$\begin{cases} -u'' = f & \text{in } \Omega, \\ -u'(0) = \alpha_0, \\ u'(1) = \alpha_1. \end{cases} \tag{85}$$

The minus sign in front of $u'(0)$ is only for convenience.

Here, the solution is not unique anymore and only determined up to a constant:

$$u + c, \quad c > 0.$$

It is trivial to see that this solution satisfies Formulation 6.68. One possibility is to add an additional condition to fix the solution. A common choice is to prescribe the mean value:

$$\int_\Omega u(x)\,\mathrm{d}x = 0. \tag{86}$$

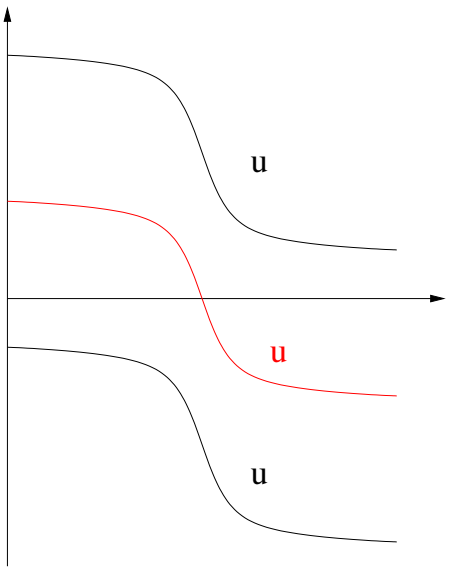

Figure 16: Sketch of possible solutions to the Poisson problem with Neumann boundary conditions. The red solution fulfills the compatibility condition (86). But any other constant to fix the solution would work as well such that one of the black solutions would be feasbile.

A variational formulation of (85) on $V = \mathcal{C}^1(\Omega) \cap \mathcal{C}(\bar{\Omega})$ reads:

Formulation 6.69 ((V)). *Find $u \in V$ such that*

$$a(u,\phi) = l(\phi) \quad \forall \phi \in V$$

where

$$\begin{aligned} a(u,\phi) &= \int_\Omega u'\phi'\,dx, \\ l(\phi) &= \int_\Omega f\phi\,dx + \alpha_1\phi(1) + \alpha_0\phi(0). \end{aligned} \tag{87}$$

Proposition 6.70. *If Formulation 6.68 has a solution $u \in \mathcal{C}^2(\bar{\Omega})$, it is unique by applying the normalization condition* (86)*. Moreover, a* ***compatibility condition****, which is sufficient and necessary, must hold:*

$$\int_0^1 f(x)\,\mathrm{d}x + \alpha_0 + \alpha_1 = 0.$$

Proof. More a sketch rather a rigorous result! It holds necessarily using integration by parts in 1D (Gauss divergence theorem to be more precise)

$$\int_\Omega f\,dx = -\int_\Omega u''\,dx = -\int_\Omega u''\cdot 1\,dx = -[u'(x)]_0^1 = -[u'(1)-u'(0)] = -\alpha_0 - \alpha_1.$$

Consequently, we have shown the compatibility condition:

$$\int_\Omega f\,dx + \alpha_0 + \alpha_1 = 0.$$

This condition is also sufficient to obtain a solution:

$$\begin{aligned} &\int_\Omega f\,dx + \alpha_0 + \alpha_1 = 0 \\ \Leftrightarrow &\int_\Omega f\cdot 1\,dx + \alpha_0\cdot 1 + \alpha_1\cdot 1 = 0 \\ \Leftrightarrow &\underbrace{\int_\Omega f\phi\,dx + \alpha_0\cdot\phi + \alpha_1\cdot\phi}_{=l(\phi)} = 0 \\ &\Leftrightarrow \underbrace{\int_\Omega u'\phi'}_{=a(u,\phi)} = 0 \quad \forall \phi = const. \end{aligned}$$

Therefore, $a(u,\phi) = l(\phi)$. □

6.7.3 Coupled problems / Transmission problems

In practice, the coupling of different PDEs or PDEs with different coefficients in different subdomains is a timely research topic (the keyword is **multiphysics** problems[6]). To explain the basic mechanism, we work again in 1D.

Formulation 6.71. *Let $\Omega = (a,b)$ be an open domain and $k := k(x) \in \mathbb{R}$ a material coefficient, which is specified as*

$$k(x) = \begin{cases} k_1, \text{ for } x \in \Omega_1 = (a,c), \\ k_2, \text{ for } x \in \Omega_2 = (c,b) \end{cases}$$

[6]The word multiphysics is self-explaining: here multiple physical phenomena interact. Examples are interaction of solids with temperature, interaction of solids with fluids, interaction of fluids with chemical interactions, interaction of solids with electromagnetic waves.

for $a < c < b$. The solution is split into two parts:

$$u(x) = \begin{cases} u_1(x), \text{ for } x \in \Omega_1, \\ u_2(x), \text{ for } x \in \Omega_2, \end{cases}$$

which is obtained by solving

$$\begin{cases} -k_1 u_1'' = f \text{ in } \Omega_1, \\ -k_2 u_2'' = f \text{ in } \Omega_2, \\ u_1(a) = u_2(b) = 0, \\ u_1(c) = u_2(c), \\ k_1 u_1'(c) = k_2 u_2'(c). \end{cases} \tag{88}$$

The third line in (88) *is well-known to us and denotes homogeneous Dirichlet conditions on the outer boundaries. Afterwards, we have two new conditions that are so-called* ***interface*** *or* ***coupling*** *conditions and which are very characteristic to coupled problems.*

Remark 6.72. *One of the most famous examples of a coupled (multiphysics) problem is fluid-structure interaction [59, 73] in which the Navier-Stokes equations (fluid flow) are coupled to elasticity (solid mechanics).*

In the following, we now derive a variational formulation. Again the first step is to formulate a function space. To this end, we define:

$$V = \{u \in \mathcal{C}^0(\bar{\Omega}) : u|_{\Omega_1} \in \mathcal{C}^1(\Omega_1), u|_{\Omega_2} \in \mathcal{C}^1(\Omega_2) \text{ and } u(a) = u(b) = 0\}.$$

Remark 6.73 (on the Dirichlet coupling conditions)**.** *Since we seek $u \in \mathcal{C}^0(\bar{\Omega})$ the first coupling condition $u_1(c) = u_2(c)$ is implicitly contained. In general, however, it holds that same rule as before: Dirichlet conditions need to be set explicitely in the function space and Neumann conditions, i.e., the second coupling condition, appears naturally in the equations through integration by parts.*

It holds:

Proposition 6.74 (Variational formulation of the coupled problem)**.** *Find $u \in V$ such that*

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V$$

with

$$a(u, \phi) = \int_a^b k(x) u'(x) \phi'(x)\, dx, \tag{89}$$

$$l(\phi) = \int_a^b f(x) \phi(x)\, dx. \tag{90}$$

Proof. As in the previous sections, we first multiply with a test function from V and integrate:

$$-\int_a^c k_1 u_1''(x) \phi_1\, dx - \int_c^b k_2 u_2''(x) \phi_1\, dx = \int_a^b f(x) \phi\, dx. \tag{91}$$

We perform integration by parts:

$$\int_a^c k_1 u_1'(x) \phi_1'(x)\, dx - k_1 \Big[u_1'(c)\phi_1(c) - u_1'(a)\phi_1(a)\Big] \tag{92}$$

$$+ \int_c^b k_2 u_2'(x) \phi_2'(x)\, dx - k_2 \Big[u_2'(b)\phi_2(b) - u_2'(c)\phi_2(c)\Big] = \int_a^b f(x)\phi(x)\, dx \tag{93}$$

using the continuity (first coupling condition) of v on the interface c, we obtain:

$$\int_a^b k(x) u'(x) \phi'(x)\, dx - \underbrace{\Big[k_1 u_1'(c) - k_2 u_2'(c)\Big]}_{=0} \phi(c) + k_1 u_1'(a) \underbrace{\phi_1(a)}_{=0} - k_2 u_2'(b) \underbrace{\phi_2(b)}_{=0} = \int_a^b f(x)\phi(x)\, dx. \tag{94}$$

On the outer boundaries, the boundary terms will vanish because the test functions are zero. On the interface, the terms will vanish as well since we can employ the second coupling condition. Consequently, we obtain (89) with $u \in V$. □

We next show the equivalence of solutions between the strong form and the variational forms:

Proposition 6.75. *Let $u \in C^2$ and $u|_{\Omega_i} \in \mathcal{C}^2(\overline{\Omega}_1), i = 1, 2$. Then, u is a solution of the strong form (88) if and only if u is a solution of the variational problem.*

Proof. As usually, we first show $(D) \to (V)$. Let u be a solution of (88). Moreover, u has sufficient regularity such that the previous calculations (multiplication with a test function, integration, integration by parts) hold true and $u \in V$. Therefore, u is a solution of (V).

We now discuss $(V) \to (D)$ and consider $u \in V$ to be solution of (V) plus the assumption that u is twice differentiable in each subdomain Ω_i. Consequently, we can integrate by parts backwards. Let $\phi \in V$. Then:

$$-\int_a^c k_1 u_1''(x)\phi_1(x)\,dx + k_1\Big[u_1'(c)\phi_1(c) - u_1'(a)\phi_1(a)\Big] \tag{95}$$

$$-\int_c^b k_2 u_2''(x)\phi_2(x)\,dx + k_2\Big[u_2'(b)\phi_2(b) - u_2'(c)\phi_2(c)\Big] = \int_a^b f(x)\phi(x)\,dx \tag{96}$$

On the boundary $\bar{\Omega}$ we have zero test functions V, and obtain

$$-\int_a^c k_1 u_1''(x)\phi_1(x)\,dx + k_1 u_1'(c)\phi_1(c) - \int_c^b k_2 u_2''(x)\phi_2(x)\,dx - k_2 u_2'(c)\phi_2(c) = \int_a^b f(x)\phi(x)\,dx. \tag{97}$$

In the following we now discuss term by term on each interval:

- We choose a subspace of V such that ϕ_2 is zero and ϕ_1 has a compact support on Ω_1 (yielding specifically that $\phi_1(a) = \phi_1(c) = 0$). Such a subspace is indeed a subspace of V since it is continuous on c and $\phi(a) = \phi(b) = 0$. Consequently, we obtain

  $$-k_1 u_1''(x) = f(x) \quad a < x < c.$$

  The other way around holds true as well from which we get

  $$-k_2 u_2''(x) = f(x) \quad c < x < b.$$

  Consequently, we have recovered the two differential equations inside Ω_1 and Ω_2.

- We now discuss the second interface conditions. In particular, the previous findings from (97) and eliminating the boundary terms give us for $\phi \in V$:

  $$k_1 u_1'(c)\phi_1(c) - k_2 u_2'(c)\phi_2(c) = 0.$$

  The continuity of ϕ at c becomes now essential such that we can write:

  $$\Big[k_1 u_1'(c) - k_2 u_2'(c)\Big]\phi(c) = 0, \quad \forall \phi \in V.$$

  We now construct $\phi \in V$ such that $\phi(c) = 1$ (take for instance piece-wise linear functions on each subdomain). This yields the transmission conditions:

  $$k_1 u_1'(c) - k_2 u_2'(c) = 0.$$

- Finally, we also get the missing conditions on the outer boundary and the first interface conditions because $u \in V$:

  $$u(a) = u(b) = 0$$

  for $u \in V$. Finally, since $u \in V$, i.e., $u \in C^0$ yields

  $$u_1(c) = u_2(c).$$

In summary, u is a solution of (88), which shows that the variational formulation yields the classical solution. □

Exercise 7. *Exercise 2.8 in [43].*

6.8 Variational formulations of elliptic problems in higher dimensions

In the previous sections of this chapter, we have restricted ourselves to 1D derivations in order to show the principle mechanism. In the forthcoming sections, we extend everything to $\mathbb{R}^n$ and introduce a deeper mathematical theory for elliptic problems.

The model problem is

Formulation 6.76. *Let $\Omega \subset \mathbb{R}^n$. Find $u \in C^2 \cap C(\bar{\Omega})$ such that*

$$-\Delta u = f \quad \text{in } \Omega \tag{98}$$

$$u = 0 \quad \text{on } \partial\Omega. \tag{99}$$

The Laplace-Operator Δ has been previously defined in Definition 3.2.

6.8.1 Comments on the domain Ω and its boundaries $\partial\Omega$

We recall the basic definitions of a domain and its boundaries. Let $\Omega \subset \mathbb{R}^n$ be a domain (its properties are specified in a minute) and $\partial\Omega$ its boundary. The closure of Ω is denoted by $\bar{\Omega}$. The boundary may be split into non-overlapping parts

$$\partial\Omega = \partial\Omega_D \cup \partial\Omega_N \cup \partial\Omega_R,$$

where $\partial\Omega_D$ represents a Dirichlet boundary, $\partial\Omega_N$ represents a Neumann boundary and $\partial\Omega_R$ represents a Robin boundary.

- By $C(\Omega)$ we denote the function space of continuous functions in Ω.
- Let $k \geq 0$ (an integer). Then we denote by $C^k(\Omega)$ the functions that are k-times continuously differentiable in Ω.
- By $C^\infty(\Omega)$ we denote the space of infinitely differentiable functions.

In order to be able to define a normal vector n, we assume that Ω is sufficiently regular, here Ω is of class C^1 (or we also say $\partial\Omega \in C^1$ or Ω has a Lipschitz boundary). The normal vector points outward of the domain.

Remark 6.77. *Depending on the properties of Ω (bounded or not), its regularity, and also the properties of its boundary, one can obtain very delicate mathematical results. We refer exemplarily to Grisvard [31] or Wloka [76]. We always assume in the remainder that Ω is open and bounded and sufficiently regular.*

6.8.2 Integration by parts and Green's formulae

From the divergence Theorem 3.30, we obtain immediately:

Proposition 6.78 (Integration by parts)**.** *Let $u, v \in C^1(\bar{\Omega})$. Then:*

$$\int_\Omega u_{x_i} v \, dx = -\int_\Omega u v_{x_i} \, dx + \int_{\partial\Omega} u v n_i \, ds, \quad \text{for } i = 1, \ldots, n.$$

In compact notation:

$$\int_\Omega \nabla u v \, dx = -\int_\Omega u \nabla v \, dx + \int_{\partial\Omega} u v n \, ds.$$

Proof. Apply the divergence theorem to uv. Exercise. □

We obtain now some further results, which are very useful, but all are based directly on the integration by parts. For this reason, it is more important to know the divergence theorem and integration by parts formula.

Proposition 6.79 (Green's formulas)**.** *Let $u, v \in C^2(\bar{\Omega})$. Then:*

$$\int_\Omega \Delta u \, dx = \int_{\partial\Omega} \partial_n u \, ds,$$

$$\int_\Omega \nabla u \cdot \nabla v \, dx = -\int_\Omega \Delta u \, v \, dx + \int_{\partial\Omega} v \, \partial_n u \, ds.$$

Proof. Apply integration parts. □

6.8.3 Variational formulations

We recall the details of the notation that we shall use in the following. In $\mathbb{R}^n$, it holds:

$$(\nabla u, \nabla \phi) = \int_\Omega \nabla u \cdot \nabla \phi \, dx = \int_\Omega \begin{pmatrix} \partial_1 u \\ \vdots \\ \partial_n u \end{pmatrix} \cdot \begin{pmatrix} \partial_1 \phi \\ \vdots \\ \partial_n \phi \end{pmatrix} dx = \int_\Omega \Big(\partial_1 u \partial_1 \phi + \ldots + \partial_n u \partial_n \phi \Big) \, dx.$$

We have

Proposition 6.80. *Let $u \in C^2(\bar{\Omega})$ and $f \in C(\bar{\Omega})$. Let V defined by*

$$V = \{\phi \in C^1(\bar{\Omega}) |\ \phi = 0 \text{ on } \partial\Omega\}.$$

The function u is a solution to (D) if and only if $u \in V$ such that

$$(\nabla u, \nabla \phi) = (f, \phi) \quad \forall \phi \in V.$$

Proof. The proof is the same as in the 1D case. Let u be the solution of (D). We multiply the PDE by a test function $\phi \in V$, integrate and perform integration by parts:

$$-(\Delta u, \phi) = -\int_{\partial\Omega} \partial_n u \phi \, ds + (\nabla u, \nabla \phi)$$

where $\partial_n u = \nabla u \cdot n$, where n is the normal vector as before. Since $\phi = 0$ on $\partial\Omega$, we obtain

$$-(\Delta u, \phi) = (\nabla u, \nabla \phi)$$

thus

$$(\nabla u, \nabla \phi) = (f, \phi).$$

In the backward direction we show $(V) \to (D)$:

$$(\nabla u, \nabla \phi) = -(\Delta u, \phi) + \int_{\partial\Omega} \partial_n u \phi \, ds.$$

As before $\phi = 0$ on $\partial\Omega$ and we get

$$(\nabla u, \nabla \phi) = -(\Delta u, \phi)$$

thus

$$-(\Delta u, \phi) = (f, \phi).$$

Consequently,

$$(-\Delta u + f, \phi) = 0.$$

Since $-\Delta u + f$ is continuous on Ω we can employ the fundamental lemma of calculus of variations in higher dimensions from which we obtain

$$-\Delta u + f = 0 \quad \forall x \in \Omega.$$

We recover the boundary conditions since we started from $u \in V$, which includes that $u = 0$ on $\partial\Omega$. □

Proposition 6.81 (Fundamental lemma of calculus of variations in $\mathbb{R}^n$)**.** *Let $\Omega \subset \mathbb{R}^n$ be an open domain and let w be a continuous function. Let $\phi \in C^\infty(\Omega)$ have a compact support in Ω. If*

$$\int_\Omega w(x)\phi(x) \, dx = 0 \quad \forall \phi \in C^\infty(\Omega)$$

then, $w \equiv 0$ in Ω.

Proof. Similar to the 1D version. □

Remark 6.82. *We have not yet shown well-posedness of (D) and (V), which is the topic of the next section. Only we have shown so far that if solutions to (D) and (V) exist, then these solutions are equivalent.*

6.8.4 A short excursus to analysis, linear algebra, functional analysis and Sobolev spaces

Functional analysis is the combination of classical analysis (i.e., calculus) and linear algebra. The latter one is introduced for finite dimensional spaces in first semester classes. In functional analysis, infinite dimensional vector spaces are considered, which require tools from analysis such as functions, operators, and convergence results. The extension to infinite dimensions leads to many nontrivial results. Classical books are [3, 15, 46, 69, 78].

We cannot provide a complete introduction, but only the most important concepts necessary for the further understanding of the present notes. At the beginning on the next two pages, we partially follow in the spirit of Tröltzsch [67][p. 17-19] since therein a nice, compact, introduction to the basic tools is provided. The analysis results (in particlar the Lebuesgue integral) can be studied in more detail with the help of Königsberger [45].

Before we begin, we refer the reader to Section 3.9 in which the definition of a normed space is recapitulated.

Definition 6.83 (Convergence)**.** *A sequence $(u_n)_{n\in\mathbb{N}} \subset \{V, \|\cdot\|\}$ **converges**, if there exists a limit $u \in V$ such that*

$$\lim_{n\to\infty} \|u_n - u\| = 0$$

or in other notation

$$\lim_{n\to\infty} u_n = u.$$

Definition 6.84 (Cauchy sequence)**.** *A sequence is called a **Cauchy sequence** if for all $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that*

$$\|u_n - u_m\| \leq \varepsilon \quad \forall n > N, \quad \forall m > N$$

Any sequence that converges is a Cauchy sequence. The opposite is in general not true.

Example 6.85. *Let $\Omega = (0, 2)$. We consider the space $\{C(\Omega), \|\cdot\|_{L^2}\}$ with*

$$u_n(x) = \min\{1, x^n\}.$$

We see (check yourself)

$$\|u_n - u_m\|_{L^2}^2 = \int_0^2 (u_n - u_m)^2 \, dx = \ldots = \ldots \leq \frac{2}{2m+1}$$

for $m \leq n$. It is clear that this is a Cauchy sequence. However the limit u is not contained in the space $\{C(\Omega), \|\cdot\|_{L^2}\}$ because it is not a continuous function:

$$u(x) = \lim_{n\to\infty} u_n(x) = \begin{cases} 0, & 0 \leq x \leq 1 \\ 1, & 1 \leq x \leq 0 \end{cases}$$

A very important class of function spaces are those in which all Cauchy sequences do converge:

Definition 6.86 (Complete space, Banach space)**.** *A normed space $\{V, \|\cdot\|_V\}$ is a **complete space**, if all Cauchy sequences converge (i.e., the Cauchy sequence has a limit that belongs to V). A complete, normed space is a so-called **Banach space**.*

Example 6.87. *We continue our examples:*

1. *The set $\mathbb{Q}$ of rational numbers is **not** complete (e.g., [46]).*
2. *$\mathbb{R}^n$ with the euclidian norm $|x| = (\sum_{i=1}^n x_i^2)^{1/2}$ is a Banach space.*
3. *$C(\Omega)$ endowed with the maximum norm*

$$\|u\|_{C(\Omega)} = \max_{x\in\Omega} \|u(x)\|$$

is a Banach space.

4. The space $\{C(\Omega), \|\cdot\|_{L^2}\}$ *with*

$$\|u\|_{L^2} = \Big(\int_\Omega u(x)^2\,dx\Big)^{1/2}$$

is ***not*** *a Banach space; please see the previous example in which we have shown that the limit* u *does not belong to* $C(\Omega)$ *when we work with the* L^2 *norm.*

Remark 6.88. *The last example from before has already several consequences on our variational formulations. So far, we have mainly worked and proven our results using classical function spaces* C, C^k, C^∞*. On the other hand, we work with integrals, i.e,* $(\nabla u, \nabla\phi) = \int_\Omega \nabla u \cdot \nabla\phi\,dx$*. Observing that* $\{C(\Omega), \|\cdot\|_{L^2}\}$ *is not a Banach space and that the limit may be not contained in* $C(\Omega)$ *will influence wellposedness results and convergence results (recall that we try to proof later results similar as for finite differences).*

In $\mathbb{R}^n$ a very important concept is associated to **orthogonality**, which does not come automatically with the Banach-space property. Those spaces in which a **scalar product** (inner product) can be defined, allow the definition of orthogonality. These spaces play a crucial role in PDEs and functional analysis.

Definition 6.89 (Pre-Hilbert space)**.** *Let* V *be a linear space (see Section 3.8). A mapping* $(\cdot,\cdot) : V \to \mathbb{R}$ *is called a* ***scalar product*** *if for all* $u, v, w \in V$ *and* $\lambda \in R$ *it holds*

$$\begin{aligned} (u,u) \geq 0 \quad \textit{and} \quad (u,u) = 0 \quad &\Leftrightarrow \quad u = 0 \\ (u,v) &= (v,u) \\ (u+v,w) &= (u,w) + (v,w) \\ (\lambda u, v) &= \lambda(u,v) \end{aligned}$$

If a scalar product is defined in V*, this space is called a* ***pre-Hilbert space****.*

Definition 6.90 (Hilbert space)**.** *A complete space endowed with an inner product is called a* ***Hilbert space****. The norm is defined by*

$$\|u\| := \sqrt{(u,u)}.$$

Example 6.91. *The space* $\mathbb{R}^n$ *from before has a scalar product and is complete, thus a Hilbert space. The space* $\{C(\Omega), \|\cdot\|_{L^2}\}$ *has a scalar product, but is not complete, and therefore not a Hilbert space. The space* $\{C(\Omega), \|\cdot\|_{C(\Omega)}\}$ *is complete, but the norm is not induced by a scalar product and is therefore not a Hilbert space, but only a Banach space.*

Specifically, for studying PDEs we provide the following further examples:

Definition 6.92 (The L^2 space in 1D)**.** *Let* $\Omega = (a,b)$ *be an interval (recall 1D Poisson). The space of square-integrable functions on* Ω *is defined by*

$$L^2(\Omega) = \{v : \int_\Omega v^2\,dx < \infty\}$$

The space L^2 *is a Hilbert space equipped with the scalar product*

$$(v,w) = \int_\Omega vw\,dx$$

and the induced norm

$$\|v\|_{L^2} := \sqrt{(v,v)}.$$

Using Cauchy's inequality

$$|(v,w)| \leq \|v\|_{L^2}\|w\|_{L^2},$$

we observe that the scalar product is well-defined when $v, w \in L^2$*. A mathematically very correct definition must include in which sense (Riemann or Lebesgue) the integral exists. In general, all L spaces are defined in the sense of the Lebuesgue integral (see for instance [45][Chapter 7]).*

Definition 6.93 (The H^1 space in 1D)**.** *We define the $H^1(\Omega)$ space with $\Omega = (a, b)$ as*

$$H^1(\Omega) = \{v : \ v \text{ and } v' \text{ belong to } L^2\}$$

This is space is equipped with the following scalar product:

$$(v, w)_{H^1} = \int_\Omega (vw + v'w')\, dx$$

and the norm

$$\|v\|_{H^1} := \sqrt{(v, v)_{H^1}}.$$

Definition 6.94 (The H_0^1 space in 1D)**.** *We define the $H_0^1(\Omega)$ space with $\Omega = (a, b)$ as*

$$H_0^1(\Omega) = \{v \in H^1(\Omega) : \ v(a) = v(b) = 0\}.$$

The scalar product is the same as for the H^1 space.

Remark 6.95. *With the help of the space H_0^1 we are now able to re-state the 1D Poisson problem. Rather than working with the space V (see (49)), we work with H_0^1. In fact this is the largest space that allows to define the 1D Poisson problem:*

$$\text{Find } u \in H_0^1(\Omega) : \quad a(u, \phi) = l(\phi) \quad \forall \phi \in H_0^1(\Omega)$$

with

$$a(u, \phi) = (u', \phi'), \quad l(\phi) = (f, \phi).$$

We also recall Definition 3.20 of a dual space. For instance, the dual space of H^1 is the space H^{-1}.

Definition 6.96 (Notation)**.** *When there is no confusion with the domain, we often write*

$$L^2 := L^2(\Omega), \quad H^1 := H^1(\Omega)$$

and so forth.

6.8.4.1 Sobolev spaces To formulate variational formulations, we have worked with function spaces so far, mostly denoted by V, that contain integral information and classical continuous functions.

However, to enjoy the full beauty of variational formulations, the Riemann integral is too limited (see textbooks to analysis such as [45][Chapter 7]) and the Lebuesgue integral is usually used. The latter one does hold for a larger class of integrable functions, which is the biggest advantage in comparison to the Riemann integral. In consequence, one also obtains deep results in exchanging limiting processes such as passing to the limit with sequences and integration (see Chapter 8 in [45]). For instance, when does

$$\int f\, dx = \lim_{k\to\infty} \int f_k\, dx \quad \text{for } f_k \to f$$

hold true?

Based on the Lebuesgue integral, we can then formulate function spaces with derivative information, so-called **Sobolev spaces**, which become **Hilbert spaces** when a scalar product can be defined. The key feature of the Lebesgue integral is that Lebuesgue-measurable functions are considered for integration [45][Chapter 7]. Here, the set of Lebuesgue measure zero plays an important role.

Definition 6.97 (Lebesgue measurable, [45], Chapter 7.5)**.** *A set $\Omega \subset \mathbb{R}^n$ is **Lebesgue measurable** (or in short **measurable**) when the characteristic function 1 can be integrated over Ω. The number*

$$V(\Omega) := V^{(n)}(\Omega) := \int_\Omega 1\, dx = \int_{\mathbb{R}^n} 1_\Omega\, dx$$

is the n-dimensional volume or Lebesgues measure of A. In the case $n = 1$, this is the length of the interval. For $n = 2$ it is the area. For $n = 3$ it is the classical three-dimensional volume. The empty set has Lebesgue measure zero.

Definition 6.98 (Zero set in the sense of Lebesgue, [45], Chapter 7.6)**.** *A set $N \subset \mathbb{R}^n$ is a Lebesgue-set of measure zero if N is measurable with measure zero: $V(N) = 0$.*

Example 6.99. *We give two examples:*

- *The set $\mathbb{Q}$ of rational numbers has Lebesgue measure zero in $\mathbb{R}$ (see for instance [45][Chapter 7.6]).*
- *In $\mathbb{R}^n$ all subsets $\mathbb{R}^s$ with dimension $s < n$ have Lebesgue measure zero. (see for instance [45][Chapter 7.6]).*

The second example has important consequences for PDEs: boundaries $\Gamma \subset \mathbb{R}^{n-1}$ of domains $\Omega \subset \mathbb{R}^n$ have Lebesgue measure zero.

Definition 6.100 (Almost everywhere, a.e.)**.** *Let $\Omega \subset \mathbb{R}^n$ be a Lebesgue-measurable open domain. Measurable functions are defined **almost everywhere**, in short **a.e.**, in Ω. When we change the function value on a subset of Lebuesgue measure zero, then we do not change the function value. In other words*

$$f(x) = g(x) \quad \text{almost everywhere in } \Omega$$

when there is $T \subset \Omega$ such that the Lebesgue measure of T is zero and

$$f(x) = g(x) \quad \text{for all } x \in \Omega \setminus T.$$

Short: Almost everywhere means except on a set of Lebesgue measure zero.

Definition 6.101 (L^2 space in $\mathbb{R}^n$)**.** *Let $\Omega \subset \mathbb{R}^n$ be a Lebesgue-measurable open domain. The space $L^2 = L^2(\Omega)$ contains all square-integrable functions in Ω:*

$$L^2 = \{v \text{ is Lebesgue measurable} \mid \int_\Omega v^2 \, dx < \infty\}.$$

The space L^2 is complete (each Cauchy sequence converges in the norm defined below) and thus a Banach-space. For a proof see for instance [46][Section 2.2-7] or [69][Kapitel I]. Moreover, we can define a scalar product:

$$(u, v) = \int_\Omega vu \, dx$$

such that L^2 is even a Hilbert space with norm

$$\|u\|_{L^2} = \sqrt{(u, u)}.$$

In the same way, one obtains all L^p spaces:

Definition 6.102 (L^p spaces)**.** *Let $1 \leq p < \infty$. Then,*

$$L^p = \{v \text{ is Lebesgue measurable} \mid \int_\Omega v^p \, dx < \infty\}$$

equipped with the norm:

$$\|u\|_{L^p} := \Big(\int_\Omega |v|^p \, dx\Big)^{\frac{1}{p}}.$$

For $p = \infty$, we define the norm via:

$$\|u\|_{L^\infty} := ess \sup |u|.$$

which are the essentially bounded functions.

Proposition 6.103. *Let Ω be bounded. Then:*

$$L^\infty \subset \ldots \subset L^p \subset L^{p-1} \subset \ldots \subset L^2 \subset L^1.$$

We recall several important results from analysis that have an important impact on variational formulations of PDEs and numerics. The first result is:

Definition 6.104 (Density, e.g., [46]). *A metric space U is **dense** in V when*

$$\bar{U} = V.$$

Here $\bar{U}$ means the closure of U.

Example 6.105. *Two examples:*

- *The set $\mathbb{Q}$ of rational numbers is dense in $\mathbb{R}$ (e.g., [46])*
- *The space of polynomial functions P_k is dense in $C[a,b]$ (see lectures on Analysis 1, e.g. [44]. The key word is Weierstrass approximation theorem).*

Theorem 6.106 (Density result for L^2). *The space $C_c^\infty(\Omega)$ is **dense** in $L^2(\Omega)$. In other words: for all functions $f \in L^2$, there exists a sequence $(f_n)_{n\in\mathbb{N}} \subset C_c^\infty(\Omega)$ such that*

$$\lim_{n\to\infty} \|f - f_n\| = 0.$$

Remark 6.107. *The previous result means that we can approximate any L^2 function by a sequence of 'very nice', smooth, functions defined in the usual classical sense.*

Remark 6.108 (Density results). *In analysis and functional analysis, density arguments are often adopted since it allows to work with simple functions in order to approximate 'abstract' and complicated functions.*

As second result, we generalize the fundamental lemma of calculus of variations:

Theorem 6.109 (Fundamental lemma of calculus of variations). *Let $f \in L^2$. If for all $\phi \in C_c^\infty(\Omega)$ we have*

$$\int_\Omega f\phi \, dx = 0,$$

then $f(x) = 0$ almost everywhere in Ω.

The definition of the space L^2 now allows us to define a more general derivative expression, the so-called **weak derivative**. In fact, this is the form of derivatives that we have implicitly used so far in our variational formulations.

Definition 6.110 (Weak derivative). *Let $\Omega \subset \mathbb{R}^n$ be an open, measurable, domain. Let $u \in L^2$. We say that u is **weakly differentiable** in L^2, when functions $w_i \in L^2, i = 1, \dots, n$ exist such that for all functions $\phi \in C_c^\infty(\Omega)$, we have*

$$\int_\Omega u\partial_{x_i}\phi \, dx = -\int_\Omega w_i(x)\phi(x) \, dx.$$

In our usual compact form:

$$(u, \partial_{x_i}\phi) = -(w_i, \phi).$$

Each w_i is the ith partial derivative of u. Thus

$$\partial_{x_i} u = w_i \quad \text{in the weak sense.}$$

Remark 6.111 (Example). *As the terminology indicates, the weak derivative is weaker than the usual (strong) derivative expression. In classical function spaces both coincide.*

Example 6.112 ([67]). *Consider $u(x) = |x|$ in $\Omega = (-1, 1)$. Then, the weak derivative is given by:*

$$u'(x) := w(x) = \begin{cases} -1 & x \in (-1, 0) \\ +1 & x \in [0, 1) \end{cases}.$$

Indeed it holds that for all $v \in C_c^\infty(\Omega)$ we have

$$\int_{-1}^1 |x|v'(x) \, dx = \int_{-1}^0 (-x)v'(x) + \int_0^1 xv'(x) \, dx = \ldots = -\int_{-1}^1 w(x)v(x) \, dx.$$

We notice that it is not important which value u' has in zero since it is a point of measure zero; see Example 6.99.

We can simply extend the weak derivative to higher-order derivatives. Here we remind the reader to the multiindex notation presented in Section 3.5. It holds:

Definition 6.113. *Let $u \in C^k(\bar{\Omega})$ and $v \in C_c^k(\Omega)$ and α a multiindex with $|\alpha| \leq k$. Then employing multiple times partial integration, we obtain*

$$\int_\Omega u(x) D^\alpha v(x)\, dx = (-1)^{|\alpha|} \int_\Omega v(x) D^\alpha u(x)\, dx.$$

Definition 6.114 (Higher-order weak derivative)**.** *Let $\Omega \subset \mathbb{R}^n$ be an open, measurable, domain. Let u be locally integrable in the sense of Lebesgue and α be a multiindex. If there exists a locally integrable function w with*

$$\int_\Omega u(x) D^\alpha v(x)\, dx = (-1)^{|\alpha|} \int_\Omega v(x) w(x)\, dx,$$

then $w = D^\alpha u(x)$ and the so-called weak derivative (of higher-order).

With the help of weak derivatives, we now define Lebuesgue spaces with (weak) derivatives: **Sobolov spaces.** When their norms are induced by a scalar product, a Sobolev space is a Hilbert space.

Definition 6.115 (The space H^1)**.** *Let $\Omega \subset \mathbb{R}^n$ be an open, measurable, domain. The space $H^1 := H^1(\Omega)$ is defined by*

$$H^1 := \{v \in L^2(\Omega) |\ \partial_{x_i} v \in L^2, \quad i = 1, \ldots, n\}$$

In compact form:

$$H^1 := \{v \in L^2(\Omega) |\ \nabla v \in L^2\}.$$

The nabla-operator has been defined in Section 3.6.

Remark 6.116. *In physics and mechanics, the space H^1 is also called the **energy space** . The associated norm is called **energy norm**.*

Proposition 6.117 (H^1 is a Hilbert space)**.** *We define the scalar product*

$$(u, v)_{H^1} := \int_\Omega (uv + \nabla u \cdot \nabla v)\, dx$$

which induces the norm:

$$\|u\|_{H^1} = \sqrt{(u, u)_{H^1}}.$$

The space H^1 equipped with the norm $\|u\|_{H^1}$ is a Hilbert space.

Proof. It is tivial to see that $(u, v)_{H^1}$ defines a scalar product. It remains to show that H^1 is complete. Let $(u_n)_{n \in N}$ be a Cauchy sequence in H^1. Specifically $(u_n)_{n \in N}$ and $(\partial_{x_i} u_n)_{n \in N}$ are Cauchy sequences in L^2. Since L^2 is complete, see Definition 6.101, there exist two limits u and w_i with

$$\begin{aligned} u_n &\to u \quad \text{in } L^2 \quad \text{for } n \to \infty \\ \partial_{x_i} u_n &\to w_i \quad \text{in } L^2 \quad \text{for } n \to \infty. \end{aligned}$$

We employ now the weak derivative:

$$\int_\Omega u_n(x) \partial_{x_i} \phi(x)\, dx = -\int_\Omega \partial_{x_i} u_n(x) \phi(x)\, dx.$$

Passing to the limit $n \to \infty$ yields

$$\int_\Omega u(x) \partial_{x_i} \phi(x)\, dx = -\int_\Omega w_i(x) \phi(x)\, dx.$$

This shows that u has a weak derivative w. Therefore, the element w is contained in H^1 and $(u_n)_{n \in N} \subset H^1$, which shows the assertion. □

Remark 6.118. *The space H^1 is the closure of $C^1(\bar{\Omega})$ with respect to the H^1 norm. It also holds for Ω that is regular, open, bounded that $C_c^\infty(\bar{\Omega})$ is dense in $H^1(\Omega)$.*

Even so we can approximate Lebuesgue-measurable functions in L^2 or H^1 with functions from $C_c^\infty(\Omega)$, the limit function may not have nice properties. The most well-known example is the following:

Proposition 6.119 (Singularities of H^1 functions)**.** *Let $\Omega \subset \mathbb{R}^n$ be open and measurable. It holds:*

- *For $n = 1$, $H^1(\Omega) \subset C(\Omega)$.*
- *For $n \geq 2$, functions in $H^1(\Omega)$ are in general neither continuous nor bounded.*

Proof. Let $n = 1$. Let $\Omega = [a, b]$ and $C^\infty(\Omega)$. Then, we have for $|x - y| < \delta$ and using Cauchy-Schwarz

$$|u(x) - u(y)| = \int_x^y v'(t)\,dt \leq \Big(\int_x^y 1^2\,dt\Big)^{1/2}\Big(\int_x^y |v'(t)|^2\,dt\Big)^{1/2} \leq \sqrt{\delta}\|v\|_{H^1}.$$

Consequently, each Cauchy sequence in $H^1 \cap C^\infty$ is equicontinuous and bounded. Using Arcelà-Ascoli (for a general version we refer to [69]), the limit function is continuous.

Now let $n \geq 2$. Let the unit circle be defined as

$$D := \{(x, y) \in \mathbb{R}^2 |\ x^2 + y^2 < 1\}.$$

We define the function

$$u(x, y) = \log\log\frac{2}{r}$$

where r is the radius with $r^2 = x^2 + y^2$. It is obvious that u is unbounded. On the other hand, $u \in H^1$ because

$$\int_0^{1/2} \frac{1}{r\log^2 r}\,dr < \infty.$$

For $n \geq 3$, we have

$$u(x) = r^{-\alpha}, \quad \alpha < (n - 2)/2.$$

It holds $u \in H^1$ with a singularity in the point of origin. It seems that the higher n is, the more significant the singularities are. □

Definition 6.120 (H_0^1)**.** *The space $H_0^1(\Omega)$ contains vanishing function values on the boundary $\partial\Omega$. The space $H_0^1(\Omega)$ is the closure of $C_c^\infty(\Omega)$ in H^1.*

One often writes:

Definition 6.121 (H_0^1)**.** *The space H_0^1 (spoken: 'H,1,0' and **not** 'H,0,1') is defined by:*

$$H_0^1(\Omega) = \{v \in H^1 | v = 0 \text{ on } \partial\Omega\}.$$

We finally introduce higher-order Sobolev spaces:

Definition 6.122 (Higher-order Sobolev spaces)**.** *Let $1 \leq p < \infty$ and $k \in \mathbb{N}$. The normed space $W^{k,p}(\Omega)$ contains all functions $v \in L^p$, which derivatives $D^\alpha y$ with $|\alpha| \leq k$ also belong to L^p. The norm is defined by*

$$\|v\|_{W^{k,p}} = \Big(\sum_{|\alpha|\leq k}\int_\Omega |D^\alpha v(x)|^p\Big)^{1/p}.$$

For $p = \infty$ we define

$$\|v\|_{W^{k,\infty}} = \max_{|\alpha|\leq k}\|D^\alpha v\|_{L^\infty}.$$

The spaces $W^{k,p}$ are Banach spaces. For $p = 2$, we obtain Hilbert spaces. For instance choosing $k = 1$ and $p = 2$ yields

$$H^1 := W^{1,2}.$$

6.8.4.2 Inequalities

The Hölder inequality reads:

Proposition 6.123 (Hölder's inequality). *Let $1 \leq p \leq \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$ (with the convention $\frac{1}{\infty} = 0$). Let $f \in L^p$ and $g \in L^q$. Then,*

$$fg \in L^1,$$

and

$$\|fg\|_{L^1} \leq \|f\|_{L^p} \|g\|_{L^q}.$$

Proof. See Werner [69]. □

Corollary 6.124 (Cauchy-Schwarz inequality). *Set $p = q = 2$. Then*

$$\|fg\|_{L^1} \leq \|f\|_{L^2} \|g\|_{L^2}.$$

Proposition 6.125 (Minkowski's inequality). *Let $1 \leq p \leq \infty$ and let $f \in L^p$ and $g \in L^p$. Then,*

$$\|f + g\|_{L^p} \leq \|f\|_{L^p} + \|g\|_{L^p}.$$

Proposition 6.126 (Cauchy's inequality with ε).

$$ab \leq \varepsilon a^2 + \frac{b^2}{4\varepsilon}$$

for $a, b > 0$ and $\varepsilon > 0$. For $\epsilon = \frac{1}{2}$, the original Cauchy inequality is obtained.

Proposition 6.127 (Young's inequality). *Let $1 < p, q < \infty$ and $\frac{1}{p} + \frac{1}{q} = 1$. Then,*

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}, \quad a, b > 0.$$

6.8.5 The Lax-Milgram lemma

We present in this section a result that ensures well-posedness (existence, uniqueness, stability) of linear variational problems.

Formulation 6.128 (Abstract model problem). *Let V be a Hilbert space with norm $||\cdot||_V$. Find $u \in V$ such that*

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V.$$

Definition 6.129 (Assumptions). *We suppose:*

1. *$l(\cdot)$ is a bounded linear form:*
$$|l(u)| \leq C\|u\| \quad \text{for all } u \in V.$$
2. *$a(\cdot, \cdot)$ is a bilinear form on $V \times V$ and continuous:*
$$|a(u, v)| \leq \gamma \|u\|_V \|v\|_V, \quad \gamma > 0, \quad \forall u, v \in V.$$
3. *$a(\cdot, \cdot)$ is coercive (or V-elliptic):*
$$a(u, u) \geq \alpha \|u\|_V^2, \quad \alpha > 0, \quad \forall u \in V.$$

Lemma 6.130 (Lax-Milgram). *Let $a(\cdot, \cdot) : V \times V \to \mathbb{R}$ be a continuous, V-elliptic bilinear form. Then, for each $l \in V^*$ the variational problem*

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V$$

has a unique solution $u \in V$. Moreover, we have the stability estimate:

$$\|u\| \leq \frac{1}{\alpha} \|l\|_{V^*}.$$

with

$$\|l\|_{V^*} := \sup_{\varphi \neq 0} \frac{|l(\varphi)|}{||\varphi||_V}.$$

Proof. The proof contains typical arguments often used in optimization and calculus of variations. Consequently, we work with the minimization formulation (M) rather than (V) and finally transfer the result to (V) as in the equivalent characterizations. The goal is to construct a sequence of solutions and to pass to the limit. Then we must show that the limit is contained in the space V.

We define $J(\cdot) = \frac{1}{2}a(\cdot,\cdot) - l(\cdot)$. We choose (recall the best approximation proofs and the remarks regarding minimizing sequences therein) again a minimizing sequence $(u_n)_{n\in\mathbb{N}}$ in V is characterized by the property

$$\lim_{n\to\infty} J(u_n) = d := \inf_{\varphi\in V} J(\varphi).$$

Hence

$$2J(\varphi) = a(\varphi,\varphi) - 2l(\varphi) \geq \alpha||\varphi||_V^2 - 2||l||\,||\varphi||_V, \tag{100}$$

where, in the last term, the operator norm $|l(\varphi)| \leq ||l||||\varphi||_V$ is used. In particular, the linear functional l is bounded and consequently $\|l\|_{V*} < \infty$. This shows $d > -\infty$ because

$$(\|\varphi\|_V^2 - 2\|\,\|\varphi\|_V) \to \infty \quad \text{for } \|\phi\| \to \infty.$$

For further computations the parallelogram rule is defined as

$$||u-\varphi||_V^2 + ||u+\varphi||_V^2 = 2||u||_V^2 + 2||\varphi||_V^2 \quad \forall u,\varphi \in V.$$

Constructing a norm via the bilinear form gives

$$a(u-\varphi,u-\varphi) = (u-\varphi,u-\varphi)_V^2 = ||u-\varphi||_V^2$$

it holds

$$a(u-\varphi,u-\varphi) + a(u+\varphi,u+\varphi) = 2a(u,u) + 2a(\varphi,\varphi). \tag{101}$$

Previous work is used in the next estimation. We begin at the V-ellipticity of a:

$$\begin{aligned}\alpha||u_n-u_m||_V^2 &\leq a(u_n-u_m,u_n-u_m)\\ &\overset{(6.198)}{=} 2a(u_n,u_n) + 2a(u_m,u_m) - a(u_n+u_m,u_n+u_m)\\ &= 2a(u_n,u_n) + 2a(u_m,u_m) - 4a\left(\frac{u_n+u_m}{2},\frac{u_n+u_m}{2}\right).\end{aligned}$$

Adding zero in form of

$$-4l(u_n) - 4l(u_m) + 4l(u_n) + 4l(u_m) = -4l(u_n) - 4l(u_m) + 8l\left(\frac{u_n+u_m}{2}\right)$$

yields

$$\alpha||u_n-u_m||_V^2 \leq 2a(u_n,u_n) + 2a(u_m,u_m) - 4a\left(\frac{u_n+u_m}{2},\frac{u_n+u_m}{2}\right) \tag{102}$$

$$- 4l(u_n) - 4l(u_m) + 8l\left(\frac{u_n+u_m}{2}\right) \tag{103}$$

$$= 2a(u_n,u_n) - 4l(u_n) + 2a(u_m,u_m) - 4l(u_m) \tag{104}$$

$$- 4a\left(\frac{u_n+u_m}{2},\frac{u_n+u_m}{2}\right) + 8l\left(\frac{u_n+u_m}{2}\right) \tag{105}$$

$$\overset{(118)}{=} 4J(u_n) + 4J(u_m) - 8J\left(\frac{u_n+u_m}{2}\right). \tag{106}$$

Hence $J\left(\frac{u_n+u_m}{2}\right) \geq d$ and

$$\lim_{n\to\infty} J(u_n) + \lim_{m\to\infty} J(u_m) = d + d = 2d.$$

Therefore in the limit:

$$\alpha||u_n-u_m||_V^2 \leq 4d + 4d - 8d = 0,$$

which shows the Cauchy property:

$$\alpha||u_n - u_m||_V^2 \to 0 \quad (n, m \to \infty).$$

Therefore, the limit u exists in the strong sense in V (recall that V is a Hilbert space and therefore complete) and it holds $\lim_{n\to\infty} u_n = u$. Since $a(\cdot,\cdot)$ and $l(\cdot)$ are linear and bounded (i.e., continuous), the functional $J(\cdot)$ is linear and continuous as well, and it follows:

$$J(u) = \lim_{n\to\infty} J(u_n) = d$$

showing that the limit u exists and is an element of V.

Remark 6.131. *Be careful when the functional $J(u)$ is not linear anymore. Here, the continuity of $J(\cdot)$ is not sufficient to proof convergence. One needs to go to* ***weak convergence*** *and deeper results in functional analysis. A nice idea for quadratic optimization problems in Hilbert spaces recapitulating the basic ingredients of functional analysis is provided in [67][Section 2.4 and Section 2.5].*

(M) YIELDS (V)
It remains to show that the minimum of (M) is equivalent to the solution of (V). The following strategy for the minimizing problem is used:

$$J(u) = \min_{\varphi\in V} J(\varphi) \quad \Leftrightarrow \quad J(u) \le J(u+\varepsilon\varphi), \quad \varepsilon \in \mathbb{R}. \tag{107}$$

A minimum is achieved by differentiation with respect to ε and setting the resulting equation to zero:

$$\frac{d}{d\varepsilon} J(u+\varepsilon\varphi)\bigg|_{\varepsilon=0} = 0. \tag{108}$$

The assertion is shown with the help of the equivalence of the variational problem (V) and the previous minimization formulation (107). Thus,

$$a(u,\varphi) = l(\varphi) \quad \forall\varphi \in V. \tag{109}$$

UNIQUENESS.
Consider two solutions $u_1, u_2 \in V$ of the variational equation (137) which imply

$$a(u_1,\varphi) = l(\varphi) \quad \text{and} \quad a(u_2,\varphi) = l(\varphi) \quad \text{for all } \varphi \in V.$$

Subtraction yields

$$a(u_1 - u_2, \varphi) = 0 \quad \text{for all } \varphi \in V.$$

Choosing $\varphi = u_1 - u_2$ gives us

$$a(u_1 - u_2, u_1 - u_2) = \|u_1 - u_2\|_V^2$$

and therefore

$$\alpha\|u_1 - u_2\|_V^2 = 0 \quad \Leftrightarrow \quad u_1 - u_2 = 0 \quad \Leftrightarrow \quad u_1 = u_2.$$

STABILITY.
Follows from consideration of

$$\alpha||u||_V^2 \le a(u,u) = l(u) \le C||u||_V.$$

Hence

$$||u||_V \le \frac{C}{\alpha} \qquad \text{with } C = \|l\|_{V^*} = \sup_{u\neq 0} \frac{|l(u)|}{||u||_V}.$$

All assertions were shown. □

Remark 6.132. *For a symmetric bilinear form, i.e.,*

$$a(u, \phi) = a(\phi, u)$$

*the **Riesz representation Theorem** 6.133 can be applied. The key observation is that in this case the mapping $u \mapsto \sqrt{a(u,u)}$ defines a scalar product; and consequently a norm because we work in a Hilbert space. Then, there exists a unique element $u \in V$ such that*

$$l(\phi) = a(u, \phi) \quad \forall \phi \in V.$$

The Lax-Milgram lemma is in particular useful for non-symmetric bilinear forms.

Theorem 6.133 (Riesz representation theorem)**.** *Let $l : V \to \mathbb{R}$ be a continuous linear functional on a Hilbert space V. Then there exists a unique element $u \in V$ such that*

$$(u, \phi)_V = l(\phi) \quad \forall \phi \in V.$$

Moreover, this operation is an isometry, that is to say, $\|l\|_* = \|u\|_V$. This works since $(u, \phi)_V = a(u, \phi)$.

Proof. For a proof the Riesz representation theorem, we refer to [14]. □

Proposition 6.134. *In the symmetric case, $a(u, \phi) = (\nabla u, \nabla \phi)$ defines a scalar product on the Hilbert space V. Moreover, the right hand side functional $l(\cdot)$ is linear and continuous. Thus, the assumptions of the Riesz representation theorem are fulfilled and there exists a unique solution to the Poisson problem.*

Exercise 8. *Show that the assumptions of the Lax-Milgram lemma are verified for the 1D Poisson model problem. Thus, the variational formulation is well-posed (existence, uniqueness and stability of a solution).*

6.8.6 The energy norm

As we have seen before, the (symmetric) bilinear form defines a scalar product from which we can define another norm. The continuity and coercivity of the bilinear form yield the **energy norm**:

$$\|v\|_a^2 := a(v, v), \quad v \in V.$$

This norm is aquivalent to the V-norm of the space V, i.e.,

$$c\|v\|_V \leq \|v\|_a \leq C\|v\|_V, \quad \forall v \in V$$

and two positive constants c and C. We can even precisely determine these two constants:

$$\alpha||u||_V^2 \leq a(u, u) \leq \gamma\|u\|_V^2$$

yielding $c = \sqrt{\alpha}$ and $C = \sqrt{\gamma}$. The corresponding scalar product is defined by

$$(v, w)_a := a(v, w).$$

Finally, the best approximation property can be written in terms of the a-norm:

$$(u - u_h, v)_a = 0 \quad \forall v \in V_h.$$

6.8.7 The Poincaré inequality

In order to apply the Lax-Milgram lemma to partial differential equations complemented with homogeneous boundary conditions, we need another essential tool, the so-called **Poincaré inequality**.

Proposition 6.135. *Let $\Omega \subset \mathbb{R}^n$ be a bounded, open, domain. There exists a constant d_Ω (depending on the domain) such that*

$$\int_\Omega |v(x)|^2 \, dx \leq d_\Omega \int_\Omega |\nabla v(x)|^2 \, dx.$$

holds true for each $v \in H_0^1(\Omega)$.

Remark 6.136. *It is clear that the Poincaré inequality specifically holds true for functions from H_0^1 but not H^1. A counter example is $v \equiv 1$, which is in H^1, but violates the Poincaré inequality.*

Proof. The proof can be found in all textbooks, e.g., Rannacher [56], Wloka [76] or Braess [12], page 29. □

Corollary 6.137. *Let $\Omega \subset \mathbb{R}^n$ be an open, bounded, domain. The semi-norm*

$$|v|_{H_0^1} = \Big(\int_\Omega |\nabla v|^2 \, dx \Big)^{1/2}$$

is a norm on H_0^1 and equivalent to the usual H^1 norm.

Proof. Let $v \in H_0^1$. It holds on the one hand (trivial!):

$$|v|_{H_0^1} \leq \|v\|_{H_0^1}.$$

On the other hand, using Poincaré's inequality, we obtain:

$$\|v\|_{H^1}^2 \leq (d_\Omega + 1) \int_\Omega |\nabla v|^2 \, dx = (d_\Omega + 1)|v|_{H_0^1}^2,$$

yielding

$$|v|_{H_0^1}^2 \leq \|v\|_{H^1}^2 \leq (d_\Omega + 1)|v|_{H_0^1}^2.$$

□

6.8.8 Trace theorems

We have seen that H^1 functions are not continuous when $n \geq 2$. Consequently, we cannot define pointwise values and secondly, boundaries of measure zero. Thus, it is a priori a bit difficult to define boundary values in Sobolev spaces. We face the question in which sense boundary values can be defined when working with Sobolev spaces.

We have:

Theorem 6.138 (Trace theorem)**.** *Let Ω be an open, boundary domin of class C^1. We define the* ***trace*** *$\gamma_0 : H^1 \cap C(\bar{\Omega}) \to L^2(\partial\Omega) \cap C(\partial\bar{\Omega})$ as*

$$v \mapsto \gamma_0(v) = v|_{\partial\Omega}.$$

By continuity γ_0 can be extended to the entire domain Ω. It holds

$$\|v\|_{L^2(\partial\Omega)} \leq C\|v\|_{H^1(\Omega)}.$$

Proof. See Rannacher [56], page 30. □

Theorem 6.139 (Trace theorem for derivatives)**.** *Let Ω be an open, boundary domin of class C^1. We define the* ***trace*** *$\gamma_0 : H^2 \cap C^1(\bar{\Omega}) \to L^2(\partial\Omega) \cap C(\partial\bar{\Omega})$ as*

$$v \mapsto \gamma_1(v) = \partial_n v|_{\partial\Omega}.$$

By continuity, γ_1 can be extended from $H^2(\Omega)$ mapped into $L^2(\partial\Omega)$. It holds

$$\|\partial_n v\|_{L^2(\partial\Omega)} \leq C\|v\|_{H^2(\Omega)}.$$

Proof. The existence of γ_1 follows from the first trace theorem previously shown. Since $v \in H^2$, we have $\nabla v \in (H^1)^n$. Consequently, we can define the trace of ∇v on $\partial\Omega$, which is then a trace on $(L^2(\partial\Omega))^n$. Since the normal vector is a bounded, continuous function on $\partial\Omega$, we have $\nabla v \cdot n \in L^2(\partial\Omega)$. □

Remark 6.140. *The second trace theorem has important consequences on the regularity of the boundary functions. In general, for variational problems we can only proof that the solution u is from H^1. In this case, the normal derivative is only from H^{-1} (a space with very low regularity). Only under additional assumptions on the data and the domain, we can show $u \in H^2$, which would yield better regularity of the normal derivative $\nabla u \cdot n$ on $\partial\Omega$.*

6.8.9 A compactness result

A fundamental result in prooving well-posedness of PDEs in infinite dimensional spaces (recall that boundedness and closetness are not equivalent to compactness for infinite-dimensional spaces) is the following result:

Theorem 6.141 (Rellich)**.** *Let Ω be an open, bounded domain of class C^1. The embedding of H^1 into L^2 is compact. In other words: from each bounded sequence $(u_n) \subset H^1$, we can extract a subsequence $(u_{n_l}) \subset L^2$ with a limit in L^2.*

Proof. See for instance Wloka [76]. □

6.9 Theory of elliptic problems

Gathering all results from the previous sections, we are now in a state to proof well-posedness of the Poisson problem.

6.9.1 The formal procedure

In most cases, one starts with the differential problem (D). In order to establish well-posedness, one can work in three steps:

- Derive formally a variational problem (V).
- Proof well-posedness for the variational problem (here for linear problems using Lax-Milgram).
- Go back and show $(V) \to (D)$ (see also Section 6.1.2) using the fundamental lemma of calculus of variations.

6.9.2 Poisson's problem: homogeneous Dirichlet conditions

Proposition 6.142. *Let Ω be a bounded domain of class C^1 and let $f \in L^2$. Let $V := H_0^1$. Then, the Poisson problem has a unique solution $u \in V$ and it exists a constant c_p (independant of f) such that the stability estimate*

$$\|u\|_{H^1} \leq c_p \|f\|_{L^2}$$

holds true.

Proof. We use the Lax-Milgram lemma and check its assumptions. Since H_0^1 is a subspace of H^1, we work with the H^1 norm.

Continuity of $a(\cdot,\cdot)$

Using Cauchy-Schwarz, we obtain:

$$\begin{aligned} |a(u,\phi)| = |\int_\Omega \nabla u \cdot \nabla \phi \, dx| &\leq \int_\Omega |\nabla u| \, |\nabla \phi| \, dx \leq \Big(\int_\Omega |\nabla u|^2 \, dx\Big)^{1/2} \Big(\int_\Omega |\nabla \phi|^2 \, dx\Big)^{1/2} \\ &\leq \Big(\int_\Omega (u^2 + |\nabla u|^2) \, dx\Big)^{1/2} \Big(\int_\Omega (\phi^2 + |\nabla \phi|^2) \, dx\Big)^{1/2} = \|u\|_{H^1} \|\phi\|_{H^1}. \end{aligned}$$

Coercivity of $a(\cdot,\cdot)$

Using the Poincaré inequality, we obtain

$$\begin{aligned} a(u,u) = \int_\Omega |\nabla u|^2 \, dx &= \frac{1}{2} \int_\Omega |\nabla u|^2 \, dx + \frac{1}{2} \int_\Omega |\nabla u|^2 \, dx \\ &\geq \frac{1}{2} \int_\Omega |\nabla u|^2 \, dx + \frac{1}{2d_\Omega} \int_\Omega u^2 \, dx \geq \frac{1}{2} \min(1, \frac{1}{d_\Omega}) \|u\|_{H^1}^2. \end{aligned}$$

Continuity of $l(\cdot)$

Using again Cauchy-Schwarz and Poincaré, we obtain:

$$|l(\phi)| = |(f,\phi)_{L^2}| \leq \|f\|_{L^2}\|\phi\|_{L^2} \leq \|f\|_{L^2}\|\phi\|_{H^1}.$$

Stability estimate

From Lax-Milgram we know:

$$\|u\| \leq \frac{1}{\alpha}\|l\|_{V^*} \leq \frac{|l(\phi)|}{||\phi||_V} \leq \frac{\|f\|_{L^2}\|\phi\|_{H^1}}{\|\phi\|_{H^1}} = \frac{1}{\alpha}\|f\|_{L^2}.$$

Everything is shown now.

□

Remark 6.143 (On the dual norm)**.** *Previously we have had:*

$$|l(\phi)| = |(f,\phi)| \leq \|f\|_{L^2}\|\phi\|_{L^2}.$$

When $f \in L^2$. But this estimate works also for $f \in H^{-1}$ the dual space of H^1. Then:

$$|l(\phi)| = |(f,\phi)| \leq \|f\|_{H^{-1}}\|\phi\|_{H^1}.$$

The smallest possible choice for $C = \|l\|_{V^*} = \|f\|_{H^{-1}}$ (see Section 3.10) is

$$C = \sup_{\phi \in H_0^1, \phi \neq 0} \frac{|(f,\phi)|}{\|\phi\|_{H^1}},$$

which is indeed a norm and in general denoted by $\|f\|_{H^{-1}}$. In the L^2 case we have

$$\|f\|_{L^2} = \sup_{\phi \in L^2, \phi \neq 0} \frac{|(f,\phi)|}{\|\phi\|_{L^2}}.$$

Here, we take the supremum over a larger space; recall that $H^1 \subset L^2$. Consequently

$$\|f\|_{H^{-1}} \leq \|f\|_{L^2}$$

and therefore

$$L^2 \subset H^{-1}$$

and finally

$$H^1 \subset L^2 \subset H^{-1}.$$

Under stronger conditions, we have the following regularity result:

Proposition 6.144 (Regularity of the exact solution)**.** *Let Ω be bounded, open and of class C^2. Then, the solution of the Poisson problem is $u \in H^2(\Omega)$ and it holds the stability estimate:*

$$\|u\|_{H^2} \leq C\|f\|_{L^2}.$$

Proof. See for instance [47]. □

Proposition 6.145 (Higher regularity of the exact solution)**.** *Let Ω be bounded, open and of class C^k (i.e., very smooth). Then, the solution of the Poisson problem is $u \in H^{k+2}(\Omega)$ and it holds the stability estimate:*

$$\|u\|_{H^{k+2}} \leq C\|f\|_{H^k}.$$

If the boundary is not smooth enough (even when f is sufficiently smooth), this result will not hold true anymore.

6.9.3 Nonhomogeneous Dirichlet conditions

In this section, we investigate non-homogeneous Dirichlet boundary conditions. Despite one only needs to add formally a value or a function for the boundary conditions, the mathematical formulations are non-trivial.

We have

Formulation 6.146. *Let $\Omega \subset \mathbb{R}^n$ an open and bounded domain. Let $f \in L^2(\Omega)$. Find u such that*

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= h \quad \text{on } \partial\Omega. \end{aligned}$$

Since we seek $u \in H^1(\Omega)$ we assume that there exists a function $u_0 \in H^1$ such that

$$trace(u_0) = h \quad \text{on } \partial\Omega.$$

We know that $h \in L^2(\partial\Omega)$.

In the following, we concentrate first on the derivation of a weak formulation. The idea is to work with the space H^1 by extracting the nonhomogeneous Dirichlet condition. We write

$$u = u_0 + \tilde{u}, \quad \tilde{u} \in H_0^1.$$

Then:

$$\int_\Omega \nabla u \nabla \phi \, dx = \int_\Omega \nabla(u_0 + \tilde{u}) \nabla \phi \, dx = \int_\Omega (\nabla u_0 \nabla \phi + \tilde{u} \nabla \phi) \, dx$$

This yields the variational problem:

Formulation 6.147. *Find $\tilde{u} \in H_0^1$ such that*

$$(\nabla \tilde{u}, \nabla \phi) = (f, \phi) - (\nabla u_0, \nabla \phi) \quad \forall \phi \in H_0^1.$$

Having found $\tilde{u}$, we obtain $u \in H^1$ via

$$u = u_0 + \tilde{u}.$$

The equivalent formulation, often found in the literature, is:

Formulation 6.148. *Find $u \in \{h + H_0^1\}$ such that*

$$(\nabla u, \nabla \phi) = (f, \phi) \quad \forall \phi \in H_0^1.$$

Example 6.149 (Poisson in 1D)**.** *Let*

$$\begin{aligned} -u''(x) &= f, \\ h &= 4 \quad \text{on } \partial\Omega. \end{aligned}$$

Here $u_0 = 4$ in Ω. When we solve the Poisson problem with finite elements, we proceed as shown before and obtain $\tilde{u} \in H_0^1$ and finally simply

$$u = \tilde{u} + 4$$

to obtain the final u. For nonconstant u_0 (thus nonconstant h on $\partial\Omega$, please be careful in the variational formulation since the term

$$(\nabla u_0, \nabla \phi)$$

will not vanish and needs to be assembled. For instance let $u(0) = 0$ and $u(1) = 1$ in $\Omega = (0, 1)$. Then, $u_0 = x$ and $trace(u_0) = h$ with $h(0) = 1$ and $h(1) = 1$. Thus we solve Formulation 6.147 to obtain $\tilde{u}$ and finally:

$$u = \tilde{u} + u_0 = \tilde{u} + x.$$

Proposition 6.150. *Formulation 6.147 yields a unique solution.*

Proof. The proof is similar to Formulation 6.142. □

6.9.4 Neumann conditions

We have seen in the 1D case that the pure Neumann problem has solutions defined up to a constant value.

Formulation 6.151 (Pure Neumann problem)**.** *Let $\Omega \subset \mathbb{R}^n$ be an open, bounded, connected, domain and $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$. We consider*

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ \partial_n u &= g \quad \text{on } \partial\Omega. \end{aligned}$$

Remark 6.152. *We call the problem 'pure' Neumann problem because it is of course possible to have different boundary conditions on different boundary parts. But once we have Dirichlet conditions on some part of the boundary $\partial\Omega_D$ with $\partial\Omega_D \neq \emptyset$ and on the remaining boundary $\partial\Omega_N$ Neumann conditions, the problem of non-uniqueness is gone since the solution is fixed on the Dirichlet part.*

It only exists a solution if the following compatibility condition is fulfilled:

$$\int_\Omega f(x)\,dx + \int_{\partial\Omega} g(x)\,ds = 0. \tag{110}$$

The compatibility condition is necessary and sufficient. We have the following result:

Proposition 6.153. *Let $\Omega \subset \mathbb{R}^n$ an open, bounded, connected domain of class C^1. Let $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$, which satisfy the compabitiblity condition* (110)*. Then, it exists a weak solution $u \in H^1(\Omega)$ of Formulation 6.151, which is unique up to a constant value.*

Proof. The variational formulation is obtained as usually:

$$(\nabla u, \nabla\phi) = (f, \phi) + \langle g, \phi\rangle \quad \text{for all admissible } \phi.$$

Here, we have the first difficulty, since the Hilbert space H^1 will not yield the coercivity of the corresponding bilinear form (in fact this is related to the non-uniqueness). For this reason, we define the space:

$$V := \{v \in H^1 |\ \int_\Omega v\,dx = 0\}.$$

Hence, the correct variational formulation is:

$$\text{Find } u \in V: \quad (\nabla u, \nabla\phi) = (f, \phi) + \langle g, \phi\rangle \quad \forall \phi \in V.$$

As before, we adopt the Lax-Milgram Lemma to show existence and uniqueness of the solution. The continuity of the bilinear form and right hand side are obvious; see the proof of Proposition 6.142.

The only delicate step is the coercivity. Here, Poincaré's inequality will not work and we have to adopt a variant of Friedrichs inequality

$$\|v\|_{L^2} \leq c(|\tilde{v}| + |v|_{H^1}) \quad \text{for } v \in H^1$$

with $c := c(\Omega)$ and with $\tilde{v} = \int_\Omega v\,dx/\mu(\Omega)$ being the mean value of v. Thus, it holds:

$$a(u,u) = \int_\Omega |\nabla u|^2\,dx = |u|^2_{H^1} \geq \alpha \|u\|^2_{H_1}.$$

Thus, the Lax-Milgram lemma yields a unique solution in the space V. We notice that this solution must satisfy the compatibility condition (110) because Gauss' divergence theorem yields

$$\int_\Omega f\,dx = -\int_\Omega div(\nabla u) = -\int_{\partial\Omega} \nabla u \cdot n\,ds = -\int_{\partial\Omega} g\,ds.$$

Hence

$$\int_\Omega f\,dx + \int_{\partial\Omega} g\,ds = 0.$$

Interestingly this condition is also sufficient. Lax-Milgram say that we obtain the solution $u \in V$ via solving

$$a(u,\phi) = (f,\phi) + \langle g,\phi\rangle \quad \forall \phi \in V.$$

Using the compatibility condition we see the following:

$$\begin{aligned} & \int_\Omega f\,dx + \int_{\partial\Omega} g\,ds = 0, \\ \Leftrightarrow \quad & c\int_\Omega f\,dx + c\int_{\partial\Omega} g\,ds = 0 \\ \Leftrightarrow \quad & \int_\Omega f\cdot c\,dx + \int_{\partial\Omega} g\cdot c\,ds = 0 \\ \Leftrightarrow \quad & \int_\Omega f\cdot \phi\,dx + \int_{\partial\Omega} g\cdot \phi\,ds = 0 \\ \Leftrightarrow \quad & a(u,\phi) = (f,\phi) + \langle g,\phi\rangle. \end{aligned}$$

We observe that this relation holds in particular for $\phi = const$. For more information see Braess [12]. □

6.9.5 Robin conditions

Given:

Formulation 6.154. *Let $\Omega \subset \mathbb{R}^n$ and $\partial\Omega$ its boundary. Furthermore, $f \in L^2(\Omega)$ and $g \in L^2(\partial\Omega)$ and $c \in L^\infty(\Omega)$ and $b \in L^\infty(\partial\Omega)$. Then: Find u such that*

$$\begin{aligned} -\Delta u + cu &= f, \quad \text{in } \Omega, \\ \partial_n u + bu &= g, \quad \text{on } \partial\Omega. \end{aligned}$$

We first formulate a weak formulation. As function space, we use $V := H^1$. The bilinear and linear forms are given by:

$$\begin{aligned} a(u,\phi) &= \int_\Omega \nabla u\cdot\nabla\phi\,dx + \int_\Omega cu\phi\,dx + \int_{\partial\Omega} au\phi\,ds, \\ l(\phi) &= \int_\Omega fv\,dx + \int_{\partial\Omega} g\phi\,ds. \end{aligned}$$

We notice that the functional l is not anymore a L^2 function, but only a linear functional on the space V.

To proof coercivity, we need the following general form of Friedrichs inequality:

Lemma 6.155. *Let $\Omega \subset \mathbb{R}^n$ a bounded domain of class C^1 and $\Gamma \subset \partial\Omega$ a measurable set with $|\Gamma| > 0$. Then there exists a constant C (independent of u) such that*

$$\|u\|_{H^1}^2 \le C(|u|_{H^1}^2 + \Big(\int_\Gamma u\,ds)^2\Big) \quad \forall y \in H^1(\Omega).$$

Proof. For a proof see [76]. □

Proposition 6.156. *Let Ω be a bounded domain of class C^1 and functions $c \in L^\infty(\Omega)$ and $b \in L^\infty(\partial\Omega)$ almost everywhere non-negative with the condition:*

$$\int_\Omega c^2\,dx + \int_{\partial\Omega} b^2\,dx > 0.$$

Then, the variational Robin problem has for each $f \in L^2, g \in L^2(\partial\Omega)$ a unique solution $u \in H^1 =: V$. Furthermore, it holds the stability estimate:

$$\|u\|_{H^1} \le C\big(\|f\|_{L^2} + \|g\|_{L^2(\partial\Omega)}\big).$$

Proof. We verify again the assumptions of the Lax-Milgram lemma. We work with the Cauchy-Scharz inequality, the generalization of Friedrichs inequality and also the trace lemma since we have deal with functions on the boundary $\partial\Omega$.

PRELEMINARIES

First:

$$|\int_\Omega cu\phi\,dx| \leq \|c\|_{L^\infty}\|u\|_{L^2}\|\phi\|_{L^2} \leq \|c\|_{L^\infty}\|u\|_{H^1}\|\phi\|_{H^1}.$$

With the trace lemma, we have:

$$|\int_{\partial\Omega} bu\phi\,dx| \leq \|b\|_{L^\infty(\partial\Omega)}\|u\|_{L^2(\partial\Omega)}\|\phi\|_{L^2(\partial\Omega)} \leq C\|b\|_{L^\infty(\partial\Omega)}\|u\|_{H^1(\partial\Omega)}\|\phi\|_{H^1(\partial\Omega)}.$$

CONTINUITY OF $a(\cdot,\cdot)$

We establish the continuity:

$$|a(u,\phi)| = |(\nabla u,\nabla\phi) + (cu,\phi)_{L^2} + (bu,\phi)_{L^2(\partial\Omega)}| \leq C\|u\|_{H^1}\|\phi\|_{H^1}.$$

In the last estimates we used the inequalities derived in the preleminaries.

COERCIVITY OF $a(\cdot,\cdot)$

$$a(u,u) \geq \frac{\min(1,C_1)}{C_2\max(1,|\Gamma|)}\|u\|^2_{H^1}$$

where $b(x) \geq C_1$. We left out several steps, which can be verified by the reader.

CONTINUITY OF $l(\phi)$

We estimate:

$$|l(\phi)| = |\int_\Omega fv\,dx + \int_{\partial\Omega} g\phi\,ds| \leq C(\|f\|_{L^2} + \|g\|_{L^2})\|\phi\|_{H^1}.$$

STABILITY ESTIMATE

From Lax-Milgram we know:

$$\|u\|_{H^1} \leq \frac{1}{\alpha}\|l\|_{V^*} \leq \frac{|l(\phi)|}{||\phi||_V} \leq \frac{C(\|f\|_{L^2} + \|g\|_{L^2})}{\|\phi\|_{H^1}} = C(\|f\|_{L^2} + \|g\|_{L^2}).$$

Everything is shown now.

□

6.10 Finite element spaces

We have now a theoretical basis of the finite element method and for 1D problems we explained in Section 6.3 how to construct explicitely finite elements and how to implement them in a code.

It remains to give a flavour of the construction of finite elements in higher dimensions. Since the major developments have been taken place some time ago, there are nice books including all details:

- Johnson [43] provides an elementary introduction that is easily accessible.
- The most classical book is by Ciarlet [20].
- Another book including many mathematical aspects is by Rannacher [56].

Consequently, the aim of this chapter is only to provide the key ideas to get the reader started. For further studies, we refer to the literature. In addition, to the three above references the reader may consult the books mentiond in Chapter 1 of these lecture notes.

We follow chapter 3 in [43] to provide an elementary introduction. The generalization is nicely performed by Ciarlet [20][Chapter 2]. Moreover, Ciarlet also introduces in details rectangular finite elements and parametric and isoparametric finite elements in order to approximate curved boundaries. Isoparametric elements are also explained in detail by Rannacher [56].

Finite element spaces consist of piecewise polynomials functions on a decomposition (historically still called triangulation), i.e., a mesh,

$$\mathcal{T}_h := \bigcup_i K_i$$

of a bounded domain $\mathbb{R}^n, n = 1, 2, 3$. The K_i are called **mesh elements**.

In particular:

- $n = 1$: The elements K_i are intervals;
- $n = 2$: The elements K_i are triangles or quadrilaterals;
- $n = 3$: The elements $K - i$ are tetrahedrons or hexahedra.

A **conforming** FEM scheme requires $V_h \subset V$ with $V := H^1$ for instance. Since V_h consists of piecewise polynomials, we have

$$V_h \subset H^1(\Omega) \quad \Leftrightarrow \quad V_h \subset C^0(\bar{\Omega}).$$

As we have seen in the 1D case, to specify a **finite element** (see Section 6.3.6 - we remind that this section holds not only for 1D, but also higher dimensions), we need three ingrediants:

- A mesh $\mathcal{T}_h$ representing the domain Ω.
- Specification of $v \in V_h$ on each element K_i.
- Parameters (degrees of freedom; DoFs) to describe v uniquely on K_i.

6.10.1 Example: a triangle in 2D

Let us follow [43][Chapter 3] and let us work in $2D$. Let K be an element of $\mathcal{T}_h$. A $P_1(K)$ function is defined as

$$v(x) = a_{00} + a_{10}x_1 + a_{01}x_2, \quad x = (x_1, x_2) \in K$$

and coefficients $a_{ij} \in \mathbb{R}$.

Definition 6.157 (A basis of P_1 (linear polynomials))**.** *A basis of the space P_1 is given by*

$$P_1 = \{\phi_1, \phi_2, \phi_3\}$$

with the basis functions

$$\phi_1 \equiv 1, \quad \phi_2 = x_1, \quad \phi_3 = x_2.$$

The dimension is $dim(P_1) = 3$. Clearly, any polynomial from P_1 can be represented through the linear combination:

$$p(x) = a_{00}\phi_1 + a_{10}\phi_2 + a_{01}\phi_3.$$

Definition 6.158 (A basis of P_2 (quadratic polynomials))**.** *A basis of the space P_1 is given by*

$$P_2 = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6\}$$

with the basis functions

$$\phi_1 \equiv 1, \quad \phi_2 = x_1, \quad \phi_3 = x_2, \quad \phi_4 = x_1^2, \quad \phi_5 = x_2^2, \quad \phi_6 = x_1x_2.$$

The dimension is $dim(P_1) = 6$. Again, any polynomial from P_2 can be represented through the linear combination:

$$p(x) = a_{00}\phi_1 + a_{10}\phi_2 + a_{01}\phi_3 + a_{20}\phi_4 + a_{02}\phi_5 + a_{11}\phi_6.$$

Definition 6.159 (A basis of P_s)**.** *A basis of the space P_1 is given by*

$$P_s = \{v|\ v(x) = \sum_{0\leq i+j\leq s} a_{ij} x_1^i x_2^j\}$$

with the dimension

$$dim(P_s) = \frac{(s+1)(s+2)}{2}.$$

In the following, we explicitely construct the space of linear functions on K. Let

$$V_h = \{v \in C^0(\bar{\Omega})|\ v|_K \in P_1(K), \quad \forall K \in \mathcal{T}_h\},$$

which we still know from our 1D constructions. To uniquely describe the functions in V_h we choose the global degrees of freedom, here the values of the nodal points of $\mathcal{T}_h$.

Remark 6.160. *Choosing the nodal points is the most common choice and is a Lagrangian scheme. Using for instance derivative information in the nodal points, we arrive at Hermite polynomials.*

We need to show that the values of the nodal points yield a unique polynomial in $P_1(K)$.

Definition 6.161 (Local DoFs)**.** *Let $n = 2$ and let K be a triangle with vertices $a^i, i = 1, 2, 3$. These values are called* ***local degrees of freedom****.*

Proposition 6.162. *Let $K \in \mathcal{T}_h$ be an element (a triangle) with vertices $a^i = (a_1^i, a_2^i) \in K$. A function on K is uniquely determined by the local degrees of freedom from Definition 6.161. In other words, for given values α_i, there exists a unique function $v \in P_1(K)$ such that*

$$v(a^i) = \alpha_i, \quad i = 1, 2, 3.$$

Proof. Recalling the definition of a P_1 polynomial on K, we have

$$v(x) = a_{00} + a_{10}x_1 + a_{01}x_2.$$

In the support points a^i, we have:

$$v(a^i) = a_{00} + a_{10}a_1^i + a_{01}a_2^i = \alpha_i, \quad i = 1, 2, 3.$$

This yields a quadratic system with three equations for three unknowns. Using matrix notation, we can plug the coefficients into a system

$$Ba = b$$

with

$$B = \begin{pmatrix} 1 & a_1^1 & a_2^1 \\ 1 & a_1^2 & a_2^2 \\ 1 & a_1^3 & a_2^3 \end{pmatrix}, \quad a = \begin{pmatrix} a_{00} \\ a_{10} \\ a_{01} \end{pmatrix}, \quad b = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}.$$

The system $Ba = b$ has a unique solution when $det(B) \neq 0$, which can be easily checked here and completes the proof. □

Algorithm 6.163 (Construction of P_1 shape functions)**.** *For the explicit construction of such P_1 shape function, we simply need to solve*

$$v(a^i) = a_{00} + a_{10}a_1^i + a_{01}a_2^i = \alpha_i, \quad i = 1, 2, 3,$$

for the unknown coefficients a_{ij}. This means, we need to construct three basis functions $\psi_i \in P_1(K)$ with

$$\psi_i(a^j) = \delta_{ij},$$

where δ_{ij} is the already introduced Kronecker delta. Thus, we have to solve the system $Ba = b$ three times:

$$\psi_i(a^j) = a_{00} + a_{10}a_1^j + a_{01}a_2^j = \alpha_j, \quad i, j = 1, 2, 3.$$

Then, we can build the local basis functions via

$$\psi_i(x) = a_{00}^i + a_{10}^i x_1 + a_{01}^i x_2$$

with all quantities explicitely given.

We finally check that the continuity at the nodes a^i is sufficient to obtain a globally continuous function, which is in particular also continuous at the edges between two elements. Let K_1 and K_2 be two elements in $\mathcal{T}_h$ with a common edge Γ and the end points b_1 and b_2. Let $v_i = v|_{K_i} \in P_1(K_i)$ be the restriction of v to K_i. Then, the function $w = v_1 - v_2$ defined on Γ vanishes at the end points b_1 and b_2. Since w is linear on Γ the entire function w must vanish on Γ. Consequently, v is continuous across Γ. Extending to all edges of $\mathcal{T}_h$ be conclude that the function v is a globally continuous function and that indeed $v \in C^0(\bar{\Omega})$.

Definition 6.164 (C^0 elements)**.** *The class of finite elements that fulfills the continuity condition at the edges are called C^0 elements.*

Remark 6.165. *The generalization to n-simplicies can be found in [20][Chapter 2].*

6.10.2 Example: a quadrilateral in 2D

In this section, we briefly present how quadrilaterals can be constructed. The work program of this section is the same as in Section 6.10.1 and is based on [43][Chapter 3].

Let K be an element with four vertices $a^i, i = 1, \dots 4$ and the sides parallel to the coordinate axes in $\mathbb{R}^2$.

Definition 6.166 (A basis of Q_1 (bilinear polynomials))**.** *A basis of the space $Q_1(K)$ on an element K is given by*

$$Q_1 := Q_1(K) = \{\phi_1, \phi_2, \phi_3, \phi_4\}$$

with the basis functions

$$\phi_1 \equiv 1, \quad \phi_2 = x_1, \quad \phi_3 = x_2, \quad \phi_4 = x_1 x_2.$$

The dimension is $dim(P_1) = 3$. Clearly, any polynomial from P_1 can be represented through the linear combination:

$$p(x) = a_{00}\phi_1 + a_{10}\phi_2 + a_{01}\phi_3 + a_{11}\phi_4.$$

Proposition 6.167. *A Q_1 function is uniquely determined by the values of the nodal points a^i.*

Proof. The same as for triangles. □

Proposition 6.168. *The definition of $Q_1(K)$ yields globally continuous functions.*

Proof. Same as for triangles. □

Remark 6.169. *As for triangles, we can easily defined higher-order spaces.*

Remark 6.170. *Be careful, when quadrilateral elements are turned or shifted. Then, the nature of the local polynomials will change. This is one major reason why in practice we work with a master element and all other elements are obtained via transformations from this master element. To this end, we are not restricted to a specific coordinate system and leaves the desired freedom for deformations of the single elements. A lot of details can be found in [56][Pages 105 ff].*

6.10.3 Well-posedness of the discrete problem

With the help of the Lax-Milgram lemma, we obtain:

Proposition 6.171. *Let V a real Hilbert space and $V_h \subset V$. Let $a(u, v)$ be a bilinear form and $l(v)$ a linear form. Then, the discrete problem*

$$\text{Find } u_h \in V_h : a(u_h, \phi_h) = l(\phi_h) \quad \forall \phi_h \in V_h$$

has a unique solution. This discrete solution $u_h \in V_h$ is obtained by solving a linear equation system.

Proof. Existence and uniqueness are obtained from the Lax-Milgram lemma with the same arguments as for the continuous problem. In particular, we have

$$AU = b$$

with

$$(A)_{ij=1}^n = a(\phi_j, \phi_i), \quad (b)_{i=1}^n = l(\phi_i).$$

The coercivity yields the positive definitness of the matrix A and therefore (see lectures on linear algebra), A is invertible:

$$AU \cdot U \geq \alpha \| \sum_{j=1}^n u_j \phi_j \| \geq C \|U\|^2.$$

In fact, we investigate the homogeneous system

$$a(u_h, \phi_h) = 0$$

and show that this only yields the zero solution, which is immediately true using the coercivity.

Furthermore, the symmetry of $a(u, \phi)$ yields the symmetry of A. Linear algebra arguments on matrices (symmetric, positive, definite) yield a unique solution U. □

6.11 Numerical analysis: error estimates

In this section, we perform the numerical analysis of our finite element derivations. We work as before and focus on elementary aspects working out 1D results in order to show the mechanism and to explain the principle ideas. For higher dimensions, we only provide the results and refer to the literature for detailed proofs.

6.11.1 The Céa lemma

The Céa lemma is the only result in this section that holds for higher dimensions. We investigate the best approximation error (recall our 1D findings).

Proposition 6.172 (Céa lemma)**.** *Let V be a Hilbert space and $V_h \subset V$ a finite dimensional subspace. Let the assumptions, Def. 6.129, of the Lax-Milgram Lemma 6.130 hold true. Let $u \in V$ and $u_h \in V_h$ be the solutions of the variational problems. Then:*

$$\|u - u_h\|_V = \frac{\gamma}{\alpha} \inf_{\phi_h \in V_h} \|u - \phi_h\|_V$$

Proof. It holds Galerkin orthogonality:

$$a(u - u_h, w_h) = 0 \quad \forall w_h \in V_h.$$

We choose $w_h := u_h - \phi_h$ and we obtain:

$$\alpha \|u - u_h\|^2 \leq a(u - u_h, u - u_h) = a(u - u_h, u - \phi_h) \leq \gamma \|u - u_h\| \|u - \phi_h\|.$$

This yields

$$\|u - u_h\| \leq \frac{\gamma}{\alpha} \|u - \phi_h\|.$$

Passing to the infimum yields:

$$\|u - u_h\| = \inf_{\phi_h \in V_h} \frac{\gamma}{\alpha} \|u - \phi_h\|.$$

□

Corollary 6.173. *When $a(\cdot, \cdot)$ is symmetric, which is the case for Poisson, then the improved estimate holds true:*

$$\|u - u_h\|_V = \sqrt{\frac{\gamma}{\alpha}} \inf_{\phi_h \in V_h} \|u - \phi_h\|_V$$

Based on the Céa lemma, we have the first (still non-quantitative) error estimate. The key aspect is that we work with a dense subspace U of V which contains simpler functions than V itself from which we can interpolate from U to the discrete space V_h. We have [2]:

Proposition 6.174. *We assume that the hypotheses from before hold true. Furthermore, we assume that $U \subset V$ is dense. We construct an interpolation operator $i_h : U \to V_h$ such that*

$$\lim_{h\to 0} \|v - i_h(v)\| = 0 \quad \forall v \in U$$

holds true. Then, for all $u \in V$ and $u_h \in V_h$:

$$\lim_{h\to 0} \|u - u_h\| = 0.$$

This result shows that the Galerkin solution $u_h \in V_h$ converges to the continuous solution u.

Proof. Let $\varepsilon > 0$. Thanks to the density, for each $u \in V$, it exists a $v \in U$ such that $\|u - v\| \leq \varepsilon$. Moreover, there is an $h_0 > 0$, depending on the choice of ε, such that

$$\|v - i_h(v)\| \leq \varepsilon \quad \forall h \leq h_0.$$

The Céa lemma yields now:

$$\|u - u_h\| \leq C\|u - i_h(v)\| = C(\|u - v + v - i_h(v)\|) \leq C(\|u - v\| + \|v - i_h(v)\|) \leq C(\varepsilon + \varepsilon) = 2C\varepsilon.$$

These proofs employs the trick, very often seen for similar calculations, that at an appropriate place an appropriate function, here v is inserted, and the terms are split thanks to the triangular inequality. Afterwards, the two separate terms can be estimated using the assumptions of other known results. □

6.11.2 H^1 and L^2 estimates in 1D

First, we need to construct an interpolation operator in order to approximate the continuous solution at certain nodes.

Definition 6.175. *Let $\Omega = (0, 1)$. A P_1 interpolation operator $i_h : H^1 \to V_h$ is defined by*

$$(i_h v)(x) = \sum_{j=0}^{n+1} v(x_j)\phi_j(x) \quad \forall v \in H^1.$$

This definition is well-defined since H^1 functions are continuous in 1D and are pointwise defined. The interpolation i_h creates a piece-wise linear function that coincides in the support points x_j with its H^1 function.

Remark 6.176. *Since H^1 functions are not continuous any more in higher dimensions, we need more assumptions here.*

The convergence of a finite element method in 1D relies on

Lemma 6.177. *Let $i_h : H^1 \to V_h$ be given. Then:*

$$\lim_{h\to 0} \|u - i_h u\|_{H^1} = 0.$$

If $u \in H^2$, there is a constant C such that

$$\|u - i_h u\|_{H^1} \leq Ch|u|_{H^2}.$$

Proof. Since

$$\|u - i_h u\|_{H^1}^2 = \|u - i_h u\|_{L^2}^2 + |u - i_h u|_{H^1}^2,$$

the result follows immediately from the next two lemmas; namely Lemma 6.179 and Lemma 6.180. □

Definition 6.178. *The global norm $\|\cdot\|$ is obtained by locally integrating and then summing up the results on all elements. For instance, the L^2 norm is defined by*

$$\|u\|_{L^2} = \Big(\sum_{K\in T_h}\int_K |u|^2\,dx\Big)^{1/2}.$$

Lemma 6.179. *For a function $u \in H^2$, it exists a constant C (independent of h) such that*

$$\|u - i_h u\|_{L^2} \le Ch^2\|u''\|_{L^2},$$
$$|u - i_h u|_{H^1} \le Ch\|u''\|_{L^2}.$$

Proof. We choose $u \in C_c^\infty(\Omega)$ (the result will hold true for H^2 because of a density argument). The interpolant $i_h u$ is a linear function on each element $K_j = (x_j, x_{j+1})$. We calculate:

$$\begin{aligned} u(x) - i_h u(x) &= u(x) - \Big(u(x_j) + \frac{u(x_{j+1}) - u(x_j)}{x_{j+1} - x_j}(x - x_j)\Big) \\ &= u(x) - u(x_j) - \frac{x - x_j}{x_{j+1} - x_j}\Big(u(x_{j+1}) - u(x_j)\Big) \\ &= \int_{x_j}^{x} u'(t)\,dt - \frac{x - x_j}{x_{j+1} - x_j}\int_{x_j}^{x_{j+1}} u'(t)\,dt. \end{aligned}$$

Since u' is continuous, we can apply the mean value theorem from calculus (see e.g., [44][Chapter 11.3]) and further obtain:

$$\begin{aligned} u(x) - i_h u(x) &= (x - x_j)u'(x_j + \alpha) - \frac{x - x_j}{x_{j+1} - x_j}(x_{j+1} - x_j)u'(x_j + \beta) \\ &= (x - x_j)\int_{x_j+\beta}^{x_j+\alpha} u''(t)\,dt, \end{aligned}$$

with $0 \le \alpha \le x - x_j$ and $0 \le \beta \le h, h = x_{j+1} - x_j$. We now take the square and apply the Cauchy-Schwarz inequality (recall $|(u,v)|^2 \le \|u\|^2\|v\|^2$):

$$\begin{aligned} |u(x) - i_h u(x)|^2 &= \Big|(x - x_j)\int_{x_j+\beta}^{x_j+\alpha} u''(t)\,dt\Big|^2 \\ &\le \Big|(x - x_j)\int_{x_j}^{x_{j+1}} u''(t)\,dt\Big|^2 \\ &\le h^2\Big|\int_{x_j}^{x_{j+1}} u''(t)\,dt\Big|^2 \\ &\le h^2\Big(\int_{x_j}^{x_{j+1}} 1^2\,dt\Big)\Big(\int_{x_j}^{x_{j+1}} |u''(t)|^2\,dt\Big) \\ &= h^3\Big(\int_{x_j}^{x_{j+1}} |u''(t)|^2\,dt\Big). \end{aligned}$$

When we now integrate $|u(x) - i_h u(x)|^2$ on K_j, we obtain a norm-like object what we wish:

$$\int_{x_j}^{x_{j+1}} |u(x) - i_h u(x)|^2\,dx \le h^4\Big(\int_{x_j}^{x_{j+1}} |u''(t)|^2\,dt\Big).$$

Summing up over all elements $K_j = [x_j, x_{j+1}], j = 0, \dots, n$, we obtain the global norm (see Definition 6.178):

$$\sum_j \int_{x_j}^{x_{j+1}} |u(x) - i_h u(x)|^2\,dx = \|u - i_h u\|_{L^2}^2 \le h^4|u|_{H^2}^2.$$

Taking the square root yields the desired result:

$$\|u - i_h u\|_{L^2} \leq Ch^2 \|u''\|_{L^2} \quad \text{for } u \in C_c^\infty(\Omega).$$

By density the result also holds true for H^2.

The second estimate is obtained by the same calculations. Here, the initial idea is:

$$\begin{aligned} u'(x) - i_h u'(x) &= u'(x) - \frac{u(x_{j+1} - u(x_j)}{h} \\ &= \frac{1}{h} \int_{x_j}^{x_{j+1}} (u'(x) - v'(t))\, dt \\ &= \frac{1}{h} \int_{x_j}^{x_{j+1}} \int_t^x u''(s)\, ds. \end{aligned}$$

As before, taking the square root, then using Cauchy-Schwarz, then integrating over each element K_j, and finally summing over all elements K_j will yield the desired result

$$|u - i_h u|_{H^1} \leq Ch \|u''\|_{L^2}.$$

□

Lemma 6.180. *There exists a constant C (independent of h) such that for all $u \in H^1(\Omega)$, it holds*

$$\|i_h u\|_{H^1} \leq C \|u\|_{H^1}$$

and

$$\|u - i_h u\|_{L^2} \leq Ch |u|_{H^1}.$$

Moreover:

$$\lim_{n \to \infty} \|u' - i_h u'\|_{L^2} = 0.$$

Remark 6.181. *The generalization of the estimate of the interpolation operator is known as the Bramble-Hilbert lemma from 1970 (see e.g. [12]).*

Proof. The proofs use similar techniques as before. However, to get started, let $u \in H^1$. In 1D we discussed that H^1 functions are continuous. We therefore have:

$$\|i_h u\|_{L^2} \leq \max_x |i_h u(x)| \leq \max_x |u(x)| \leq C \|u\|_{H^1}.$$

Furthermore, we have

$$|i_h u|_{H^1}^2 = \int_{x_j}^{x_{j+1}} |(i_h u)'(x)|^2\, dx = \frac{\Big(u(x_{j+1}) - u(x_j)\Big)^2}{h} = \frac{1}{h}\Big(\int_{x_j}^{x_{j+1}} u'(x)\, dx\Big)^2 \leq \int_{x_j}^{x_{j+1}} |u'(x)|^2\, dx = |u|_{H^1}^2.$$

As in the other Lemma 6.179, we sum over all elements and obtain

$$\|i_h u\|_{H^1} \leq C |u|_{H^1} \leq C \|u\|_{H^1}.$$

To obtain the second result, we use again (see the other Lemma 6.179):

$$\begin{aligned} u(x) - i_h u(x) &= u(x) - \Big(u(x_j) + \frac{u(x_{j+1}) - u(x_j)}{x_{j+1} - x_j}(x - x_j)\Big) \\ &= \int_{x_j}^{x} u'(t)\, dt - \frac{x - x_j}{x_{j+1} - x_j} \int_{x_{j+1}}^{x_j} u'(t)\, dt \\ &\leq 2 \int_{x_j}^{x_{j+1}} |u'(x)|^2\, dx. \end{aligned}$$

We then take again the square, integrate, use Cauchy-Schwarz and finally sum over all elements to obtain

$$\|u - i_h u\|_{L^2} \leq Ch|u|_{H^1}.$$

To establish the third result, let $\varepsilon > 0$. Since C_c^∞ is dense in H^1, we have for all $u \in H^1$:

$$\|u' - v'\|_{L^2} \leq \varepsilon, \quad \text{for } v \in C_c^\infty.$$

With our first estimate on the interpolation error we have:

$$\|i_h u' - i_h v'\|_{L^2} \leq C\|u' - v'\|_{L^2} \leq C\varepsilon.$$

For sufficiently small h we also have

$$|v - i_h v|_{H^1} \leq \varepsilon.$$

These results can be used in our final estimate:

$$\|u' - (i_h u)'\|_{L^2} \leq \|u' - v'\|_{L^2} + \|v' - (i_h v)'\|_{L^2} + \|(i_h u)' - (i_h v)'\|_{L^2} \leq 3C\varepsilon,$$

which yields the desired result. □

With the help of the previous three lemmas, we obtain the main result:

Theorem 6.182. *Let $u \in H_0^1$ and $u_h \in V_h$ be the solutions of the continuous and discrete Poisson problems. Then, the finite element method using linear shape functions converges:*

$$\lim_{h \to 0} \|u - u_h\|_{H^1} = 0.$$

Moreover, if $u \in H^2$ (for instance when $f \in L^2$ and in higher dimensions when the domain is sufficiently smooth or polygonal and convex), we have

$$\|u - u_h\|_{H^1} \leq Ch\|u''\|_{L^2} = Ch\|f\|_{L^2}.$$

Thus the convergence in the H^1 norm (the energy norm) is linear and depends continuously on the problem data.

Proof. The first part is proven by using Lemma 6.177 applied to Lemma 6.174, which yields the first part of the assertion. The estimate is based on the Céa lemma:

$$\|u - u_h\|_{H^1} \leq C\|u - \phi_h\| \leq C\|u - i_h u\|_{H^1} \leq Ch|u|_{H^2} = O(h).$$

In the last estimate, we used again Lemma 6.177. □

Corollary 6.183. *We have*

$$\|u - u_h\|_{L^2} \leq Ch\|u''\|_{L^2} = Ch\|f\|_{L^2} = O(h).$$

Proof. Follows immediately from

$$\|u - u_h\|_{H^1} \leq Ch\|u''\|_{L^2} = Ch\|f\|_{L^2},$$

and then applying the Poincaré inequality to the left hand side term. □

Remark 6.184. *Here the L^2 estimate seems to have order h. In the next section, we see that this result can be improved.*

6.11.3 An improved L^2 estimate - the Aubin-Nitsche trick

Proposition 6.185. *Let $u \in H_0^1$ and $u_h \in V_h$ be the solutions of the continuous and discrete Poisson problems. Then:*

$$\|u - u_h\|_{L^2} \leq Ch^2 \|u''\|_{L^2} = Ch^2 \|f\|_{L^2} = O(h^2).$$

Proof. The goal is to obtain one order better than in the previous subsection. To this end, we define an auxiliary problem (the Aubin-Nitsche trick) that considers the error between the exact solution u and the discrete solution u_h: Find $w \in V$ such that

$$a(w, \phi) = \int_\Omega (u - u_h)\phi \, dx \quad \text{for all } \phi \in V.$$

We set $\phi = u - u_h \in V$, indeed since $u \in V$ and $V_h \subset V$ the variable ϕ is in V, and obtain

$$a(w, u - u_h) = \int_\Omega (u - u_h)^2 \, dx = \|u - u_h\|_{L^2}^2.$$

On the other hand we still have Galerkin orthogonality:

$$a(u - u_h, \phi_h) = 0 \quad \forall \phi_h \in V_h.$$

Herein, we take the interpolant $\phi_h := i_h w \in V_h$ and obtain

$$a(u - u_h, i_h w) = 0 \quad \forall \phi_h \in V_h.$$

Combining these two findings, we get:

$$a(w, u - u_h) = a(w - i_h w, u - u_h) = \|u - u_h\|_{L^2}^2.$$

To get a better estimate we now proceed with the continuity of the bilinear form:

$$\|u - u_h\|_{L^2}^2 = a(w - i_h w, u - u_h) \leq \gamma \|w - i_h w\|_{H^1} \|u - u_h\|_{H^1}. \tag{111}$$

The previous interpolation estimate in Theorem 6.182 can now be applied to the auxiliary variable $w \in V$:

$$\|w - i_h w\|_{H^1} \leq Ch \|w''\|_{L^2} = Ch \|u - u_h\|_{L^2}. \tag{112}$$

The last estimate holds true thanks to the auxiliary problem, which reads in strong form:

$$-w'' = u - u_h \quad \text{in } \Omega, \quad w = 0 \text{ on } \partial\Omega.$$

and in which we need again that Ω is sufficiently regular, which still holds from the original problem of course, and also that the right hand side is in L^2. But the latter one is also okay since $u \in V$ (in particular in L^2 and $u_h \in V_h$ with $V_h \subset V$.

Plugging (112) into (111) yields on the one hand:

$$\|u - u_h\|_{L^2}^2 \leq \gamma Ch \|u - u_h\|_{L^2} \|u - u_h\|_{H^1}$$

therefore

$$\|u - u_h\|_{L^2} \leq \gamma Ch \|u - u_h\|_{H^1}.$$

On the other hand, we know from Theorem 6.182 that

$$\|u - u_h\|_{H^1} \leq Ch \|f\|_{L^2}.$$

Plugging the last equation into the previous equation brings us to:

$$\|u - u_h\|_{L^2} \leq \gamma Ch \|u - u_h\|_{H^1} \leq C^2 h^2 \|f\|_{L^2} = O(h^2).$$

□

6.11.4 Analogy between finite differences and finite elements

Having established the numerical analysis, we can draw a short comparison to finite differences:

- FD: convergence = consistency + stability
- FE: convergence = interpolation + coercivity

6.11.5 2D and higher dimensions

The 2D and higher dimension results are much more involved and for the detailed proofs, we refer to the literature [12, 20, 56].

Definition 6.186 ([12], Chapter 5)**.** *Let Ω be a polygonal domain in $\mathbb{R}^2$ such that we can approximate this domain with quadrilaterals or triangles. A triangulation $\mathcal{T}_h$ with M elements $K_i, i = 1, \dots, M$ is* ***admissible*** *when*

1. $\bar{\Omega} = \bigcup_{i=1}^{M} K_i$.
2. *If $K_i \cap K_j$ results in one single point, it is a corner point of K_i and K_j.*
3. *If $K_i \cap K_j$ results in more than point, then $K_i \cap K_j$ is an edge (or face) of K_i and K_j.*

Definition 6.187 (Quasi-uniform, shape regular)**.** *A family of triangulations $\mathcal{T}_h$ is called quasi-uniform or shape regular, when there exists a $\kappa > 0$ such that each $K_i \in \mathcal{T}_h$ contains a circle with radius ρ_K with*

$$\rho_K \geq \frac{h_K}{\kappa}$$

where h_K is half diameter of the element K. The diameter of K is defined as the longest side of K.

Definition 6.188 (Uniform triangulation)**.** *A family of triangulations $\mathcal{T}_h$ is called uniform, when there exists a $\kappa > 0$ such that each $K_i \in \mathcal{T}_h$ contains a circle with radius ρ_K with*

$$\rho_K \geq \frac{h}{\kappa}$$

where $h = \max_{K \in \mathcal{T}_h} h_K$.

Remark 6.189. *Geometrically speaking, the above conditions characterize a* ***minimum angle condition*** *, which tells us that the largest angle must not approach $180°$ in order to avoid that triangles become too thin. The constant β is a measure for the smallest angle in K.*

Remark 6.190. *For adaptive mesh refinement (see Section 7.10.1) in higher dimensions, i.e., $n \geq 2$, one has to take care that the minimum angle condition is fulfilled when refining the mesh locally.*

Proposition 6.191 ([12], Chapter 6)**.** *Let $r \geq 2$ and $\mathcal{T}_h$ a quasi-uniform (i.e., shape-regular) triangulation of Ω. Then the interpolation via piecewise polynomials of degree $r - 1$ yields*

$$\|u - i_h u\|_{H^m} \leq c h^{r-m} |u|_{H^r}, \quad u \in H^r, \quad 0 \leq m \leq r.$$

Proof. See [12][Chapter 6]. Be careful, these are long and nontrivial proofs in which one learns however a lot of things. □

6.12 Numerical analysis: influence of numerical quadrature, first Strang lemma

Previously, we studied the Céa-Lemma in terms of approximation theory. Now, a second error estimate for $\tilde{u}_h$ becomes important which is a consequence of **numerical quadrature**, we denote it **error of consistence**, named **first Lemma from Strang**.

Our focus is combining both errors and give some answers about the order of convergence in the H_1-norm $||\cdot||_1$, that means

$$||u - \tilde{u}_h||_1 \leq \underbrace{||u - u_h||_1}_{\text{Céa}} + \underbrace{||u_h - \tilde{u}_h||_1}_{\text{Strang}}$$

This investigation leads to practical results in solving integrals with numerical integration. One could say which degree of quadrature formula is necessary integrating finite elements at a given degree.

This discussion covers two possible ways: First, we study interpolation by Lagrange with exact integration. The second way illustrates approximation theory for numerical quadrature formulae, the most common way solving integrals in FEM.

Therefore, we study an elliptic variational problem in two dimensions. This includes the model problem (Poisson problem). The results also hold for elliptic PDE's of second order in 1D.

Formulation 6.192. *Let $\Omega \subset \mathbb{R}^2$ be a bounded, convex, polygonal domain. The task is finding a weak solution $u \in H_0^1(\Omega)$ for an elliptic boundary value problem in $\mathbb{R}^2$ with homogeneous Dirichlet conditions*

$$u \in H_0^1(\Omega): \quad a(u,v) = (f,v)_0 \quad \text{for } v \in H_0^1(\Omega)$$

with

$$a(u,v) = \sum_{i,k=1}^{2} \int_\Omega a_{ik} \partial_k u \partial_i v \, dx \tag{113}$$

and coefficient functions $a_{ik} := a_{ik}(x)$.

We assume a regular triangulation $\mathbb{T}_h = \{T\}$ from $\overline{\Omega}$ with properties

i) $\overline{\Omega} = \bigcup_i T_i$

ii) $T_i^\circ \cap T_\nu^\circ = \varnothing\,, \quad \nu \neq i$

For more information see [12] p. 58.

6.12.1 Approximate solution of A and b

There is often no chance or an enormous computational cost solving matrix A and right-hand side b. In principle we have the reasons

i) The first case puts $a_{ij} = \delta_{ij}$ in (113), so the computation uses integration formulae for polynomials of degree $2m-2$. Exact formulae for higher m need many computations but convergence is sure for quadrature rules of lower order.

ii) In general, the coefficient functions a_{ij} are not constant. Therefore, exact computations for the elements of A_{ij} and b_i are not possible.

iii) Numerical integration is the favorite way to determine isoparametric elements.

The next ideas where introduced by Strang and generalize the Céa-Lemma. Already known is the error of approximation and now a second estimate becomes important, named *error of consistence*.

Calculations will be done on a reference element subsequently affine-transformed on each cell (triangle).

6.12.1.1 Idea, deriving an approximate solution The matrix $A = (A_{ij})_{i,j=1}^N$ and right-hand-side $b = (b_i), i = 1, \ldots, N$ leads to the linear equation system

$$Au = b$$

Instead of solving this one, an approximate linear equation system is given by

$$\tilde{A}\tilde{u} = \tilde{b} \tag{114}$$

with $\tilde{A} = (\tilde{A}_{ij})_{i,j=1}^N$ and $\tilde{b} = (\tilde{b}_i), i = 1, \ldots, N$. The solution of (114) brings us the approximate solution vector

$$\tilde{u}_h = \sum_{i=1}^N \tilde{\alpha}_i w_i \in S_h^m .$$

Now the error

$$e_h = u_h - \tilde{u}_h$$

arises. Next we give an idea for the corresponding variational formulation of (114). We work with two linear combinations of S_h^m:

$$v_h = \sum_{i=1}^N \xi_i w_i \quad \text{and} \quad w_h = \sum_{j=1}^N \eta_j w_j$$

These are necessary for the approximate bilinear form $\tilde{a}(v_h, w_h)$ and the approximate right-hand-side $\tilde{l}(v_h)$:

$$\tilde{a}(v_h, w_h) = \sum_{i,j=1}^N \tilde{A}_{ij}\xi_i\eta_j \quad \text{and} \quad \tilde{l}(v_h) = \sum_{i=1}^N \tilde{b}_i\xi_i \tag{115}$$

This leads to the variational formulation $(\tilde{V})$

$$\tilde{u}_h \in S_h^m : \quad \tilde{a}(\tilde{u}_h, v_h) = \tilde{l}(v_h) \quad \forall v_h \in S_h^m \qquad \Leftrightarrow \qquad \tilde{A}\tilde{u} = \tilde{b}$$

The following lemma describes an error estimate for $u_h - \tilde{u}_h$.

Lemma 6.193. *(first Lemma from Strang)*
The approximate bilinear form $\tilde{a}$ and the linear functional $\tilde{l}$ satisfying the condition

$$||v_h||_1^2 \leq \gamma\tilde{a}(v_h) \qquad \textit{(Uniform ellipticity of } \tilde{a}) \tag{116}$$

uniformly in $0 < h \leq h_0$ and also holds the estimate

$$|(a - \tilde{a})(u_h, v_h)| + |(f, v_h)_0 - \tilde{l}(v_h)| \leq ch^\tau \cdot ||v_h||_1 , \quad v_h \in S_h^m \tag{117}$$

Then the problems $(\tilde{V})$ have unique solutions $\tilde{u} \in S_h^m$. The quantitative error estimate of the Ritzapproximation $u_h \in S_h^m$ is given by

$$||u_h - \tilde{u}_h||_1 \leq c\gamma h^\tau$$

Proof.
The unique solvability may be proofed with the Theorem of Lax-Milgram. See [12] p. 37 respectively [57] p.41. We derive the quantitative error estimation by the following calculation. Denote $e_h = v_h = u_h - \tilde{u}_h$,

$$\begin{aligned}
\tilde{a}(e_h) &= \tilde{a}(e_h, e_h) = \tilde{a}(u_h - \tilde{u}_h, e_h) = \tilde{a}(u_h, e_h) - \tilde{a}(\tilde{u}_h, e_h) \\
&= \tilde{a}(u_h, e_h) - a(u_h, e_h) + a(u_h, e_h) - \tilde{a}(\tilde{u}_h, e_h) \\
&= (\tilde{a} - a)(u_h, e_h) + a(u_h, e_h) - \tilde{a}(\tilde{u}_h, e_h) \\
&= (\tilde{a} - a)(u_h, e_h) + (f, e_h)_0 - \tilde{l}(e_h) \\
&\leq ch^\tau ||e_h||_1
\end{aligned}$$

The last estimation follows from second assumption (117). Since we have (116), we can conclude:

$$||e_h||_1^2 \leq \gamma\tilde{a}(e_h) \leq \gamma ch^\tau ||e_h||_1$$

Division by $||e_h||_1$ shows the assertion.

□

With the aid of (115) we have a corresponding formulation for uniform ellipticity of $\tilde{a}$:

$$\tilde{a}(v_h) = \sum_{i,j} \tilde{A}_{ij}\xi_i\xi_j \geq \gamma_1|\xi|^2 \quad \forall \xi \in \mathbb{R}^n,\, \gamma_1 > 0 \tag{118}$$

Hence the coefficient matrix is uniformly definite in the variable x.

We point out two different ways for an approximate determination of A and b. In this context, we assume that the coefficient functions a_{ij} and the power vector f are sufficiently smooth,

$$a_{ij}, f \in L^\infty(E), \quad i,j = 1,2$$

Secondly, we work with an *affine family* of finite elements, so we have the polynomial space $P_m(E)$ for the reference element. Interpolation and quadrature formulae were derived on that reference element and later transformed on each cell of Ω.

6.12.2 Interpolation with exact polynomial integration

There is a triangulation $\mathbb{T}_h$. Each triangle $T \in \mathbb{T}_h$ satisfies polynomial interpolation for a_{ij} and f with

$$\tilde{a}_{ij}, \tilde{f} \in P_{r-1}(T), \quad i,j = 1,2$$

The error is determined by the remainder and we find the uniform estimation

$$||a_{ij} - \tilde{a}_{ij}||_\infty + ||f - \tilde{f}||_\infty = \mathcal{O}(h^r) \tag{119}$$

This error is independent from step width h and holds for functions $a_{ij}, f \in C^r(T)$. Higher differentiability leads not to a better error estimation. See [35] p. 194ff.

We compute the elements of $\tilde{A}$ and right-hand-side $\tilde{b}$ with

$$\tilde{A}_{ij} = \int_\Omega \sum_{\nu,\mu=1}^{2} \tilde{a}_{\nu\mu}\partial_\mu w_i\, \partial_\mu w_j\, dx, \quad i,j = 1,\dots,N\,,$$

$$\text{and} \quad \tilde{b}_i = \int_\Omega \tilde{f} w_i\, dx, \quad i = 1,\dots,N\,.$$

The advantage of these elements is exact integration. Therefore we have to evaluate the following integrals,

$$\int_T x^\beta\, dx, \quad 0 \leq |\beta| \leq 2m + r - 3$$

Proof. Sketch. We give an answer that the degree of the polynomials must be $0 \leq |\beta| \leq 2m + r - 3$. The assumption says $w_i \in S_h^m$ which means $w_i|_T \in P_m(T)$. The coefficient function $\tilde{a}_{ij}$ is a polynomial of $\in P_{r-1}(T)$. Then

$$\tilde{a}_{\nu,\mu}\partial_\nu w_i\, \partial_\mu w_j \in P_{(r-1)+2m-2}(T)$$

The proof is finished. □

An error estimation is given by Lemma (6.193). First task is to check the assumptions. Consider $v_h, w_h \in S_h^m$:

$$
\begin{aligned}
|(a-\tilde{a})(v_h,w_h)| &= |a(v_h,w_h)-\tilde{a}(v_h,w_h)| && (120)\\
&= \left| \int_G \sum_{\nu,\mu=1}^{2} a_{\nu\mu}\partial_\mu v_h\, \partial_\nu w_h\, dx - \int_G \sum_{\nu,\mu=1}^{2} \tilde{a}_{\nu\mu}\partial_\mu v_h\, \partial_\nu w_h\, dx \right| && (121)\\
&= \left| \int_G \sum_{\nu,\mu=1}^{2} (a_{\nu\mu}-\tilde{a}_{\nu\mu})\partial_\mu v_h\, \partial_\nu w_h\, dx \right| && (122)\\
&\leq c||a-\tilde{a}||_\infty \cdot \left| \int_G \sum_{\nu,\mu=1}^{2} \partial_\mu v_h\, \partial_\nu w_h\, dx \right| && (123)\\
&\overset{\text{C.S.}}{\leq} c||a-\tilde{a}||_\infty \cdot ||v_h||_1 \cdot ||w_h||_1 && (124)\\
&\overset{(119)}{\leq} ch^r||v_h||_1 \cdot ||w_h||_1 && (125)
\end{aligned}
$$

We proof the second part

$$
\begin{aligned}
|(f,v_h)-\tilde{l}(v_h)| &= \left| \int_G f\, v_h\, dx - \int_G \tilde{f}\, v_h\, dx \right| && (126)\\
&= \left| \int_G (f-\tilde{f})\, v_h\, dx \right| && (127)\\
&\leq c||f-\tilde{f}||_\infty \cdot ||v_h||_\infty && (128)\\
&\overset{(119)}{\leq} ch^r \cdot ||v_h||_1 && (129)
\end{aligned}
$$

We complement the proof showing (116),

$$
\begin{aligned}
||v_h||_1^2 &\leq c\, a(v_h) && \text{(V-ellipticity)} && (130)\\
&\leq c|a(v_h)-\tilde{a}(v_h)+\tilde{a}(v_h)| && \text{(expansion with } \tilde{a}(v_h)\text{)} && (131)\\
&\leq c|a(v_h)-\tilde{a}(v_h)|+c|\tilde{a}(v_h)| && \text{(triangle inequality)} && (132)\\
&= c|(a-\tilde{a})(v_h)|+c|\tilde{a}(v_h)| && && (133)\\
&\leq ch^r||v_h||_1^2 + c\,\tilde{a}(v_h) && \text{(set } v_h = w_h \text{ in (125))} && (134)
\end{aligned}
$$

The last estimate is a result from $\tilde{a}(v_h) > 0$. For sufficient small step size $0 < h \leq h_0$ we can conclude

$$||v_h||_1^2 \leq c\,\tilde{a}(v_h)$$

Now, Lemma (6.193) delivers the error of the approximate Ritzapproximation

$$||u_h - \tilde{u}_h||_1 \leq ch^r$$

6.12.2.1 Total error We recall the roles of the different solutions

- u : original function which is to approximate,
- the function u is approached by the solution u_h of the Ritz-procedure,
- for lower computational cost, u_h is approximated itself by $\tilde{u}_h$.

The triangle inequality shows

$$||u-\tilde{u}_h||_1 \leq ||u-u_h||_1 + ||u_h-\tilde{u}_h||_1 = \mathcal{O}(h^m + h^r)$$

The order for optimal convergence is $r \geq m$. The error of the interpolating functions can be neglected in comparison with the procedure-error if $r > m$.

6.12.3 Integration with numerical quadrature

In this context numerical integration formulae based on interpolated functions. The most common formulae is

$$f_T(g) = \int_T g\,dx \sim Q_T(g) = \sum_{i=1}^{L} \omega_i g(\xi_i) \tag{135}$$

with distinct quadrature points $\xi_i \in T$ and quadrature weights ω_i. For construction of these formulae, please visit other literature.

6.12.3.1 Affine transformation rules We define the unit triangle E as the reference element. Let $\sigma_T : E \to T$ be affine-linear mapping of E onto the triangle T. Then we have the properties

$$x = \sigma_T(\hat{x}) = B_T\hat{x} + b_T, \quad x \in T,\, \hat{x} \in E$$

with the functional matrix B_T.

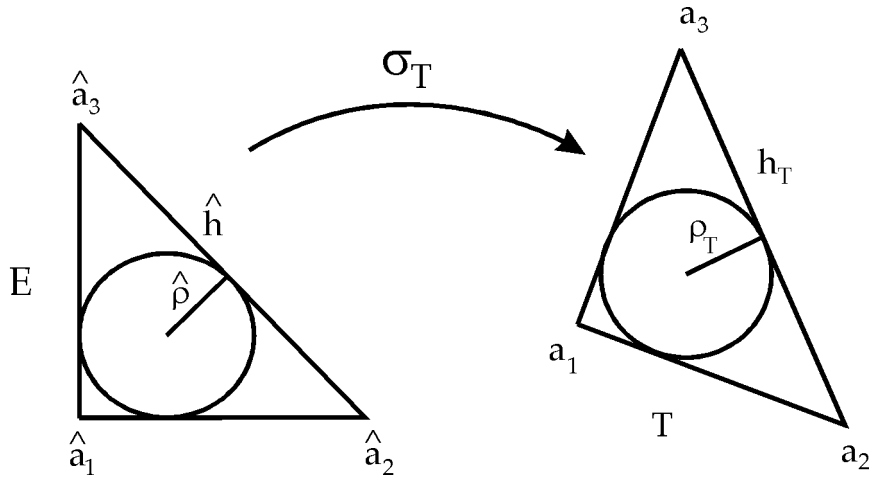

Figure 17: Reference element E and arbitrary cell (triagle) T

Example 6.194. *Let $E = conv\{\hat{a}_1, \hat{a}_2, \hat{a}_3\}$ with $\hat{a}_1 = (0,0)^T, \hat{a}_2 = (1,0)^T, \hat{a}_3 = (0,1)^T$. The triangle T has the coordinates $a_1 = (3, 1/4), a_2 = (4, -1/2), a_3 = (17/4, 1)$.*

A bijective mapping between E and a triangle T is given by $\sigma_T : E \to T$ with $\sigma_T(\hat{x}) = B_T\hat{x} + b$, where $b = a_1$. The functional matrix is defined by

$$B_T = (a_2 - a_1, a_3 - a_1) = \begin{pmatrix} 1 & \frac{5}{4} \\ -\frac{3}{4} & \frac{3}{4} \end{pmatrix}$$

The determinant of B_T is of order h^2. That means $detB_T = \mathcal{O}(h^2)$.

We obtain the relation, also known as "affine-equivalence" between two Lagrange elements,

$$\hat{v}(\hat{x}) := v(x), \quad \hat{x} = \sigma_T^{-1}x,\, x \in T$$

between functions on E and T. The polynomial space for the reference element is charakterized by $P_m(E)$. A (linear) quadrature rule on E satisfies

$$Q_E(g) = \sum_{i=1}^{L} \hat{\omega}_i g(\hat{\xi}_i), \quad \hat{\omega}_i := \int_E \hat{L}_i(\hat{x})d\hat{x},\, \hat{L}_i(\hat{x}) = \prod_{k=0}^{n} \frac{\hat{x} - x_k}{x_i - x_k}$$

with quadrature points $\hat{\xi}_i \in E$ and positive weights $\hat{\omega}_i$, $i = 1, \ldots, L$.

Affine transformation of QF on E gives quadrature rules on each element T with

$$Q_T(g) = \sum_{i=1}^{L} \omega_i g(\xi_i)$$

with quadrature points $\xi_i = \sigma_T \hat{\xi}_i$ and weights $\omega_i = |\mathrm{det} B_T| \hat{\omega}_i, i = 1, \ldots, L$. Then

$$Q_T(g) := \sum_{i=1}^{L} \omega_i g(\xi_i) = \sum_{i=1}^{L} |\mathrm{det} B_T| \hat{\omega}_i \hat{g}(\hat{\xi}_i) = |\mathrm{det} B_T| \, |Q_E(\hat{g})|$$

There is a short introduction of the very important transformation rule for integrals. It follows from the substitution rule in 1D.

Lemma 6.195. *Let T_1 and T_2 open subsets of $\mathbb{R}^n$ and $\sigma : T_1 \to T_2$. Then, and only then, the function f on T_2 is integrable if $(f \circ \sigma) \cdot |det\sigma'|$ is integrable over T_1. It is*

$$\int_{T_1} f(\sigma(x)) \cdot |det\sigma'| \, dx = \int_{T_2} f(y) \, dy$$

Proof. See [45] p. 299. It follows directly for an affine mapping $\sigma\hat{x} = B\hat{x} + b$, with derivation $D(\sigma\hat{x}) = D(B\hat{x} + b) = B$, that

$$\int_T v(x) \, dx = |\mathrm{det} B_T| \int_E \hat{v}(\hat{x}) \, d\hat{x}$$

□

The next two estimations can directly derived from the geometry of our triangulation. Remember the use of triangles in a regular triangulation. "Regular" means, that

$$\varrho_T \geq c h_T \quad \forall h, T, \, c > 0$$

where h_T charakterises the maximum diameter for any triangle, that means the longest side. In general we have the following two situations

$$||B_T||_2 \leq \frac{h_T}{\hat{\varrho}} = c_1 h_T \quad \text{and} \quad ||B_T^{-1}||_2 \leq \frac{\hat{h}}{\varrho_T} = c_0 \varrho_T^{-1} \tag{136}$$

The symbol $||B||_2$ means

$$||B||_2 = \left(\sum_{i,k} |b_{ik}|^2 \right)^{\frac{1}{2}}$$

Furthermore there is an estimation for determinants as follows

$$||B_T^{-1}||_2^{-n} \leq |\mathrm{det} B_T| \leq ||B_T||_2^n$$

which may be proofed with the Laplace expansion theorem. Then

$$c_0 \varrho_T^n \leq |\mathrm{det} B_T| \leq c_1 h_T^n$$

and

$$||B_T||_2 \, ||B_T^{-1}||_2 \leq C \frac{h}{\varrho}$$

It follows in 2D ($n = 2$)

$$C_0 h_T^2 \leq \frac{1}{||B_T^{-1}||_2^2} \leq |\mathrm{det}(B_T)| \leq ||B_T||_2^2 \leq C_1 h_T^2 \tag{137}$$

At last we mention

$$c_0 |\widehat{\nabla} \hat{v}(\hat{x})| \leq h_T |\nabla v(x)| \leq c_1 |\widehat{\nabla} \hat{v}(\hat{x})|, \quad x \in T, \hat{x} = \sigma_T^{-1} x \tag{138}$$

Proof. Sketch. Estimation (138) follows from the compound rule of three

$$\widehat{\nabla}\hat{v}(\hat{x}) = B_T^T \nabla v(x) \tag{139}$$

We have

$$|\nabla v(x)| = |\nabla v(\sigma(\hat{x}))| = |B^{-T}\widehat{\nabla}\hat{v}(\hat{x})| \leq ||B^{-T}||_2 |\widehat{\nabla}\hat{v}(\hat{x})|$$

and on the other hand

$$|\widehat{\nabla}\hat{v}(\hat{x})| = |B^T \nabla v(x)| \leq ||B^T||_2 |\nabla v(x)|$$

Working with (136) show (138). □

6.12.3.2 Numerical quadrature We set QF as an abbreviation for *quadrature formula*.

Definition 6.196. *(Order of a QF)*
A QF $Q_T(\cdot)$ on T is said to be of order r if it integrates exactly all polynomials of degree $r-1$,

$$Q_T(p) = \int_T p\,dx, \quad p \in P_{r-1}(T)$$

Another equivalent formulation is given by

$$Q_T(g) = \int_T P_T g\,dx, \quad P_T g \in P_{r-1}(T)$$

where P_T is an interpolating operator for g. Then, $Q_T(\cdot)$ is named as an interpolatary approximation of $f_T(\cdot)$.
The first result gives an estimation for the error of numerical quadrature. Therefore we need the seminorm

$$|u|_{r,1,T} = \sum_{|\alpha|=r} \int_T |\partial^\alpha u|\ dx$$

and we mention that $C^r(T)$ is dense in $H^{r,1}(T)$.

Lemma 6.197. *(Quantitative error of numerical quadrature)*
Assume $Q_T(g)$ is a QF of order $r \geq 3$ of the function $g \in H^{r,1}(T)$. Then

$$|f_T(g) - Q_T(g)| \leq ch^r |g|_{r,1,T} \quad \forall T \tag{140}$$

for constant c which is independent of T and h

Proof. The requirement $r \geq 3$ will be proofed with the inequality from Sobolev which is a consequence of the embedding theorem. Note, that the case $r = 2$ is also possible but difficult to prove. The linear functional f_T is continuous with respect to the L^1-norm,

$$|f_T(g)| = \left| \int_T g\,dx \right| \leq \int_T |g|\,dx = |g|_{0,1,T}\,, \quad g \in L^1(T) \tag{141}$$

It follows for $g \in H^{r,1}(T)$ and all T

$$\begin{aligned} |f_T(g) - Q_T(g)| &= |f_T(g) - f_T(P_T g)| \qquad \text{(interpolated approximation of } f_T\text{)} \\ &= |f_T(g - P_T g)| \\ &\overset{(141)}{\leq} |g - P_T g|_{0,1} \\ &\leq ch^r |g|_{r,1} \qquad \text{(interpolation estimation)} \end{aligned}$$

We finished proof. □

Definition 6.198. *(Admissibility of a QF)*
Each polynomial $q \in P_{m-1}(E)$, integrated by Q_E, has the property

$$\text{For} \quad q(\hat{\xi}_i) = 0,\ i = 1, \dots, L \quad \text{follows} \quad q \equiv 0 \tag{142}$$

This property ensures that the Lagrange interpolation has a unique solution. The necessary condition for (142) requires that the number of interpolation points L is greater or equal to $\dim P_{m-1}$.

We need further results like a new norm, and after that definition, we give an estimation between QF and integral.

Definition 6.199. *(Norm on $P_{m-1}(E)/P_0(E)$)*
On the finite dimensional quotient space $P_{m-1}(E)/P_0(E)$ is a norm defined by

$$|||\hat{q}||| := \left(\sum_{i=1}^{L} \hat{\omega}_i \sum_{j=1}^{2} (\partial_j q)^2(\hat{\xi}_i) \right)^{\frac{1}{2}}$$

Proof. We have to proof all attributes of a norm. But the friendly reader wants to show three properties. We only check,

Definiteness.

Property (142) holds. Since the quadrature weights $\hat{\omega}_i$ are positive it follows for $q \in P_{m-1}(E)$,

$$\begin{aligned} &\sum_{i=1}^{L} \hat{\omega}_i \sum_{j=1}^{2} (\partial_j q)^2(\hat{\xi}_i) = 0 \\ \Rightarrow \quad &\partial_j q(\hat{\xi}_i) = 0, \quad 1 \le j \le 2, \quad 1 \le i \le L \end{aligned}$$

Remark that any $\partial_j q$ is an element of $P_{m-2}(E)$ because $q \in P_{m-1}(E)$. Since (142), $\partial_j q$ vanishes at $\hat{\xi}_i$. This implies $\partial_j q \equiv 0$, $j = 1, 2$ which shows the definiteness.

The reader may check

$$||q|| \ge 0, \quad ||\alpha q|| = |\alpha|\,||q||,\ \alpha \in \mathbb{R}, \quad ||p + q|| \le ||p|| + ||q||$$

□

Lemma 6.200. *(Ellipticity of QF)*
For some constant c which is independent of h we have

$$Q_T(|\nabla v|^2) \ge c \int_T |\nabla v|^2 dx \quad \forall v \in S_h^{m-1} \quad \forall T \tag{143}$$

Proof. There is some work to do. First we have the norm of definition (6.199). A second norm on $P_{m-1}(E)/P_0(E)$ is given by $|\hat{q}|_{1,E}$. The finite dimension of $P_{m-1}(E)/P_0(E)$ leads to the equivalence of the two norms. It exists some constant $\hat{C}$, that

$$\hat{C}|\hat{q}|_1^2 \le |||\hat{q}|||, \quad \hat{q} \in P_{m-1}(E)/P_0(E) \tag{144}$$

We compute

$$\begin{aligned} Q_T(|\nabla p|^2) &\overset{(135)}{=} \sum_{i=1}^{L} \omega_i |\nabla p(\xi_i)|^2 \\ &= |\det B_T| \sum_{i=1}^{L} \widehat{\omega}_i |\nabla p(\xi_i)|^2 \quad (\text{with } \omega_i = |\det B_T| \hat{\omega}_i) \\ &\overset{(138)}{\ge} c_0 |\det B_T| \frac{1}{h^2} \sum_{i=1}^{L} \hat{\omega}_i |\widehat{\nabla} \hat{q}(\hat{\xi}_i)|^2 \\ &\ge c_0' \cdot Q_E(|\widehat{\nabla} \hat{q}|^2) \\ &= c_0' |||\hat{q}|||, \quad \hat{q} \in P_{m-1}(E)/P_0(E) \end{aligned}$$

We conclude with the aid of equivalence of the two norms (144)

$$\begin{aligned}
Q_T(|\nabla p|^2) &\geq c_0'|||\hat{q}|||^{(144)}_{\geq} c_0'\hat{C}|\hat{q}|^2_{1,E} \\
&= c_0|\mathrm{det}B_T|h^{-2}|\hat{q}|^2_{1,E} = c_0|\mathrm{det}B_T|h^{-2}\int_E |\widehat{\nabla}\hat{q}|^2\,d\hat{x} \\
&= c_0|\mathrm{det}B_T|h^{-2}\int_E |B_T^T\,\nabla q|^2\,d\hat{x} \\
&\geq c_0h^{-2}\int_T ||B_T^T||^2|\nabla q|^2\,dx \\
&\overset{(137)}{\geq} c_0h^{-2}\,C_1h^2\int_T |\nabla q|^2\,dx \\
&= C\int_T |\nabla p|^2\,dx
\end{aligned}$$

for $p \in S_h^{m-1}$. Which completes the proof. □

6.12.3.3 Error estimation The central theorem leads to an error estimation of the FEM for numerical quadrature formulae.

We give the definitions of the approximate elements of the matrix $\tilde{A}_{ij}$ and the right-hand side $\tilde{b}_i$

$$\tilde{A}_{ij} = \sum_T Q_T\left(\sum_{\nu,\mu=1}^{2} a_{\mu\nu}\partial_\mu w_i\,\partial_\nu w_j\right), \quad i,j = 1,\ldots,N$$

and

$$\tilde{b}_i = \sum_T Q_T(fw_i), \quad i = 1,\ldots,N$$

The main Theorem is a consequence of the Lemma from Strang (6.193). First, we proof the conditions, then we have the error estimation for $u_h - \tilde{u}_h$.

Theorem 6.201. *Assume the quadrature formula, of order $r \geq 3$,*

$$Q_E(g) = \sum_{i=1}^{L} \hat{\omega}_i g(\hat{\xi}_i)$$

on E, which is admissible for $P_{m-1}(E)$. Let $r \geq m-1$, $m-2$. The degree of the finite elements is $m-1$ that is $u_h, v_h \in S_h^{m-1}$. The coefficient functions $a_{\nu\mu} \in L^\infty(E)$ are sufficiently smooth. Then we have

$$|(a-\tilde{a})(u_h,v_h)| \leq Ch^{r-m+2}||v_h||_1, \quad v_h \in S_h^{m-1}$$

and

$$|(f,v_h)_0 - \tilde{l}(v_h)| \leq Ch^{r-m+2}||v_h||_1, \quad v_h \in S_h^{m-1}$$

and also uniform ellipticity

$$c||v_h||_1^2 \leq \tilde{a}(v_h), \quad v_h \in S_h^{m-1}$$

The constants c, C depending on u.

Proof. Set $m' := m - 1$. Denote $v_h \in S_h^{m'}$:

$$\begin{aligned}
|(a-\tilde{a})(u_h, v_h)| &= |a(u_h, v_h) - \tilde{a}(u_h, v_h)| \\
&= \left| \sum_T \left[\int_T \sum_{\nu,\mu=1}^{2} a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \, dx - Q_T \left(\sum_{\nu,\mu=1}^{2} a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \right) \right] \right| \\
&= \left| \sum_T \sum_{\nu,\mu=1}^{2} \left[\int_T a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \, dx - Q_T \left(a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \right) \right] \right| \\
&\overset{\text{TRI}}{\leq} \sum_T \left| \sum_{\nu,\mu=1}^{2} \left[\int_T a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \, dx - Q_T \left(a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \right) \right] \right| \\
&\overset{(140)}{\leq} ch^r \sum_T \left| \sum_{\nu,\mu=1}^{2} a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \right|_{r,1,T}
\end{aligned}$$

The functions $a_{\nu\mu}$ are bounded with $||a_{\nu\mu}||_\infty \leq c$ which implies

$$|(a-\tilde{a})(u_h, v_h)| \overset{(140)}{\leq} ch^r \sum_T \left| \sum_{\nu,\mu=1}^{2} a_{\nu\mu} \partial_\mu u_h \, \partial_\nu v_h \right|_{r,1,T} \tag{145}$$

$$\overset{\text{C.S.}}{\leq} ch^r \sum_T \left(\sum_{\mu=1}^{2} |\partial_\mu u_h|_{r,1,T}^2 \right)^{\frac{1}{2}} \cdot \left(\sum_{\nu=1}^{2} |\partial_\nu v_h|_{r,1,T}^2 \right)^{\frac{1}{2}} \tag{146}$$

$$\leq ch^r \sum_T ||u_h||_{r+1,T} \cdot ||v_h||_{r+1,T} \tag{147}$$

Now, a function $w_h \in S_h^{m'}$ on each triangle $T \in \mathbb{T}_h$ is a polynomial of $P_{m'}(T)$. Because of differentiability, we compute

$$||w_h||_{r+1,T} = ||w_h||_{m,T} \quad \text{for } r \geq m' \tag{148}$$

The inverse estimation delivers

$$||w_h||_{m',T} \leq ch^{1-m'} ||w_h||_{1,T} \tag{149}$$

Using (148) and (149) for the function v_h show us

$$||v_h||_{r+1,T} \leq ch^{1-m'} ||v_h||_{1,T} \tag{150}$$

We obtain with (148) for u_h

$$||u_h||_{r+1,T} = ||u_h||_{m',T} \tag{151}$$

We put the last two results and the Cauchy-Schwarz inequality in (147),

$$\begin{aligned}
|(a-\tilde{a})(u_h, v_h)| &\leq ch^r \sum_T ||u_h||_{r+1,T} \cdot ||v_h||_{r+1,T} \\
&\overset{\text{C.S.}}{\leq} ch^r \left(\sum_T ||v_h||_{r+1,T}^2 \right)^{\frac{1}{2}} \cdot \left(\sum_T ||u_h||_{r+1,T}^2 \right)^{\frac{1}{2}} \\
&\overset{(150),(151)}{\leq} ch^{r-m'+1} ||v_h||_1 \cdot \left(\sum_T ||u_h||_{m',T}^2 \right)^{\frac{1}{2}} \\
&= ch^{r-m'+1} ||v_h||_1 \cdot ||u_h||_{m'}
\end{aligned}$$

We specify the last term with the Ritzapproximation u_h. Taking an interpolation operator $I_h u \in S_h^{m'}$ for u we conclude

$$\begin{aligned}
||u_h||_{m',T} &= ||u_h - I_h u + I_h u - u + u||_{m',T} \\
&\leq ||u_h - I_h u||_{m',T} + ||I_h u - u||_{m',T} + ||u||_{m',T} \\
&\overset{(149)}{\leq} ch^{1-m'}||u_h - I_h u||_{1,T} + ||I_h u - u||_{m',T} + ||u||_{m',T} \\
&= ch^{1-m'}||u_h - u + u - I_h u||_{1,T} + ||I_h u - u||_{m',T} + ||u||_{m,T} \\
&\leq ch^{1-m'}||u_h - u||_{1,T} + ch^{1-m'}||u - I_h u||_{1,T} + ||I_h u - u||_{m',T} + ||u||_{m',T}
\end{aligned}$$

Remember the result from approximation with I_h:

$$||u - I_h u||_{m'} \leq ch^{t-m'}|u|_t \leq ch^{t-m'}||u||_t \quad \text{for } u \in H^t(\Omega),\ 0 \leq m' \leq t.$$

Then, we have

$$\begin{aligned}
||u_h||_{m',T} &\leq ch^{1-m'}||u_h - u||_{1,T} + c_1||u||_{m',T} + c_2||u||_{m',T} + ||u||_{m',T} \\
&\leq ch^{1-m'}||u_h - u||_{1,T} + C||u||_{m',T}
\end{aligned}$$

The extension on Ω is

$$||u_h||_{m'} = \left(\sum_T ||u_h||^2_{m',T}\right)^{\frac{1}{2}} \leq ch^{1-m'}||u - u_h||_1 + c||u||_{m'}$$

Since $||u - u_h||_1 = \mathcal{O}(h^{m'})$, gives

$$\begin{aligned}
|(a - \tilde{a})(u_h, v_h)| &\leq ch^{r-m'+1}||v_h||_1 \cdot ||u_h||_{m'} \\
&\leq ch^{r-m'+1}||v_h||_1 \left(ch^{1-m'}||u - u_h||_1 + c||u||_{m'}\right) \\
&= ch^{r-2m'+2}||u - u_h||_1||v_h||_1 + ch^{r-m'+1}||v_h||_1 \cdot ||u||_{m'} \\
&= ch^{r-m'+2}||v_h||_1 + c(u)h^{r-m'+1}||v_h||_1 \\
&\leq c(u)h^{r-m'+1}||v_h||_1
\end{aligned}$$

We explain in words: The third line uses the order of convergence $||u - u_h|| = \mathcal{O}(h^{m'})$. The number $||u||_{m'}$ is not important for convergence, so it is some constant factor $c(u)$. The order of convergence of $h^{r-m'+1}$ is not as good as $h^{r-m'+2}$ which completes the last line. Remark that $r \geq 3$ and $r \geq m'$. Short summary of the result ($m' = m - 1$):

$$|(a - \tilde{a})(u_h, v_h)| \leq ch^{r-m+2}||v_h||_1, \quad c = c(u)$$

Analogous computation verifies

$$|(f, v_h)_0 - \tilde{l}(v_h)| \leq ch^{r-m+2}||v_h||_1$$

The last estimation shows uniform ellipticity of $\tilde{a}$. Let $v_h \in S_h^{m'}$ which implies $v_h|_T \in P_{m'}(T)$. Also, numerical stability requires positive weights and we refrain that the bilinear form a is elliptic. Then

$$\begin{aligned}
\tilde{a}(v_h) &= \sum_T Q_T \left(\sum_{\nu,\mu=1}^{2} a_{\nu\mu}\partial_\mu v_h\, \partial_\nu v_h\right) \\
&\overset{(115),(118)}{\geq} c\sum_T Q_T(|\nabla v_h|^2) \qquad \text{(uniform ellipticity of } a\text{)} \\
&\overset{(143)}{\geq} c\sum_T \int_T |\nabla v_h|^2\, dx \\
&= c|v_h|_1^2 \\
&\geq c||v_h||_1^2 \qquad \text{(Poincaré for functions of } H_0^1(\Omega)\text{)}
\end{aligned}$$

The last estimation follows from the equivalence of $||\cdot||_1$ and $|\cdot|_1$ on H_0^1.

The conditions of Lemma (6.193) are true and the proof is finished.

□

The previous theorem yields the order convergence of the FEM for numerical integration.

Corollary 6.202. *The conditions of Theorem* (6.201) *hold. Then, Lemma* (6.193) *leads to a quantitative error estimation, with* $\tau = r - m + 2$*,*

$$||u_h - \tilde{u}_h||_1 = \mathcal{O}(h^{r-m+2})$$

The total error is given by

$$||u - \tilde{u}_h||_1 \leq ||u - u_h||_1 + ||u_h - \tilde{u}_h||_1 = \mathcal{O}(h^{m-1} + h^{r-m+2})$$

Convergence is assured if $r \geq m - 1$*. In other words*

$$r' \geq m' \tag{152}$$

Remark that r' *denotes the degree of the quadrature rule and* m' *the degree of the finite elements.*

Example 6.203. *For* $r = m - 1$ *we have,*

$$||u - \tilde{u}_h||_1 = \mathcal{O}(h)$$

Optimal convergence is given if $r \geq 2m - 3$*. In example* $r = 2m - 3$*:*

$$||u - \tilde{u}_h||_1 = \mathcal{O}(h^{m-1})$$

The error of quadrature can be neglected if $r > 2m - 3$*, because*

$$||u - \tilde{u}_h||_1 = \mathcal{O}(h^{m-1})$$

is more important for the order of convergence.

Remark 6.204. *The inequality* (152) *is the main assertion of the report. But remind that* $r' >> 2m' - 1$ *(degree of quadrature rule very large in comparison to the optimal order of convergence) is not recommended because the calculation of the coefficient functions* $a_{ik}(x)$ *and the gradients* $\partial_\mu w_i \, \partial_\nu w_j, i, j = 1, \ldots N$*,* $\mu, \nu = 1, 2$ *could be very expensive for higher order quadrature rules.*

6.12.3.4 Concluding remarks The central result is the Lemma of Strang (6.193) which gives a quantitative error estimation for convergence and leads to Theorem (6.201).

The main condition is the uniform ellipticity of the approximate bi linear form $\tilde{a}$.

If the conditions are not satisfied it may be possible that the linear equation system $Au = b$ is not solvable because matrix A is singular. We study one example for this situation.

6.12.4 Numerical tests

6.12.4.1 Example 1 We discuss one simple example for a situation where the system matrix will be singular. Since $r' \geq m$ we have convergence of the procedure. Where r' means the degree of the quadrature rule and m is the degree of the FEM polynomials.

We assume the one-dimensional Poisson problem, take quadratic finite elements and use the middle-point rule as integration formula. So we solve

$$-u'' = f, \quad u(0) = (1) = 0$$

and take quadratic basic functions in the interval $[0, 1]$ with equidistant step size $h = x_i - x_{i-1}$. There is an additional point required, so we take the middle point of the cells T_i,

$$x_{i-1/2} := \frac{1}{2}(x_{i-1} - x_i)$$

The construction of the quadratic function v satisfies

$$v(x) = \sum_{i=0}^{N} v_i \psi_i(x) + \sum_{i=1}^{N} v_{i-1/2} \psi_{i-1/2}(x)$$

with the following properties

i) $\psi_i(x_k) = \delta_{ik}, \quad \psi_i(x_{k-1/2}) = 0$

ii) $\psi_{i-1/2}(x_k) = 0, \quad \psi_{i-1/2}(x_{k-1/2}) = \delta_{ik}$

Now, we get three quadratic basic functions

$$\psi_i(x) = \begin{cases} \frac{(x-x_{i-1})(x-x_{i-1/2})}{(x_i-x_{i-1})(x_i-x_{i-1/2})}, & x \in T_i \\ \frac{(x-x_{i+1})(x-x_{i+1/2})}{(x_i-x_{i+1})(x_i-x_{i+1/2})}, & x \in T_{i+1} \\ 0, & \text{otherwise} \end{cases}$$

and

$$\psi_{i-1/2}(x) = \begin{cases} \frac{(x-x_{i-1})(x-x_i)}{(x_{i-1/2}-x_i)(x_{i-1/2}-x_{i+1})}, & x \in T_i \\ 0, & \text{otherwise} \end{cases}$$

Next task is constructing the derivatives and give the variational formulation of the one-dimensional Laplace problem. Then solving integrals with middle-point rule,

$$\int_a^b f(x)\,dx \approx (b-a)\, f\left(\frac{a+b}{2}\right) + \frac{1}{24} f''(\xi)(b-a)^3 \tag{153}$$

One know that the middle-point rule has degree 1. Using this quadrature formula for quadratic elements fails and the resulting systemmatrix A will have some zeros, so it is singular. For a local element we obtain

$$\begin{aligned} \partial_x \psi_{i-1}(x) &= \frac{2}{h_i^2}(2x - x_{i-1/2} - x_i) \\ \partial_x \psi_{i-1/2}(x) &= \frac{2}{h_i^2}(x_i + x_{i-1} - 2x) \\ \partial_x \psi_i(x) &= \frac{2}{h_i^2}(2x - x_{i-1/2} - x_{i-1}) \end{aligned}$$

and the following integrals

$$\begin{aligned} A_{11} &= \int_{x_{i-1}}^{x_i} (\partial_x \psi_{i-1})^2\,dx \\ A_{22} &= \int_{x_{i-1}}^{x_i} (\partial_x \psi_{i-1/2})^2\,dx \approx h_i (\partial_x \psi_{i-1/2}(x_{i-1/2}))^2 = 0 \\ A_{33} &= \int_{x_{i-1}}^{x_i} (\partial_x \psi_i)^2\,dx \\ A_{12} = A_{21} &= \int_{x_{i-1}}^{x_i} \partial_x \psi_{i-1} \cdot \partial_x \psi_{i-1/2}\,dx \approx h_i \left[(\partial_x \psi_{i-1} \cdot \partial_x \psi_{i-1/2})(x_{i-1/2})\right] = 0 \\ A_{13} = A_{31} &= \int_{x_{i-1}}^{x_i} \partial_x \psi_{i-1} \cdot \partial_x \psi_i\,dx \\ A_{23} = A_{32} &= \int_{x_{i-1}}^{x_i} \partial_x \psi_{i-1/2} \cdot \partial_x \psi_i\,dx \approx h_i \left[(\partial_x \psi_{i-1/2} \cdot \partial_x \psi_i)(x_{i-1/2})\right] = 0 \end{aligned}$$

At last we write the results in our matrix,

$$A = \begin{pmatrix} A_{11} & 0 & A_{13} \\ 0 & 0 & 0 \\ A_{31} & 0 & A_{33} \end{pmatrix} \tag{154}$$

6.12.4.2 Example 2 The second example handles also with the Laplace equation but in 2D implemented again in deal.II [4]. Our focuss is on two lines in this program:

```
// m' : degree of the finite elements
fe (m'),
```

and

```
// Gauss formula, where n gives the number of notes in each direction
QGauss<2> quadrature_formula(n)
```

The degree of Gauss quadrature is given by $r' = 2n - 1$. We made some calculations for different m' and r'. The results are

n	1	2	3	4
r'	1	3	5	7
m'				
1	OK	OK	OK	OK
2	F	OK	OK	OK
3	F	ok	OK	OK
4	F	F	ok	OK
5	F	F	ok	ok
6	F	F	F	ok
7	F	F	F	ok
8	F	F	F	F

We shortly explain the notation:

- ok: (normal) convergence takes place
- OK: Optimal convergence is given with $r' \geq 2m - 1$
- F: no convergence

6.13 Numerical tests and computational convergence analysis

We finish this long chapter with several numerical tests in 2D and 3D.

6.13.1 2D Poisson

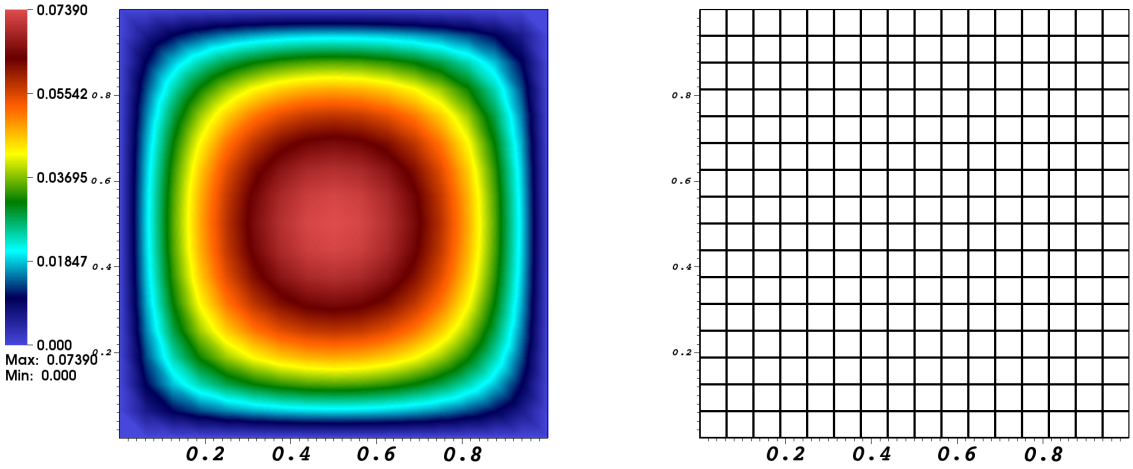

Figure 18: The graphical solution (left) and the corresponding mesh (right).

We compute Poisson in 2D on $\Omega = (0,1)^2$ and homogeneous Dirichlet conditions on $\partial\Omega$. The force is $f = 1$. We use a quadrilateral mesh using Q_1 elements (in an isoparametric FEM framework). The number of mesh elements is 256 and the number of DoFs is 289. We need 26 CG iterations (see Section 8.4 for the development of the CG scheme), without preconditioning, for the linear solver to converge.

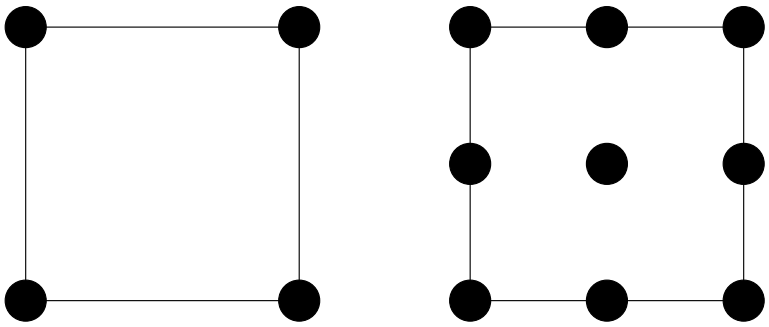

Figure 19: Q_1 (bilinear) and Q_2 (biquadratic) elements on a quadrilateral in 2D.

6.13.2 Numerical test: 3D

```
Number of elements:   4096
Number of DoFs:       4913
Number of iterations: 25 CG steps without preconditioning.
```

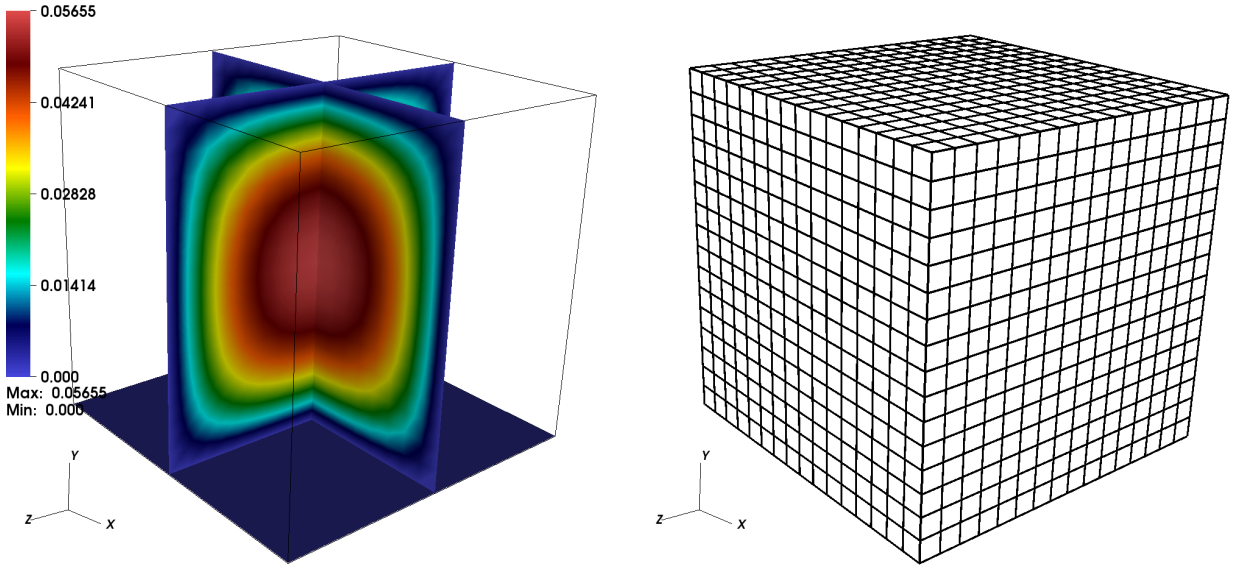

Figure 20: The graphical solution (left) and the corresponding mesh (right).

6.13.3 Checking programming code and convergence analysis for linear and quadratic FEM

We present a general algorithm and present afterwards 2D results.

Algorithm 6.205. *Given a PDE problem. E.g.* $-\Delta u = f$ *in* Ω *and* $u = 0$ *on the boundary* $\partial\Omega$.

1. *Construct by hand a solution* u *that fulfills the boundary conditions.*
2. *Insert* u *into the PDE to determine* f.
3. *Use that* f *in the finite element simulation to compute numerically* u_h.
4. *Compare* $u - u_h$ *in relevant norms (e.g.,* L^2, H^1*).*
5. *Check whether the desired* h *powers can be obtained for small* h.

We demonstrate the previous algorithm for a $2D$ case in $\Omega = (0, \pi)^2$:

$$\begin{aligned} -\Delta u(x, y) &= f \quad \text{in } \Omega, \\ u(x, y) &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

and we constuct $u(x, y) = \sin(x)\sin(y)$, which fulfills the boundary conditions (trivial to check! But please do it!). Next, we compute the right hand side f:

$$-\Delta u = -(\partial_{xx}u + \partial_{yy}u) = 2sin(x)sin(y) = f(x, y).$$

6.13.3.1 2D Poisson: Linear FEM We then use f in the above program from Section 6.13.1 and evaluate the L^2 and H^1 norms using linear FEM. The results are:

Level	Elements	DoFs	h	L2 err	H1 err
2	16	25	1.11072	0.0955104	0.510388
3	64	81	0.55536	0.0238811	0.252645
4	256	289	0.27768	0.00597095	0.126015
5	1024	1089	0.13884	0.00149279	0.0629697
6	4096	4225	0.06942	0.0003732	0.0314801
7	16384	16641	0.03471	9.33001e-05	0.0157395
8	65536	66049	0.017355	2.3325e-05	0.00786965
9	262144	263169	0.00867751	5.83126e-06	0.00393482
10	1048576	1050625	0.00433875	1.45782e-06	0.00196741
11	4194304	4198401	0.00216938	3.64448e-07	0.000983703

In this table we observe that we have quadratic convergence in the L^2 norm and linear convergence in the H^1 norm. For a precise (heuristic) computation, we also refer to Chapter 12 in which a formula for computing the convergence orders α is derived.

6.13.3.2 2D Poisson: Quadratic FEM We use again f in the above program from Section 6.13.1 and evaluate the L^2 and H^1 norms using quadratic FEM. The results are:

Level	Elements	DoFs	h	L2 err	H1 err
2	16	81	1.11072	0.00505661	0.0511714
3	64	289	0.55536	0.000643595	0.0127748
4	256	1089	0.27768	8.07932e-05	0.00319225
5	1024	4225	0.13884	1.01098e-05	0.000797969
6	4096	16641	0.06942	1.26405e-06	0.000199486
7	16384	66049	0.03471	1.58017e-07	4.98712e-05
8	65536	263169	0.017355	1.97524e-08	1.24678e-05
9	262144	1050625	0.00867751	2.46907e-09	3.11694e-06
10	1048576	4198401	0.00433875	3.08687e-10	7.79235e-07
11	4194304	16785409	0.00216938	6.14696e-11	1.94809e-07

In this table we observe that we have cubic convergence $O(h^3)$ in the L^2 norm and quadratic convergence $O(h^2)$ in the H^1 norm. This confirms the theory; see for instance [12]. For a precise (heuristic) computation, we also refer to Chapter 12 in which a formula for computing the convergence orders α is derived.

We next plot the DoFs versus the errors in Figure 21 in order to highlight the convergence orders. For the relationship between h and DoFs versus the errors in different dimensions, we refer again to Chapter 12.

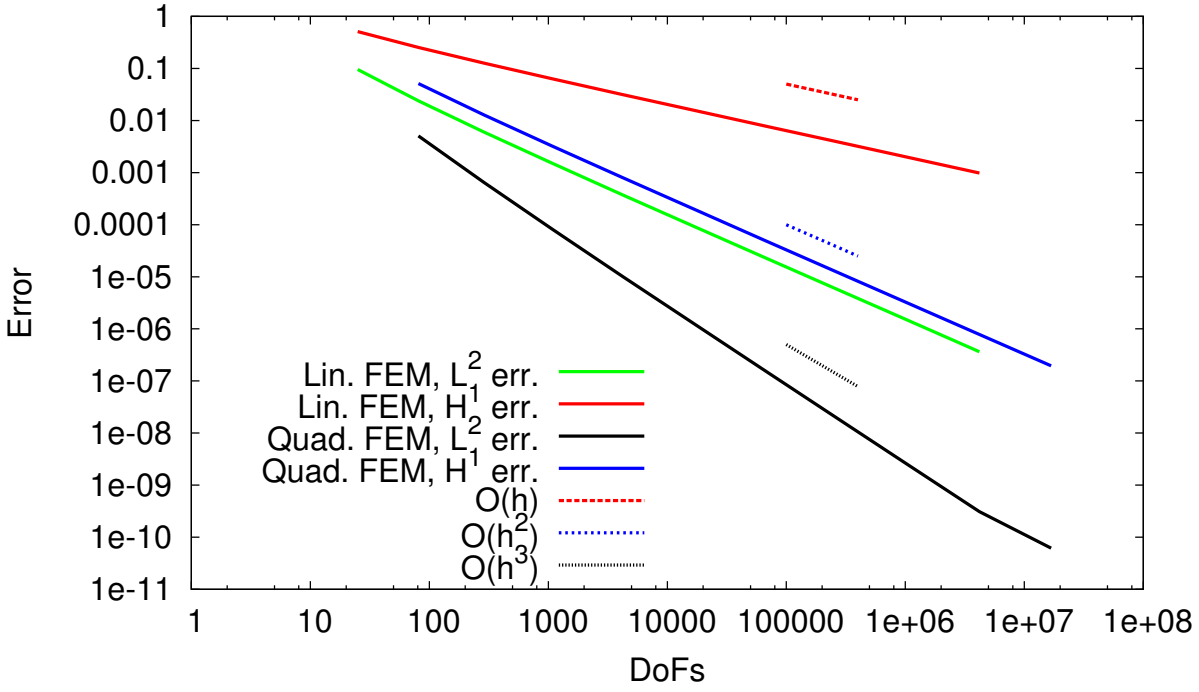

Figure 21: Plotting the DoFs versus the various errors for the 2D Poisson test using linear and quadratic FEM. We confirm numerically the theory: we observe $\|u-u_h\|_{L^2} = O(h^{r+1})$ and $\|u-u_h\|_{H^1} = O(h^r)$ for $r = 1, 2$, where r is the FEM degree.

6.13.4 Convergence analysis for 1D Poisson using linear FEM

We continue Section 6.3.8. As manufactured solution we use

$$u(x) = \frac{1}{2}(-x^2 + x),$$

on $\Omega = (0, 1)$ with $u(0) = u(1) = 0$, which we derived in Section 6.1.1. In the middle point, we have $u(0.5) = -0.125$ (in theory and simulations). In the following table, we plot the L^2 and H^1 error norms, i.e., $\|u - u_h\|_X$ with $X = L^2$ and $X = H^1$, respectively.

Level	Elements	DoFs	h	L2 err	H1 err
1	2	3	0.5	0.0228218	0.146131
2	4	5	0.25	0.00570544	0.072394
3	8	9	0.125	0.00142636	0.0361126
4	16	17	0.0625	0.00035659	0.0180457
5	32	33	0.03125	8.91476e-05	0.00902154
6	64	65	0.015625	2.22869e-05	0.0045106
7	128	129	0.0078125	5.57172e-06	0.00225528
8	256	257	0.00390625	1.39293e-06	0.00112764
9	512	513	0.00195312	3.48233e-07	0.000563819
10	1024	1025	0.000976562	8.70582e-08	0.000281909

Using the formulae from Chapter 12, we compute the convergence order. Setting for instance for the L^2 error we have

$$P(h) = 1.39293e-06$$
$$P(h/2) = 3.48233e-07$$
$$P(h/4) = 8.70582e-08.$$

Then:

$$\alpha = \frac{1}{log(2)} log\Big(\Big|\frac{1.39293e-06 - 3.48233e-07}{3.48233e-07 - 8.70582e-08}\Big|\Big) = 1.99999696186957,$$

which is optimal convergence that confirm the a priori error estimates from Section 6.11.3. The octave code is:

```
alpha = 1 / log(2) * log(abs(1.39293e-06 - 3.48233e-07) / abs(3.48233e-07 - 8.70582e-08))
```

For the H^1 convergence order we obtain:

```
alpha = 1 / log(2) * log(abs(0.00112764 - 0.000563819) / abs(0.000563819 - 0.000281909))
      = 1.00000255878430,
```

which is again optimal convergence; we refer to Section 6.11.2 for the theory.

6.13.5 Computing the error norms $\|u - u_h\|$

To evaluate the global error norms $\|u - u_h\|_{L^2(\Omega)}$ and $\|u - u_h\|_{H^1(\Omega)}$, we proceed element-wise and define:

Definition 6.206 (See for example [12](Chapter 2, Section 6)). *For a decomposition $\mathcal{T}_h = \{K_1, \dots, K_M\}$ of the domain Ω and $m \geq 1$ we define*

$$\|u\|_{H^m(\Omega)} := \sqrt{\sum_{K_j} \|u\|^2_{H^m(K_j)}}.$$

Therefore we compute element-wise the error contributions:

$$\|u\|^2_{H^m(K_j)} = \int_{K_j} \sum_{|\alpha| \leq m} \|\partial^\alpha u\|^2_{L^2(K_j)},$$

where α is a multiindex representing the degree of the derivatives; see Section 3.5.

Example 6.207 (L^2 and H^1). *For L^2 it holds:*

$$\|u\|^2_{L^2(K_j)} = \int_{K_j} u^2 \, dx$$

For H^1 it holds:

$$\|u\|^2_{H^1(K_j)} = \int_{K_j} (u^2 + \nabla u^2) \, dx$$

Consequently for the errors:

Example 6.208 (Errors in L^2 and H^1). *For L^2 it holds:*

$$\|u - u_h\|^2_{L^2(K_j)} = \int_{K_j} (u - u_h)^2 \, dx$$

For H^1 it holds:

$$\|u - u_h\|^2_{H^1(K_j)} = \int_{K_j} ((u - u_h)^2 + (\nabla u - \nabla u_h)^2) \, dx,$$

which allow directly the evaluation of the integrals, or better, since u_h only lives in the discrete nodes (or quadrature points), we use directly a quadrature rule.

Then we obtain:

Proposition 6.209. *It holds:*

$$\|u-u_h\|_{L^2(\Omega)} = \sqrt{\sum_{K_j} \|u-u_h\|_{L^2(K_j)}},$$
$$\|u-u_h\|_{H^1(\Omega)} = \sqrt{\sum_{K_j} \|u-u_h\|_{H^1(K_j)}}.$$

6.14 Chapter summary and outlook

In this key chapter, we have introduced the theoretical background of finite elements, namely variational formulations. We discussed the properties and characteristic features of the finite element method in one dimension and higher dimensions. The discretization is based on uniform meshes (grids). For large-scaled problems and practical applications, uniform meshes are however unfeasible. In the next chapter, we introduce a posteriori error estimation that can be used to extract localized error indicators, which are then used for adaptive mesh refinement such that only 'important' parts of the numerical solution are resolved with a desired accuracy. Less important solution parts can be still treated with relatively coarse meshes. In Chapter 8, we discuss some numerical solution techniques such as the CG scheme, which has already been employed in the current chapter.

7 A posteriori error estimation and mesh adaptivity

So far, we have discretized PDE problems without paying a lot of attention on the efficiency and numerical cost. Often, only a part of the numerical solution u or the evaluation of a functional of interest that contains parts of the solution or its derivative(s) is of interest. In these cases, we need a high accuracy of the numerical solution, but uniform (i.e., global) mesh refinement is very expensive for large problems and three-dimensional computations. Here, memory requirements and CPU wall clock simulation times are too high to be justified. Mesh adaptivity tries to refine only a few mesh elements such that the envisaged discretization error reduces below a given tolerance. Simultaneously, mesh regions out of interest for the total accuracy may be coarsened.

7.1 Principles of error estimation

In general, we distinguish between **a priori** and **a posteriori error estimation**. In the first one, which we already have discussed for finite differences and finite elements, the (discretization) error is estimated before we start a numerical simulation/computation. Often, there are however constants c and (unknown) higher-order derivatives $|u|_{H^m}$ of the (unknown) solution:

$$\|u - u_h\| \leq ch^m |u|_{H^m}.$$

Such estimates yield

- the asymptotic information of the error for $h \to 0$.
- In particular, they provide the expected order of convergence, which can be adopted to verify programming codes and the findings of numerical simulations for (simple?!) model problems.

However, a priori estimates do not contain useful information during a computation. Specifically, the determination of better, reliable, adaptive step sizes h is nearly impossible (except for very simple cases).

On the other hand, a posteriori error estimation uses information of the already computed u_h during a simulation. Here, asymptotic information of u is not required:

$$\|u - u_h\| \leq c\eta(u_h)$$

where $\eta(u_h)$ is a computable quantity, because, as said, they work with the known discrete solution u_h. Such estimates cannot in general predict the asymptotic behavior, the reason for which a priori estimates remain important, but they can be evaluated during a computation with two main advantages:

- We can **control** the error during a computation;
- We can possibly **localize** the error estimator in order to obtain local error information on each element $K_i \in \mathcal{T}_h$. The elements with the highest error information can be decomposed into smaller elements in order to reduce the error.

7.2 Preliminaries

Following Becker/Rannacher [8, 9], Bangerth/Rannacher [7], and Rannacher [54], we derive a posteriori error estimates not only for norms, but for a general differentiable functional of interest $J : V \to \mathbb{R}$. This allows in particular to estimate technical quantities arising from applications (mechanical and civil engineering, fluid dynamics, solid dynamics, etc.). Of course, norm-based error estimation, $\|u - u_h\|$ can be expressed in terms of $J(\cdot)$; for hints see Section 7.9.

When designing a posteriori error estimates, we are in principle interested in the following relationship:

$$C_1\eta \leq |J(u) - J(u_h)| \leq C_2\eta \tag{155}$$

where $\eta := \eta(u_h)$ is the **error estimator** and C_1, C_2 are positive constants. Moreover, $J(u) - J(u_h)$ is the **true error**.

Definition 7.1 (Efficiency and reliablity)**.** *A good estimator $\eta := \eta(u_h)$ should satisfy two bounds:*

1. *An error estimator η of the form* (155) *is called* ***efficient*** *when*

$$C_1\eta \leq |J(u) - J(u_h)|,$$

which means that the error estimator is bounded by the error itself.

2. *An error estimator η of the form* (155) *is called* ***reliable*** *when*

$$|J(u) - J(u_h)| \leq C_2\eta.$$

Here, the true error is bounded by the estimator.

Remark 7.2. *For general goal functionals $J(\cdot)$, it is much more simpler to derive* ***reliable*** *estimators rather than prooving their efficiency.*

Remark 7.3 (AFEM)**.** *Finite element frameworks working with a posteriori error estimators applied to local mesh adaptivity, are called* ***adaptive FEM****.*

Definition 7.4 (Basic algorithm of AFEM)**.** *The basic algorithm for AFEM is always the same:*

1. ***Solve*** *the PDE on the current mesh $\mathcal{T}_h$;*
2. ***Estimate*** *the error via a posteriori error estimation to obtain η;*
3. ***Mark*** *the elements by localizing the error estimator;*
4. ***Refine/coarsen*** *the elements with the highest/lowest error contributions using a certain refinement strategy.*

A prototype situation on three meshes is displayed in Figure 22.

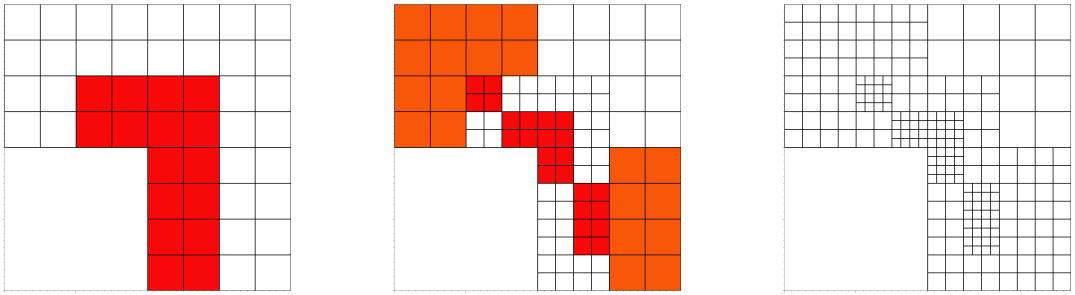

Figure 22: Meshes, say, on level 0, level 1 and 2. The colored mesh elements indicate high local errors and are marked for refinement using bisection and hanging nodes.

7.3 Goal-oriented error estimation using duality arguments: dual-weighted residuals (DWR)

7.3.1 Motivation

The goal of this section is to derive an error estimator that is based on duality arguments and can be used for norm-based error estimation (residual-based) as well as estimating more general error functionals $J(\cdot)$.

The key idea is based on numerical optimization, but goes back to classical mechanics in the 17th century. From our previous considerations, our goal is to reduce the error in the functional of interest with respect to a given PDE:

$$\min\big(J(u) - J(u_h)\big) \quad \text{s.t.} \quad a(u,\phi) = l(\phi), \tag{156}$$

where $J(\cdot)$ and $a(u,\phi)$ can be linear or nonlinear, but need to be differentiable (in Banach spaces).

Remark 7.5. *Of course for linear functionals, we can write*

$$J(u) - J(u_h) = J(u - u_h) = J(e), \quad e = u - u_h.$$

Such minimization problems as in (156) can be treated with the help of the so-called **Lagrangian** $L : V \times V \to \mathbb{R}$ in which the functional $J(\cdot)$ is of main interest subject to the constraint $a(\cdot,\cdot) - l(\phi)$ (here the PDE in variational form). Here, we deal with the **primal variable** $u \in V$ and a **Lagrange multiplier** $z \in V$, which is the so-called **adjoint variable** and which is assigned to the constraint $a(u,z) - l(z) = 0$. We then obtain

Definition 7.6. *The Lagrangian $L : V \times V \to \mathbb{R}$ is defined as*

$$L(u,z) = \big(J(u) - J(u_h)\big) - a(u,z) + l(z).$$

Before we proceed, we briefly recapitulate an important property of the Lagrangian. Let V and Z be two Banach spaces. A Lagrangian is a mapping $L(v,q) : U \times Y \to \mathbb{R}$, where $U \times Y \subset V \times Z$.

Definition 7.7 (Saddle-point)**.** *A point $(u,z) \in U \times Y$ is a saddle point of L on $U \times Y$ when*

$$\forall y \in Y : \quad L(u,y) \le L(u,z) \le L(v,z) \quad \forall v \in U.$$

A saddle-point is also known as min-max point. Geometrically one may think of a horse saddle.

Remark 7.8. *One can show that a saddle point yields under certain (strong) conditions a global minimum of u of $J(\cdot)$ in a subspace of U.*

The Lagrangian has the following important property to clear away the constraint (recall the constraint is the PDE!):

Lemma 7.9. *The problem*

$$\inf_{u \in V, -a(u,z)+l(z)=0} \big(J(u) - J(u_h)\big)$$

is equivalent to

$$\inf_{u \in V, -a(u,z)+l(z)=0} = \inf_{u \in V} \sup_{z \in V} L(u,z)$$

Proof. If $a(u,z) + l(z) = 0$ we clearly have $J(u) - J(u_h) = L(u,z)$ for all $z \in V$. If $a(u,z) + l(z) \neq 0$, then $\sup_{z \in V} L(u,z) = +\infty$, which shows the result. □

Remark 7.10. *In the previous lemma, we prefer to work with the infimum (inf) since it is not clear a priori whether the minimum (min) is taken.*

7.3.2 Excursus: Variational principles and Lagrange multipliers in mechanics

Variational principles - as motivated earlier in Section 6.1.2 - have been developed in physics and more precisely in **classical mechanics**, e.g., [26, 29]. This section shall serve two purposes:

1. it is another motivation of variational principles and thus the finite element method;
2. we give a mechanics-based motivation of Lagrange multipliers and constrained optimization problems as they form the background of the current chapter.

7.3.2.1 Variational principles in mechanics One of the first variational problems was designed by Jacob Bernoulli in the year 1696: how does a mass point reach from a point p_1 in the shortest time T another point p_2 under graviational forces? In consequence, we seek a minimal time T along a curve $u(x)$:

$$\min J(u) = \min(T)$$

with respect to the boundary conditions $u(x_1) = u_1$ and $u(x_2) = u_2$. To obtain the solution $u(x)$, we start from energy conservation, which yields for the kinetic and the potential energies:

$$\frac{mv^2}{2} = mg(u_1 - u),$$

where m is the mass, v the velocity of the mass, g the graviational force, u_1 the boundary condition, and $u = u(x)$ the sought solution curve. We does have the functional:

$$J(u) = T = \int_1^2 \frac{ds}{v} = \int_{x_1}^{x_2} \sqrt{\frac{1 + u'(x)^2}{2g(u_1 - u(x))}}\, dx.$$

Proposition 7.11. *The solution to this problem is the so-called* ***Brachistochrone*** *formulated by Jacob Bernoulli in 1696 [26].*

More generally, we formulate the unconstrained problem:

Formulation 7.12. *Find $u = u(x)$ such that*

$$\min J(u)$$

with

$$J(u) = \int_{x_1}^{x_2} F(u, u', x)\, dx.$$

Here, we assume that the function $F(u, u', x)$ and the boundary values $u(x_1) - u_1$ and $u(x_2) = u_2$ are known.

The idea is that we vary $J(u + \delta u)$ with a small increment δu as we have formulated in Section 6.1.2. Then, we arrive at the **Euler-Lagrange equations**:

Definition 7.13. *The (strong form of the) Euler-Lagrange equations are obtained as stationary point of the functional $J(u)$ and are nothing else, but the PDE in differential form:*

$$\frac{d}{dx}\frac{\partial F(u, u', x)}{\partial u'} = \frac{\partial F(u, u', x)}{\partial u}.$$

Remark 7.14 (Weak form of Euler-Lagrange equations)**.** *An equivalent statement is related to the weak form of the Euler-Lagrange equations in Banach spaces. Here, the functional $J(u)$ is differentiated w.r.t. u into the direction ϕ yielding $J'_u(u)(\phi)$ as used in Section 6.1.2.*

Example 7.15. *To compute the length of a curve $u(x)$ between two points $(x_1, u(x_1))$ and $(x_2, u(x_2))$, the following functional is used:*

$$J(u) = \int_1^2 ds = \int_{x_1}^{x_2} \sqrt{1 + (u')^2}\, dx.$$

This brings us to the question for which function $u(x)$ the functional $J(u)$ atteins a minimum, i.e., measuring the shortest distance between $(x_1, u(x_1))$ and $(x_2, u(x_2))$. This functional is the starting point to derive the clothesline problem.

Moreover, we identify $F(u, u', x) = \sqrt{1 + (u')^2}$. The Euler-Lagrange equation is then given by

$$\frac{d}{dx}\frac{u'(x)}{\sqrt{1 + (u')^2}} = 0.$$

When no external forces act (right hand side is zero), we can immediately derive the solution:

$$\frac{d}{dx}\frac{u'(x)}{\sqrt{1 + (u')^2}} = 0 \quad \Rightarrow \quad \frac{d}{dx}u'(x) = 0 \quad \Rightarrow \quad u'(x) = const \quad \Rightarrow \quad u(x) = ax + c.$$

The boundary conditions (not specified here) will determine the constants a and c. Of course, the solution, i.e., the shortest distance between two points, is a linear function. When external forces act (e.g., gravity), we will arrive at a solution similar to Section 6.1.1.

7.3.2.2 Variational principles in mechanics subject to constraints - Lagrange multipliers We come now to the second goal and address a physical interpretation of **Lagrange multipliers**. We have motivated the Poisson problem as the clothesline problem in Section 5.1. The derivation based on first principles in physics, namely conservation of potential energy is as follows. Let a clothesline be subject to graviational forces g. We seek the solution curve $u = u(x)$. A final equilibrium is achieved for minimal potential energy. This condition can be expressed as:

Formulation 7.16 (Minimal potential energy - an unconstrained variational problem)**.** *Find u such that*

$$\min J(u)$$

with

$$J(u) = E_{pot} = \underbrace{\int_1^2 gu\,dm}_{right\ hand\ side} = \underbrace{\rho g \int_{x_1}^{x_2} u\sqrt{1+(u')^2}\,dx}_{left\ hand\ side},$$

where g is the gravity, u the sought solution, dm mass elements, ρ the mass density. By variations δu we obtain solutions $u + \delta u$ and seek the optimal solution such that $J(u)$ is minimal. Of course, the boundary conditions u_1 and u_2 are not varied.

In the following, we formulate a constrained minimization problem. We ask that the length L of the clothesline is fixed:

$$K(u) = L = \int_{x_1}^{x_2} \sqrt{1+(u')^2}\,dx = \text{const.}$$

Formulation 7.17 (A constrained minimization problem)**.** *Find u such that*

$$\min J(u) \quad s.t. \quad K(u) = const.$$

The question is how we address Formulation 7.17 in practice? We explain the derivation in terms of a 1D situation in order to provide a basic understanding as usually done in these lecture notes.

The task is:

$$\min J(x, u(x)) \quad \text{s.t.} \quad K(x, u(x)) = 0. \tag{157}$$

Let us assume for a moment, we can explicitly compute $u = u_K(x)$ from $K(x, u(x)) = 0$ such

$$K(x, u_K(x)) = 0.$$

The minimal value of $J(x, u)$ on the curve $u_K(x)$ can be computed as minimum of $J(x, u_K(x))$. For this reason, we use the first derivative to compute the stationary point. With the help of the chain rule, we obtain:

$$0 = \frac{d}{dx} J(x, u_K) = J'_x(x, u_K) + J'_u(x, u_K) u'_K(x). \tag{158}$$

With this equation, we obtain the solution x_1. The solution u_1 is then obtain from $u_1 = u_K(x_1)$.

Using **Lagrange multipliers**, we avoid the explicit construction of $u_K(x)$ because such an expression is only easy to obtain for simple model problems. To this end, we introduce the variable z as a Lagrange multiplier.

We build the Lagrangian

$$L(x, u, z) = J(x, u) - zK(x, u)$$

and consider the problem:

$$\min L(x, u, z) \quad \text{s.t. } K(x, u) = 0.$$

Again, to find the optimal points, we differentiate w.r.t. to the three solution variables x, u, z:

$$\begin{aligned} J'_x(x, u) - zK'_x(x, u) &= 0 \\ J'_u(x, u) - zK'_u(x, u) &= 0 \\ K(x, u) &= 0 \end{aligned} \tag{159}$$

to obtain a first-order optimality system for determining x, u, z.

Proposition 7.18. *The formulations* (158) *and* (159) *are equivalent.*

Proof. We show (159) yields (158). We assume that we know $u = u_K(x)$, but we do not need the explicit construction of that $u_K(x)$. Then, $K(x,u) = 0$ is equivalent to

$$K(x,u) = u - u_K(x) = 0.$$

We differentiate w.r.t. x and u and insert the resulting expressions into the first two equations in (159):

$$J'_x(x,u) + zu'_K(x) = 0, \tag{160}$$
$$J'_u(x,u) - z = 0. \tag{161}$$

The second condition is nothing else, but

$$z = J'_u(x,u).$$

Here, we easily see that the Lagrange multiplier measures the sensitivity (i.e., the variation) of the solution curve u with respect to the functional $J(x,u)$. Inserting $z = J'_u(x,u)$ into (159) yields (158). The backward direction can be shown in a similar fashion. □

Remark 7.19. *The previous derivation has been done for a 1D problem in which $u(x)$ with $x \in \mathbb{R}$ is unknown. The method can be extended to $\mathbb{R}$ and Banach spaces, the latter one being addressed in Section 7.3.3.*

Remark 7.20. *We emphasize again that the use of Lagrange multipliers seems more complicated, but avoids the explicit construction of $u_K(x)$, which can be cumbersome. This is the main reason of the big success of* ***adjoint methods****, working with the* ***adjoint variable*** *z in physics and numerical optimization. Again, here, the Lagrangian $L(x,u,z)$ is minimized and the solution $u = u(x,z)$ contains a parameter z. This parameter is automatically determined such that the constraint $K(x,u) = 0$ is satisfied. The prize to pay is a higher computational cost since more equations need to be solved using* (159) *in comparison to* (158).

7.3.3 First-order optimality system

As motivated in the previous subsections, we seek minimal points and therefore we look at the first-order necessary conditions. Now we work in Banach spaces rather than $\mathbb{R}$. Differentiation with respect to $u \in V$ and $z \in V$ yields the optimality system:

Proposition 7.21 (Optimality system: Primal and adjoint problems)**.** *Differentiating (see Section 7.3.4) the Lagrangian in Definition 7.6 yields:*

$$L'_u(u,z)(\phi) = J'_u(u)(\phi) - a'_u(u,z)(\phi) \quad \forall \phi \in V,$$
$$L'_z(u,z)(\psi) = -a'_z(u,z)(\psi) + l'_z(\psi) \quad \forall \psi \in V.$$

The first equation is called the ***adjoint problem*** *and the second equation is nothing else than our PDE, the so-called* ***primal problem****. We also observe that the trial and test functions switch in a natural way in the adjoint problem.*

Proof. Trivial with the methods presented in Section 7.3.4. □

Remark 7.22 (on the notation)**.** *We abuse a bit the standard notation for semi-linear forms. Usually, all linear and nonlinear arguments are distingished such that $a'_u(u,z)(\phi)$ would read $a'_u(u)(z,\phi)$ because u is nonlinear and z and ϕ are linear. We use in these notes however $a'_u(u,z)(\phi)$ in order to emphasize that u and z are the main variables.*

Corollary 7.23 (Primal and adjoint problems in the linear case)**.** *In the linear case, we obtain from the general formulation:*

$$L(\phi,z) = J(\phi) - a(\phi,z)$$
$$L(u,\psi) = -a(u,\psi) + l(\psi).$$

Definition 7.24. *When we discretize both problems using for example a finite element scheme (later more), we define the residuals for $u_h \in V_h$ and $z_h \in V_h$. The primal and adjoint residuals read, respectively:*

$$\rho(u_h, \cdot) = -a'_z(u_h, z)(\cdot) + l'_z(\cdot)$$
$$\rho^*(z_h, \cdot) = J'_u(u)(\cdot) - a'_u(u, z_h)(\cdot).$$

In the linear case:

$$\rho(u_h, \cdot) = -a(u_h, \cdot) + l(\cdot)$$
$$\rho^*(z_h, \cdot) = J(\cdot) - a(\cdot, z_h).$$

Proposition 7.25 (First-order optimality system)**.** *To determine the optimal points $(u, z) \in V \times V$, we set the first-order optimality conditions to zero:*

$$0 = J'_u(u)(\phi) - a'_u(u)(z, \phi)$$
$$0 = -a'_z(u)(z, \psi) + l'_z(\psi).$$

Here, we easily observe that the primal equation is nothing else than the bilinear form $a(\cdot, \cdot)$ we have worked with so far. The adjoint problem is new (well known in optimization though) and yields sensitivity measures z of the primal solution u with respect to the goal functional $J(\cdot)$.

Example 7.26. *Let $a(u, \phi) = (\nabla u, \nabla \phi)$. Then: $a(u, z) = (\nabla u, \nabla z)$. Then,*

$$a'_u(u, z)(\phi) = (\nabla \phi, \nabla z),$$

and

$$a'_z(u, z)(\psi) = (\nabla u, \nabla \psi).$$

These derivatives are computed with the help of directional derivatives (Gâteaux derivatives) in Banach spaces.

7.3.4 Excursus: Differentiation in Banach spaces

We discuss in this section how to differentiate in Banach spaces.

Definition 7.27 (Directional derivative in a Banach space)**.** *Let V and W be normed vector spaces and let $U \subset V$ be non-empty. Let $f : U \to W$ be a given mapping. If the limit*

$$f'(v)(h) := \lim_{\varepsilon \to 0} \frac{f(v + \varepsilon h) - f(v)}{\varepsilon}, \quad v \in U, h \in V$$

exists, then $f'(v)(h)$ is called the directional derivative of the mapping f at v in direction h. If the directional derivative exists for all $h \in V$, then f is called directionally differentiable at v.

Remark 7.28 (on the notation)**.** *Often, the direction h is denoted by δv in order to highlight that the direction is related with the variable v. This notation is useful, when several solution variables exist and several directional derivatives need to be computed.*

Remark 7.29. *The definition of the directional derivative in Banach spaces is in perfect agreement with the definition of derivatives in $\mathbb{R}$ at $x \in \mathbb{R}$ (see [44]):*

$$f'(x) := \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon) - f(x)}{\varepsilon},$$

and in $\mathbb{R}^n$ we have (see [45]):

$$f'(x)(h) := \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon h) - f(x)}{\varepsilon}.$$

*A function is called differentiable when all directional derivatives exist in the point $x \in \mathbb{R}^n$ (similar to the Gâteaux derivative). The derivatives in the directions $e_i, i = 1, \ldots, n$ of the standard basis are the well-known **partial derivatives**.*

Definition 7.30 (Gâteaux derivative)**.** *Let the assumptions hold as in Definition 7.27. A directional-differentiable mapping as defined in Definition 7.27, is called Gâteaux differentiable at $v \in U$, if there exists a linear continuous mapping $A : U \to W$ such that*

$$f'(v)(h) = A(h)$$

holds true for all $h \in U$. Then, A is the Gâteaux derivative of f at v and we write $A = f'(v)$.

Remark 7.31. *The definition of the directional derivative immediately carries over to bilinear forms and for this reason the derivatives in Example 7.26 are well-defined. Therein, the semilinear form $a(u, z)$ is differentiated with respect to the first variable u into the direction $\phi \in V$.*

Remark 7.32. *The Gâteaux is computed with the help of directional deriatives and it holds $f'(v) \in L(U, W)$. If $W = \mathbb{R}$, then $f'(v) \in U^*$.*

Definition 7.33 (Fréchet derivative)**.** *A mapping $f : U \to W$ is Fréchet differentiable at $u \in U$ if there exists an operator $A \in L(U, W)$ and a mapping $r(u, \cdot) : U \to W$ such that for each $h \in U$ with $u + h \in U$, it holds:*

$$f(u + h) = f(u) + A(h) + r(u, h)$$

with

$$\frac{\|r(u,h)\|_W}{\|h\|_U} \to 0 \quad \text{for } \|h\|_U \to 0.$$

The operator $A(\cdot)$ is the Fréchet derivative of f at u and we write $A = f'(u)$.

Example 7.34. *The above bilinear form $a(u, \phi) = (\nabla u, \nabla\phi)$ is Fréchet differentiable in the first argument u (of course also in the second argument! But u is the variable we are interested in):*

$$a(u + h, \phi) = (\nabla(u + h), \nabla\phi) = \underbrace{(\nabla u, \nabla\phi)}_{=a(u,\phi)} + \underbrace{(\nabla h, \nabla\phi)}_{=a'_u(u,\phi)(h)}.$$

Here the remainder term is zero, i.e., $r(u, h) = 0$, because the bilinear form is linear in u. Thus, the Fréchet derivative of $a(u, \phi) = (\nabla u, \nabla\phi)$ is $a'_u(u, \phi)(h) = (\nabla h, \nabla\phi)$.

Second example is $J(u) = \int u^2\,dx$. Here:

$$J(u + h) = \int (u + h)^2\,dx = \underbrace{\int u^2\,dx}_{J(u)} + \underbrace{\int 2uh\,dx}_{=A(h)} + \underbrace{\int h^2\,dx}_{=r(u,h)}.$$

That $J(u)$ is really Fréchet differentiable we need to check whether

$$\frac{\|r(u,h)\|_W}{\|h\|_U} \to 0 \quad \text{for } \|h\|_U \to 0.$$

Here we have

$$\int h^2\,dx = \|h\|_V^2$$

and therefore:

$$\frac{\|h\|_V^2}{\|h\|_V} = \|h\|_V$$

For $h \to 0$ we clearly have $\|h\|_V \to 0$. Consequently the directional derivative of $J(u) = \int u^2\,dx$ is

$$J'(u)(h) = A(h) = \int_\Omega 2uh\,dx.$$

7.3.5 Linear problems and linear goal functionals (Poisson)

We explain our developments in terms of the linear Poisson problem and linear goal functionals.

Formulation 7.35. *Let $f \in L^2(\Omega)$, and we assume that the problem and domain are sufficiently regular such that the trace theorem (see Theorem 6.139 and also the literature, e.g., [76]) holds true, i.e., $h \in H^{-\frac{1}{2}}(\Gamma_N)$, and finally $u_D \in H^{\frac{1}{2}}(\Omega)$. Find $u \in \{u_D + V\}$:*

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V,$$

where

$$a(u, \phi) = (\alpha \nabla u, \nabla \phi)$$

and

$$l(\phi) := \int_\Omega f\phi \, dx + \int_{\Gamma_N} g\phi \, ds,$$

and the diffusion coefficient $\alpha := \alpha(x) \in L^\infty(\Omega)$. In this setting $\int_{\Gamma_N} g\phi \, ds$ has to be understood as duality product as for instance in [33]. If $g \in L^2(\Gamma_N)$ then it coincides with the integral.

As previously motivated, the aim is to compute a certain quantity of interest $J(u)$ with a desired accuracy at low computational cost.

Example 7.36. *Examples of goal functionals are mean values, line integration or point values:*

$$J(u) = \int_\Omega u \, dx, \quad J(u) = \int_\Gamma u \, ds, \quad J(u) = \int_\Gamma \partial_n u \, ds, \quad J(u) = u(x_0, y_0, z_0).$$

The first goal functional is simply the mean value of the solution. The third and fourth goal functionals are a priori not well defined. In case of the second functional we know the $\nabla u \in [L^2(\Omega)]^d$. Using the trace theorem (see Theorem 6.139), we can deduce that the trace in normal direction belongs to $H^{-\frac{1}{2}}(\partial\Omega)$. This however leads to the problem that the second functional is not always well defined. Concerning the third functional, we have previously shown in Section 6.8.4 that H^1 functions with dimension $d > 1$ the solution u is not any more continuous and the last evaluation is not well defined. If the domain and boundaries are sufficiently regular in 2D, the resulting solution is however H^2 regular and thanks to Sobolev embedding theorems (e.g., [19, 24]) also continuous.

Example 7.37. *Let $J(u) = \int_\Omega u \, dx$. Then the Fréchet derivative is given by*

$$J(\phi) = J'_u(u)(\phi) = \int_\Omega \phi \, dx$$

and, of course $J'_\lambda(u)(\psi) \equiv 0$.

The above goal functionals are however computed with a numerical method leading to a discrete version $J(u_h)$. Thus the key goal is to control the error $J(e) := J(u) - J(u_h)$ in terms of local residuals, which are computable on each mesh cell $K_i \in \mathcal{T}_h$.

Proposition 7.38 (Adjoint problem)**.** *Based on the optimality system, here Corollary 7.23, we seek the adjoint variable $z \in V$:*

$$a(\phi, z) = J(\phi) \quad \forall \phi \in V. \tag{162}$$

Specifically, the adjoint bilinear form for the Poisson problem is given by

$$a(\phi, z) = (\alpha \nabla \phi, \nabla z).$$

For symmetric problems, the adjoint bilinear form $a(\cdot, \cdot)$ is the same as the original one, but differs for non-symmetric problems like transport for example.

Proof. Apply the first-order necessary condition. □

Remark 7.39. *Existence and uniqueness of this adjoint solution follows by standard arguments provided sufficient regularity of the goal functional and the domain are given. The regularity of $z \in V$ depends on the regularity of the functional J. For $J \in H^{-1}(\Omega)$ it holds $z \in H^1(\Omega)$. Given a more regular functional like the L^2-error $J(\phi) = \|e\|^{-1}(e_h, \phi)$ (where $e := u - u_h$) with $J \in L^2(\Omega)^*$ (denoting the dual space), it holds $z \in H^2(\Omega)$ on suitable domains (convex polygonal or smooth boundary with C^2-parameterization).*

We now work with the techniques as adopted in Section 6.11. Inserting as special test function $\psi := u - u_h \in V$, recall that $u_h \in V_h \subset V$ into (162) yields:

$$a(u - u_h, z) = J(u - u_h),$$

and therefore we have now a representation for the error in the goal functional.

Next, we use the Galerkin orthogonality $a(u - u_h, \psi_h) = 0$ for all $\psi_h \in V_h$, and we obtain (compare to Aubin-Nitsche in Section 6.11.3):

$$a(u - u_h, z) = a(u - u_h, z) - \underbrace{a(u - u_h, \psi_h)}_{=0} = a(u - u_h, z - \psi_h) = J(u - u_h). \tag{163}$$

The previous step allows us to choose ψ_h in such a way that $z - \psi_h$ can be bounded using interpolation estimates. Indeed, since ψ_h is an arbitrary discrete test function, we can for example use a projection $\psi_h := i_h z \in V_h$ in (163), which is for instance the nodal interpolation.

Definition 7.40 (Error identity)**.** *Choosing $\psi_h := i_h z \in V_h$ in* (163) *yields:*

$$a(u - u_h, z - i_h z) = J(u - u_h). \tag{164}$$

Thus, the error in the functional $J(u - u_h)$ can be expressed in terms of a residual, that is weighted by adjoint sensitivity information $z - i_h z$.

However, since $z \in V$ is an unknown itself, we cannot yet simply evaluate the error identity because z is only known analytically in very special cases. In general, z is evaluated with the help of a finite element approximation yielding $z_h \in V_h$.

However, this yields another difficulty since we inserted the interpolation $i_h : V \to V_h$ for $z \in V$. When we approximate now z by z_h and use a linear or bilinear approximation $r = 1$: $z_h \in V_h^{(1)}$, then the interpolation i_h does nothing (in fact we interpolate a linear/bilinear function z_h with a linear/bilinear function $i_h z_h$, which is clearly

$$z_h - i_h z_h \equiv 0.$$

For this reason, we need to approximate z_h with a scheme that results in a higher-order representation: here at least something of quadratic order: $z_h \in V_h^{(2)}$.

7.3.6 Nonlinear problems and nonlinear goal functionals

In the nonlinear case, the PDE may be nonlinear (e.g., p-Laplace, nonlinear elasticity, Navier-Stokes, and Section 4.4) and also the goal functional may be nonlinear, e.g.,

$$J(u) = \int_\Omega u^2 \, dx.$$

These nonlinear problems yield a semi-linear form $a(u)(\phi)$ (not bilinear any more), which is nonlinear in the first variable u and linear in the test function ϕ. Both the semi-linear form and the goal functional $J(\cdot)$ are assumed to be (Fréchet) differentiable.

We start from Definition 7.21. Assuming that we have discretized both problems using finite elements (for further hints on the adjoint solution see Section 7.4). We suppose that all problems have unique solutions. We define:

Definition 7.41 (Primal and adjoint residuals)**.** *We define the primal and adjoint residuals, respectively:*

$$\rho(u_h)(\phi) = l(\phi) - a(u_h)(\phi) \quad \forall \phi \in V,$$
$$\rho^*(z_h)(\phi) = J'_u(u_h)(\phi) - a'_u(u_h)(\phi, z) \quad \forall \phi \in V.$$

It holds:

Theorem 7.42 ([9])**.** *For the Galerkin approximation of the first-order necessary system Definition 7.21, we have the* ***combined*** *a posteriori error representation:*

$$J(u) - J(u_h) = \eta = \frac{1}{2} \min_{\phi_h \in V_h} \rho(u_h)(z - \phi_h) + \frac{1}{2} \min_{\phi_h \in V_h} \rho^*(z_h)(u - \phi_h) + R.$$

The remainder term is of third order in $J(\cdot)$ and second order in $a(\cdot)(\cdot)$. Thus, for linear $a(\cdot)(\cdot)$ and quadratic $J(\cdot)$ the remainder term R vanishes. In practice the remainder term is neglected anyway and assumed to be small. However, in general this assumption should be justified for each nonlinear problem.

Proof. We refer the reader to [9]. □

Corollary 7.43 (Linear problems)**.** *In the case of linear problems the two residuals coincide. Then, it is sufficient to only solve the primal residual:*

$$\begin{aligned} J(u) - J(u_h) = J(u - u_h) &= \underbrace{\min_{\phi_h \in V_h} \rho(u_h)(z - \phi_h)}_{Primal} \\ &= \underbrace{\min_{\phi_h \in V_h} \rho^*(z_h)(u - \phi_h)}_{Adjoint} \\ &= \underbrace{\frac{1}{2} \min_{\phi_h \in V_h} \rho(u_h)(z - \phi_h) + \frac{1}{2} \min_{\phi_h \in V_h} \rho^*(z_h)(u - \phi_h)}_{Combined} \end{aligned}$$

Indeed the errors are exactly the same for linear goal functionals using the primal, adjoint or combined error estimator. The only difference are the resulting meshes. Several examples have been shown in Example 2 in [60].

Proof. It holds for the adjoint problem in the linear case:

$$J(\phi) = a(\phi, z).$$

Then:

$$\begin{aligned} J(u - u_h) = J(e) = & \underbrace{a(u - u_h, z)}_{= l(z) - a(u_h, z)} \\ & \underbrace{\qquad\qquad}_{=\rho(u_h, z)} \\ &= a(e, z) = a(e, z - z_h) = a(e, e^*) = a(u - u_h, e^*) \\ &= \underbrace{a(u, e^*)}_{=a(u,z) - a(u,z_h) = \underbrace{J(u) - a(u, z_h)}_{=\rho^*(z_h, u)}} \\ &= J(e^*), \end{aligned}$$

where $e^* := z - z_h$ and where $a(e, z) = a(e, z - z_h)$ and $a(u - u_h, e^*) = a(u, e^*)$ hold true thanks to Galerkin orthogonality. □

Definition 7.44 (A formal procedure to derive the adjoint problem for nonlinear equations)**.** *We summarize the previous developments. Based on Proposition 7.21, we set-up the adjoint problem as follows:*

- *Given $a(u)(\phi)$*
- *Differentiate w.r.t. $u \in V$ such that*

  $$a'_u(u)(\delta u, \phi)$$

  with the direction $\delta u \in V$.

- *Switch the trial function $\delta u \in V$ and the test function $\phi \in V$ such that:*

$$a'_u(u)(\phi, \delta u)$$

- *Replace $\delta u \in V$ by the adjoint solution $z \in V$:*

$$a'_u(u)(\phi, z)$$

- *This procedure also shows that the primal variable $u \in V$ enters the adjoint problem in the nonlinear case. However $u \in V$ is now given data and $z \in V$ is the sought unknown. This also means that in nonlinear problems, that primal solution needs to be stored in order to be accessed when solving the adjoint problem.*

7.4 Approximation of the adjoint solution for the primal estimator $\rho(u_h)(\cdot)$

In order to obtain a computable error representation, $z \in V$ is approximated through a finite element function $z_h \in V_h$, that is obtained from solving a discrete adjoint problem:

Formulation 7.45 (Discrete adjoint problem for linear problems)**.**

$$a(\psi_h, z_h) = J(\psi_h) \quad \forall \psi_h \in V_h. \tag{165}$$

Then the primal part of the error estimator reads:

$$a(u - u_h, z_h - i_h z_h) = \rho(u_h)(z_h - i_h z_h) \approx J(u) - J(u_h). \tag{166}$$

The difficulty is that if we compute the adjoint problem with the same polynomial degree as the primal problem, then $z_h - i_h z_h \equiv 0$, and thus the whole error identity defined in (166) would vanish, i.e., $J(u) - J(u_h) \equiv 0$. This is clear from a theoretical standpoint and can be easily verified in numerical computations.

In fact normally an interpolation operator i_h interpolates from infinite dimensional spaces V into finite-dimensional spaces V_h or from higher-order spaces into low-order spaces.

Thus:

$$\begin{aligned} &i_h : V \to V_h^{(1)} && \text{so far...} \\ &i_h : V_h^{(1)} \to V_h^{(1)} && \text{first choice, but trivial solution, useless} \\ &i_h : V_h^{(2)} \to V_h^{(1)} && \text{useful choice with nontrivial solution} \end{aligned}$$

From these considerations it is also clear that

$$i_h : V_h^{(1)} \to V_h^{(2)}$$

will even be worse (this would arise if we approximate the primal solution with $u_h \in V_h^{(2)}$ and $z_h \in V_h^{(1)}$).

7.4.1 Summary and alternatives for computing the adjoint

In summary:

- the adjoint solution needs to be computed either with a global higher-order approximation (using a higher order finite element of degree $r+1$ when the primal problem is approximated with degree r),
- or a solution on a finer mesh,
- or local higher-order approximations using a patch-wise higher-order interpolation [7, 9, 60].

Clearly, the last possibility is the cheapest from the computational cost point of view, but needs some efforts to be implemented. For the convenience of the reader we tacitly work with a global higher-order approximation in the rest of this chapter.

We finally end up with the primal error estimator:

Definition 7.46 (Primal error estimator)**.** *The primal error estimator is given by:*

$$a(u - u_h, z_h^{(r+1)} - i_h z_h^{(r+1)}) = \rho(u_h)(z_h^{(r+1)} - i_h z_h^{(r+1)}) =: \eta \approx J(u) - J(u_h).$$

7.5 Approximation of the primal solution for the adjoint estimator $\rho^*(z_h)(\cdot)$

To evaluate the adjoint estimator $\rho(z_h)(\cdot)$, we need to construct

$$u - i_h u$$

with $i_h : V \to V_h$ for $u \in V$ and $u_h \in V_h$. Here we encounter the opposite problem to the previous section. We need to solve the primal problem with higher accuracy using polynomials of degree $r+1$ in order to construct a useful interpolation, yielding

$$u - i_h u \neq 0 \quad \text{a.e.}$$

Then:

$$J(u) - J(u_h) \approx \eta := \rho^*(z_h)(u_h^{(r+1)} - i_h u_h^{(r+1)}).$$

7.6 Measuring the quality of the error estimator η

As quality measure how well the estimator approximates the true error, we use the effectivity index I_{eff}:

Definition 7.47 (Effectivity index)**.** *Let η be an error estimator and $J(u) - J(u_h)$ the true error. The effectivity index is defined as:*

$$I_{eff} := I_{eff}(u_h, z_h) = \Big| \frac{\eta}{J(u) - J(u_h)} \Big|. \tag{167}$$

Problems with good I_{eff} satisfy asymptotically $I_{eff} \to 1$ for $h \to 0$. We say that

- *for $I_{eff} > 1$, we have an **over estimation** of the error,*
- *for $I_{eff} < 1$, we have an **under estimation** of the error.*

7.7 Localization techniques

In the previous sections, we have derived an error approximation η with the help of duality arguments that lives on the entire domain Ω. In order to use the error estimator for mesh refinement we need to localize the error estimator η to single mesh elements $K_j \in \mathcal{T}_h$ or degrees of freedom (DoF) i. We present two techniques:

- A classical procedure using integration by parts resulting in an element-based indicator η_K;
- A more recent way by employing a partition-of-unity (PU) yielding PU-DoF-based indicators η_i.

We recall that the primal estimator starts with $z \in V$ from:

$$\rho(u_h)(z - i_h z) = a(u - u_h, z - i_h z) = a(u, z - i_h z) - a(u_h, z - i_h z) = l(z - i_h z) - a(u_h, z - i_h z),$$

and the adjoint estimator

$$\rho^*(z_h)(u - i_h u) = J(u - i_h u) - a(u - i_h u, z).$$

According to Theorem 7.42, we have for linear problems

$$\begin{aligned} J(u) - J(u_h) &= \frac{1}{2}\rho + \frac{1}{2}\rho^* + R \\ &= \frac{1}{2}\big(l(z - i_h z) - a(u_h, z - i_h z)\big) + \frac{1}{2}\big(J'(u - i_h u) - a'(u - i_h u, z_h)\big) + R \end{aligned}$$

where R is the remainder term. Furthermore on the discrete level, we have (for linear problems):

$$J(u) - J(u_h) \approx \eta := \frac{1}{2}\big(l(z_h - i_h z_h) - a(u_h, z_h - i_h z_h)\big) + \frac{1}{2}\big(J(u_h - i_h u_h) - a(u_h - i_h u_h, z_h)\big). \tag{168}$$

We understand that the discrete solutions z_h in the first part and u_h in the second part have to be understood computed in terms of higher-order approximations $r+1$.

In the following we localize these error estimators on a single element $K_i \in \mathcal{T}_h$. Here, the influence of neighboring elements $K_j, j \neq i$ is important [18]. In order to achieve such an influence, we consider the error

estimator on each cell and then either integrate back into the strong form (the classical way) or keep the weak form and introduce a partition-of-unity (a more recent way). Traditionally, there is also another way with weak form, proposed in [11], which has been analyzed theoretically in [60], but which we do not follow in these notes further.

In the following both localization techniques we present, we start from:

$$J(u-u_h) = l(z_h - i_h z_h) - a(u_h, z_h - i_h z_h) \tag{169}$$

For the Poisson problem, we can specify as follows:

$$J(u-u_h) = (f, z_h - i_h z_h) - (\nabla u_h, \nabla(z_h - i_h z_h)) \tag{170}$$

The next step will be to localize both terms either on a cell $K \in \mathcal{T}_h$ (Section 7.7.1) or on a degree of freedom (Section 7.7.3).

7.7.1 The classical way of error localization of the primal estimator for linear problems

In the classical way, the error identity (164) is treated with integration by parts on every mesh element $K \in \mathcal{T}_h$, which yields:

Proposition 7.48. *It holds:*

$$J(u-u_h) \approx \eta = \sum_{K\in\mathcal{T}_h} (f + \nabla\cdot(\alpha\nabla u_h), z_h - i_h z_h)_K + (\alpha\partial_n u_h, z_h - i_h z_h)_{\partial K} \tag{171}$$

Proof. Idea: Take (168), insert specific form of the PDE, reduce to an element K, integrate by parts. Now in detail: Let $\alpha = 1$ for simplicity. We start from

$$J(u-u_h) = (f, z_h - i_h z_h) - (\nabla u_h, \nabla(z_h - i_h z_h))$$

and obtain further

$$\begin{aligned}
J(u-u_h) &= (f, z_h - i_h z_h) - (\nabla u_h, \nabla(z_h - i_h z_h)) \\
&= \sum_K (f, z_h - i_h z_h)_K - (\nabla u_h, \nabla(z_h - i_h z_h))_K \\
&= \sum_K (f, z_h - i_h z_h)_K + (\Delta u_h, z_h - i_h z_h)_K - (\partial_n u_h, z_h - i_h z_h)_{\partial K} \\
&= \sum_K (f + \Delta u_h, z_h - i_h z_h)_K - \frac{1}{2}([\partial_n u_h]_K, z_h - i_h z_h)_{\partial K}
\end{aligned}$$

with $[\partial_n u_h] := [\partial_n u_h]_K = \partial_n u_h|_K + \partial_{n'} u_h|_{K'}$ where K' is a neighbor cell of K. On the outer (Dirichlet) boundary we set $[\partial_n u_h]_{\partial\Omega} = 2\partial_n u_h$. □

With the notation from the proof, we can define local residuals:

$$\begin{aligned}
R_T &:= f + \Delta u_h, \\
r_{\partial K} &:= -[\partial_n u_h]
\end{aligned}$$

Here, R_T are **element residuals** that measures the 'correctness' of the PDE. The $r_{\partial K}$ are so-called **face residuals** that compute the jumps over element faces and consequently measure the smoothness of the discrete solution u_h.

Further estimates are obtained as follows:

$$|J(u-u_h)| \le \Big|\sum_{K\in\mathcal{T}_h} ...\Big| \le \sum_{K\in\mathcal{T}_h} |...|$$

where we assume $z_h - \phi_h = 0$ on $\partial\Omega$. With Cauchy-Schwarz we further obtain:

$$|J(u-u_h)| \leq \eta = \sum_K \Big[\|f + \Delta u_h\|_K \|z_h - i_h z_h\|_K + \frac{1}{2} \|[\partial_n u_h]\|_{\partial K \setminus \partial\Omega} \|z_h - i_h z_h\|_{\partial K} \Big]$$
$$= \sum_K \rho_K(u_h)\omega_K(z_h) + \rho_{\partial K}(u_h)\omega_{\partial K}(z_h).$$

In summary we have shown:

Proposition 7.49. *We have:*

$$|J(u) - J(u_h)| \leq \eta := \sum_{K \in \mathcal{T}_h} \rho_K \omega_K, \tag{172}$$

with

$$\rho_K := \|f + \nabla \cdot (\alpha \nabla u_h)\|_K + \frac{1}{2} h_K^{-\frac{1}{2}} \|[\alpha \partial_n u_h]\|_{\partial K}, \tag{173}$$

$$\omega_K := \|z - i_h z\|_K + h_k^{\frac{1}{2}} \|z - i_h z\|_{\partial K}, \tag{174}$$

where by $[\alpha \partial_n u_h]$ we denote the jump of the u_h derivative in normal direction. The residual part ρ_K only contains the discrete solution u_h and the problem data. On Dirichlet boundaries Γ_D, we set $[\alpha \partial_n u_h] = 0$ and on the Neumann part we evaluate $\alpha \partial_n u_h = g_N$. Of course, we implicitly assume here that $g_N \in L^2(\Gamma_N)$ such that these terms are well-defined.

Remark 7.50. *In practice, this primal error estimator needs to be evaluated in the dual space. Here, we proceed as follows:*

- *Prolongate the primal solution u_h into the dual space;*
- *Next, we compute the interpolation $i_h z_h^{(r+1)} \in Q_r$ w.r.t. to the primal space;*
- *Then, we compute $z_h^{(r+1)} - i_h z_h^{(r+1)}$ (here, $i_h z_h^{(r+1)}$ is prolongated to Q_{r+1} in order to compute the difference);*
- *Evaluate the duality product $\langle \cdot, \cdot \rangle$ and face terms.*

Remark 7.51. *When $V_h = V_h^{(1)}$, then $\nabla \cdot \nabla u_h \equiv 0$. This also demonstrates heuristically that face terms are important.*

7.7.2 The classical way for the combined estimator

The combined estimator reads:

Proposition 7.52. *It holds:*

$$J(u) - J(u_h) \approx \sum_{K \in \mathcal{T}_h} \frac{1}{2}\eta_K + \frac{1}{2}\eta_K^*$$

with

$$\eta_K = \frac{1}{2}\big(\langle f + \nabla \cdot (\alpha \nabla u_h), z_h - i_h z_h \rangle_K + \int_{\partial K} \alpha \partial_n u_h \cdot (z_h - i_h z_h)\, ds \big)$$
$$\eta_K^* = \frac{1}{2}\big(J(u_h - i_h u_h) - (\int_K ... + \int_{\partial K} ...) \big)$$

7.7.3 A variational primal-based error estimator with PU localization

An alternative way is an DoF-based estimator, which is the first difference to before. The second difference to the classical approach is that we continue to work in the variational form and do not integrate back into the strong form. Such an estimator has been developed and analyzed in [60]. This idea combines the simplicity of the approach proposed in [11] (as it is given in terms of variational residuals) in terms of a very simple structure, which makes it particularly interesting for coupled and nonlinear PDE systems (see further comments below). Variational localizations are useful for nonlinear and coupled problems as we do not need to derive the strong form.

To this end we need to introduce a partition-of-unity (PU), which can be realized in terms of another finite element function. The procedure is therefore easy to realize in existing codes.

Definition 7.53 (PU - partition-of-unity)**.** *The PU is given by:*

$$V_{PU} := \{\psi_1, \dots, \psi_M\}$$

with $dim(V_{PU}) = M$. The PU has the property

$$\sum_{i=1}^{M} \psi_i \equiv 1.$$

Remark 7.54. *The PU can be simply chosen as the lowest order finite element space with linear or bilinear elements, i.e.,*

$$V_{PU} = V_h^{(1)}.$$

To understand the idea, we recall that in the classical error estimator the face terms are essential since they gather information from neighboring cells. When we work with the variational form, no integration by parts (fortunately!) is necessary. Therefore, the information of the neighboring cells is missing. Using the PU, we touch different cells per PU-node and consequently we gather now information from neighboring cells. Therefore, the PU serves as localization technique.

In the following, we now describe how the PU enters into the global error identity (164):

Proposition 7.55 (Primal error estimator)**.** *For the finite element approximation of Formulation 7.35, we have the a posteriori error estimate:*

$$|J(u) - J(u_h)| \leq \eta := \sum_{i=1}^{M} |\eta_i|, \tag{175}$$

where

$$\eta_i = a(u - u_h, (z - i_h z)\psi_i) = l((z - i_h z)\psi_i) - a(u_h, (z - i_h z)\psi_i),$$

and more specifically for the Poisson problem:

$$\eta_i = \Big\{ \langle f, (z - i_h z)\psi_i \rangle - (\alpha \nabla u_h, \nabla (z - i_h z)\psi_i) \Big\}. \tag{176}$$

7.7.4 PU localization for the combined estimator

Proposition 7.56 (The combined primal-dual error estimator)**.** *The combined estimator reads:*

$$|J(u) - J(u_h)| \leq \eta := \sum_{i=1}^{M} \frac{1}{2}|\eta_i| + \frac{1}{2}|\eta_i^*|$$

with

$$\eta_i = \frac{1}{2}\big(l((z_h - i_h z_h) - a(u, (z_h - i_h z_h)\psi_i)\psi_i)\big)$$
$$\eta_i^* = \frac{1}{2}\big(J'_u((u_h - i_h u_h)\psi_i) - a'_u((u_h - i_h u_h)\psi_i, z)\big)$$

7.8 Comments to adjoint-based error estimation

Adjoint-based error estimation allows to measure precisely at a low computational cost specific functionals of interest $J(u)$. However, the prize to pay is:

- We must compute a second solution $z \in V$.
- This second solution inside the primal estimator must be of higher order, which means more computational cost in comparison to the primal problem.
- For the full error estimator in total we need to compute four problems.
- From the theoretical point of view, we cannot proof convergence of the adaptive scheme for general goal functionals.

For nonlinear problems, one has to say that the primal problem is subject to nonlinear iterations, but the adjoint problem is always a linearized problem. Here, the computational cost may become less significant of computing an additional adjoint problem. Nonetheless, there is no free lunch.

7.9 Residual-based error estimation

Setting

$$J(\phi) = \|\nabla e_h\|^{-1}(\nabla\phi, \nabla e_h)$$

we obtain an estimator for the energy norm. And setting

$$J(\phi) = \|e_h\|^{-1}(\phi, e_h)$$

we obtain an estimator for the L^2 norm. For details we refer to [9] and [56].

7.10 Mesh refinement strategies

We have now on each element $K_j \in \mathcal{T}_h$ or each PU-DoF i an error value. It remains to set-up a strategy which elements shall be refined to enhance the accuracy in terms of the goal functional $J(\cdot)$.

Let an error tolerance TOL be given. Mesh adaption is realized using extracted local error indicators from the a posteriori error estimate on the mesh T_h. A cell-wise assembling reads:

$$|J(u) - J(u_h)| \leq \eta := \sum_{K \in T_h} \eta_K \quad \text{for all cells } K \in T_h.$$

Alternatively, the PU allows for a DoF-wise assembling:

$$|J(u) - J(u_h)| \leq \eta := \sum_{i} \eta_i \quad \text{for all DoFs } i \text{ of the PU.}$$

This information is used to adapt the mesh using the following strategy:

1. Compute the primal solution u_h and the adjoint solution z_h on the present mesh T_h.
2. Determine the cell indicator η_K at each cell K.
   Alternatively, determine the DoF-indicator η_i at each PU-DoF i.
3. Compute the sum of all indicators $\eta := \sum_{K \in T_h} \eta_K$.
   Alternatively, $\eta := \sum_i \eta_i$.
4. Check, if the stopping criterion is satisfied: $|J(u) - J(u_h)| \leq \eta \leq TOL$, then accept u_h within the tolerance TOL. Otherwise, proceed to the following step.
5. Mark all cells K_i that have values η_{K_i} above the average $\frac{\alpha\eta}{N}$ (where N denotes the total number of cells of the mesh $\mathbb{T}_h$ and $\alpha \approx 1$).
   Alternatively, all PU-DoFs are marked that are above, say, the average $\frac{\alpha\eta}{N}$.

Other mesh adaption strategies are discussed in the literature [7, 9]. For instance:

- Refining/coarsening a fixed fraction of elements. Here all elements are ordered with respect to their error values:
$$\eta_1 \geq \eta_2 \geq ... \geq \eta_N.$$
Then, for instance 30% of all elements are refined and for instance 2% of all cells are coarsened.
- Refining/coarsening according to a reduction of the error estimate (also known as bulk criterion or Dörfler marking [23]). Here, the error values are summed up such that a prescribed fraction of the total error is reduced. All elements that contribute to this fraction are refined.

Remark 7.57. *We emphasize that the tolerance TOL should be well above the tolerances of the numerical solvers. Just recall $TOL = 0.01$ would mean that the goal functional is measured up to a tolerance of 1%.*

When the DoF-based estimator is adopted, the error indicators η_i are node-wise contributions of the error. Mesh adaptivity can be carried out in two ways:

- in a node-wise fashion: if a node i is picked for refinement, all elements touching this node will be refined;
- alternatively, one could also first assemble element wise for each $K \in \mathcal{T}_h$ indicators by summing up all indicators belonging to nodes of this element and then carry out adaptivity in the usual element-wise way.

On adaptive meshes with hanging nodes, the evaluation of the PU indicator is straightforward: First, the PU is assembled in (176) employing the basis functions $\psi_i \in V_{PU}$ for $i = 1, \ldots, M$. In a second step, the contributions belonging to hanging nodes are condensed in the usual way by distribution to the neighboring indicators.

7.10.1 How to refine marked cells

It remains to explain how marked cells are finally refined.

7.10.1.1 Quads and hexs Using quadrilateral or hexahedra meshes, simply bisection can be used. Here, a cell is cut in the middle and split into 4 (in 2d) or 8 (in 3d) sub-elements. When the neighboring cell is not refined, we end up with so-called **hanging nodes**. These are degrees of freedom on the refined cell, but on the coarse neighboring cell, these nodes lie on the middle point of faces or edges and do not represent true degrees of freedom. Their values are obtained by interpolation of the neighboring DoFs. Consequently, condition No. 3 in Definition 6.186 is weakened in the presence of hanging nodes. For more details, we refer to Carey and Oden [16].

7.10.1.2 Triangles and prims For triangles or prisms, we have various possibilies how to split elements into sub-elements. Here, common ways are red and green refinement strategies. Here a strategy is to use red refinement and apply green refinement when hanging nodes would occur in neighboring elements.

7.10.1.3 Conditions on the geometry While refining the mesh locally for problems in $n \geq 2$, we need to take care that the minimum angle condition (see Section 6.11.5) is fulfilled.

7.10.2 Convergence of adaptive algorithms

The first convergence result of an adaptive algorithm was shown in [23] for the Poisson problem. The convergence of adaptive algorithms of generalized problems is subject to current research. Axioms of adaptivity have been recently formulated in [17].

7.11 Numerical test: Poisson with mean value goal functional

We consider the following test first: Let $\Omega = (0,1)^2$ and find $u \in H_0^1(\Omega)$ such that

$$(\nabla u, \nabla \phi) = (1, \phi) \quad \forall \phi \in H_0^1.$$

The goal functional is

$$J(u) = \frac{1}{|\Omega|} \int_\Omega u \, dx.$$

The reference value has been computed on a very fine mesh (despite here a manufactured solution would have been easy to derive) and is given by

$$J_{ref}(u) = 3.5144241236e-02.$$

The above goal functional yields the adjoint problem:

$$a(\phi, z) = J(\phi)$$

and in particular:

$$(\nabla \phi, \nabla z) = \int_\Omega \phi \, dx = (1, \phi).$$

The parameter α in Section 7.10 is chosen $\alpha = 4$. The solution of the adjoint problem is the same as the primal problem as also illustrated in Figure 23.

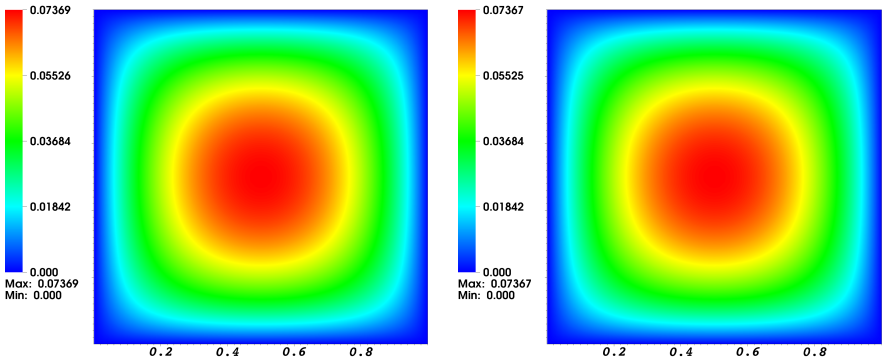

Figure 23: Section 7.11: Solutions of the primal and adjoint problems.

The first goal in our computations is to investigate I_{eff} using the PU localization from Proposition 7.55; again in short form:

$$|J(u) - J(u_h)| \leq \eta := \sum_{i=1}^{M} \left| \langle f, (z_h^{(2)} - i_h z_h^{(2)}) \psi_i \rangle - (\nabla u_h, \nabla (z_h^{(2)} - i_h z_h^{(2)}) \psi_i) \right|, \tag{177}$$

A second goal is to show that a pure weak form without partial integration or without PU does yield an over-estimation of the error:

$$|J(u) - J(u_h)| \leq \eta := \sum_{i=1}^{M} \left| \langle f, (z_h^{(2)} - i_h z_h^{(2)}) \rangle - (\nabla u_h, \nabla (z_h^{(2)} - i_h z_h^{(2)})) \right|, \tag{178}$$

```
Findings using PU localization
Dofs    True error      Estimated error     Effectivity
=====================================================
25      3.17e-03        3.14e-03            9.92e-01
69      1.66e-03        2.23e-03            1.35e+00
153     7.15e-04        1.02e-03            1.43e+00
329     2.01e-04        2.19e-04            1.09e+00
829     1.07e-04        1.43e-04            1.34e+00
1545    4.83e-05        5.62e-05            1.16e+00
4049    1.68e-05        2.28e-05            1.35e+00
6673    1.18e-05        1.35e-05            1.15e+00
15893   4.15e-06        5.35e-06            1.29e+00
24853   2.96e-06        3.29e-06            1.11e+00
62325   1.05e-06        1.38e-06            1.31e+00
96953   7.34e-07        8.21e-07            1.12e+00
=====================================================

Findings using a pure weak form without part. int. nor PU
Dofs    True error      Estimated error     Effectivity
=====================================================
25      3.17e-03        1.26e-02            3.97e+00
69      1.66e-03        8.79e-03            5.31e+00
165     6.92e-04        4.12e-03            5.95e+00
313     3.47e-04        2.42e-03            6.98e+00
629     1.77e-04        1.02e-03            5.73e+00
1397    8.36e-05        5.64e-04            6.75e+00
2277    4.57e-05        2.37e-04            5.19e+00
4949    2.14e-05        1.18e-04            5.54e+00
8921    1.04e-05        5.10e-05            4.90e+00
22469   4.71e-06        2.73e-05            5.78e+00
37377   2.66e-06        1.36e-05            5.12e+00
80129   1.83e-06        8.85e-06            4.83e+00
=====================================================
```

In the second table, we clearly observe an over-estimation of the error by a factor of 5. In the case of the PU localization, we see I_{eff} around 1, what we expected due to the regularity of the problem.

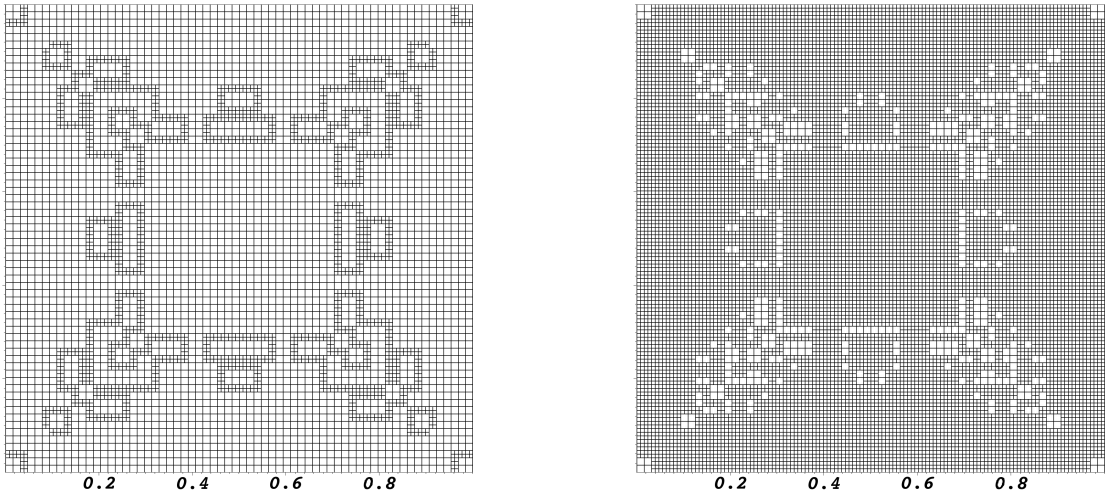

Figure 24: Section 7.11: Locally adapted meshes using PU localization for the refinement levels 7 and 8.

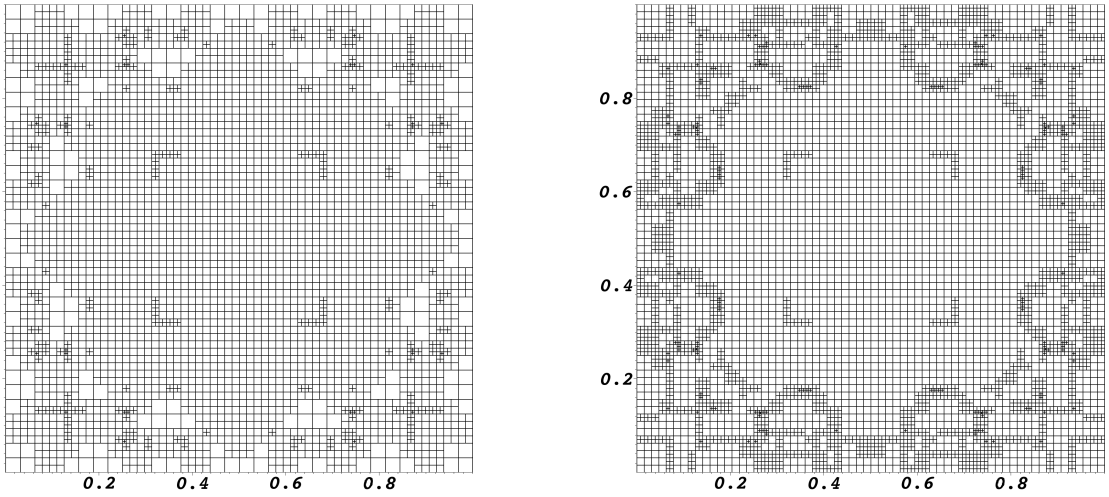

Figure 25: Section 7.11: Locally adapted meshes using a pure weak form without partial integration nor PU localization for the refinement levels 7 and 8.

7.12 Numerical test: L-shaped domain with Dirac rhs and Dirac goal functional

We redo Example 2 from [60]. Specifically, we consider Poisson's equation on an L-shaped domain $\Omega_L = (-1,1)^2 \setminus (-1,0)^2$ with bounddary $\partial\Omega_L$, where the right hand side is given by a Dirac function in $x_0 = (-0.5, 0.5)$

$$-\Delta u = \delta_{x_0} \text{ in } \Omega_L, \quad u = 0 \text{ on } \partial\Omega_L. \tag{179}$$

As goal functional, we take the point evaluation in $x_1 = (0.5, -0.5)$ such that $J(\phi) = \phi(x_1)$. The adjoint problem corresponds to solving Poisson's equation $-\Delta z = \delta_{x_1}$ with a Dirac right hand side in x_1. Both the primal problem and the adjoint problem lack the required minimal regularity for the standard finite element theory (see the previous chapter), such that a regularization by averaging is required, e.g. by averaging over a small subdomain:

$$J_\epsilon(\phi) = \frac{1}{2\pi\epsilon^2} \int_{|x-x_1|<\epsilon} \phi(x)\,\mathrm{d}x, \tag{180}$$

where $\epsilon > 0$ is a small parameter not depending on h. As reference functional quantity we compute the value

$$J_{ref} = 2.134929 \cdot 10^{-3} \pm 10^{-7} \tag{181}$$

obtained on a sufficiently globally refined mesh.

Dofs	True error	Estimated error	Effectivity
21	1.24e-03	1.41e-03	1.14e+00
65	2.74e-04	2.45e-04	8.97e-01
225	7.50e-05	6.32e-05	8.42e-01
677	2.65e-05	2.11e-05	7.95e-01
2021	9.93e-06	7.70e-06	7.75e-01
6089	3.59e-06	2.80e-06	7.81e-01
20157	1.38e-06	1.18e-06	8.51e-01
61905	5.16e-07	5.20e-07	1.01e+00
177221	1.34e-07	2.10e-07	1.56e+00

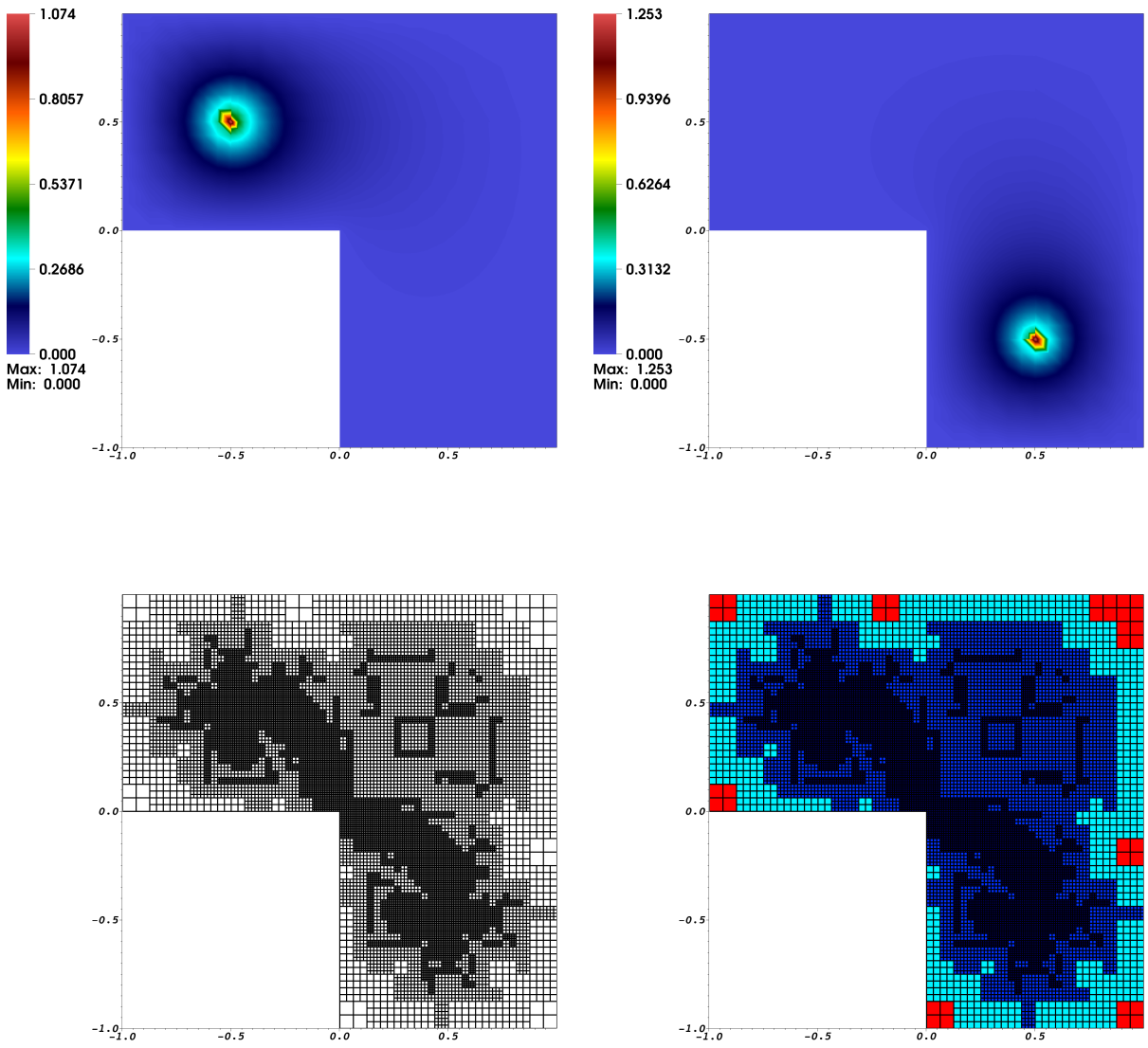

Figure 26: Section 7.12: Poisson problem with Dirac right hand side (left figure on top) and Dirac goal functional (right figure on top) on the l-shaped domain. For mesh refinement the primal error estimator with PU localization is used. The primal problem is discretized with Q_1 elements and the adjoint problem with Q_2 elements. The PU is based on Q_1 elements. The mesh is refined around the two Dirac functions and in the corner of the domain (bottom figure left). In order to highlight the various mesh levels within one computation, the different element areas are colorized (bottom figure right).

7.13 Final comments to error estimation in numerical simulations

We give some final comments to error estimation. One should keep in mind several aspects. First:

- Numerical mathematics is its own discipline and we should aim to develop methods with best accuracy as possible.
- On the other hand, numerical mathematics is a very useful tool for many other disciplines (engineering, natural sciences, etc.). Here, an improvement of a process (i.e., approximation) by several percent is already very important, while the total error may be still large though and in the very sense of numerical mathematics the result is not yet satisfying.

Second (w.r.t. the first bullet point): A posteriori error estimation, verification of the developed methods, and code validation are achieved in three steps (with increasing level of difficulty) after having constructed an a posteriori error estimator η that can be localized and used for local mesh adaptivity.

1. If possible test your code with an acknowledged benchmark configuration and verify whether $J(u_h)$ matches the benchmark values in a given range and on a sequence of at least three meshes. This first step can be performed with uniform and adaptive mesh refinement.
2. Compute $J(u)$ either on a uniformly-refined super-fine mesh or even analytically. Compute the error $J(u) - J(u_h)$ and observe whether the error is decreasing. If a priori estimates are available, see if the orders of convergence are as expected. But be careful, often goal functionals are nonlinear, for which rigorous a priori error estimates are not available.
3. Compare η and $J(u) - J(u_h)$ in terms of the effectivity index I_{eff}.

In the very sense of numerical mathematics we must go for all three steps, but in particular the last step No. 3 is often very difficult when not all theoretical requirements (smoothness of data and boundary conditions, regularity of the goal functional, smoothness of domains and boundaries) are fulfilled. Also keep in mind that all parts of error estimators are implemented and that still the remainder term R may play a role. For instance, in practice, very often when working with the DWR method, only the primal error estimator is implemented and the adjoint part neglected; see the following section.

7.14 Example: Stationary Navier-Stokes 2D-1 benchmark

We demonstrate the developements in this chapter with a well-known benchmark in fluid mechanics: the 2D-1 benchmark [62]. The underlying equations are the stationary incompressible Navier-Stokes equations [27, 52, 66], which are vector-valued (more details in Section 9) and nonlinear (more details in Section 11).

The goals are the accurate measurements of drag, lift and a pressure difference.

7.14.1 Equations

Formulation 7.58. *Let $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma_{wall} \cup \Gamma_{in} \cup \Gamma_{out}$. Find vector-valued velocities $v : \Omega \to \mathbb{R}^2$ and a scalar-valued pressure $p : \Omega \to \mathbb{R}$ such that*

$$\begin{aligned} \rho(v \cdot \nabla)v - div(\sigma) &= \rho f \quad in\ \Omega, \\ div(v) &= 0 \quad in\ \Omega, \\ v &= 0 \quad on\ \Gamma_{wall}, \\ v &= g \quad on\ \Gamma_{in}, \\ \nu\partial_n v - pn &= 0 \quad on\ \Gamma_{out}, \end{aligned}$$

where $f : \Omega \to \mathbb{R}^2$ is a force, $g : \Gamma \to \mathbb{R}^2$ a Dirichlet inflow profile, n the normal vector, ρ the density and

$$\sigma = -pI + \rho\nu(\nabla v + \nabla v^T), \quad [\sigma] = Pa = N/m^2$$

is the Cauchy stress tensor for a Newtonian fluid. Moreover, I is the identity matrix and ν the kinematic viscosity.

The Reynolds number Re is given by the dimensionless expression

$$Re = \frac{LU}{\nu}$$

where L is a characteristic length and U a characteristic velocity.

Remark 7.59 (Stokes)**.** *Neglecting the convection term $(v \cdot \nabla)v$ for viscous flow, we obtain the linear Stokes equations.*

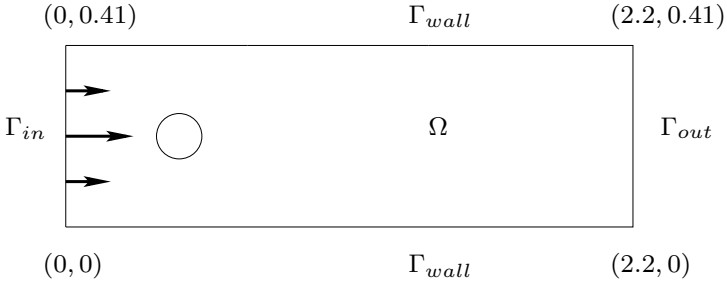

Figure 27: 2D-1 benchmark configuration.

For a variational formulation, we define the spaces (for details see [27, 52, 66]):

$$V_v := \{v \in H^1(\Omega)^2 |\ v = 0 \text{ on } \Gamma_{in} \cup \Gamma_{wall}\},$$
$$V_p = L^2(\Omega)/\mathbb{R}.$$

Here, V_v is a vector-valued space. The pressure is only defined up to a constant. For existence of the pressure, the inf-sup condition is adopted [27, 52, 66].

On the outflow boundary Γ_{out}, we apply a natural boundary condition:

$$\rho\nu\partial_n v - pn = 0.$$

This type of condition implicitly normalizes the pressure [38, 51] to a unique solution. Thus, the pressure space can be written as

$$V_p = L^2(\Omega).$$

Remark 7.60. *Applying $\rho\nu\partial_n v - pn = 0$ on Γ_{out} is not consistent with the total stress $\sigma \cdot n = -pI + \rho\nu(\nabla v + \nabla v^T) \cdot n$. The symmetric part $\rho\nu\nabla v^T \cdot n$ needs to be subtracted on Γ_{out}. For more details see [38, 73]. For a better concentration on the main parts in this section, we neglect this term in the following.*

As variational formulation, we obtain:

Formulation 7.61. *Find $v \in \{g + V_v\}$ and $p \in V_p$ such that*

$$(\rho(v \cdot \nabla)v, \phi) + (\sigma, \nabla\phi) = (\rho f, \phi) \quad \forall \phi \in V_v,$$
$$(div(v), \psi) = 0 \quad \forall \psi \in V_p.$$

For the prescription of non-homogeneous Dirichlet conditions, we refer to Section 6.9.3.

For the finite element spaces we have to satisfy a discrete inf-sup condition [27]. Basically speaking, the finite element space for the velocities must be large enough. A stable FE pair is the so-called Taylor-Hood element $Q_2 \times Q_1$; here defined on quadrilaterals. The velocities are bi-quadratic and the pressure bi-linear. Hence we seek $(v_h, p_h) \in V_h^2 \times V_h^1$ with

$$V_h^s := \{u_h \in C(\bar{\Omega}) |\ u_h|_K \in Q_s(K), \forall K \in \mathcal{T}_h\}.$$

This is a conforming finite element space, i.e., $V_h^s \subset H^1(\Omega)$.

Formulation 7.62. *Find $v_h \in \{g + V_h^2\}$ and $p_h \in V_h^1$ such that*

$$(\rho(v_h \cdot \nabla)v_h, \phi_h) + (\sigma, \nabla\phi_h) = (\rho f, \phi_h) \quad \forall \phi_h \in V_h^2,$$
$$(div(v_h), \psi_h) = 0 \quad \forall \psi_h \in V_h^1.$$

A compact semi-linear form is formulated as follows:

Formulation 7.63. *Find $U_h := (v_h, p_h) \in \{g + V_h^2\} \times V_h^1$ such that*

$$a(U_h)(\Psi_h) = (\rho(v_h \cdot \nabla)v_h, \phi_h) + (\sigma, \nabla\phi_h) - (\rho f, \phi_h) + (div(v_h), \psi_h) \quad \forall \Psi_h = (\phi_h, \psi_h) \in V_h^2 \times V_h^1$$

7.14.2 Functionals of interest

The goal functionals are drag, lift, and pressure difference, respectively:

$$J_1(U) = \int_S \sigma \cdot ne_1 \, ds, \quad J_2(U) = \int_S \sigma \cdot ne_2 \, ds, \quad J_3(U) = p(x_0, y_0) - p(x_1, y_1)$$

with $x_0, x_1, y_0, y_1 \in \Omega$ and where e_1 and e_2 are the unit vectors in x and y direction. We are interested in the drag and lift coefficients:

$$c_D = \frac{2J_1}{\rho V^2 D}, \quad c_L = \frac{2J_2}{\rho V^2 D}$$

with $D = 0.1m$ (the cylinder diameter) and the mean velocity

$$V = V(t) = \frac{2}{3} v_x(0, H/2)$$

where v_x is defined below and with the height of the channel $H = 0.41m$.

7.14.3 A duality-based a posteriori error estimator

We develop an error estimator only based on the primal part. The adjoint bi-linear form is given by:

Formulation 7.64. *From the optimality system, we obtain:*

$$a'_U(U_h)(\Psi_h, Z_h) = (\rho(\phi_h \cdot \nabla)v_h + \rho(v_h \cdot \nabla)\phi_h, z_h^v) + (-\psi_h I + \nu\rho(\nabla\phi_h + \nabla\phi_h^T), \nabla z_h^v) + (div(\phi_h), z_h^p).$$

Proposition 7.65 (Adjoint problem)**.** *The adjoint problem depends on the specific goal functional $J_i(U), i = 1, 2, 3$. For example, for the drag, it holds: Find $Z_h = (z_h^v, z_h^p) \in V_h^3 \times V_h^2$ such that*

$$a'_U(U_h)(\Psi_h, Z) = J_1(\Psi_h) \quad \forall \Psi_h = (\phi_h, \psi_h) \in V_h^3 \times V_h^2$$

with $a'_U(U_h)(\Psi_h, Z)$ defined in Formulation 7.64 and

$$J_1(\Psi_h) = \int_S \Big(-\psi_h I + \rho\nu(\nabla\phi_h + \nabla\phi_h^T)\Big) \cdot ne_1 \, ds, \quad \Psi_h = (\phi_h, \psi_h).$$

Recall that the adjoint solution must be approximated with a higher-order method. Therefore, we choose the space $V^3 \times V^2$.

The primal PU-DWR error estimator reads:

Proposition 7.66. *It holds:*

$$|J(U) - J(U_h)| \leq \eta := \sum_i |\eta_i|,$$

with

$$\eta_i = -a(U_h)((Z_h - i_h Z_h)\chi_i), \quad Z_h = (z_h^v, z_h^p), \quad \chi_i \in V_{PU}$$

and the PU is realized as $V_{PU} := V_h^1$ and where $a(U_h)(\cdot)$ has been defined in Formulation 7.63.

Remark 7.67. *In the error estimator the non-homogeneous Dirichlet data g on Γ_{in} are neglected and assumed to be small. Also, the outflow correction condition, see Remark 7.60, is neglected.*

7.14.4 2D-1 configuration

The configuration is displayed in [62][Figure 1].

7.14.5 Boundary conditions

As boundary conditions, we specified all boundaries previously except the inflow:

$$g = (g_1, g_2)^T = (v_x(0, y), v_y)^T = (4v_m y \frac{H - y}{H^2}, 0)^T$$

with $v_m = 0.3m/s$. The resulting Reynolds number is $Re = 20$ (use D and V as defined above).

7.14.6 Parameters and right hand side data

We use $\rho = 1kg/m^3, \nu = 10^{-3}m^2/s$ and $f = 0$.

7.14.7 Step 1: Verification of benchmark values

In Table 3 in [62], the following bounds are given:

- Drag, c_D: $5.57 - 5.59$.
- Lift, c_L: $0.0104 - 0.0110$.
- Pressure difference, Δp: $0.1172 - 0.1176$.

We obtain on level 3 with 1828 elements and with 15868 velocity DoFs and 2044 pressure DoFs:

```
Functional     Value
=====================================
Drag:          5.5754236905876873e+00
Lift:          1.0970590684824442e-02
Pressure diff: 1.1745258793930979e-01
=====================================
```

7.14.8 Step 2 and Step 3: Computing $J(u)$ on a fine mesh, $J(u) - J(u_h)$ and I_{eff}

The reference value $J_1(U)$ for the drag is:

```
5.5787294556197073e+00
```

and was obtained on a four times (Level 4) uniformely refined mesh.

We obtain:

```
Level   Exact err       Est err           I_eff
================================================
0       3.51e-01        1.08e-01          3.07e-01
1       9.25e-02        2.08e-02          2.25e-01
2       1.94e-02        6.07e-03          3.12e-01
3       3.31e-03        2.45e-03          7.40e-01
================================================
```

We observe that the true error and the estimated error η both decrease. The effectivity indicies are less than 1 and indicate an underestimation of the true error (clearly seen in columns No. 2 and 3 as well). Because of our several assumptions and relatively coarse meshes, the results are more or less satisfying w.r.t. Step 3. Regarding Step 1 and 2, our findings are excellent.

7.14.9 More findings - graphical solutions

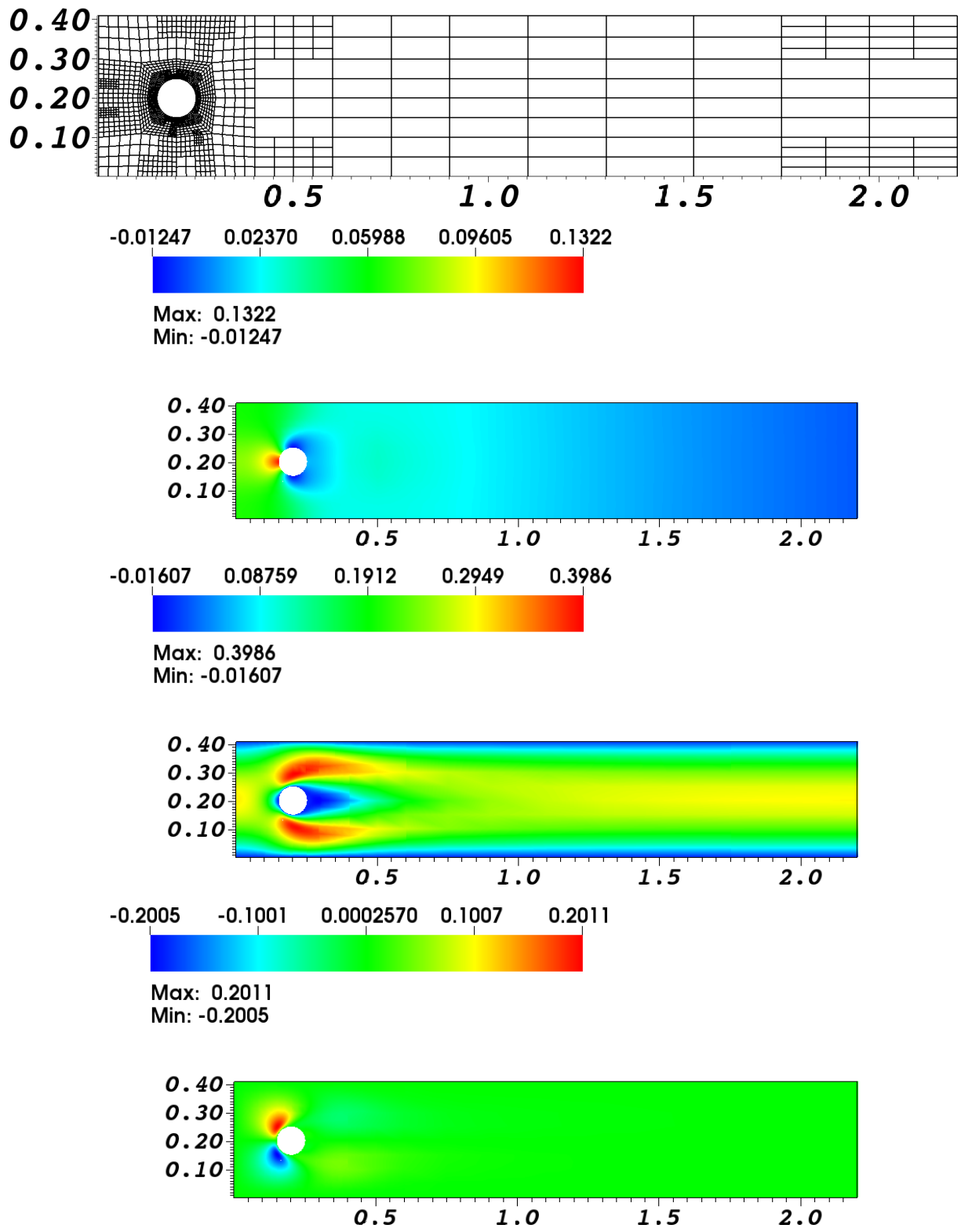

Figure 28: 2D-1 fluid benchmark: adaptively refined mesh with respect to $J_1(U)$, the pressure solution, x-velocity, y-velocity.

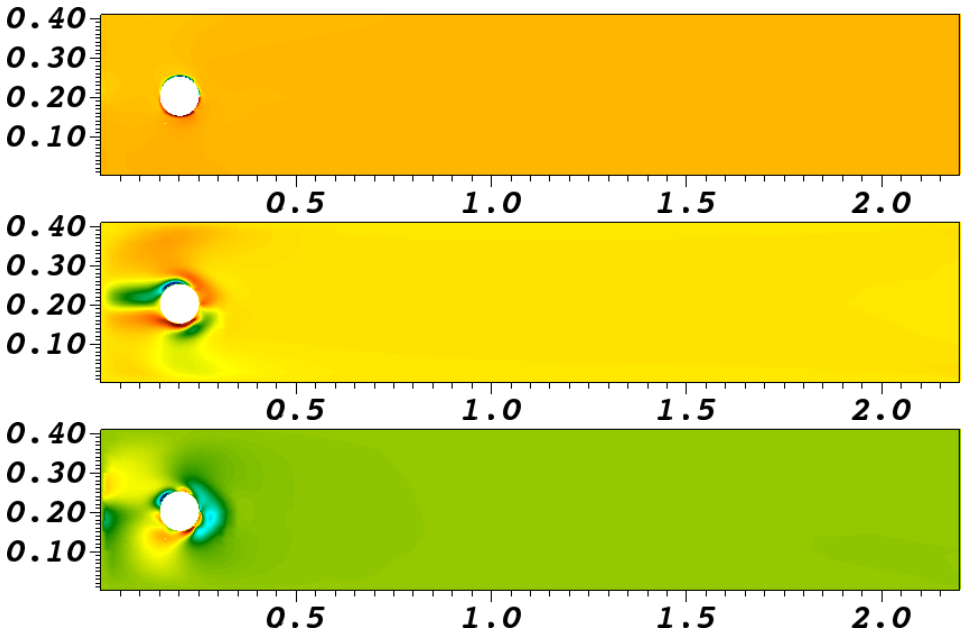

Figure 29: 2D-1 fluid benchmark: adjoint solutions of the pressure solution, x-velocity, y-velocity.

7.15 Chapter summary and outlook

In this chapter, we introduced goal-oriented a posteriori error estimation to control the error during a finite element simulation a posteriori. Furthermore, these estimates can be localized on each mesh element and then be used for mesh refinement to obtain a higher accuracy of the numerical solution with low computational cost. The discrete systems in the present and the previous chapter lead to large linear equation systems that can in general not be solved anymore with a direct decomposition (e.g., LU). We shall briefly introduce some iterative methods in the next chapter.

8 Numerical solution of the discretized problems

In this chapter, we provide some ideas how to solve the arising linear systems

$$Ax = b$$

where

$$A \in \mathbb{R}^{n\times n}, \quad x = (u_1, \dots, u_n)^T \in \mathbb{R}^n, \quad b \in \mathbb{R}^n.$$

when discretizing a PDE using finite differences or finite elements. We notice that to be consistent with the previous notation, we assume that the boundary points x_0 and x_{n+1} are not assembled.

For a moderate number of degrees of freedom, direct solvers such as Gaussian elimination, LU or Cholesky (for symmetric A) can be used.

More efficient schemes for large problems in terms of

- computational cost (CPU run time);
- and memory consumptions

are **iterative solvers.** Illustrative examples of floating point operations and CPU times are provided in [61][Pages 68-69, Tables 3.1 and 3.2].

We notice that most of the material in this chapter is taken from [61][Chapter 7] and translated from German into English.

8.1 On the condition number of the system matrix

Proposition 8.1. *Let $\mathcal{T}_h$ be a quasi-uniform triangulation. The condition numbers $\chi(A)$ of the system matrix A, and the condition number $\chi(M)$ of a mass matrix M can be estimated by:*

$$\chi(A) = O(h^{-2}), \qquad \chi(M) = O(1), \quad h \to 0.$$

Here

$$A = a_{ij} = a(\phi_j, \phi_i) = \int_\Omega \nabla\phi_j \nabla\phi_i \, dx,$$
$$M = m_{ij} = m(\phi_j, \phi_i) = \int_\Omega \phi_j \phi_i \, dx.$$

Proof. See [56], Section 3.5 or also [43], Section 7.3. □

8.2 Fixed-point schemes: Richardson, Jacobi, Gauss-Seidel

A large class of schemes is based on so-called **fixed point** methods:

$$g(x) = x.$$

We provide in the following a brief introduction that is based on [61]. Starting from

$$Ax = b$$

we write

$$0 = b - Ax$$

and therefore

$$x = \underbrace{x + (b - Ax)}_{g(x)}.$$

Introducing a scaling matrix C (in fact C is a preconditioner) and an iteration, we arrive at

$$x^k = x^{k-1} + C(b - Ax^{k-1}).$$

Summarizing, we have

Definition 8.2. *Let $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $C \in \mathbb{R}^{n\times n}$. To solve*

$$Ax = b$$

we choose an initial guess $x^0 \in \mathbb{R}^n$ and we iterate for $k = 1, 2, \ldots$:

$$x^k = x^{k-1} + C(b - Ax^{k-1}).$$

Please be careful that k does not denote the power, but the current iteration index. Furthermore, we introduce:

$$B := I - CA \quad \textit{and} \quad c := Cb.$$

Then:

$$x^k = Bx^{k-1} + c.$$

Thanks to the construction of

$$g(x) = Bx + c = x + C(b - Ax)$$

it is trivial to see that in the limit $k \to \infty$, it holds

$$g(x) = x$$

with the solution

$$Ax = b$$

Remark 8.3. *Thanks to Banach's fixed point theorem (see again [61]), we can investigate under which conditions the above scheme will converge. We must ensure that*

$$\|g(x) - g(y)\| \leq \|B\| \, \|x - y\|$$

with $\|B\| < 1$. In fact we have:

$$g(x) = Bx + c, \quad g(y) = By + c$$

from which the above estimate is easily to see. A critical issue (which is also true for the ODE cases) is that different norms may predict different results. For instance it may happen that

$$\|B\|_2 < 1 \quad \textit{but} \quad \|B\|_\infty > 1.$$

For this reason, one often works with the spectral norm spr(B). More details can be found for instance in [61]. That we have a chance to achieve $\|B\| < 1$ is the reason why we introduced the matrix C.

We concentrate now on the algorithmic aspects. The two fundamental requirements for the matrix C (defined above) are:

- It should hold $C \approx A^{-1}$ and therefore $\|I - CA\| \ll 1$;
- It should be simple to construct C.

Of course, we easily see that these two requirements are conflicting statements. As always in numerics we need to find a trade-off that is satisfying for the developer and the computer.

Definition 8.4 (Richardson iteration)**.** *The simplest choice of C is the identity matrix, i.e.,*

$$C = \omega I.$$

Then, we obtain the Richardson iteration

$$x^k = x^{k-1} + \omega(b - Ax^{k-1})$$

with a relaxation parameter $\omega > 0$.

Further schemes require more work and we need to decompose the matrix A first:

$$A = L + D + U.$$

Here, L is a lower-triangular matrix, D a diagional matrix, and U an upper-triangular matrix. In more detail:

$$A = \underbrace{\begin{pmatrix} 0 & & \dots & 0 \\ a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{pmatrix}}_{=:L} + \underbrace{\begin{pmatrix} a_{11} & & \dots & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & \dots & & a_{nn} \end{pmatrix}}_{=:D} + \underbrace{\begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & \dots & & 0 \end{pmatrix}}_{=:U}.$$

With this, we can now define two very important schemes:

Definition 8.5 (Jacobi method)**.** *To solve $Ax = b$ with $A = L + D + R$ let $x^0 \in \mathbb{R}^n$ be an initial guess. We iterate for $k = 1, 2, \dots$*

$$x^k = x^{k-1} + D^{-1}(b - Ax^{k-1})$$

or in other words $J := -D^{-1}(L + R)$:

$$x^k = Jx^{k-1} + D^{-1}b.$$

Definition 8.6 (Gauß-Seidel method)**.** *To solve $Ax = b$ with $A = L + D + R$ let $x^0 \in \mathbb{R}^n$ be an initial guess. We iterate for $k = 1, 2, \dots$*

$$x^k = x^{k-1} + (D + L)^{-1}(b - Ax^{k-1})$$

or in other words $H := -(D + L)^{-1}R$:

$$x^k = Hx^{k-1} + (D + L)^{-1}b.$$

To implement these two schemes, we provide the presentation in index-notation:

Theorem 8.7 (Index-notation of the Jacobi- and Gauß-Seidel methods)**.** *One step of the Jacobi method and Gauß-Seidel method, respectively, can be carried out in $n^2 + O(n)$ operations. For each step, in index-notation for each entry it holds:*

$$x_i^k = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{k-1} \right), \quad i = 1, \dots, n,$$

i.e., (for the Gauss-Seidel method):

$$x_i^k = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^k - \sum_{j>i} a_{ij} x_j^{k-1} \right), \quad i = 1, \dots, n.$$

8.3 Gradient descent

An alternative class of methods is based on so-called **descent** or **gradient** methods, which further improve the previously introduced methods. So far, we have:

$$x^{k+1} = x^k + d^k, \quad k = 1, 2, 3, \dots$$

where d^k denotes the **direction** in which we go at each step. For instance:

$$d^k = D^{-1}(b - Ax^k), \quad d^k = (D + L)^{-1}(b - Ax^k)$$

for the Jacobi and Gauss-Seidel methods, respectively. To improve these kind of iterations, we have two possiblities:

- Introducing a relaxation (or so-called damping) parameter $\omega^k > 0$ (possibly adapted at each step) such that

$$x^{k+1} = x^k + \omega^k d^k,$$

and/or to improve the search direction d^k such that we reduce the error as best as possible. We restrict our attention to positive definite matrices as they appear in the discretization of elliptic PDEs studied previously in this section. A key point is another view on the problem by regarding it as a minimization problem for which $Ax = b$ is the first-order necessary condition and consequently the sought solution. Imagine for simplicity that we want to minimize $f(x) = \frac{1}{2}ax^2 - bx$. The first-order necessary condition is nothing else than the derivative $f'(x) = ax - b$. We find a possible minimum via $f'(x) = 0$, namely

$$ax - b = 0 \quad \Rightarrow \quad x = a^{-1}b, \quad \text{if } a \neq 0.$$

That is exactly the same how we would solve a linear matrix system $Ax = b$. By regarding it as a minimum problem we understand better the purpose of our derivations: How does minimizing a function $f(x)$ work in terms of an iteration? Well, we try to minimize f at each step k:

$$f(x^0) > f(x^1) > \ldots > f(x^k)$$

This means that the direction d^k (to determine $x^{k+1} = x^k + \omega^k d^k$) should be a descent direction. This idea can be applied to solving linear equation systems. We first define the quadratic form

$$Q(y) = \frac{1}{2}(Ay, y)_2 - (b, y)_2,$$

where $(\cdot, \cdot)$ is the Euclidian scalar product. Then, we can define

Algorithm 8.8 (Descent method - basic idea)**.** *Let $A \in \mathbb{R}^{n\times n}$ be positive definite and $x^0, b \in \mathbb{R}^n$. Then for $k = 0, 1, 2, \ldots$*

- *Compute d^k;*
- *Determine ω^k as minimum of $\omega^k = argmin\, Q(x^k + \omega^k d^k)$;*
- *Update $x^{k+1} = x^k + \omega^k d^k$.*

For instance d^k can be determined via the Jacobi or Gauss-Seidel methods.

Another possibility is the gradient method in which we use the gradient to obtain search directions d^k. This brings us to the gradient method:

Algorithm 8.9 (Gradient descent)**.** *Let $A \in \mathbb{R}^{n\times n}$ positive definite and the right hand side $b \in \mathbb{R}^n$. Let the initial guess be $x^0 \in \mathbb{R}$ and the initial search direction $d^0 = b - Ax^0$. Then $k = 0, 1, 2, \ldots$*

- *Compute the vector $r^k = Ad^k$;*
- *Compute the relaxation*

$$\omega^k = \frac{\|d_k\|_2^2}{(r^k, d^k)_2}$$

- *Update the solution vector $x^{k+1} = x^k + \omega^k d^k$.*
- *Update the search direction vector $d^{k+1} = d^k - \omega^k r^k$.*

One can show that the gradient method converges to the solution of the linear equation system $Ax = b$ (see for instance [61]).

Proposition 8.10 (Descent directions)**.** *Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite. Then, two subsequent search directions d^k and d^{k+1} of the gradient descent scheme are orthogonal, i.e., $(d^k, d^{k+1}) = 0$.*

Proof. We have

$$d^{k+1} = d^k - \omega^k r^k = d^k - \frac{(d^k, d^k)}{(Ad^k, d^k)} Ad^k.$$

Therefore,

$$(d^{k+1}, d^k) = (d^k, d^k) - \frac{(d^k, d^k)}{(Ad^k, d^k)}(Ad^k, d^k) = (d^k, d^k) - (d^k, d^k) = 0.$$

□

8.4 Conjugate gradients (a Krylov space method)

This section is copy and paste from [61][Section 7.8] and translation from German into English. The relationship in Proposition 8.10 is only true for pairwise descent directions. In general, we have $d^k \not\perp d^{k+2}$. For this reason ,the gradient descent scheme converges slowly in most cases.

8.4.1 Formulation of the CG scheme

In order to enhance the performance of gradient descent, the conjugate gradient (CG) scheme was developed. Here, the search directions $\{d^0, \dots, d^{k-1}\}$ are pairwise orthogonal. The measure of orthogonality is achievend by using the A scalar product:

$$(Ad^r, d^s) = 0 \quad \forall r \neq s$$

At step k, we seek the approximation $x^k = x^0 + \sum_{i=0}^{k-1} \alpha_i d^i$ as the minimum of all $\alpha = (\alpha_0, \dots, \alpha_{k-1})$ with respect to $Q(x^k)$:

$$\min_{\alpha \in \mathbb{R}^k} Q\left(x^0 + \sum_{i=0}^{k-1} \alpha_i d^i\right) \quad = \quad \min_{\alpha \in \mathbb{R}^k} \left\{ \frac{1}{2} \left(Ax^0 + \sum_{i=0}^{k-1} \alpha_i A d^i, x^0 + \sum_{i=0}^{k-1} \alpha_i d^i \right) - \left(b, x^0 + \sum_{i=0}^{k-1} \alpha_i d^i \right) \right\}$$

The stationary point is given by

$$0 \stackrel{!}{=} \frac{\partial}{\partial \alpha_j} Q(x^k) = \left(Ax^0 + \sum_{i=0}^{k-1} \alpha_i A d^i, d^j \right) - (b, d^j) = -\left(b - Ax^k, d^j\right), \quad j = 0, \dots, k-1.$$

Therefore, the new residual $b - Ax^k$ is perpendicular to all search directions d^j for $j = 0, \dots, k-1$. The resulting linear equation system

$$(b - Ax^k, d^j) = 0 \quad \forall j = 0, \dots, k-1 \tag{182}$$

has the feature of Galerkin orthogonality, which we know as property of FEM schemes.

While constructing the CG method, new search directions should be linearly independent of the current d^j. Otherwise, the space would not become larger and consequently, the approximation cannot be improved.

Definition 8.11 (Krylov space)**.** *We choose an initial approximation $x^0 \in \mathbb{R}^n$ with $d^0 := b - Ax^0$ the Krylov space $K_k(d^0, A)$ such that*

$$K_k(d^0, A) := \operatorname{span}\{d^0, Ad^0, \dots, A^{k-1}d^0\}.$$

Here, A^k means the k-th power of A.

It holds:

Lemma 8.12. *Let $A^k d^0 \in K_k$. Then, the solution $x \in \mathbb{R}^n$ of $Ax = b$ is an element of the k-th Krylov space $K_k(d_0, A)$.*

Proof. Let K_k be given and $x^k \in x_0 + K_k$ the best approximation, which fulfills the Galerkin equation (182) Let $r^k := b - Ax^k$. Since

$$r^k = b - Ax^k = \underbrace{b - Ax^0}_{=d^0} + A \underbrace{(x^0 - x^k)}_{\in K_k} \in d^0 + AK_k$$

it holds $r^k \in K_{k+1}$. Supposing that $K_{k+1} \subset K_k$, we obtain $r^k \in K_k$. The Galerkin equation yields $r^k \perp K_k$, from which we obtain $r^k = 0$ and $Ax^k = b$. □

If the CG scheme aborts since it cannot find new search directions, the solution is found. Let us assume that the A-orthogonal search directions $\{d^0, d^1, \dots, d^{k-1}\}$ have been found, then we can compute the next CG approximation using the basis representation $x^k = x^0 + \sum \alpha_i d^i$ and employing the Galerkin equation:

$$\left(b - Ax^0 - \sum_{i=0}^{k-1} \alpha_i A d^i, d_j \right) = 0 \quad \Rightarrow \quad (b - Ax^0, d^j) = \alpha_j (Ad^j, d^j) \quad \Rightarrow \quad \alpha_j = \frac{(d^0, d^j)}{(Ad^j, d^j)}$$

The A-orthogonal basis $\{d^0, \ldots, d^{k-1}\}$ of the Krylov space $K_k(d^0, A)$ can be computed with the Gram-Schmidt procedure. However, this procedure has a high computational cost; see e.g., [61]. A better procedure is a two-step recursion formula, which is efficient and stable:

Lemma 8.13 (Two-step recursion formula)**.** *Let $A \in \mathbb{R}^{n\times n}$ symmetric positive definite and $x^0 \in \mathbb{R}^n$ and $d^0 := b - Ax^0$. Then, for $k = 1, 2, \ldots$, the iteration*

$$r^k := b - Ax^k, \quad \beta_{k-1} := -\frac{(r^k, Ad^{k-1})}{(d^{k-1}, Ad^{k-1})}, \quad d^k := r^k - \beta_{k-1} d^{k-1}$$

constructs an A-orthogonal basis with $(Ad^r, d^s) = 0$ for $r \neq s$. Here x^k in step k defines the new Galerkin solution $(b - Ax^k, d^j) = 0$ for $j = 0, \ldots, k-1$.

Proof. See [61]. □

We collect now all ingredients to construct the CG scheme. Let x^0 be an initial guess and $d^0 := b - Ax^0$ the resulting defect. Suppose that $K_k := \operatorname{span}\{d^0, \ldots, d^{k-1}\}$ and $x^k \in x^0 + K_k$ and $r^k = b - Ax^k$ have been computed. Then, we can compute the next iterate d^k according to Lemma 8.13:

$$\beta_{k-1} = -\frac{(r^k, Ad^{k-1})}{(d^{k-1}, Ad^{k-1})}, \quad d^k = r^k - \beta_{k-1} d^{k-1}. \tag{183}$$

For the new coefficient α_k in $x^{k+1} = x^0 + \sum_{i=0}^{k} \alpha_i d^i$ holds with testing in the Galerkin equation (182) with d^k:

$$\left(\underbrace{b - Ax^0}_{=d^0} - \sum_{i=0}^{k} \alpha_i A d^i, d^k \right) = (b - Ax^0, d^k) - \alpha_k (Ad^k, d^k) = (b - Ax^0 + \underbrace{A(x^0 - x^k)}_{\in K_k}, d^k) - \alpha_k (Ad^k, d^k).$$

That is

$$\alpha_k = \frac{(r^k, d^k)}{(Ad^k, d^k)}, \quad x^{k+1} = x^k + \alpha_k d^k. \tag{184}$$

This allows to compute the new defect r^{k+1}:

$$r^{k+1} = b - Ax^{k+1} = b - Ax^k - \alpha_k Ad^k = r^k - \alpha_k Ad^k \tag{185}$$

We summarize (183 − 185) and formulate the classical CG scheme:

Algorithm 8.14. *Let $A \in \mathbb{R}^{n\times n}$ symmetric positive definite and $x^0 \in \mathbb{R}^n$ and $r^0 = d^0 = b - Ax^0$ be given. Iterate for $k = 0, 1, \ldots$:*

1. $\alpha_k = \frac{(r^k, d^k)}{(Ad^k, d^k)}$
2. $x^{k+1} = x^k + \alpha_k d^k$
3. $r^{k+1} = r^k - \alpha_k Ad^k$
4. $\beta_k = \frac{(r^{k+1}, Ad^k)}{(d^k, Ad^k)}$
5. $d^{k+1} = r^{k+1} - \beta_k d^k$

Without round-off errors, the CG scheme yields after (at most) n steps the solution of a n-dimensional problem and is in this sense a direct method rather than an iterative scheme. However, in practice for huge n, the CG scheme is usually stopped earlier, yiedling an approximate solution.

Proposition 8.15 (CG as a direct method)**.** *Let $x^0 \in \mathbb{R}^n$ be any initial guess. Assuming no round-off errors, the CG scheme terminates after (at most) n steps with $x^n = x$. At each step, we have:*

$$Q(x^k) = \min_{\alpha \in \mathbb{R}} Q(x^{k-1} + \alpha d^{k-1}) = \min_{y \in x^0 + K_k} Q(y)$$

i.e.,

$$\|b - Ax^k\|_{A^{-1}} = \min_{y \in x^0 + K_k} \|b - Ay\|_{A^{-1}}$$

with the norm

$$\|x\|_{A^{-1}} = (A^{-1}x, x)_2^{\frac{1}{2}}.$$

Proof. That the CG scheme is a direct scheme follows from Lemma 8.12.

The iterate is given by

$$Q(x^k) = \min_{y \in x^0 + K_k} Q(y),$$

with is equivalent to (182). The ansatz

$$x^k = x^0 + \sum_{k=0}^{k-1} \alpha_k d^{k-1} = x^0 + \underbrace{y^{k-1}}_{\in K_{t-1}} + \alpha_{t-1} d^{k-1}$$

yields

$$(b - Ax^k, d^j) = (b - Ay^{k-1}, d_j) - \alpha_{t-1}(Ad^{k-1}, d^j) = 0 \quad \forall j = 0, \ldots, t-1$$

that is

$$(b - Ay^{k-1}, d_j) = 0 \; \forall j = 0, \ldots, t-2,$$

and therefore $y^{k-1} = x^{k-1}$ and

$$Q(x^k) = \min_{\alpha \in \mathbb{R}} Q(x^{k-1} + \alpha d^{k-1}).$$

Finally, employing symmetry $A = A^T$, we obtain:

$$\begin{aligned} \|b - Ay\|_{A^{-1}}^2 = (A^{-1}[b - Ay], b - Ay)_2 &= (Ay, y)_2 - (A^{-1}b, Ay)_2 - (y, b)_2 \\ &= (Ay, y)_2 - 2(b, y)_2, \end{aligned}$$

i.e., the relationship $\|b - Ay\|_{A^{-1}}^2 = 2Q(y)$. □

Remark 8.16 (The CG scheme as iterative scheme)**.** *As previosously mentioned, in pratice the CG scheme is (always) used as iterative method rather than a direct method. Due to round-off errors the search directions are never* 100% *orthogonal.*

8.4.2 Convergence analysis of the CG scheme

We now turn our attention to the convergence analysis, which is a nontrivial task. The key is the following characterization of one iteration $x^k = x^0 + K_k$ by

$$x^k = x^0 + p_{k-1}(A)d^0,$$

where $p_{k-1} \in P_{k-1}$ is a polynomial in A:

$$p_{k-1}(A) = \sum_{i=0}^{k-1} \alpha_i A^i$$

The characterization as minimization of Proposition 8.15 can be written as:

$$\|b - Ax^k\|_{A^{-1}} = \min_{y \in x^0 + K_k} \|b - Ay\|_{A^{-1}} = \min_{q \in P_{k-1}} \|b - Ax^0 - Aq(A)d^0\|_{A^{-1}}.$$

When we employ the $\|\cdot\|_A$ norm, we obtain with $d^0 = b - Ax^0 = A(x - x^0)$

$$\|b - Ax^k\|_{A^{-1}} = \|x - x^k\|_A = \min_{q \in P_{k-1}} \|(x - x^0) - q(A)A(x - x^0)\|_A,$$

that is

$$\|x - x^k\|_A = \min_{q \in P_{k-1}} \|[I - q(A)A](x - x^0)\|_A.$$

In the sense of the best approximation property (recall e.g., Section 6.6), we can formulate this task as:

$$p \in P_{k-1}: \quad \|[I - p(A)A](x - x^0)\|_A = \min_{q \in P_{k-1}} \|[I + q(A)A](x - x^0)\|_A. \tag{186}$$

The characterization as best approximation is key in the convergence analysis of the CG scheme. Let $q(A)A \in P_k(A)$. We seek a polynomial $q \in P_k$ with $q(0) = 1$, such that

$$\|x^k - x\|_A \leq \min_{q \in P_k,\, q(0)=1} \|q(A)\|_A \|x - x^0\|_A. \tag{187}$$

The convergence of the CG method is related to the fact whether we can construct a polynomial $q \in P_k$ with $p(0) = 1$ such that the A norm is as small as possible. First, we have:

Lemma 8.17 (Bounds for matrix polynomials)**.** *Let $A \in \mathbb{R}^{n \times n}$ symmetric positive definite with the eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$, and $p \in P_k$ a polynomials with $p(0) = 1$:*

$$\|p(A)\|_A \leq M, \quad M := \min_{p \in P_k,\, p(0)=1} \sup_{\lambda \in [\lambda_1, \lambda_n]} |p(\lambda)|.$$

Proof. See [61]. □

Employing the previous result and the error estimate (187), we can now derive a convergence result for the CG scheme.

Proposition 8.18 (Convergence of the CG scheme)**.** *Let $A \in \mathbb{R}^{n \times n}$ be symmetric positive definite. Let $b \in \mathbb{R}^n$ a right hand side vector and let $x^0 \in \mathbb{R}^n$ be an initial guess. Then:*

$$\|x^k - x\|_A \leq 2 \left(\frac{1 - 1/\sqrt{\kappa}}{1 + 1/\sqrt{\kappa}} \right)^k \|x^0 - x\|_A, \quad k \geq 0,$$

with the spectral condition $\kappa = \mathrm{cond}_2(A)$ of the matrix A.

Remark 8.19. *We see immediately that a large condition number $\kappa \gg 1$ yields*

$$\frac{1 - 1/\sqrt{\kappa}}{1 + 1/\sqrt{\kappa}} \to 1$$

and deteriorates significantly the convergence rate of the CG scheme. This is the key reason why preconditioners of the form $P^{-1} \approx A^{-1}$ are introduced that re-scale the system; see Section 8.5:

$$\underbrace{P^{-1}A}_{\approx I} x = P^{-1} b.$$

Computations to substantiate these findings are provided in Section 8.7.

Proof. Of Prop. 8.18.
From the previous lemma and the estimate (187) it follows

$$\|x^k - x\|_A \leq M \|x^0 - x\|_A$$

with

$$M = \min_{q \in P_k,\, q(0)=1} \max_{\lambda \in [\lambda_1, \lambda_n]} |q(\lambda)|.$$

We have to find a sharp estimate of the size of M. That is to say, we seek a polynomial $q \in P_k$ which takes at the origin the value 1, i.e., $q(0) = 1$ and which simultaneously has values near 0 in the maximum norm in the interval $[\lambda_1, \lambda_n]$. To this end, we work with the Tschebyscheff approximation (see e.g., [61] and references therein for the original references). We seek a best approximation $p \in P_k$ of the zero function on $[\lambda_1, \lambda_n]$. Such a polynomial should have the property $p(0) = 1$. For this reason, the trivial solution $p = 0$ is not valid. A Tschebyscheff polynomial reads:

$$T_k = \cos\big(k \arccos(x)\big)$$

and has the property:

$$2^{-k-1} \max_{[-1,1]} |T_k(x)| = \min_{\alpha_0,\ldots,\alpha_{k-1}} \max_{[-1,1]} |x^k + \sum_{i=0}^{k-1} \alpha_i x^k|.$$

We choose now the transformation:

$$x \mapsto \frac{\lambda_n + \lambda_1 - 2t}{\lambda_n - \lambda_1}$$

and obtain with

$$p(t) = T_k\left(\frac{\lambda_n + \lambda_1 - 2t}{\lambda_n - \lambda_1}\right) T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)^{-1}$$

a polynomial of degree k, which is minimal on $[\lambda_1, \lambda_n]$ and can be normalized by

$$p(0) = 1.$$

It holds:

$$\sup_{t\in[\lambda_1,\lambda_n]} |p(t)| = T_k\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)^{-1} = T_k\left(\frac{\kappa+1}{\kappa-1}\right)^{-1} \tag{188}$$

with the spectral condition:

$$\kappa := \frac{\lambda_n}{\lambda_1}.$$

We now employ the Tschebyscheff polynomials outside of $[-1, 1]$:

$$T_n(x) = \frac{1}{2}\Big[(x + \sqrt{x^2-1})^n + (x - \sqrt{x^2-1})^n\Big]$$

For $x = \frac{\kappa+1}{\kappa-1}$, it holds:

$$\frac{\kappa+1}{\kappa-1} + \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} = \frac{\kappa + 2\sqrt{\kappa} + 1}{\kappa - 1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}$$

and therefore

$$\frac{\kappa+1}{\kappa-1} - \sqrt{\left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}.$$

Using this relationship, we can estimate (188):

$$T_k\left(\frac{\kappa+1}{\kappa-1}\right) = \frac{1}{2}\Big[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\Big] \geq \frac{1}{2}\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k$$

It follows that

$$\sup_{t\in[\lambda_1,\lambda_n]} T_k\left(\frac{\kappa+1}{\kappa-1}\right)^{-1} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k = 2\left(\frac{1-\frac{1}{\sqrt{\kappa}}}{1+\frac{1}{\sqrt{\kappa}}}\right)^k.$$

This finishes the proof. □

8.5 Preconditioning

This section is copy and paste from [61][Section 7.8.1] and translation from German into English.

The rate of convergence of iteratice schemees depends on the condition number of the system matrix. For instance, we recall that for second-order operators (such as Laplace) we have a dependence on the mesh size $O(h^{-2}) = O(N)$ (in 2D) (see Section 12 for the relation between h and N). For the CG scheme it holds:

$$\rho_{CG} = \frac{1 - \frac{1}{\sqrt{\kappa}}}{1 + \frac{1}{\sqrt{\kappa}}} = 1 - \frac{2}{\sqrt{\kappa}} + O\left(\frac{1}{\kappa}\right).$$

Preconditioning reformulates the original system with the goal of obtaining a moderate condition number for the modified system. Let $P \in \mathbb{P}^{n\times n}$ be a matrix with

$$P = KK^T.$$

Then:

$$Ax = b \quad \Leftrightarrow \quad \underbrace{K^{-1}A(K^T)^{-1}}_{=:\tilde{A}} \underbrace{K^T x}_{=:\tilde{x}} = \underbrace{K^{-1}b}_{=:\tilde{b}},$$

which is

$$\tilde{A}\tilde{x} = \tilde{b}.$$

In the case of

$$\mathrm{cond}_2(\tilde{A}) \ll \mathrm{cond}_2(A)$$

and if the application of K^{-1} is cheap, then the consideration of a preconditioned system $\tilde{A}\tilde{x} = \tilde{b}$ yields a much faster solution of the iterative scheme. The condition $P = KK^T$ is necessary such that the matrix $\tilde{A}$ keeps its symmetry.

The preconditioned CG scheme (PCG) can be formulated as:

Algorithm 8.20. *Let $A \in \mathbb{R}^{n\times n}$ symmetric positive definite and $P = KK^T$ a symmetric preconditioner. Choosing an initial guess $x^0 \in \mathbb{R}^n$ yields:*

1. $r^0 = b - Ax^0$
2. $Pp^0 = r^0$
3. $d^0 = p^0$
4. *For* $k = 0, 1, \ldots$
   a) $\alpha_k = \frac{(r^k, d^k)}{(Ad^k, d^k)}$
   b) $x^{k+1} = x^k + \alpha_k d^k$
   c) $r^{k+1} = r^k - \alpha_k A d^k$
   d) $Pp^{k+1} = r^{k+1}$
   e) $\beta_k = \frac{(r^{k+1}, p^{k+1})}{(r^k, g^k)}$
   f) $d^{k+1} = p^{k+1} + \beta_k d^k$

At each step, we have as additional cost the application of the preconditioner P. We recall that P allows the decomposition into K and K^T even if they are not explicitely used.

We seek P such that

$$P \approx A^{-1}.$$

On the other hand

$$P \approx I,$$

such that the construction of P is not too costly. Obviously, these are two conflicting requirements. Typical preconditioners are:

- *Jacobi preconditioning*
  We choose $P \approx D^{-1}$, where D is the diagonal part of A. It holds

  $$D = D^{\frac{1}{2}}(D^{\frac{1}{2}})^T,$$

  which means that for $D_{ii} > 0$, this preconditioner is admissible. For the preconditioned matrix, it holds

  $$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad \Rightarrow \quad \tilde{a}_{ii} = 1$$

- *SSOR preconditioning*

  The SSOR scheme is a symmetric variant of the SOR method (successive over-relaxation) and is based on the decomposition:

  $$P = (D + \omega L)D^{-1}(D + \omega R) = \underbrace{(D^{\frac{1}{2}} + \omega L D^{-\frac{1}{2}})}_{K}\underbrace{(D^{\frac{1}{2}} + \omega D^{-\frac{1}{2}} R)}_{=K^T}.$$

  For instance, for the Poisson problem, we can find an optimal ω (which is a non-trivial task) such that

  $$\mathrm{cond}_2(\tilde{A}) = \sqrt{\mathrm{cond}_2(A)}$$

  can be shown. Here, the convergence improves significantly. The number of necessary steps to achieve a given error reduction by a factor of ϵ improves to

  $$t_{CG}(\epsilon) = \frac{\log(\epsilon)}{\log(1 - \kappa^{-\frac{1}{2}})} \approx -\frac{\log(\epsilon)}{\sqrt{\kappa}}, \quad \tilde{t}_{CG}(\epsilon) = \frac{\log(\epsilon)}{\log(1 - \kappa^{-\frac{1}{4}})} \approx \frac{\log(\epsilon)}{\sqrt[4]{\kappa}}.$$

  Rather than having 100 steps, we only need 10 steps for instance, in case an optimal ω can be found.

8.6 Comments on other Krylov space methods such as GMRES and BiCGStab

For non-symmetric systems the CG method cannot be used anymore. In practice in most cases, one works with the GMRES (generalized minimal residuals) method or the BiCGStab (biconjugate gradient stabilized) scheme. Both also belong to Krylov space methods. Typically, both schemes need to be preconditioned in order to yield satisfactory results.

Let $A \in \mathbb{R}^{n \times n}$ be a regular matrix, but not necessarily symmetric. A symmetric version of the problem

$$Ax = b$$

can be achieved by multiplication with A^T:

$$A^T A x = A^T b.$$

The matrix $B = A^T A$ is positive definite since

$$(Bx, x)_2 = (A^T A x, x)_2 = (Ax, Ax)_2 = \|Ax\|_2.$$

In principle, we could now apply the CG scheme to $A^T A$. Instead of one matrix-vector multiplication, we would need two such multiplications per step. However, using $A^T A$, the convergence rate will deteriorate since

$$\kappa(B) = \mathrm{cond}_2(A^T A) = \mathrm{cond}_2(A)^2.$$

For this reason, the CG scheme is not really an option.

The GMRES method, *generalized minimal residual* transfers the idea of the CG scheme to general matrices. Using the Krylov space

$$K_k(d^0, A) = \{d^0, Ad^0, \ldots, A^{k-1}d^0\},$$

we first construct an orthonormal basis. Employing the GMRES method, orthogonality will be achieved with respect to the euclidian scalar product:

$$(d^i, d^j)_2 = \delta_{ij}, \quad i, j = 0, \ldots, k-1$$

To work with the A scalar product does not go anymore because A may not be symmetric.

The approximation $x^k \in x^0 + K_k$ is computed with the help of the Galerkin equation:

$$(b - Ax^k, Ad^j)_2 = 0, \quad j = 0, \ldots, k-1. \tag{189}$$

The computational cost is higher because a two-step procedure as in the CG scheme cannot be applied for the GMRES scheme. Therefore, the orthogonal basis is constructed with the help of the Arnoldi procedure.

The weak point of the GMRES method is the increasing computational cost with increasing iteration indices because the orthogonalization needs to be re-done until the first step. In practice, often a restart is employed and only a fixed number of search directions N_o is saved. After N_o iterations, a new Krylov space is constructed.

8.7 Numerical tests

We continue the numerical tests from Section 6.13 and consider again the Poisson problem in 2D and 3D on the unit square and with force $f = 1$ and homogeneous Dirichlet conditions. We use as solvers:

- CG
- PCG with SSOR preconditioning and $\omega = 1.2$
- GMRES
- GMRES with SSOR preconditioning and $\omega = 1.2$
- BiCGStab
- BiCGStab with SSOR preconditioning and $\omega = 1.2$

The tolerance is chosen as $TOL = 1.0e-12$. We also run on different mesh levels in order to show the dependency on n.

Dimension	Elements	DoFs	CG	PCG	GMRES	GMRES prec.	BiCGStab	BiCGStab prec.
2	256	289	23	19	23	18	16	12
2	1024	1089	47	33	83	35	33	21
2	4096	4225	94	60	420	78	66	44
3	4096	4913	25	19	25	21	16	11
3	32768	35937	51	32	77	38	40	23
3	262144	274625	98	57	307	83	69	46

Three main messages:

- GMRES performs worst.
- PCG performs better than pure CG, but less significant than one would wish.
- BiCGStab performs best - a bit surprising.

9 Applications in solid mechanics: linearized elasticity and briefly Stokes

In this chapter, we undertake a short excursus to important prototype applications: linearized elasticity (solid mechanics) and the Stokes problem (fluid mechanics). Both models lead to systems of equations since they are vector-valued, but are still linear and consequently fit into the lines of our previous developments. We try to provide an introduction that touches the aspects, which can be understood with the help of our previous developments. For instance, linearized elasticity is still of elliptic type and we show, how the well-posedness can be proven using Lax-Milgram (i.e., Riesz since the problem is symmetric). On the other hand, the maximum principle does not hold anymore. These notes cannot provide all possible details since they fill entire books such as [19] and [65]. In regard of fluid mechanics, the Navier-Stokes equations have been introduced in Section 7.14 to demonstrate mesh adaptivity with respect to drag forces as goal functional.

9.1 Modeling

Since linearized elasticity is vector-valued, we design now the space H^1 for three solution components $\hat{u}_x, \hat{u}_y, \hat{u}_z$:

$$\hat{u} = (\hat{u}_x, \hat{u}_y, \hat{u}_z) \in [H^1(\widehat{\Omega})]^3$$

Specifically, we define:

$$\hat{V}_s^0 := [H_0^1(\widehat{\Omega})]^3.$$

Remark 9.1. *In linearized elasticity we deal with small displacements and consequently the two principle coordinate systems, Lagrangian and Eulerian are identified; we also refer the reader to Remark 4.6. For large displacements, one needs to distingish between a* ***reference configuration*** *in which the actual computations are carried out. This reference configuration is indicated in a 'hat' notation, i.e., $\widehat{\Omega}$. Via a transformation, the solution is mapped to the* ***current (i.e., physical or deformed) configuration*** *Ω.*

Problem 9.2 (Stationary linearized elasticity)**.** *Let $\Omega \subset \mathbb{R}^d, d = 3$. Given $\hat{f} : \Omega \to \mathbb{R}^d$, find a vector-valued displacement $\hat{u} \in \hat{V}_s^0 = H_0^1$ such that*

$$(\widehat{\Sigma}_{lin}, \widehat{\nabla}\hat{\varphi}) = (\hat{f}, \hat{\varphi}) \quad \forall \hat{\varphi} \in \hat{V}_s^0,$$

where

$$\widehat{\Sigma}_{lin} = 2\mu\widehat{E}_{lin} + \lambda tr\widehat{E}_{lin}\hat{I}.$$

Here, $\widehat{\Sigma}_{lin}$ is a matrix and specified below. Furthermore, the gradient $\widehat{\nabla}\hat{\varphi}$ is a matrix. Consequently, the scalar product is based on the Frobenius Definition 6.12. The material parameters $\mu, \lambda > 0$ are the so-called Lamé parameters. Finally $tr(\cdot)$ is the trace operator. For further details on the entire model, we refer to Ciarlet [19].

Definition 9.3 (Green-Lagrange strain tensor $\widehat{E}$)**.**

$$\widehat{E} = \frac{1}{2}(\widehat{C} - \hat{I}) = \frac{1}{2}(\widehat{F}^T\widehat{F} - \hat{I}) = \frac{1}{2}(\widehat{\nabla}\hat{u} + \widehat{\nabla}\hat{u}^T + \widehat{\nabla}\hat{u} \cdot \widehat{\nabla}\hat{u}^T),$$

which is again symmetric and positive definite for all $\hat{x} \in \widehat{\Omega}$ since $\widehat{C}$ and of course, $\hat{I}$ have these properties. $\diamond$

Performing geometric linearization, which is reasonable for example when $\|\widehat{\nabla}\hat{u}\| \ll 1$, we can work with the linearized Green-Lagrange strain tensor

Definition 9.4 (Linearized Green-Lagrange strain tensor $\widehat{E}_{lin}$)**.**

$$\widehat{E}_{lin} = \frac{1}{2}(\widehat{\nabla}\hat{u} + \widehat{\nabla}\hat{u}^T).$$

$\diamond$

Proposition 9.5. *Let the following linear boundary value problem be given:*

$$-div(\widehat{\Sigma}_{lin}) = f \quad in\ \widehat{\Omega}, \tag{190}$$

$$\hat{u} = 0 \quad on\ \partial\widehat{\Omega}_D, \tag{191}$$

$$\widehat{\Sigma}_{lin}\hat{n} = \hat{g} \quad on\ \partial\widehat{\Omega}_N. \tag{192}$$

Finding a solution $\hat{u}$ of this strong form is formally equivalent to finding a solution of Problem 9.2.

Proof. We employ Green's formula for any sufficiently smooth tensor field $\widehat{\Sigma}$ and vector field $\hat{\varphi}$, we obtain

$$\int_{\widehat{\Omega}} \widehat{\nabla} \cdot \widehat{\Sigma} \cdot \hat{\varphi}\, d\hat{x} = -\int_{\widehat{\Omega}} \widehat{\Sigma} : \widehat{\nabla}\hat{v}\, d\hat{x} + \int_{\partial\widehat{\Omega}_N} \widehat{\Sigma}\hat{n} \cdot \hat{\varphi}\, d\hat{s} = -\int_{\widehat{\Omega}} \widehat{\Sigma} : \widehat{E}_{lin}\, d\hat{x} + \int_{\partial\widehat{\Omega}_N} \widehat{\Sigma}\hat{n} \cdot \hat{\varphi}\, d\hat{s}.$$

The last equal sign is justified with linear algebra arguments, i.e., for a symmetric matrix A is holds:

Definition 9.6 (Proof is homework)**.** *For the Frobenius scalar product, and $A = A^T$, it holds*

$$A : B = A : \frac{1}{2}(B + B^T).$$

Furthermore, on the boundary part $\partial\widehat{\Omega}_D$ the vector field $\hat{\varphi}$ vanishes by definition. Thus, the first direction is shown. Conversely, we now assume that the variational equations are satisfied; namely,

$$\int_{\widehat{\Omega}} \widehat{\Sigma} : \widehat{\nabla}\hat{\varphi}\, d\hat{x} = \int_{\widehat{\Omega}} \hat{f} \cdot \hat{\varphi}\, d\hat{x}$$

if $\hat{\varphi} = 0$ on all $\partial\Omega$. By Green's formula we now obtain

$$\int_{\widehat{\Omega}} \widehat{\Sigma} : \widehat{\nabla}\hat{\varphi}\, d\hat{x} = -\int_{\widehat{\Omega}} \widehat{\nabla} \cdot \widehat{\Sigma}\, d\hat{x}.$$

Putting both pieces together and taking into account that the integrals do hold on arbitrary volumes yields

$$-\widehat{\nabla} \cdot \widehat{\Sigma} = \hat{f} \quad \text{in } \widehat{\Omega}.$$

Considering now the Neumann boundary, we use again Green's formula to see

$$\int_{\partial\widehat{\Omega}_N} \widehat{\Sigma}\hat{n} \cdot \hat{\varphi}\, ds = \int_{\partial\widehat{\Omega}_N} \hat{g}\, ds.$$

This holds for arbitrary boundary parts $\partial\widehat{\Omega}_N$ such that $\widehat{\Sigma}\hat{n} = \hat{g}$ can be inferred and concluded the second part of the proof. □

9.2 Well-posedness

Even so that linearized elasticity is an elliptic problem and has much in common with the scalar-valued Poisson problem, establishing existence and uniquenss for the stationary version is a non-trivial task. Why? Let us work with Problem 9.2 in 3D and since this problem is linear we aim to apply the Riesz representation theorem[7] (or the Lax-Milgram lemma - see Theorem 6.130) and need to check the assumptions:

- Determine in which space $\hat{V}$ we want to work;
- The form $A(\hat{u}, \hat{\varphi})$ is symmetric or non-symmetric
- The form $A(\hat{u}, \hat{\varphi})$ is continuous bilinear w.r.t. to the norm of $\hat{V}$;
- The bilinear form $A(\hat{u}, \hat{\varphi})$ is V-elliptic;

[7] Here, $A(\hat{u}, \hat{\varphi}) = (\widehat{\Sigma}_{lin}, \widehat{\nabla}\hat{\varphi})$ is symmetric and the Riesz representation 6.133 is sufficient for existence and uniqeness!

- The right-hand side functional $\hat{f}$ is a continuous linear form.

The ingredients to check are:

- From which space must $\hat{f}$ be chosen such that it is continuous? (This leads to Sobolev embedding theorems [19, 24]);
- How do we check the V-ellipticity? (This requires the famous Korn inequality as it is stated in a minute below).

For proofing existence of linearized elasticity, we need to check as second contition the V-ellipticity which is 'easy for scalar-valued problems' but non-trivial in the case of vector-valued elasticity. The resulting inequalities are named after Korn (1906):

Theorem 9.7 (1st Korn inequality)**.** *Let us assume displacement Dirichlet conditions on the entire boundary $\partial\Omega$. For vector-fields $\hat{u} \in H_0^1(\widehat{\Omega})^3$ it holds:*

$$\|\nabla \hat{u}\| \leq c_{Korn} \|\widehat{E}_{lin}(\hat{u})\|.$$

where $\widehat{E}_{lin}(\hat{u}) := \frac{1}{2}(\widehat{\nabla}\hat{u}_s + \widehat{\nabla}\hat{u}_s^T) \in L^2(\widehat{\Omega}_s)$.

Proof. See for example [12, 19, 52]. □

The more general form is as follows:

Theorem 9.8 (2nd Korn inequality)**.** *Let a part of the boundary of Neumann type, i.e., $\partial\Omega_N \neq 0$ and let the entire boundary be of class C^1, or a polygon, or of Lipschitz-type. For vector-fields $\hat{u} \in H^1(\widehat{\Omega})^3$ and $\widehat{E}_{lin} \in L^2$ it holds*

$$\|\hat{u}_s\|_{H^1(\widehat{\Omega}_s)} \leq C_{Korn}\big(|\hat{u}_s|^2_{L^2(\widehat{\Omega}_s)} + |\widehat{E}_{lin}|^2_{L^2(\widehat{\Omega}_s)}\big)^{1/2} \quad \forall \hat{u}_s \in H^1(\widehat{\Omega}_s),$$

and a positive constant C_K and secondly,

$$u \mapsto \big(|\hat{u}_s|^2_{L^2(\widehat{\Omega}_s)} + |\widehat{E}_{lin}|^2_{L^2(\widehat{\Omega}_s)}\big)^{1/2}$$

is a norm that is equivalent to $\|\cdot\|_{H^1}$.

Theorem 9.9 (Existence of a weak solution of linearized elasticity)**.** *Let $\widehat{\Omega} \subset \mathbb{R}^3$ and $\widehat{\Gamma}_D > 0$ and $\widehat{\Gamma}_N > 0$ Dirichlet and Neumann boundary respectively. Let the Lamé constants be $\mu > 0$ and $\lambda > 0$ and $\hat{f} \in L^{6/5}(\Omega)$ and $\hat{g} \in L^{4/3}(\widehat{\Gamma}_N)$ be given. Furthermore, let*

$$\hat{V} := \{\hat{\varphi} \in H^1(\widehat{\Omega}) |\, \hat{\varphi} = 0 \text{ on } \widehat{\Gamma}_D\}.$$

Then, there exists a unique element $\hat{u} \in \hat{V}$ such that

$$A(\hat{u}, \hat{\varphi}) = F(\hat{\varphi}) \quad \forall \hat{\varphi} \in \hat{V}.$$

Here, the bilinear form and the right hand side functional are given by

$$A(\hat{u}, \hat{\varphi}) = (\widehat{\Sigma}_{lin}, \widehat{\nabla}\hat{\varphi}), \quad \text{and} \quad F(\hat{\varphi}) = (\hat{f}, \hat{\varphi}) + \langle \hat{g}, \hat{\varphi} \rangle, \tag{193}$$

where the linearized STVK model is given by

$$\widehat{\Sigma}_{lin} := 2\mu \widehat{E}_{lin} + \lambda tr\widehat{E}_{lin}\hat{I}.$$

The function $\hat{g}$ is a prescribed traction condition on $\widehat{\Gamma}_N$; namely:

$$[2\mu \widehat{E}_{lin} + \lambda tr\widehat{E}_{lin} I]\, \hat{n} = \hat{g}.$$

Proof. The proof is as follows:

- **Continuity of the right hand side functional $F(\cdot)$.** In order to apply Hölder's inequality, we first check the correct Sobolev embedding:

$$H^1 = W^{1,2} \hookrightarrow L^{p^*} \quad \text{with } \frac{1}{p^*} = \frac{1}{p} - \frac{m}{d}$$

for $m < \frac{d}{p}$. Since we work in 3D, $d = 3$. Furthermore, $m = 1, p = 2$ in $W^{m,p}$. It can be inferred that consequently, $p^* = 6$. For the continuity of F, we calculate with Hölder's inequality:

$$|(\hat{f}, \hat{\varphi})| \leq \|\hat{f}\|_p \, \|\hat{\varphi}\|_{q=6}$$

where $\frac{1}{p} + \frac{1}{q} = 1$. Since, $q = 6$, this yields $p = \frac{6}{5}$, i.e., $\hat{f} \in L^{6/5}$. In the same spirit, we obtain $\hat{g} \in L^{4/3}$ while additionally employing the trace inequality. Let us verify this claim: From trace theorems, e.g., [19], p. 280, we know that for $1 \leq p < \infty$ we have

$$tr \in L(W^{1,p}, L^{p^*}) \quad \Leftrightarrow \quad \|\cdot\|_{L^{p^*}} \leq c\|\cdot\|_{W^{1,p}},$$

with $\frac{1}{p^*} = \frac{1}{p} - \frac{1}{(d-1)}(\frac{p-1}{p})$ if $1 \leq p < d$. As before and still, $d = 3$. This means $1 \leq p \leq 2$. Of course we want to work again with $W^{1,2}$, $p = 2$. Then,

$$\frac{1}{p^*} = \frac{1}{p} - \frac{1}{(d-1)}(\frac{p-1}{p}) = \frac{1}{2} - \frac{1}{(3-1)}(\frac{2-1}{2}) = \frac{1}{4}.$$

Consequently, $p^* = 4$. Let us estimate with Hölder's inequality and the trace theorem:

$$\int_{\widehat{\Gamma}_N} \hat{g}\hat{\varphi} \, ds \leq \|\hat{g}\|_{L^q(\widehat{\Gamma}_N)} \|\hat{\varphi}\|_{L^4(\widehat{\Gamma}_N)} \leq \|\hat{g}\|_{L^q(\widehat{\Gamma}_N)} \|\hat{\varphi}\|_{W^{1,2}(\widehat{\Omega})}.$$

In order to be able to apply Hölder's inequality we need to find the correct q. This is now easy with

$$\frac{1}{p} + \frac{1}{q} = 1 \quad \Rightarrow q = \frac{4}{3}.$$

Consequently, $\hat{g} \in L^{4/3}$.

- **Symmetry of $A(\hat{u}, \hat{\varphi})$.** Let us now verify if $A(\hat{u}, \hat{\varphi})$ is symmetric:

$$\begin{aligned}
A(\hat{u}, \hat{\varphi}) &= (\widehat{\Sigma}_{lin}, \widehat{\nabla}\hat{\varphi}) = (2\mu\widehat{E}_{lin}(\hat{u}) + \lambda tr(\widehat{E}_{lin}(\hat{u}))\hat{I}, \widehat{E}_{lin}(\hat{\varphi})) \\
&= 2\mu(\widehat{E}_{lin}(\hat{u}), \widehat{E}_{lin}(\hat{\varphi})) + \lambda(tr(\widehat{E}_{lin}(\hat{u}))\hat{I}, \widehat{E}_{lin}(\hat{\varphi})) \\
&= 2\mu(\widehat{E}_{lin}(\hat{\varphi}), \widehat{E}_{lin}(\hat{u})) + \lambda(tr(\widehat{E}_{lin}(\hat{\varphi}))\hat{I}, \widehat{E}_{lin}(\hat{u})) \\
&= (2\mu\widehat{E}_{lin}(\hat{\varphi}) + \lambda tr(\widehat{E}_{lin}(\hat{\varphi}))\hat{I}, \widehat{E}_{lin}(\hat{u})) \\
&= A(\hat{\varphi}, \hat{u}).
\end{aligned}$$

Here, we used the relation defined in Definition 9.6.

- **Continuity of $A(\hat{u}, \hat{\varphi})$.** We need to show that

$$|A(\hat{u}, \hat{\varphi})| \leq c\|\hat{u}\|_{H^1}\|\hat{\varphi}\|_{H^1} \quad \forall \hat{u}, \phi \text{ in } \hat{V}.$$

Let us start with the definition and apply first Cauchy's inequality:

$$\begin{aligned}
|A(\hat{u}, \hat{\varphi})| &= |(\widehat{\Sigma}_{lin}, \widehat{\nabla}\hat{\varphi})| = |(\widehat{\Sigma}_{lin}, \widehat{E}_{lin}(\hat{\varphi}))| \\
&\leq \Big(\int |\widehat{\Sigma}_{lin}|^2\Big)^{1/2} \Big(\int |\widehat{E}_{lin}(\hat{\varphi})|^2\Big)^{1/2} \\
&= \Big(\int |2\mu\widehat{E}_{lin}(\hat{u}) + \lambda tr(\widehat{E}_{lin}(\hat{u}))\hat{I}|^2\Big)^{1/2} \Big(\int |\widehat{E}_{lin}(\hat{\varphi})|^2\Big)^{1/2}.
\end{aligned}$$

We use $\|tr(\widehat{E}_{lin}(\hat{u}))\hat{I}\| \leq \|E_{lin}(\hat{u})\|$ and $|E(\hat{u})|_{L^2} \leq c\|\hat{u}\|_{H^1}$, where the latter one specifically holds for all $\hat{u} \in H^1$. Then,

$$\begin{aligned}
|A(\hat{u},\hat{\varphi})| &\leq \Big(\int |2\mu\widehat{E}_{lin}(\hat{u}) + \lambda tr(\widehat{E}_{lin}(\hat{u}))\hat{I}|^2\Big)^{1/2}\Big(\int |\widehat{E}_{lin}(\hat{\varphi})|^2\Big)^{1/2} \\
&\leq \Big(\int |2\mu\widehat{E}_{lin}(\hat{u}) + \lambda\widehat{E}_{lin}(\hat{u})|^2\Big)^{1/2}\Big(\int |\widehat{E}_{lin}(\hat{\varphi})|^2\Big)^{1/2} \\
&\leq c(\mu,\lambda)\Big(\int |\widehat{E}_{lin}|^2\Big)^{1/2}\Big(\int |\widehat{E}_{lin}(\hat{\varphi})|^2\Big)^{1/2} \\
&= c(\mu,\lambda)|E(\hat{u})|_{L^2}\,|E(\hat{\varphi})|_{L^2} \\
&\leq c_1\|\hat{u}\|_{H^1}\,\|\hat{\varphi}\|_{H^1}.
\end{aligned}$$

- **V-ellipticity.** First, we observe that

$$A(\hat{u},\hat{u}) = (2\mu\widehat{E}_{lin}(\hat{u}) + \lambda tr(\widehat{E}_{lin}(\hat{u}))\hat{I}, \widehat{E}_{lin}(\hat{u})) \geq (2\mu\widehat{E}_{lin}(\hat{u}), \widehat{E}_{lin}(\hat{u})) = 2\mu|\widehat{E}_{lin}|_{L^2}.$$

  because $\mu, \lambda > 0$. The V-ellipticity can be inferred if we can show that on the space $\hat{V}_s^0$, the semi-norm

$$\hat{u} \mapsto |\widehat{E}_{lin}|_{L^2}$$

  is a norm that is equivalant to $\|\cdot\|_{H^1}$. To do so, we proceed in two steps. We now employ the 2nd Korn inequality (first step), see Theorem 9.8. In detail: if $\widehat{\Gamma}_D > 0$ and $c > 0$ a given constant such that (the second step)

$$c^{-1}\|\hat{u}\|_1 \leq |\widehat{E}_{lin}|_{L^2} \leq c\|\hat{u}\|_1 \quad \forall \hat{u} \in H^1(\widehat{\Omega}_s),$$

  i.e., on $\hat{V}$, the mapping $u \mapsto |\widehat{E}_{lin}|$ is norm equivalent to $\|\hat{u}\|_1$. (This techniquality requires a seperate proof!)

- **Apply Riesz.** Having proven that A is symmetric, continuous and V-elliptic as well as that F is linear continuous, we have checked all assumptions of the Riesz representation Theorem 6.133 and consequently, a unique solution $\hat{u} \in \hat{V}$ does exist.

For further discussions, the reader might consult Ciarlet [19]. □

Remark 9.10. *Since this elasticity problem is symmetric, the problem can be also interpreted from the energy minimization standpoint as outlined in Theorem 6.133. In this respect, the weak form considered here corresponds to the first-order necessary condition, i.e., the weak form of the Euler-Lagrange equations.*

Remark 9.11. *For linearized elasticity, even that it is of elliptic type, the maximum principle does not hold anymore.*

9.3 Finite element discretization

Problem 9.12 (Discretized stationary linearized elasticity)**.** *Find vector-valued displacements $\hat{u}_h \in \hat{V}_h^0$ such that*

$$(\widehat{\Sigma}_{h,lin}, \widehat{\nabla}\hat{\varphi}_h) = (\hat{f}, \hat{\varphi}_h) \quad \forall \hat{\varphi}_h \in \hat{V}_h^0,$$

where

$$\widehat{\Sigma}_{h,lin} = 2\mu\widehat{E}_{h,lin} + \lambda tr\widehat{E}_{h,lin}\hat{I}.$$

and

$$\widehat{E}_{h,lin} = \frac{1}{2}(\widehat{\nabla}\hat{u}_h + \widehat{\nabla}\hat{u}_h^T).$$

As previsously discussed, the spatial discretization is based upon a space $\hat{V}_h^0$ with basis $\{\hat{\varphi}_1, \ldots, \hat{\varphi}_N\}$ where $N = dim(\hat{V}_h^0)$. We recall, that in 3D elasticity, we have three solution components such that the total dimension is $3N = [\hat{V}_h^0]^d, d = 3$.

We solve the problem:

Formulation 9.13. *Find $\hat{u}_h \in \hat{V}_h$ such that*

$$A(\hat{u}_h, \hat{\varphi}_h) = F(\hat{\varphi}_h) \quad \forall \hat{\varphi}_h \in \hat{V}_h,$$

with

$$A(\hat{u}_h, \hat{\varphi}_h) := (\widehat{\Sigma}_{h,lin}, \widehat{\nabla}\hat{\varphi}_h), \quad F(\hat{\varphi}_h) := (\hat{f}, \hat{\varphi}_h).$$

This relation does in particular hold true for each test function $\hat{\varphi}_i, i = 1, \ldots, N$:

$$A(\hat{u}_h, \hat{\varphi}_h^i) = F(\hat{\varphi}_h^i) \quad \forall \hat{\varphi}_h^i, i = 1, \ldots, N$$

The solution $\hat{u}_h$ we are seeking for is a linear combination of all test functions, i.e., $\hat{u}_h = \sum_{j=1}^N u_j \hat{\varphi}_h^j$. Inserting this relation into the bilinear form $A(\cdot, \cdot)$ yields

$$\sum_{j=1}^{N} A(\hat{\varphi}_h^j, \hat{\varphi}_h^i) u_j = F(\hat{\varphi}_h^i), \quad i = 1, \ldots, N.$$

It follows for the ingredients of the linear equation system:

$$\hat{u} = (\hat{u}_j)_{j=1}^N \in \mathbb{R}^N, \quad b = (F_i)_{i=1}^N \in \mathbb{R}^N, \quad B = A_{ij} = a(\hat{\varphi}_h^J, \hat{\varphi}_h^i).$$

The resulting linear equation systems reads:

$$Bu = b$$

Remark 9.14. *In the matrix B the rule is always as follows: the $\hat{\varphi}_h^i$ test function determines the row and the trial function $\hat{\varphi}_h^j$ the column. This does not play a role for symmetric problems (e.g., Poisson's problem) but becomes important for nonsymmetric problems such as for example Navier-Stokes (because of the convection term).*

Remark 9.15. *In the matrix, the degrees of freedom that belong to Dirichlet conditions (here only displacements since we assume Neumann conditions for the phase-field) are strongly enforced by replacing the corresponding rows and columns as usual in a finite element code.*

Example 9.16 (Laplacian in 3D)**.** *In this vector-valued problem, we have $3N$ test functions since the solution vector is an element of $\mathbb{R}^3$: $u_h = (u_h^{(1)}, u_h^{(2)}, u_h^{(3)})$. Thus in the boundary value problem*

$$\text{Find } u_h \in V_h: \quad (\nabla u_h, \nabla \varphi_h) = (f, \varphi_h) \quad \forall \varphi_h \in V_h,$$

the bilinear form is tensor-valued:

$$a_{Laplacian}(\varphi_h^j, \varphi_h^i) = \int_\Omega \nabla \varphi_h^j : \nabla \varphi_h^i \, dx = \int_\Omega \begin{pmatrix} \partial_1 \varphi_h^{1,j} & \partial_2 \varphi_h^{1,j} & \partial_3 \varphi_h^{1,j} \\ \partial_1 \varphi_h^{2,j} & \partial_2 \varphi_h^{2,j} & \partial_3 \varphi_h^{2,j} \\ \partial_1 \varphi_h^{3,j} & \partial_2 \varphi_h^{3,j} & \partial_3 \varphi_h^{3,j} \end{pmatrix} : \begin{pmatrix} \partial_1 \varphi_h^{1,i} & \partial_2 \varphi_h^{1,i} & \partial_3 \varphi_h^{1,i} \\ \partial_1 \varphi_h^{2,i} & \partial_2 \varphi_h^{2,i} & \partial_3 \varphi_h^{2,i} \\ \partial_1 \varphi_h^{3,i} & \partial_2 \varphi_h^{3,i} & \partial_3 \varphi_h^{3,i} \end{pmatrix} dx.$$

9.4 Numerical tests

9.4.1 3D

We use the above laws given Problem 10.49 and implement them again in deal.II [4, 6].

9.4.1.1 Configuration The domain Ω is spanned by $(0, 0, 0)$ and $(5, 0.5, 2)$ resulting in an elastic beam (see Figure 30). The vertical axis is the y-direction (see again Figure 30). The initial domain is four times globally refined resulting in 4096 and total 14739 DoFs (i.e., 4913 DoFs in each solution component).

9.4.1.2 Boundary conditions The specimen has six boundary planes. On the boundary $\Gamma = \{(x, y, z)|x = 0\}$ the elastic beam is fixed and can move at the other boundaries:

$$\begin{aligned} u = (u_x, u_y, u_z) = 0 \quad &\text{on } \Gamma \quad \text{(Homogeneous Dirichlet)}, \\ \Sigma_{lin} \cdot n = 0 \quad &\text{on } \partial\Omega \setminus \Gamma \quad \text{(Homogeneous Neumann)}. \end{aligned}$$

9.4.1.3 Material parameters and given data The material parameters are:

```
lame_coefficient_mu = 1.0e+6;
poisson_ratio_nu = 0.4;

lame_coefficient_lambda =  (2 * poisson_ratio_nu * lame_coefficient_mu)/
  (1.0 - 2 * poisson_ratio_nu);
```

and the force vector is prescribed as

$$f(x, y, z) = (0, -9.81, 0)^T.$$

9.4.1.4 Simulation results With a PCG scheme (see Section 8.7), we need 788 PCG iterations to obtain convergence of the linear solver. The resulting solution is displayed in Figure 30.

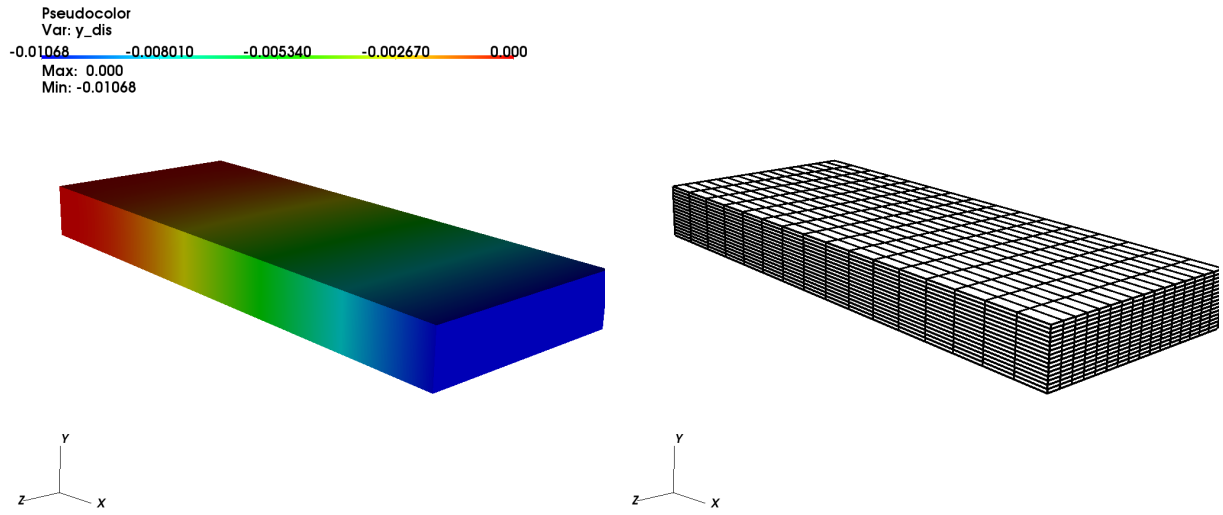

Figure 30: 3D linearized elasticity: u_y solution and mesh. The elastic beam is attached at $x = 0$ in the $y - z$ plane by $u_x = u_y = u_z = 0$. All other boundaries are traction free (homogeneous Neumann conditions).

9.4.2 2D test with focus on the maximum principle

In the corresonding 2D test (because it is easier to show), we demonstrate the violation of the maximum principle.

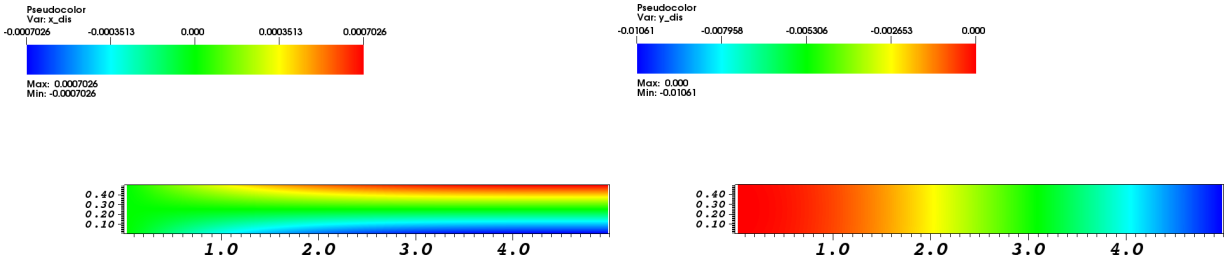

Figure 31: 2D linearized elasticity. The same test as for 3D. At left, the u_x solution is displayed. At right, the u_y solution is displayed.

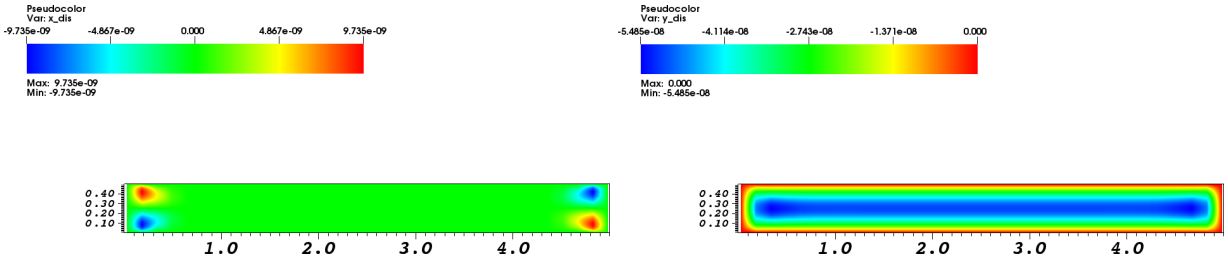

Figure 32: 2D linearized elasticity: all boundaries are subject to homogeneous Dirichlet conditions. In the u_x solution (left), we observe that the minimum and maximum are obtained inside the domain, which is a violation of the maximum principle.

9.5 Stokes - FEM discretization

In Section 7.14, we have formulated the stationary NSE system. Without convection term, we arrive at the Stokes system for viscous flow (such as honey for instance). Without going into any theoretical details, we briefly present the finite element scheme in the following. A classical reference to NSE FEM discretizations is [27].

Let $U_h = \{v_h, p_h\}$, $X_h := V_h \times L_h = H_0^1 \times L_0$ and $\Psi = \{\psi^v, \psi^p\}$. The problem reads:

$$\text{Find } U_h \in X_h \text{ such that: } A(U_h, \Psi_h) = F(\Psi_h) \quad \forall \Phi \in X_h,$$

where

$$\begin{aligned} A(U_h, \Psi_h) &= (\nabla v_h, \nabla \psi_h^v) - (p_h, \nabla \cdot \psi_h^v) + (\nabla \cdot v_h, \psi_h^p), \\ F(\Psi_h) &= (f_f, \psi_h^v). \end{aligned}$$

Let us choose as basis:

$$\begin{aligned} V_h &= \{\psi_h^{v,i}, i = 1, \ldots, N_V := dim V_h\}, \\ L_h &= \{\psi_h^{p,i}, i = 1, \ldots, N_P := dim L_h\}. \end{aligned}$$

Please pay attention that $V_h := (V_h)^d$ is a vector-valued space with dimension d. If follows:

$$\begin{aligned} (\nabla v_h, \nabla \psi_h^{v,i}) - (p_h, \nabla \cdot \psi_h^{v,i}) \quad & i = 1, \ldots, N_V, \\ (\nabla \cdot v_h, \psi_h^{p,i}) \quad & i = 1, \ldots, N_P, \end{aligned}$$

Setting:

$$v_h = \sum_{j=1}^{N_V} v_j \psi_h^{v,j}, \quad p_h = \sum_{j=1}^{N_P} p_j \psi_h^{p,j}$$

yield the discrete equations:

$$\begin{aligned} \sum_{j=1}^{N_V} (\nabla \psi_h^{v,j}, \nabla \psi_h^{v,i}) v_j - \sum_{j=1}^{N_P} (\psi_h^{p,j}, \nabla \cdot \psi_h^{v,i}) p_j, \quad & i = 1, \ldots, N_V, \\ \sum_{j=1}^{N_v} (\nabla \cdot \psi_h^{v,j}, \psi_h^{p,i}), \quad & i = 1, \ldots, N_P. \end{aligned}$$

With this, we obtain the following matrices:

$$A := (\nabla \psi_h^{v,j}, \nabla \psi_h^{v,i})_{ij=1}^{N_V, N_V}, \quad B := -(\psi_h^{p,j}, \nabla \cdot \psi_h^{v,i})_{ij=1}^{N_V, N_P}, \quad -B^T = (\nabla \cdot \psi_h^{v,j}, \psi_h^{p,i})_{ij=1}^{N_P, N_V}.$$

These matrices form the block system:

$$\begin{pmatrix} A & B \\ -B^T & 0 \end{pmatrix} \begin{pmatrix} v \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{194}$$

where $v = (v_i)_{i=1}^{N_V}, p = (p_i)_{i=1}^{N_P}$.

9.6 Numerical test - 2D-1 benchmark without convection term

We take the same material parameters as in Section 7.14, but now without the convection term. The results are displayed in Figure 33.

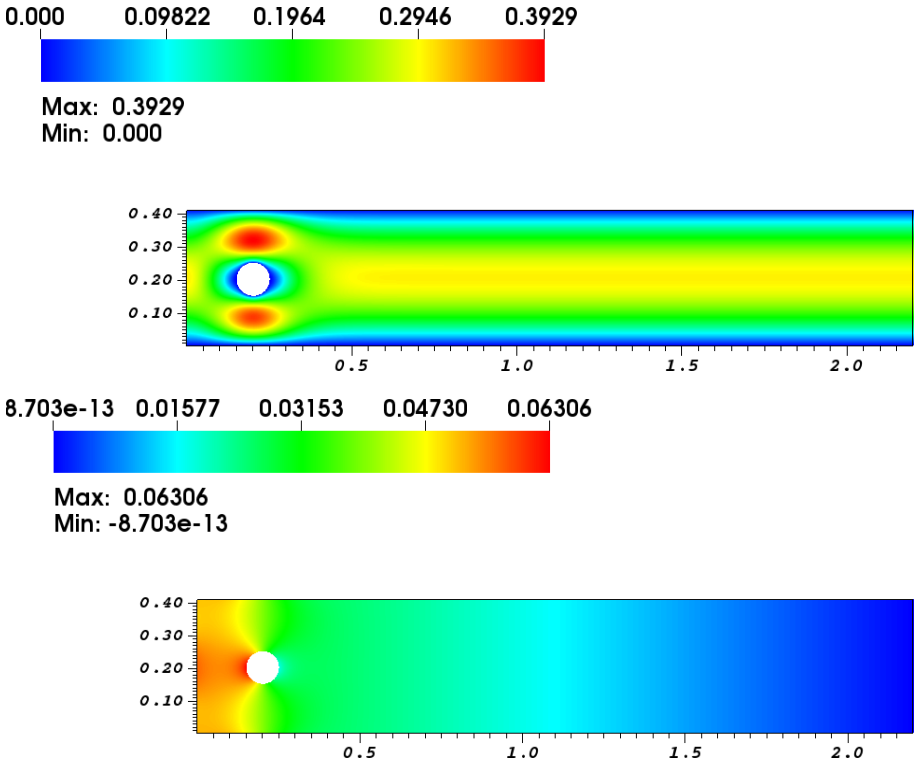

Figure 33: Stokes 2D-1 test: x velocity profile and pressure distribution.

The functional values on a three times uniformly-refined mesh are:

```
==================================
P-Diff:      4.5559514861737337e-02
P-front:     6.3063021215059634e-02
P-back:      1.7503506353322297e-02
Drag:        3.1419886302140707e+00
Lift:        3.0188981539007183e-02
==================================
```

9.7 Chapter summary and outlook

In this chapter, we extended from scalar-valued elliptic problems to vector-valued elliptic problems. The well-posedness analysis follows the same lines as the scalar cases. We notice that the maximum principle does not hold anymore in the vector-valued case (not proven though, but shown in a numerical example.). The finite element formulation is in principle the same as the scalar case, but we have to work with vector-valued function spaces now. The formal procedure is the same as before, but the computational cost increases significantly since now three variables need to discretized (in 3D) rather than only one variable.

10 Methods for parabolic and hyperbolic problems

By now, we have considered **stationary, linear** PDE problems. However, more realistic, and most notably fascinating, modeling is

- nonstationary,
- and nonlinear.

We investigate the former one in the present chapter and give a brief introduction to nonlinear problems in Chapter 11.

Therefore, the focus of this chapter is on

- time-dependent linear PDEs.

We concentrate first on parabolic PDEs and later study second-order-in-time hyperbolic PDEs. For first-order hyperbolic PDEs, we refer to [43].

A prominent example of a parabolic PDE is the heat equation that we investigate in this section. In the next section, we consider the second-order hyperbolic wave equation.

10.1 Principle procedures for discretizing time and space

Time-dependent PDE problems require require discretization in space and time. One can mainly distinguish three procedures:

- Vertical method of lines (method of lines): first space, then time.
- Horizontal method of lines (Rothe method): first time, then space.
- A full space-time Galerkin method.

Using one of the first two methods, and this is also the procedure we shall follow here, temporal discretization is based on FD whereas spatial discretization is based on the FEM. In the following we concentrate on the Rothe method and we briefly provide reasons why we want to work with the Rothe method. In fact, the traditional way of discretizing time-dependent problems is the (vertical) method of lines. The advantage of the method of lines is that we have simple data structures and matrix assembly. This leads to a (large) ODE system, which can be treated by (well-known) standard methods from ODE analysis (ordinary differential equations) can be employed for time discretization. The major disadvantage is that the spatial mesh is fixed (since we first discretize in space) and it is then difficult to represent or compute time-varying features such as certain target functionals (e.g., drag or lift).

In contrast, the Rothe method allows for dynamic spatial mesh adaptation with the price that data structures and matrix assembly are more costly.

10.2 Bochner spaces - space-time functions

For the correct function spaces for formulating time-dependent variational forms, we define the Bochner integral. Let $I := (0, T)$ with $0 < T < \infty$ a bounded time interval with end time value T. For any Banach space X and $1 \leq p \leq \infty$, the space

$$L^p(I, X)$$

denotes the space of L^p integrable functions f from the time interval I into X. This is a Banach space, the so-called Bochner space, with the norm, see [76],

$$\|v\|_{L^p(I,X)} := \Big(\int_I \|v(t)\|_X^p \, dt\Big)^{1/p}$$
$$\|v\|_{L^\infty(I,X)} := \operatorname*{ess\,sup}_{t\in I} \|v(t)\|_X.$$

For the definition of the norms of the spatial spaces, i.e., $\|v(t)\|_X$ with $X = L^2$ or $X = H^1$, we refer to Section 6.8.4.1.

Example 10.1. *For instance, we can define a H^1 space in time:*

$$H^1(I,X) = \left\{ v \in L^2(I,X) |\, \partial_t v \in L^2(I,X) \right\}.$$

Functions that are even continuous in time, i.e., $u : I \to X$, are contained in spaces like

$$C(I;X)$$

with

$$\|u\|_{C(I;X)} := \max_{0\le t\le T} \|u(t)\| < \infty.$$

Definition 10.2 (Weak derivative of space-time functions)**.** *Let $u \in L^1(I;X)$. A function $v \in L^1(I;X)$ is the weak derivative of v, denoted as*

$$\partial_t u = v$$

if

$$\int_0^T \partial_t \varphi(t) u(t)\, dt = -\int_0^T \varphi(t) v(t)\, dt$$

for all test functions $\varphi \in C_c^\infty(I)$.

In particular, the following result holds:

Theorem 10.3 ([24])**.** *Assume $v \in L^2(I, H_0^1)$ and $\partial_t v \in L^2(I, H^{-1})$. Then, v is continuous in time, i.e.,*

$$v \in C(I, L^2)$$

(after possible redefined on a set of measure zero). Furthermore, the mapping

$$t \mapsto \|v(t)\|^2_{L^2(X)}$$

is absolutely continuous with

$$\frac{d}{dt}\|v(t)\|^2_{L^2(X)} = 2\langle \frac{d}{dt} v(t), v(t)\rangle$$

for a.e. $0 \le t \le T$.

Proof. See Evans [24], Theorem 3 in Section 5.9.2. □

The importance of this theorem lies in the fact that now the point-wise prescription of initial conditions does make sense in weak formulations.

Remark 10.4. *More details of these spaces by means of the Bochner integral can be found in [21, 75] and also [24].*

10.3 Methods for parabolic problems

To illustrate the concepts of discretizing we work in 1D (one spatial dimension) in the following.

10.3.1 Problem statement

Let Ω be an open, bounded subset of $\mathbb{R}^d, d = 1$ and $I := (0, T]$ where $T > 0$ is the end time value. The IBVP (initial boundary-value problem) reads:

Formulation 10.5. *Find $u := u(x,t) : \Omega \times I \to \mathbb{R}$ such that*

$$\begin{aligned} \rho\partial_t u - \nabla\cdot(\alpha\nabla u) &= f \quad \text{in } \Omega \times I, \\ u &= a \quad \text{on } \partial\Omega \times [0,T], \\ u(0) &= g \quad \text{in } \Omega \times t = 0, \end{aligned}$$

where $f : \Omega \times I \to \mathbb{R}$ and $g : \Omega \to \mathbb{R}$ and $\alpha \in \mathbb{R}$ and ρ are material parameters, and $a \ge 0$ is a Dirichlet boundary condition. More precisely, g is the initial temperature and a is the wall temperature, and f is some heat source.

For the variational formulation, we define the Sobolev space

$$W_2^1(I, H_0^1, L^2)$$

with the norm

$$\|u\|_{W_2^1} := \left(\|u\|^2_{L^2(I,H_0^1)} + \|u\|^2_{L^2(I,H^{-1})}\right)^{1/2}$$

Formulation 10.6 ([33, 77]). *Let $\Omega \subset \mathbb{R}^n$ be bounded and sufficiently smooth boundary $\partial\Omega$. Let $f \in L^2(I, H^{-1})$. The variational form is given by: Find $u \in W_2^1(I, H_0^1, L^2)$ with the initial condition $u(0) = u_0 \in L^2(\Omega)$ such that*

$$\frac{d}{dt}(\rho u, \phi) + (\alpha \nabla u, \nabla \phi) = \langle f, \phi \rangle \quad \forall \phi \in H_0^1(\Omega),$$

where $\langle \cdot, \cdot \rangle$ is the duality product since f is taken from the dual space H^{-1}.

10.3.2 Temporal discretization

We first discretize in time and create a time grid of the time domain $I = [0, T]$ with N_T intervals and a time step size $k = \frac{T}{N_T}$:

$$0 = t_0 < t_1 < \ldots < t_N = T.$$

Furthermore we set

$$I_n = [t_{n-1}, t_n], \quad k_n = t_n - t_{n-1}, \quad k := \max_{1 \le n \le N} k_n.$$

Moreover, we denote $u^n := u(t^n)$.

Here we use finite differences and more specifically we introduce the so-called **One-Step-θ** scheme, which allows for a compact notation for three major finite difference schemes: forward Euler ($\theta = 0$), backward Euler ($\theta = 1$), and Crank-Nicolson ($\theta = 0.5$) well known from numerical methods for ODEs.

Definition 10.7 (Choice of θ). *By the choice of θ, we obtain the following time-stepping schemes:*

- *$\theta = 0$: 1st order explicit Euler time stepping;*
- *$\theta = 0.5$: 2nd order Crank-Nicolson (trapezoidal rule) time stepping;*
- *$\theta = 0.5 + k_n$: 2nd order shifted Crank-Nicolson which is shifted by the time step size $k_n = t^n - t^{n-1}$ towards the implicit side;*
- *$\theta = 1$: 1st order implicit Euler time stepping.*

To have good stability properties (namely A-stability) of the time-stepping scheme is important for temporal discretization of partial differential equations. Often (as here for the heat equation) we deal with second-order operators in space, such PDE-problems are generically (very) stiff with the order $O(h^{-2})$, where h is the spatial discretization parameter.

After these preliminary considerations, let us now discretize in time the above problem:

$$\begin{aligned}
& \partial_t u - \nabla \cdot (\alpha \nabla u) = f \\
\Rightarrow \quad & \frac{u^n - u^{n-1}}{k} - \theta \nabla \cdot (\alpha \nabla u^n) - (1-\theta) \nabla \cdot (\alpha \nabla u^{n-1}) = \theta f^n + (1-\theta) f^{n-1} \\
\Rightarrow \quad & u^n - k\theta \nabla \cdot (\alpha \nabla u^n) = u^{n-1} + k(1-\theta) \nabla \cdot (\alpha \nabla u^{n-1}) + k\theta f^n + k(1-\theta) f^{n-1}
\end{aligned}$$

where $u^n := u(t^n)$ and $u^{n-1} := u(t^{n-1})$ and $f^n := f(t^n)$ and $f^{n-1} := f(t^{n-1})$.

10.3.3 Spatial discretization

We take the temporally discretized problem and use a Galerkin finite element scheme to discretize in space as explained in Section 6. That is we multiply with a test function and integrate. We then obtain: Find $u_h \in \{a + V_h\}$ such that for $n = 1, 2, \dots N_T$:

$$(u_h^n, \varphi_h) - k\theta(\alpha \nabla u_h^n, \nabla \varphi_h) = (u^{n-1}, \varphi_h) + k(1-\theta)(\alpha \nabla u^{n-1}, \nabla \varphi_h) + k\theta(f^n, \varphi_h) + k(1-\theta)(f^{n-1}, \varphi_h)$$

for all $\varphi_h \in V_h$.

Definition 10.8 (Specification of the problem data)**.** *For simplicity let us assume that there is no heat source $f = 0$ and the material coefficients are specified by $\alpha = 1$ and $\rho = 1$.*

Furthermore, let us work with the explicit Euler scheme $\theta = 0$. Then we obtain:

$$(u_h^n, \varphi_h) = (u_h^{n-1}, \varphi_h) - k(\nabla u_h^{n-1}, \nabla \varphi_h)$$

The structure of this equation, namely

$$y^n = y^{n-1} + k f(t^{n-1}, y^{n-1}),$$

is the same as we know from ODEs. On the other hand, the single terms in the weak form are akin to our derivations in Section 6. This means that we seek at each time t^n a finite element solution u_h^n on the spatial domain Ω. With the help of the basis functions $\varphi_{h,j}$ we can represent the spatial solution:

$$u_h^n(x) := \sum_{j=1}^{N_x} u_{h,j} \varphi_{h,j}, \quad u_{h,j} \in \mathbb{R}.$$

Then:

$$\sum_{j=1}^{N_x} u_{h,j}^n (\varphi_{h,j}, \varphi_{h,i}) = (u^{n-1}, \varphi_{h,i}) - k(\nabla u_h^{n-1}, \nabla \varphi_{h,i})$$

which results in a linear equation system: $MU = B$. Here

$$\begin{aligned}(M_{ij})_{i,j=1}^{N_x} &:= (\varphi_{h,j}, \varphi_{h,i}),\\ (U_j)_{j=1}^{N_x} &:= (u_{h,1}, \dots, u_{h,N})\\ (B_i)_{i=1}^{N_x} &:= (u_h^{n-1}, \varphi_{h,i}) - k(\nabla u_h^{n-1}, \nabla \varphi_{h,i}).\end{aligned}$$

Furthermore the old time step solution can be written as well in matrix form:

$$(u^{n-1}, \varphi_{h,i}) = \sum_{j=1}^{N_x} u_{h,j}^{n-1} \underbrace{(\varphi_{h,j}, \varphi_{h,i})}_{=:M}$$

and similarly for the Laplacian term:

$$(\nabla u^{n-1}, \nabla \varphi_{h,i}) = \sum_{j=1}^{N_x} u_{h,j}^{n-1} \underbrace{(\nabla \varphi_{h,j}, \nabla \varphi_{h,i})}_{=:K}.$$

The mass matrix M (also known as the Gramian matrix) is always the same for fixed h and can be explicitely computed.

Proposition 10.9. *Formally we arrive at: Given u_h^{n-1}, find u_h^n, such that*

$$M u_h^n = M u_h^{n-1} - kK u_h^{n-1} \quad \Rightarrow \quad u_h^n = u_h^{n-1} - kM^{-1} K u_h^{n-1}.$$

for $n = 1, \dots, N_T$.

Remark 10.10. *We emphasize that the forward Euler scheme is not recommended to be used as time stepping scheme for solving PDEs. The reason is that the stiffness matrix is of order $\frac{1}{h^2}$ and one is in general interested in $h \to 0$. Thus the coefficients become very large for $h \to 0$ resulting in a stiff system. For stiff systems, as we learned before, one should better use implicit schemes, otherwise the time step size k has to be chosen too small in order to obtain stable numerical results; see the numerical analysis in Section 10.3.8.*

10.3.4 Evaluation of the integrals (1D in space)

We need to evaluate the integrals for the stiffness matrix K:

$$K = (K_{ij})_{ij=1}^{N_x} = \int_\Omega \varphi'_{h,j}(x)\varphi'_{h,i}(x)\,dx = \int_{x_{j-1}}^{x_{j+1}} \varphi'_{h,j}(x)\varphi'_{h,i}(x)\,dx$$

resulting in

$$A = h^{-1}\begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{pmatrix}$$

Be careful and do not forget the material parameter α in case it is not $\alpha = 1$.

For the mass matrix M we obtain:

$$M = (M_{ij})_{ij=1}^{N_x} = \int_\Omega \varphi_{h,j}(x)\varphi_{h,i}(x)\,dx = \int_{x_{j-1}}^{x_{j+1}} \varphi_{h,j}(x)\varphi_{h,i}(x)\,dx.$$

Specifically on the diagonal, we calculate:

$$\begin{aligned} M_{ii} &= \int_\Omega \varphi_{h,i}(x)\varphi_{h,i}(x)\,dx = \int_{x_{i-1}}^{x_i} \Big(\frac{x - x_{i-1}}{h}\Big)^2 dx + \int_{x_i}^{x_{i+1}} \Big(\frac{x_{i+1} - x}{h}\Big)^2 dx \\ &= \frac{1}{h^2}\int_{x_{i-1}}^{x_i} (x - x_{i-1})^2\,dx + \frac{1}{h^2}\int_{x_i}^{x_{i+1}} (x_{i+1} - x)^2\,dx = \frac{h}{3} + \frac{h}{3} \\ &= \frac{2h}{3}. \end{aligned}$$

For the right off-diagonal, we have

$$\begin{aligned} m_{i,i+1} &= \int_\Omega \varphi_{h,i+1}(x)\varphi_{h,i}(x)\,dx = \int_{x_i}^{x_{i+1}} \frac{x - x_i}{h}\cdot\frac{x_{i+1} - x}{h}\,dx = \int_{x_i}^{x_{i+1}} \frac{x - x_i}{h}\cdot\frac{x_i - x + h}{h}\,dx \\ &= \dots \\ &= -\frac{h}{3} + \frac{h^2}{2h} = \frac{h}{6}. \end{aligned}$$

It is trivial to see that $m_{i,i+1} = m_{i-1,i}$. Summarizing all entries results in

$$M = \frac{h}{6}\begin{pmatrix} 4 & 1 & & & 0 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ 0 & & & 1 & 4 \end{pmatrix}.$$

10.3.5 Final algorithms

We first setup a sequence of discrete (time) points:

$$0 = t_0 < t_1 < \dots < t_N = T.$$

Furthermore we set

$$I_n = [t_{n-1}, t_n], \quad k_n = t_n - t_{n-1}, \quad k := \max_{1\le n\le N} k_n.$$

Forward (explicit) Euler:

Algorithm 10.11. *Given the initial condition g, solve for $n = 1, 2, 3, \ldots, N_T$*

$$Mu_h^n = Mu_h^{n-1} - kKu_h^{n-1} \quad \Rightarrow \quad u_h^n = u_h^{n-1} - kM^{-1}Ku_h^{n-1},$$

where $u_h^n, u_h^{n-1} \in \mathbb{R}^{N_x}$.

Algorithm 10.12. *The backward (implicit) Euler scheme reads:*

$$Mu_h^n + kKu_h^n = Mu_h^{n-1}$$

An example of a pseudo C++ code is

```
final_loop()
{
 // Initalization of model parameters, material parameters, and so forth
 set_runtime_parameters();

 // Make a mesh - decompose the domain into elements
 create_mesh();

 apply_initial_conditions();

 // Timestep loop
  do
    {
      std::cout << "Timestep " << timestep_number << " (" << time_stepping_scheme
                << ")" <<    ": " << time << " (" << timestep << ")"
                << "\n=========================================================================="
                << std::endl;

      std::cout << std::endl;

      // Solve for next time step: Assign previous time step solution u^{n-1}
      old_timestep_solution = solution;

      // Assemble FEM matrix A and right hand side f
      assemble_system ();

      // Solve the linear equation system Ax=b
      solve ();

      // Update time
      time += timestep;

      // Write solution into a file or similar
      output_results (timestep_number);

      // Increment n->n+1
      ++timestep_number;
    }
  while (timestep_number <= max_no_timesteps);
}
```

10.3.6 Recapitulation of ODE stability analysis using finite differences

This section is optional. We recall the stability analysis for ODEs. The principle ideas are exactly the same as in Section 5.

10.3.6.1 The Euler method The Euler method is the most simplest scheme. It is an explicit scheme and also known as forward Euler method.

We consider again the IVP from before and the right hand side satisfies again a Lipschitz condition. According to ODE theory (e.g., [53]), there exists a unique solution for all $t \geq 0$.

For a numerical approximation of the IVP, we first select a sequence of discrete (time) points:

$$t_0 < t_1 < \ldots < t_N = t_0 + T.$$

Furthermore we set

$$I_n = [t_{n-1}, t_n], \quad k_n = t_n - t_{n-1}, \quad k := \max_{1 \leq n \leq N} k_n.$$

The derivation of the Euler method is as follows: approximate the derivative with a forward difference quotient (we sit at the time point t_{n-1} and look forward in time):

$$y'(t_{n-1}) \approx \frac{y_n - y_{n-1}}{k_n}$$

Thus: $y'(t_{n-1}) = f(t_{n-1}, y_{n-1}(t_{n-1}))$. Then the ODE can be approximated as:

$$\frac{y_n - y_{n-1}}{k_n} \approx f(t_{n-1}, y_{n-1}(t_{n-1}))$$

Thus we obtain the scheme:

Algorithm 10.13 (Euler method)**.** *For a given starting point $y_0 := y(0) \in \mathbb{R}^n$, the Euler method generates a sequence $\{y_n\}_{n \in \mathbb{N}}$ through*

$$y_n = y_{n-1} + k_n f(t_{n-1}, y_{n-1}), \quad n = 1, \ldots N,$$

where $y_n := y(t_n)$.

Remark 10.14 (Notation)**.** *The chosen notation is not optimal in the sense that y_n denotes the discrete solution obtained by the numerical scheme and $y(t_n)$ the corresponding (unknown) exact solution. However, in the literature one often abbreviates $y_n := y(t_n)$, which would both denote the exact solution. One could add another index y_n^k (k for discretized solution with step size k) to explicitely distinguish the discrete and exact solutions. In these notes, we hope that the reader will not confuse the notation and we still use y_n for the discrete solution and $y(t_n)$ for the exact solution.*

10.3.6.2 Implicit schemes With the same notation as introduced in Section 10.3.6.1, we define two further schemes. Beside the Euler method, low-order simple schemes are implicit Euler and the trapezoidal rule. The main difference is that in general a nonlinear equation system needs to be solved in order to compute the solution. On the other hand we have better numerical stability properties in particular for stiff problems (an analysis will be undertaken in Section 10.3.6.3).

The derivation of the backward Euler method is derived as follows: approximate the derivative with a backward difference quotient (we sit at t_n and look back to t_{n-1}):

$$y'(t_n) = \frac{y_n - y_{n-1}}{k_n}$$

Consequently, we take the right hand side f at the current time step $y'(t_n) = f(t_n, y_n(t_n))$ and obtain as approximation

$$\frac{y_n - y_{n-1}}{k_n} = f(t_n, y_n(t_n)).$$

Consequently, we obtain a scheme in which the right hand side is unknown itself, which leads to a formulation of a nonlinear system:

Algorithm 10.15 (Implicit (or backward) Euler)**.** *The implicit Euler scheme is defined as:*

$$y_0 := y(0),$$
$$y_n - k_n f(t_n, y_n) = y_{n-1}, \quad n = 1, \ldots, N$$

In contrast to the Euler method, the 'right hand side' function f now depends on the unknown solution y_n. Thus the computational cost is (much) higher than for the (forward) Euler method. But on the contrary, the method does not require a time step restriction as we shall see in Section 10.3.6.4.

To derive the trapezoidal rule, we take again the difference quotient on the left hand side but approximate the right hand side through its mean value:

$$\frac{y_n - y_{n-1}}{k_n} = \frac{1}{2}\Big(f(t_n, y_n(t_n)) + f(t_{n-1}, y_{n-1}(t_{n-1}))\Big),$$

which yields:

Algorithm 10.16 (Trapezoidal rule (Crank-Nicolson)). *The trapezoidal rule reads:*

$$\begin{aligned} y_0 &:= y(0), \\ y_n &= y_{n-1} + \frac{1}{2}k_n\Big(f(t_n, y_n) + f(t_{n-1}, y_{n-1}\Big), \quad n = 1, \dots, N. \end{aligned}$$

It can be shown that the trapezoidal rule is of second order, which means that halving the step size k_n leads to an error that is four times smaller.

10.3.6.3 Numerical analysis In the previous section, we have constructed algorithms that yield a sequence of discrete solution $\{y_n\}_{n\in\mathbb{N}}$. In the numerical analysis our goal is to derive a convergence result of the form

$$\|y_n - y(t_n)\| \leq Ck^\alpha$$

where α is the order of the scheme. This result will tell us that the discrete solution y^n really approximates the exact solution y and if we come closer to the exact solution at which rate we come closer. To derive error estimates we work with the model problem:

$$y' = \lambda y, \quad y(0) = y_0, \quad y_0, \lambda \in \mathbb{R}. \tag{195}$$

For linear numerical schemes the convergence is composed by **stability** and **consistency**.

First of all we have from the previous section that y_n is obtained for the forward Euler method as:

$$\begin{aligned} y_n &= (1 + k\lambda)y_{n-1}, && (196) \\ &= B_E y_{n-1}, \quad B_E := (1 + k\lambda). && (197) \end{aligned}$$

Let us write the error at each time point t_n as:

$$e_n := y_n - y(t_n) \quad \text{for } 1 \leq n \leq N.$$

It holds:

$$\begin{aligned} e_n &= y_n - y(t_n), \\ &= B_E y_{n-1} - y(t_n), \\ &= B_E(e_{n-1} + y(t_{n-1})) - y(t_n), \\ &= B_E e_{n-1} + B_E y(t_{n-1}) - y(t_n), \\ &= B_E e_{n-1} + \frac{k(B_E y(t_{n-1}) - y(t_n))}{k}, \\ &= B_E e_{n-1} - k\underbrace{\frac{y(t_n) - B_E y(t_{n-1})}{k}}_{=:\eta_{n-1}}. \end{aligned}$$

Therefore, the error can be split into two parts:

Definition 10.17 (Error splitting of the model problem). *The error at step n can be decomposed as*

$$e_n := \underbrace{B_E e_{n-1}}_{Stability} - \underbrace{k\eta_{n-1}}_{Consistency}. \tag{198}$$

The first term, namely the stability, provides an idea how the previous error e_{n-1} is propagated from t_{n-1} to t_n. The second term η_{n-1} is the so-called truncation error (or local discretization error), which arises because the exact solution does not satisfy the numerical scheme and represents the consistency of the numerical scheme. Moreover, η_{n-1} yields the speed of convergence of the numerical scheme.

In fact, for the forward Euler scheme in (198), we observe for the truncation error:

$$\eta_{n-1} = \frac{y(t_n) - B_E y(t_{n-1})}{k} = \frac{y(t_n) - (1+k\lambda)y(t_{n-1})}{k} \tag{199}$$

$$= \frac{y(t_n) - y(t_{n-1})}{k} - \lambda y(t_{n-1}) \tag{200}$$

$$= \frac{y(t_n) - y(t_{n-1})}{k} - y'(t_{n-1}). \tag{201}$$

Thus,

$$y'(t_{n-1}) = \frac{y(t_n) - y(t_{n-1})}{k} - \eta_{n-1}, \tag{202}$$

which is nothing else than the approximation of the first-order derivative with the help of a difference quotient plus the truncation error. We investigate these terms further in Section 10.3.6.5 and concentrate first on the stability estimates in the very next section.

10.3.6.4 Stability The goal of this section is to control the term $B_E = (1 + k\lambda)$. Specifically, we will justify why $|B_E| \leq 1$ should hold. The stability is often related to non-physical oscillations of the numerical solution.

We recapitulate (absolute) **stability** and **A-stability**. From the model problem

$$y'(t) = \lambda y(t), \quad y(t_0) = y_0,\ \lambda \in \mathbb{C},$$

we know the solution $y(t) = y_0 \exp(\lambda t)$. For $t \to \infty$ the solution is characterized by the sign of $Re\,\lambda$:

$$\begin{aligned} Re\,\lambda < 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) \to 0, \\ Re\,\lambda = 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) = |y_0|, \\ Re\,\lambda > 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) \to \infty. \end{aligned}$$

For a *good* numerical scheme, the first case is particularly interesting whether such a scheme can produce a bounded discrete solution when the continuous solution has this property.

Definition 10.18 ((Absolute) stability)**.** *A (one-step) method is absolute stable for $\lambda k \neq 0$ if its application to the model problem produces in the case $Re\,\lambda \leq 0$ a sequence of bounded discrete solutions: $\sup_{n\geq 0} |y_n| < \infty$. To find the stability region, we work with the stability function $R(z)$ where $z = \lambda k$. The region of absolute stability is defined as:*

$$SR = \{z = \lambda k \in \mathbb{C} : |R(z)| \leq 1\}.$$

Remark 10.19. *Recall that $R(z) := B_E$.*

The stability functions to explicit, implicit Euler and trapezoidal rule are given by:

Proposition 10.20. *For the simplest time-stepping schemes forward Euler, backward Euler and the trapezoidal rule, the stability functions $R(z)$ read:*

$$\begin{aligned} R(z) &= 1 + z, \\ R(z) &= \frac{1}{1-z}, \\ R(z) &= \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}. \end{aligned}$$

Proof. We take again the model problem $y' = \lambda y$. Let us discretize this problem with the forward Euler method:

$$\frac{y_n - y_{n-1}}{k} = \lambda y_{n-1} \tag{203}$$

$$\Rightarrow y_n = (y_{n-1} + \lambda k) y_{n-1} \tag{204}$$

$$= (1 + \lambda k) y_{n-1} \tag{205}$$

$$= (1 + z) y_{n-1} \tag{206}$$

$$= R(z) y_{n-1}. \tag{207}$$

For the implicit Euler method we obtain:

$$\frac{y_n - y_{n-1}}{k} = \lambda y_n \tag{208}$$

$$\Rightarrow y_n = (y_{n-1} + \lambda k) y_n \tag{209}$$

$$\Rightarrow y_n = \frac{1}{1-ak} y_n \tag{210}$$

$$\Rightarrow y_n = \underbrace{\frac{1}{1-z}}_{=:R(z)} y_n. \tag{211}$$

The procedure for the trapezoidal rule is again the analogous. □

Definition 10.21 (A-stability)**.** *A difference method is A-stable if its stability region is part of the absolute stability region:*

$$\{z \in \mathbb{C} : Re\, z \leq 0\} \subset SR,$$

here Re denotes the real part of the complex number z. A brief introduction to complex numbers can be found in any calculus lecture dealing with those or also in the book [61].

In other words:

Definition 10.22 (A-stability)**.** *Let $\{y_n\}_n$ the sequence of solutions of a difference method for solving the ODE model problem. Then, this method is A-stable if for arbitrary $\lambda \in \mathbb{C}^- = \{\lambda : Re(\lambda) \leq 0\}$ the approximate solutions are bounded (or even contractive) for arbitrary, but fixed, step size k. That is to say:*

$$|y_{n+1}| \leq |y_n| < \infty \quad \text{for } n = 1, 2, 3, \ldots$$

Remark 10.23. *A-stability is attractive since in particular for stiff problems we can compute with arbitrary step sizes k and do not need any step size restriction.*

Proposition 10.24. *The explicit Euler scheme cannot be A-stable.*

Proof. For the forward Euler scheme, it is $R(z) = 1 + z$. For $|z| \to \infty$ is holds $R(z) \to \infty$ which is a violation of the definition of A-stability. □

Remark 10.25. *More generally, explicit schemes can never be A-stable.*

Example 10.26. *We illustrate the previous statements.*

1. *In Proposition 10.20 we have seen that for the forward Euler method it holds:*

   $$y_n = R(z) y_{n-1},$$

   where $R(z) = 1 + z$. Thus, according to Definition 10.21 and 10.22, we obtain convergence when the sequence $\{y_n\}$ is contracting:

   $$|R(z)| \leq |1 + z| \leq 1. \tag{212}$$

   Thus if the value of λ (in $z = \lambda k$) is very big, we must choose a very small time step k in order to achieve $|1 - \lambda k| < 1$. Otherwise the sequence $\{y_n\}_n$ will increase and thus diverge (recall that stability is defined with respect to decreasing parts of functions! Thus, the continuous solution is bounded and consequently the numerical approximation should be bounded, too). In conclusions, the forward Euler scheme is only conditionally stable, i.e., it is stable provided that (212) *is fulfilled.*

2. *For the implicit Euler scheme, we see that a large λ and large k even both help to stabilize the iteration scheme (but be careful, the implicit Euler scheme, stabilizes actually too much. Because it computes contracting sequences also for case where the continuous solution would grow). Thus, no time step restriction is required. Consequently, the implicit Euler scheme is well suited for stiff problems with large parameters/coefficients λ.*

Remark 10.27. *The previous example shows that a careful design of the appropriate discretization scheme requires some work: there is no a priori best scheme. Some schemes require time step size restrictions in case of large coefficients (explicit Euler). On the other hand, the implicit Euler scheme does not need step restrictions but may have in certain cases too much damping. Which scheme should be employed for which problem depends finally on the problem itself and must be carefully thought for each problem again.*

10.3.6.5 Consistency / local discretization error - convergence order We address now the second 'error source' in (198). The consistency determines the precision of the scheme and will finally carry over the local rate of consistency to the global rate of convergence. To determine the consistency we assume sufficient regularity of the exact solution such that we can apply Taylor expansion. The idea is then that all Taylor terms of combined to the discrete scheme. The lowest order remainder term determines finally the local consistency of the scheme.

We briefly formally recall Taylor expansion. For a function $f(x)$ we develop at a point $a \neq x$ the Taylor series:

$$T(f(x)) = \sum_{j=0}^{\infty} \frac{f^{(j)}(a)}{j!}(x-a)^j.$$

Let us continue with the forward Euler scheme and let us now specify the truncation error η_{n-1} in (202):

$$y'(t_{n-1}) + \eta_{n-1} = \frac{y(t_n) - y(t_{n-1})}{k}.$$

To this end, we need information about the solution at the old time step t^{n-1} in order to eliminate $y(t_n)$. Thus we use Taylor and develop $y(t^n)$ at the time point t^{n-1}:

$$y(t^n) = y(t^{n-1}) + y'(t^{n-1})k + \frac{1}{2}y''(\tau^{n-1})k^2$$

We obtain the difference quotient of forward Euler by the following manipulation:

$$\frac{y(t^n) - y(t^{n-1})}{k} = y'(t^{n-1}) + \frac{1}{2}y''(\tau^{n-1})k.$$

We observe that the first terms correspond to (201). Thus the remainder term is

$$\frac{1}{2}y''(\tau^{n-1})k$$

and therefore the truncation error η_{n-1} can be estimated as

$$\|\eta_{n-1}\| \leq \max_{t\in[0,T]} \frac{1}{2}\|y''(t)\|k = O(k).$$

Therefore, the convergence order is k (namely linear convergence speed).

10.3.6.6 Convergence With the help of the two previous subsections, we can easily show the following error estimates:

Theorem 10.28 (Convergence of implicit/explicit Euler)**.** *We have*

$$\max_{t_n\in I} |y_n - y(t_n)| \leq c(T,y)k = O(k),$$

where $k := \max_n k_n$.

Proof. The proof does hold for both schemes, except that when we plug-in the stability estimate one should recall that the backward Euler scheme is unconditionally stable and the forward Euler scheme is only stable

when the step size k is sufficiently small. It holds for $1 \le n \le N$:

$$\begin{aligned}
|y_n - y(t_n)| = \|e_n\| &= k\| \sum_{k=0}^{n-1} B_E^{n-k} \eta_k \| \\
&\le k \sum_{k=0}^{n-1} \|B_E^{n-k} \eta_k\| \quad \text{(triangle inequality)} \\
&\le k \sum_{k=0}^{n-1} \|B_E^{n-k}\| \, \|\eta_k\| \\
&\le k \sum_{k=0}^{n-1} \|B_E^{n-k}\| \, Ck \quad \text{(consistency)} \\
&\le k \sum_{k=0}^{n-1} 1 \, Ck \quad \text{(stability)} \\
&= kN \, Ck \\
&= T \, Ck, \quad \text{where we used } k = T/N \\
&= C(T,y)k \\
&= O(k)
\end{aligned}$$

□

Theorem 10.29 (Convergence of trapezoidal rule)**.** *We have*

$$\max_{t \in I} |y_n(t) - y(t)| \le c(T,y)k^2 = O(k^2),$$

The main message is that the Euler schemes both converge with order $O(k)$ (which is very slow) and the trapezoidal rule converges quadratically, i.e., $O(k^2)$.

Let us justify the convergence order for the forward Euler scheme in more detail now.

Theorem 10.30. *Let $I := [0,T]$ the time interval and $f : I \times \mathbb{R}^d \to \mathbb{R}^d$ continuously differentiable and globally Lipschitz-continuous with respect to y:*

$$\|f(t,y) - f(t,z)\|_2 \le L\|y - z\|_2$$

for all $t \in I$ and $y, z \in \mathbb{R}^d$. Let y be the solution to

$$y' = f(t,y), \quad y(0) = y_0.$$

Furthermore let $y_n, n = 1, \dots, n$ the approximations obtained with the Euler scheme at the nodal points $t_n \in I$. Then it holds

$$\|y(t_n) - y_n\|_2 \le \frac{(1+Lk)^n - 1}{2L} \|y''\| \, k \le \frac{e^{LT} - 1}{2L} \|y''\| \, k = c(T,y)k = O(k),$$

for $n = 0, \dots, N$.

Proof. The proof follows [36], but consists in working out the steps shown at the beginning of Section 10.3.6.3, Section 10.3.6.4, and Section 10.3.6.5. □

10.3.7 Stiff problems

An essential difficulty in developing stable numerical schemes for temporal discretization using finite differences is associated with **stiffness**, which we shall define in the following. Stiffness is very important in both ODE and (time-dependent) PDE applications. The latter situation will be discussed in Section 10.3.8.

Definition 10.31. *An IVP is called* ***stiff*** *(along a solution $y(t)$) if the eigenvalues $\lambda(t)$ of the Jacobian $f'(t, y(t))$ yield the stiffness ratio:*

$$\kappa(t) := \frac{\max_{Re\lambda(t)<0} |Re\lambda(t)|}{\min_{Re\lambda(t)<0} |Re\lambda(t)|} \gg 1.$$

In the above case for the model problem, the eigenvalue corresponds directly to the coefficient λ.

It is however not really true to classify any ODE with large coefficients $|\lambda| \gg 1$ always as a stiff problem. Here, the Lipschitz constant of the problem is already large and thus the discretization error asks for relatively small time steps. Rather stiff problems are characterized as ODE solutions that contain various components with (significant) different evolutions over time; that certainly appears in ODE systems in which we seek $y(t) \in \mathbb{R}^n$ rather than $y(t) \in \mathbb{R}$.

For PDEs, we know that the stiffness matrix of the Laplacian satisfies $K \sim \frac{1}{h^2}$ and therefore it holds for the condition number $\kappa \sim \frac{1}{h^2}$ (see Section 8.1). Thus, problems involving the Laplacian are always stiff.

10.3.8 Numerical analysis: stability analysis

We apply the concepts of the previous sections now to the heat equation. and simply replace y by u. The heat equation is a good example of a stiff problem (for the definition we refer back to Section 10.3.7) and therefore, implicit methods are preferred in order to avoid very, very small time steps in order to obtain stability. We substantiate this claim in the current section.

The model problem is

$$\begin{aligned} \partial_t u - \Delta u = 0 \quad & \text{in } \Omega \times I \\ u = 0 \quad & \text{on } \partial\Omega \times I, \\ u(0) = u^0 \quad & \text{in } \Omega \times \{0\}. \end{aligned}$$

Spatial discretization yields:

$$\begin{aligned} (\partial_t u_h, \phi_h) + (\nabla u_h, \nabla \phi_h) = 0, \\ (u_h^0, \phi_h) = (u^0, \phi_h). \end{aligned}$$

Backward Euler time discretization yields:

$$(u_h^n, \phi_h) + k(\nabla u_h^n, \nabla \phi_h) = (u_h^{n-1}, \phi_h)$$

It holds

Proposition 10.32. *A basic stability estimate for the above problem is:*

$$\|u_h^n\| \leq \|u_h^0\| \leq \|u^0\| \tag{213}$$

Proof. For the stability estimate we proceed as earlier and make a special choice for a test function: $\phi_h := u_h^n$. We then obtain:

$$\|u_h^n\|_{L^2}^2 + k|\nabla u_h^n|_{H^1}^2 = (u_h^{n-1}, u_h^n) \leq \frac{1}{2}\|u_h^{n-1}\|_{L^2}^2 + \frac{1}{2}\|u_h^n\|_{L^2}^2$$

which yields:

$$\frac{1}{2}\|u_h^n\|_{L^2}^2 - \frac{1}{2}\|u_h^{n-1}\|_{L^2}^2 + k|\nabla u_h^n|_{H^1}^2 \leq 0, \quad \forall n = 1, \ldots, N.$$

Of course it further holds:

$$\begin{aligned} & \frac{1}{2}\|u_h^n\|_{L^2}^2 - \frac{1}{2}\|u_h^{n-1}\|_{L^2}^2 \leq 0, \quad \forall n = 1, \ldots, N, \\ \Leftrightarrow \quad & \frac{1}{2}\|u_h^n\|_{L^2}^2 \leq \frac{1}{2}\|u_h^{n-1}\|_{L^2}^2 \end{aligned}$$

Taking the square root and applying the result back to $n = 0$ yields

$$\frac{1}{2}\|u_h^n\|_{L^2} \leq \frac{1}{2}\|u_h^0\|_{L^2}$$

It remains to show the second estimate:

$$\|u_h^0\| \leq \|u^0\|$$

In the initial condition we choose the test function $\varphi := u_h^0$ and obtain:

$$(u_h^0, u_h^0) = (u^0, u_h^0) \leq \frac{1}{2}\|u^0\| + \frac{1}{2}\|u_h^0\|$$

which shows the assertion. □

10.3.8.1 Stability analysis for backward Euler A more detailed stability analysis that brings in the spatial discretization properties is based on the matrix notation and is as follows. We recapitulate and start from (see Section 10.3.5):

$$Mu_h^n + kKu_h^n = Mu_h^{n-1}$$

Then:

$$\begin{aligned} Mu_h^n + kKu_h^n &= Mu_h^{n-1} \\ \Rightarrow u_h^n + kM^{-1}Ku_h^n &= u_h^{n-1} \\ \Rightarrow (I + kM^{-1}K)u_h^n &= u_h^{n-1} \end{aligned}$$

Compare to the ODE notation:

$$(1 + z)y^n = y^{n-1}$$

Thus, the role of $z = k\lambda$ is taken by $kM^{-1}K$ thus $\lambda \sim M^{-1}K$. We know for the condition numbers

$$\kappa(M) = O(1), \qquad \kappa(K) = O(h^{-2}), \quad h \to 0.$$

We proceed similar to ODEs, and obtain

$$u_h^n = (I + kM^{-1}K)^{-1}u_h^{n-1}$$

The goal is now to show that

$$(I + kM^{-1}K)^{-1} \leq 1.$$

To this end, let $\mu_j, j = 1, \ldots, M = dim(V_h)$ be the eigenvalues of the matrix $M^{-1}K$ ranging from the smallest eigenvalue $\mu_1 \sim O(1)$ to the largest eigenvalue $\mu_M \sim O(h^{-2})$; for a proof, we refer the reader to [43][Section 7.7]. Then:

$$|(I + kM^{-1}K)^{-1}| = \max_j \frac{1}{1 + k\mu_j} = \frac{1}{1 + k\mu_M}.$$

To have a stable scheme, we must have:

$$\frac{1}{1 + k\mu_M} < 1.$$

We clearly have then again from (213):

$$\|u_h^n\| \leq \|u_h^{n-1}\| \quad \Rightarrow \quad \|u_h^n\| \leq \|u_h^0\|$$

Furthermore, we see that the largest eigenvalue μ_M will increase as $O(h^{-2})$ when h tends to zero. Using the backward Euler or Crank-Nicolson scheme, this will be not a problem and both schemes are **unconditionally stable.**

10.3.8.2 Stability analysis for explicit Euler Here, the stability condition reads:

$$u_h^n = (I - kM^{-1}K)u_h^{n-1}$$

yielding

$$|(I - kM^{-1}K)^{-1}| = |\max_j(1 - k\mu_j)| = |1 - k\mu_M|.$$

As before $\mu_M = O(h^{-2})$. To establish a stability estimate of the form $\|u_h^n\| \leq \|u_h^{n-1}\|$, it is required that

$$|1 - k\mu_M| \leq 1.$$

Resolving this estimate, we obtain

$$k\mu_M \leq 2 \quad \Leftrightarrow \quad k \leq \frac{2}{\mu_M} = O(h^2),$$

thus

$$k \leq Ch^2, \quad C > 0. \tag{214}$$

Here, we only have **conditional stability in time**, namely the time step size k depends on the spatial discretization parameter h. Recall that in terms of accuarcy and discretization errors we want to work with small h, then the time step size has to be extremely small in order to have a stable scheme. In practice this is in most cases not attrative at all and for this reason the forward Euler scheme does not play a role despite that the actual solution is cheaper than solving an implicit scheme.

Exercise 9. *Derive stability estimates for the Crank-Nicolson scheme.*

10.3.9 Numerical tests

The series of numerical tests in this section has three goals:

- to show some figures to get an impression about simulation results;
- to show computationally that the stability results are indeed satisfied or violated;
- to show computationally satisfaction of the (discrete) maximum principle.

We consider the heat equation

$$\begin{aligned} \partial_t u - \Delta u &= 0, \quad \text{in } \Omega \times I \\ u &= 0 \quad \text{on } \partial\Omega \times I, \\ u(x,0) &= \sin(x)\sin(y) \quad \text{in } \Omega \times \{0\}. \end{aligned}$$

We compute 5 time steps, i.e., $T = 5s$, with time step size $k = 1s$. The computational domain is $\Omega = (0,\pi)^2$. We use a One-step-theta scheme with $\theta = 0$ (forward Euler) and $\theta = 1$ (backward Euler).

The graphical results are displayed in the Figures 34 - 37. Due to stability reasons and violation of the condition $k \leq ch^2$, the forward Euler scheme is unstable (Figures 36 - 37). By reducing the time step size to $k = 0.01s$ (the critical time step value could have been computed - the value $k = 0.01s$ has been found by trial and error simply), we obtain stable results for the forward Euler scheme. These findings are displayed in Figure 38. However, to reach $T = 5s$, we need to compute 500 time steps, which is finally more expensive, despite being an explicit scheme, than the implicit Euler method.

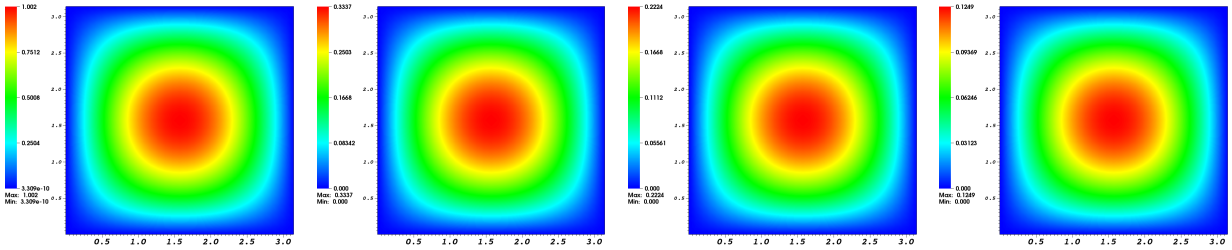

Figure 34: Heat equation with $\theta = 1$ (backward Euler) at $T = 0, 1, 2, 5$. The solution is stable and satisfies the parabolic maximum principle. The color scale is adapted at each time to the minimum and maximum values of the solution.

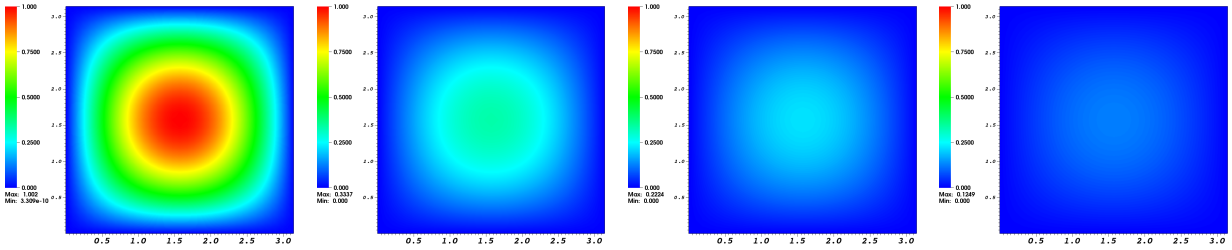

Figure 35: Heat equation with $\theta = 1$ (backward Euler) at $T = 0, 1, 2, 5$. The solution is stable and satisfies the parabolic maximum principle. The color scale is fixed between 0 and 1.

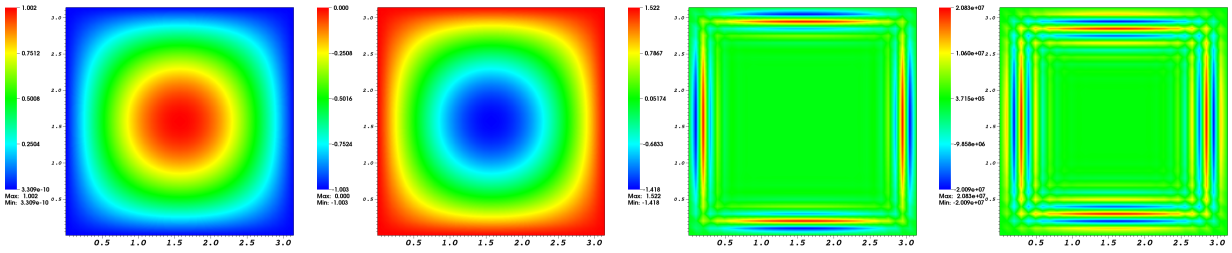

Figure 36: Heat equation with $\theta = 0$ (forward Euler) at $T = 0, 1, 2, 5$. The solution is unstable, showing non-physical oscillations, because the time step restriction (214) is violated. The color scale is adapted at each time to the minimum and maximum values of the solution.

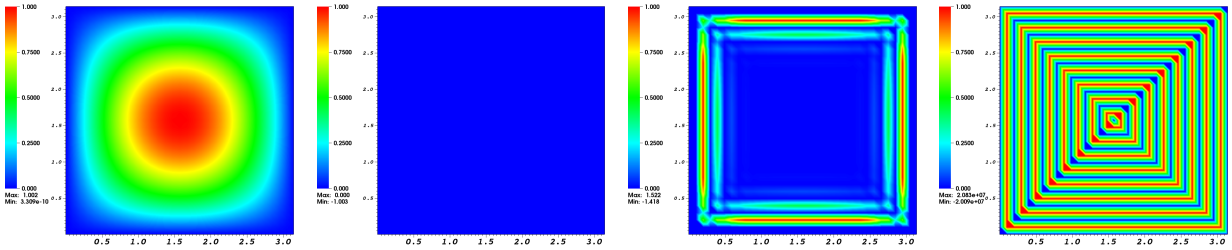

Figure 37: Heat equation with $\theta = 0$ (forward Euler) at $T = 0, 1, 2, 5$. The solution is unstable, showing non-physical oscillations, because the time step restriction (214) is violated. The color scale is fixed between 0 and 1.

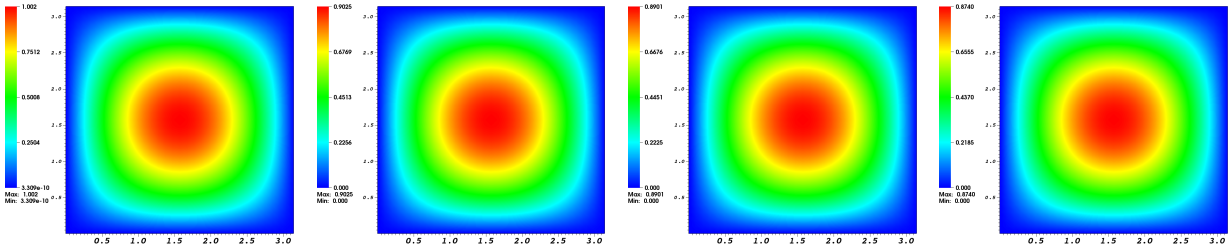

Figure 38: Heat equation with $\theta = 0$ at $T = 0, 1, 2, 5$ and time step size $k = 0.01s$. Here the results are stable since the time step size k has been chosen sufficiently small to satisfy the stability estimate (214). Of course, to obtain results at the same times T as before, we now need to compute more solutions, i.e., $N = 500$, rather than 5 as in the other tests. The color scale is adapted at each time to the minimum and maximum values of the solution.

10.4 Methods for second-order-in-time hyperbolic problems

For second-order (in time) hyperbolic problems (prototype is the wave equation) the situation is a bit more tricky since we have a second-order in time derivative.

10.4.1 Problem statement

The equation is given by; see also Section 4.2:

$$\begin{aligned} \rho\partial_{tt}^2 u - \nabla\cdot(\nabla u) &= f \quad \text{in } \Omega\times I,\\ u &= 0 \quad \text{on } \partial\Omega\times I,\\ u(x,0) &= u_0(x) \quad \text{in } \Omega\times\{0\},\\ \partial_t u(x,0) &= v_0(x) \quad \text{in } \Omega\times\{0\} \end{aligned}$$

Discretizing the wave equation directly with a One-step-theta scheme is not possible because of the second-order time derivative. Here, a common procedure is to re-write the equation into a first-order mixed system. On the other hand, we then introduce a second solution variable (the velocity), which also needs to be discretized in space and results in a higher computational cost.

Another novel feature is energy conservation. So far we were concerned with

- Stability;
- Convergence.

Now we also need

- Energy conservation.

Since the wave equation is energy conserving on the continuous level the development of numerical methods should consider property as well. As we will see, this further reduces the choices of 'good' time-stepping schemes. On the other hand, the wave equation is many important applications and for this reason a zoo of time-stepping schemes (the most prominent being the Newmark scheme) has been proposed.

10.4.2 Variational formulations

A formal derivation of a variational form reads:

$$(\partial_{tt}^2 u,\phi) + (\nabla u,\nabla\phi) = (f,\phi) \quad \forall\phi\in H_0^1.$$

From this variational formulation, we seek

$$u\in L^2(I,H_0^1), \quad \frac{du}{dt}\in L^2(I,L^2), \quad \frac{d^2u}{dt^2}\in L^2(I,H^{-1}).$$

Proposition 10.33. *Let $f\in L^2(I,L^2)$ and the initial data $u_0\in H_0^1$ and $v_0\in L^2$ be given. Then the variational problem has a unique solution*

$$u\in L^2(I,H_0^1), \quad \frac{du}{dt}\in L^2(I,L^2).$$

Moreover, the mapping

$$\{f,u_0,v_0\}\to\{u,v\}, \quad v=\frac{du}{dt}$$

from

$$L^2(I,L^2)\times H_0^1\times L^2 \quad\to\quad L^2(I,H_0^1)\times L^2(I,H^{-1})$$

is linear and continuous.

Proof. See Wloka [76]. □

We also state the promised first-order mixed system. We recall

$$\rho\partial_{tt}^2 u - \nabla\cdot(\nabla u) = f$$

yielding

$$\begin{aligned}\rho\partial_t v - \nabla\cdot(\nabla u) &= f\\ \partial_t u &= v\end{aligned}$$

Then we formulate the variational system:

Formulation 10.34. *Find $(u,v)\in L^2(I,H_0^1)\times L^2(I,L^2)$ with $u_0\in H_0^1$ and $v_0\in L^2$ such that*

$$\begin{aligned}(\rho\partial_t v,\phi)+(\nabla u,\nabla\phi) &= (f,\phi) \quad \forall\phi\in H_0^1,\\ (\partial_t u,\psi) &= (v,\psi) \quad \forall\psi\in L^2.\end{aligned}$$

10.4.3 A space-time formulation

We use the previous formulations to derive space-time variational formulations for the wave equation. We briefly introduce the concept in this section.

Let us consider the second-order hyperbolic PDE:

Formulation 10.35. *Let $\Omega\subset\mathbb{R}^n$ be open and let $I:=(0,T)$ with $T>0$. Find $u:\Omega\times I\to\mathbb{R}$ and $\partial_t u:\Omega\times I\to\mathbb{R}$ such that*

$$\begin{aligned}\rho\partial_t^2 u-\nabla\cdot(a\nabla u) &= f \quad \text{in } \Omega\times I,\\ u &= 0 \quad \text{on } \partial\Omega_D\times I,\\ a\partial_n u &= 0 \quad \text{on } \partial\Omega_N\times I,\\ u &= u_0 \quad \text{in } \Omega\times\{0\},\\ v &= v_0 \quad \text{in } \Omega\times\{0\}.\end{aligned}$$

We now prepare the functions spaces requires for the weak formulation. Let us denote L^2 and H^1 as the usual Hilbert spaces and H^{-1} the dual space to H^1. For the initial functions we assume:

- $u_0\in H_0^1(\Omega)^n$;
- $v_0\in L^2(\Omega)^n$.

For the right hand side source term we assume

- $f\in L^2(I,H^{-1}(\Omega))$, where $L^2(\cdot,\cdot)$ is a Bochner space; see Section 10.2. Specifically, we remind the reader that the initial values are a priori not well defined since they are in L-spaces. However, we have Theorem 10.3 ensuring that the initial data are continuous in time and therefore well defined.

We introduce the following short-hand notation:

- $H:=L^2(\Omega)^n$;
- $V:=H_0^1(\Omega)^n$;
- V^* is the dual space to V.
- $\bar{H}:=L^2(I,H)$;
- $\bar{V}:=\{v\in L^2(I,V)|\ \partial_t v\in\bar{H}\}$.

Theorem 10.36. *If the operator $A:=-\nabla\cdot(a\nabla u)$ satisfies the coercivity estimate:*

$$(Au,u)\geq\beta\|u\|_1^2,\quad u\in V,\quad \beta>0,$$

then there exists a unique weak solution with the properties:

- $u \in \bar{V} \cap C(\bar{I}, V)$;
- $\partial_t u \in \bar{H} \cap C(\bar{I}, H)$;
- $\partial_t^2 u \in L^2(I, V^*)$.

Proof. See Lions and Magenes, Lions or Wloka. □

Definition 10.37. *The previous derivations allow us to define a compact space-time function space for the wave equation:*

$$X := \{v \in \bar{V} \,|\, v \in C(\bar{I}, V), \partial_t v \in C(\bar{I}, H), \partial_t^2 v \in L^2(I, V^*)\}.$$

To design a Galerkin time discretization, we need to get rid of the second-order in time derivative and therefore usually the wave equation is re-written in terms of a mixed first-order system:

Formulation 10.38. *Let $\Omega \subset \mathbb{R}^n$ be open and let $I := (0, T)$ with $T > 0$. Find $u : \Omega \times I \to \mathbb{R}$ and $\partial_t u = v : \Omega \times I \to \mathbb{R}$ such that*

$$\begin{aligned}
\rho \partial_t v - \nabla \cdot (a \nabla u) &= f \quad \text{in } \Omega \times I, \\
\rho \partial_t u - \rho v &= 0 \quad \text{in } \Omega \times I, \\
u &= 0 \quad \text{on } \partial\Omega_D \times I, \\
a \partial_n u &= 0 \quad \text{on } \partial\Omega_N \times I, \\
u &= u_0 \quad \text{in } \Omega \times \{0\}, \\
v &= v_0 \quad \text{in } \Omega \times \{0\}.
\end{aligned}$$

To derive a space-time formulation, we first integrate formally in space:

$$\begin{aligned}
A_v(U)(\psi^v) &= (\rho \partial_t v, \psi^v) + (a \nabla u, \nabla \psi^v) - (f, \psi^v) \\
A_u(U)(\psi^u) &= (\rho \partial_t u, \psi^u) - (\rho v, \psi^u)
\end{aligned}$$

And then in time:

$$\begin{aligned}
\bar{A}_v(U)(\psi^v) &= \int_I \left((\rho \partial_t v, \psi^v) + (a \nabla u, \nabla \psi^v) - (f, \psi^v) \right) dt + (v(0) - v_0, \psi^v(0)) \\
\bar{A}_u(U)(\psi^u) &= \int_I \left((\rho \partial_t u, \psi^u) - (\rho v, \psi^u) \right) dt + (u(0) - u_0, \psi^u(0)).
\end{aligned}$$

The total problem reads:

Formulation 10.39. *Find $U = (u, v) \in X_u \times X_v$ with $X_u = X$ and $X_v := \{w \in \bar{H} | w \in C(\bar{I}, H), \partial_t w \in L^2(I, V^*)\}$ such that*

$$\bar{A}(U)(\Psi) = 0 \quad \forall \Psi = (\psi^u, \psi^v) \in X_u \times X_v$$

where

$$\bar{A}(U)(\Psi) := \bar{A}_v(U)(\psi^v) + \bar{A}_u(U)(\psi^u).$$

Such a formulation is the starting point for the discretization: either in space-time or using a sequential time-stepping scheme and for instance FEM in space.

10.4.4 Energy conservation and consequences for good time-stepping schemes

Another feature of the wave equation is energy conservation. In the ideal case, $f = 0, \rho = 1$, and no dissipation, build the variational form and obtain:

$$(\partial_{tt}^2 u, \phi) + (\nabla u, \nabla \phi) = 0.$$

Using $\phi = \partial_t u$ as test function, we obtain:

$$(\partial_{tt}^2 u, \partial_t u) + (\nabla u, \nabla \partial_t u) = 0,$$

which yields:

$$\frac{1}{2}\frac{d}{dt}(\partial_t u, \partial_t u) + \frac{1}{2}\frac{d}{dt}(\nabla u, \nabla u) = 0,$$

which is

$$\frac{1}{2}\frac{d}{dt}\|\partial_t u\|^2 + \frac{1}{2}\frac{d}{dt}\|\nabla u\|^2 = 0.$$

Remark 10.40. *It holds:*

$$\frac{1}{2}\frac{d}{dt}\int_\Omega \partial_t u \cdot \partial_t u \, dt = \frac{1}{2}\int_\Omega \partial_t(\partial_t u \cdot \partial_t u)\, dt = \frac{1}{2}\int_\Omega 2\partial_t^2 u \cdot \partial_t u \, dt = \int_\Omega \partial_t^2 u \cdot \partial_t u \, dt = (\partial_t^2 u, \partial_t u).$$

Thus the time variation is zero and therefore by integration in time yields

$$\frac{1}{2}\int_0^t \frac{d}{dt}\|\partial_t u\|^2 + \frac{1}{2}\int_0^t \frac{d}{dt}\|\nabla u\|^2 = 0.$$

Thus,

$$\underbrace{\underbrace{\|\partial_t u(t)\|^2}_{=E_{kin}(t)} + \underbrace{\|\nabla u(t)\|^2}_{=E_{pot}(t)}}_{=E_{tot}(t)} = const = \underbrace{\underbrace{\|v_0\|^2}_{=E_{kin}(0)} + \underbrace{\|\nabla u_0\|^2}_{=E_{pot}(0)}}_{=E_{tot}(0)}$$

which is energy conservation

$$E_{tot}(t) = E_{tot}(0)$$

at all times $t \in I$.

Consequently, the construction of **good** time-stepping schemes becomes more involved than for the parabolic case. From the ODE analysis, we recall that for the backward Euler scheme and the Crank-Nicolson scheme are both implicit and unconditionally stable. But the backward Euler scheme is very dissipative and may introduce too much artificial diffusion alternating the energy conservation on the discrete time level. Here, the Crank-Nicolson scheme is much better suited. We recall:

Definition 10.41 (Stability and properties of time stepping schemes)**.** *A good time stepping scheme for time-discretizing partial differential equations should satisfy:*

- *A-stability (local convergence): an example is the Crank-Nicolson scheme (trapezoidal rule);*
- *strict A-stability (global convergence): an example is the shifted Crank-Nicolson scheme;*
- *strong A-stability (smoothing property): examples are backward Euler (in particular even L-stable[8]) and Fractional-step-θ schemes;*
- *small dissipation (energy conservation).*

◇

Let us now describe the properties of the previous schemes in more detail:

- The explicit Euler scheme is cheap in the computational cost since no equation system needs to be solved. However, in order to be numerically stable (very, very) small time steps must be employed, which makes this scheme infeasible for most PDE problems
- A classical scheme for problems with a stationary limit is the (implicit) backward Euler scheme (BE), which is strongly A-stable (but only from first order), robust and dissipative. It is used in numerical Examples, where a stationary limit must be achieved. However, due to its dissipativity this schemes dampes high-frequent temporal parts of the solution too much and for this reason this scheme is not recommended for nonstationary PDE problems.

[8] A time-stepping scheme is L-stable if it is A-stable and the stability function satisfies $|R(z)| \to 0$ for $z \to \infty$. In this respect the Crank-Nicolson scheme is not L-stable since $R(z) \to 1$ for $z \to -\infty$.

- In contrast, the (implicit) Crank-Nicolson scheme is of second order, A-stable, and has very little dissipation but suffers from case-to-case instabilities caused by rough initial and/or boundary data. These properties are due to weak stability (it is not *strongly* A-stable). A variant of the Crank-Nicolson scheme is called *shifted* Crank-Nicolson scheme, is analyzed in Rannacher et al. [37, 50], which allows for global stability of the solution [9]. In particular, Rannacher analyzed in [49] the shifted Crank-Nicolson scheme for linear evolution equations and how they can be modified in order to make them suitable for long-term computations (without reducing second order accuracy!!).
- The fourth scheme summarizes the advantages of the previous two and is known as the Fractional-Step-θ scheme for computing unsteady-state simulations [28]. Roughly-speaking it consists of summarizing three Crank-Nicolson steps and has therefore the same accuracy and compuational cost as the Crank-Nicolson scheme. However, it is more robust, i.e., it is strongly A-stable (as backward Euler) but has 2nd order accuracy as Crank-Nicolson, and is therefore well-suited for computing solutions with rough initial data and long-term computations for problems. This property also holds for ALE-transformed fluid equations, which is demonstrated in a numerical test below. We also refer the reader to a modification of the Fractional-Step-θ scheme [68].

10.4.5 One-step-theta times-stepping for the wave equation

To finish, we aim to apply a One-Step-theta scheme and therefore, we need to work with the first-order mixed-system. Then, we apply the One-step-Theta scheme to Formulation 10.34 idea to discretize in time:

$$(\rho\frac{v^n - v^{n-1}}{k},\phi) + \theta(\nabla u^n,\nabla\phi) + (1-\theta)(\nabla u^{n-1},\nabla\phi) = \theta(f^n,\phi) + (1-\theta)(f^{n-1},\phi)$$
$$(\frac{u^n - u^{n-1}}{k},\psi) = \theta(v^n,\psi) + (1-\theta)(v^{n-1},\psi)$$

And finally, we discretize in space using a finite element scheme:

$$(\rho\frac{v_h^n - v_h^{n-1}}{k},\phi_h) + \theta(\nabla u_h^n,\nabla\phi_h) + (1-\theta)(\nabla u_h^{n-1},\nabla\phi_h) = \theta(f^n,\phi_h) + (1-\theta)(f^{n-1},\phi_h)$$
$$(\frac{u_h^n - u_h^{n-1}}{k},\psi_h) = \theta(v_h^n,\psi_h) + (1-\theta)(v_h^{n-1},\psi_h)$$

Re-arringing given data terms on the right hand side yields:

$$k^{-1}(\rho v_h^n,\phi_h) + \theta(\nabla u_h^n,\nabla\phi_h) = k^{-1}(\rho v_h^{n-1},\phi_h)\theta(f^n,\phi_h) + (1-\theta)(f^{n-1},\phi_h) - (1-\theta)(\nabla u_h^{n-1},\nabla\phi_h)$$
$$k^{-1}(u_h^n,\psi_h) - \theta(v_h^n,\psi_h) = k^{-1}(u_h^{n-1},\psi_h) + (1-\theta)(v_h^{n-1},\psi_h)$$

To obtain the system matrix and the linear system, we proceed as in the parabolic case by considering that we now seek two solutions u_h^n and v_h^n.

With the previous discretizations, we can now define

Algorithm 10.42. *Given the initial guesses u_h^0 and v_h^0 and given at each time the right hand side forces $f^n := f(t^n)$ and $k_n = t^n - t^{n-1}$ we solve for each $n = 1, 2, 3, \ldots, N$:*

$$k_n^{-1}(\rho v_h^n,\phi_h) + \theta(\nabla u_h^n,\nabla\phi_h) = k_n^{-1}(\rho v_h^{n-1},\phi_h)\theta(f^n,\phi_h) + (1-\theta)(f^{n-1},\phi_h) - (1-\theta)(\nabla u_h^{n-1},\nabla\phi_h)$$
$$k_n^{-1}(u_h^n,\psi_h) - \theta(v_h^n,\psi_h) = k_n^{-1}(u_h^{n-1},\psi_h) + (1-\theta)(v_h^{n-1},\psi_h)$$

for $(\phi_h,\psi_h) \in V_h \times W_h$ with $V_h \times W_h \subset H_0^1 \times L^2$.

Remark 10.43. *An abstract generalization of the One-Step-Theta scheme has been formulated in [70][Chapter 5].*

[9] A famous alternative to the shifted Crank-Nicolson scheme is Rannacher's time-marching technique in which the unshifted Crank-Nicolson scheme is augmented with backward Euler steps for stabilization and in particular for irregular initial data [49]. Adding backward Euler steps within Crank-Nicolson does also stabilize during long-term computations [37, 49].

10.4.6 Stability analysis / energy conservation on the time-discrete level

Similar to parabolic problems, we perform a stability analysis, but this time with a focus on energy conservation on the time-discretized level.

We set

$$0 = t_0 < t_1 < \ldots < t_N = T, \quad k = t_n - t_{n-1}$$

The model problem is:

Formulation 10.44. *Let $V = H_0^1$ and $W = L^2$. Given $u^{n-1} \in V$ and $v^{n-1} \in W$ we solve for $n = 1, \ldots, N$ using the Crank-Nicolson scheme: Find $(u^n, v^n) \in V \times W$ such that*

$$(v^n - v^{n-1}, \phi) + \frac{1}{2}k(\nabla u^n + \nabla u^{n-1}, \nabla\phi) = \frac{1}{2}(f^n + f^{n-1}, \phi) \quad \forall \phi \in V,$$
$$(u^n - u^{n-1}, \psi) - \frac{1}{2}k(v^n + v^{n-1}, \psi) = 0 \quad \forall \psi \in W.$$

Proposition 10.45. *Setting $f^n = 0$ for all $n = 1, \ldots, N$ in the previous formulation, we have*

$$\underbrace{\|\nabla u^n\|^2}_{E^n_{pot}} + \underbrace{\|v^n\|^2}_{E^n_{kin}} = \underbrace{\|\nabla u^{n-1}\|^2}_{E^{n-1}_{pot}} + \underbrace{\|v^{n-1}\|^2}_{E^{n-1}_{kin}}$$

and therefore

$$E^n_{tot} = E^{n-1}_{tot}.$$

Proof. We start from Formulation 10.44 and choose once again clever test functions:

$$\phi = u^n - u^{n-1}, \quad \psi = v^n - v^{n-1}.$$

Then:

$$(v^n - v^{n-1}, u^n - u^{n-1}) + \frac{1}{2}k(\nabla u^n + \nabla u^{n-1}, \nabla(u^n - u^{n-1})) = 0,$$
$$(u^n - u^{n-1}, v^n - v^{n-1}) - \frac{1}{2}k(v^n + v^{n-1}, v^n - v^{n-1}) = 0.$$

We subtract the second equation from the first equation. Due to the bi-linear property the first terms cancel each other:

$$\frac{1}{2}k(\nabla u^n + \nabla u^{n-1}, \nabla(u^n - u^{n-1})) + \frac{1}{2}k(v^n + v^{n-1}, v^n - v^{n-1}) = 0.$$

Next, the cross terms cancel due to different signs and it remains:

$$\frac{1}{2}k(\nabla u^n, \nabla u^n) + \frac{1}{2}k(v^n, v^n) = \frac{1}{2}k(\nabla u^{n-1}, \nabla u^{n-1}) + \frac{1}{2}k(v^{n-1}, v^{n-1}),$$

therefore we easily see

$$\|\nabla u^n\| + \|v^n\| = \|\nabla u^{n-1}\| + \|v^{n-1}\|.$$

Voilà! We are finished. □

We next show that indeed the implicit Euler scheme introduces artificial viscosity and loose energy over time:

Formulation 10.46. *Let $V = H_0^1$ and $W = L^2$. Given $u^{n-1} \in V$ and $v^{n-1} \in W$ we solve for $n = 1, \ldots, N$ using the implicit Euler scheme: Find $(u^n, v^n) \in V \times W$ such that*

$$(v^n - v^{n-1}, \phi) + k(\nabla u^n, \nabla\phi) = (f^n, \phi) \quad \forall \phi \in V,$$
$$(u^n - u^{n-1}, \psi) - k(v^n, \psi) = 0 \quad \forall \psi \in W.$$

Proposition 10.47. *Setting $f^n = 0$ for all $n = 1, \dots, N$ in the previous formulation, we have*

$$\underbrace{\|\nabla u^n\|^2}_{E^n_{pot}} + \underbrace{\|v^n\|^2}_{E^n_{kin}} \leq \underbrace{\|\nabla u^{n-1}\|^2}_{E^{n-1}_{pot}} + \underbrace{\|v^{n-1}\|^2}_{E^{n-1}_{kin}}$$

and therefore

$$E^n_{tot} \leq E^{n-1}_{tot}.$$

Therefore, energy is dissipated due to the 'wrong' numerical scheme.

Proof. The proof is basically the same as before. We start from Formulation 10.46 and choose the same test functions as before:

$$\phi = u^n - u^{n-1}, \quad \psi = v^n - v^{n-1}.$$

Then:

$$(v^n - v^{n-1}, u^n - u^{n-1}) + k(\nabla u^n, \nabla(u^n - u^{n-1})) = 0,$$
$$(u^n - u^{n-1}, v^n - v^{n-1}) - k(v^n, v^n - v^{n-1}) = 0.$$

We subtract the second equation from the first equation. Due to the bi-linear property the first terms cancel each other:

$$k(\nabla u^n, \nabla(u^n - u^{n-1})) + k(v^n, v^n - v^{n-1}) = 0.$$

Next,

$$k(\nabla u^n, \nabla u^n) - k(\nabla u^n, \nabla u^{n-1}) + k(v^n, v^n) - k(v^n, v^{n-1}) = 0.$$

Here, bring the cross terms on the right hand side and apply once again Youngs' inequality:

$$k(\nabla u^n, \nabla u^n) + k(v^n, v^n) \leq \frac{1}{2}\|\nabla u^n\| + \frac{1}{2}\|\nabla u^{n-1}\| + \frac{1}{2}\|v^n\| + \frac{1}{2}\|\nabla v^{n-1}\|.$$

Re-arranging terms yields the assertion. □

10.4.7 Summary of stability, convergence and energy conservation

We summarize our findings (see for more details in [5, 10, 32]). Our first conclusions is: In extension to elliptic and parabolic problems, the numerical discretization of the wave equation asks for temporal schemes that satisfy energy conservation.

Summarizing everything yields:

- *Stability in the L^2-norm:* the One-Step-θ scheme is unconditionally stable, i.e., there is no time step restriction on k if and only if $\theta \in [\frac{1}{2}, 1]$.
- *Convergence* It holds (can be proven using tools from ODE numerical analysis):
  - $\theta \in [0, 0.5) \cup (0, 5, 1]$: linear convergence $O(k)$
  - $\theta = 0.5$: quadratic convergence $O(k^2)$.
- *Energy conservation:* the one-step-θ scheme preserves energy only for the choice $\theta = \frac{1}{2}$. For $\theta > \frac{1}{2}$ (e.g., the implicit Euler scheme for the choice $\theta = 1$) the scheme dissipates energy as shown in the previous subsection. For $\theta < \frac{1}{2}$ energy conservation depends, but in all cases schemes will be unstable.

Consequently, the Crank-Nicolson scheme is an optimal time-stepping scheme for hyperbolic equations.

When explicit schemes are taken, possible restrictions with respect to the time-step size are weaker for hyperbolic problems than for parabolic differential equations [32]; namely

Definition 10.48 (Courant-Friedrichs-Levy - CFL)**.** *A necessary condition for explicit time stepping schemes is the Courant-Friedrichs-Levy (CFL) condition, i.e., the time step size k is dependent on material parameters and the spatial discretization parameter h:*

- *Parabolic problems: $k \leq \frac{1}{2}ch^2$;*
- *Hyperbolic problems: $k \leq a^{-1}h$, where a is of order of the elasticity constant in the wave operator.*

10.5 Numerical tests: scalar wave equation

We provide some impressions about the scalar-valued wave equation. The setting is simular to Section 6.13.1. We take $T = 10s$ and $k = 1s$ using the Crank-Nicolson scheme. The initial solutions u^0 and v^0 are taken to be zero. In the Figures 39 and 40 we show the evolution of the displacements and the velocity.

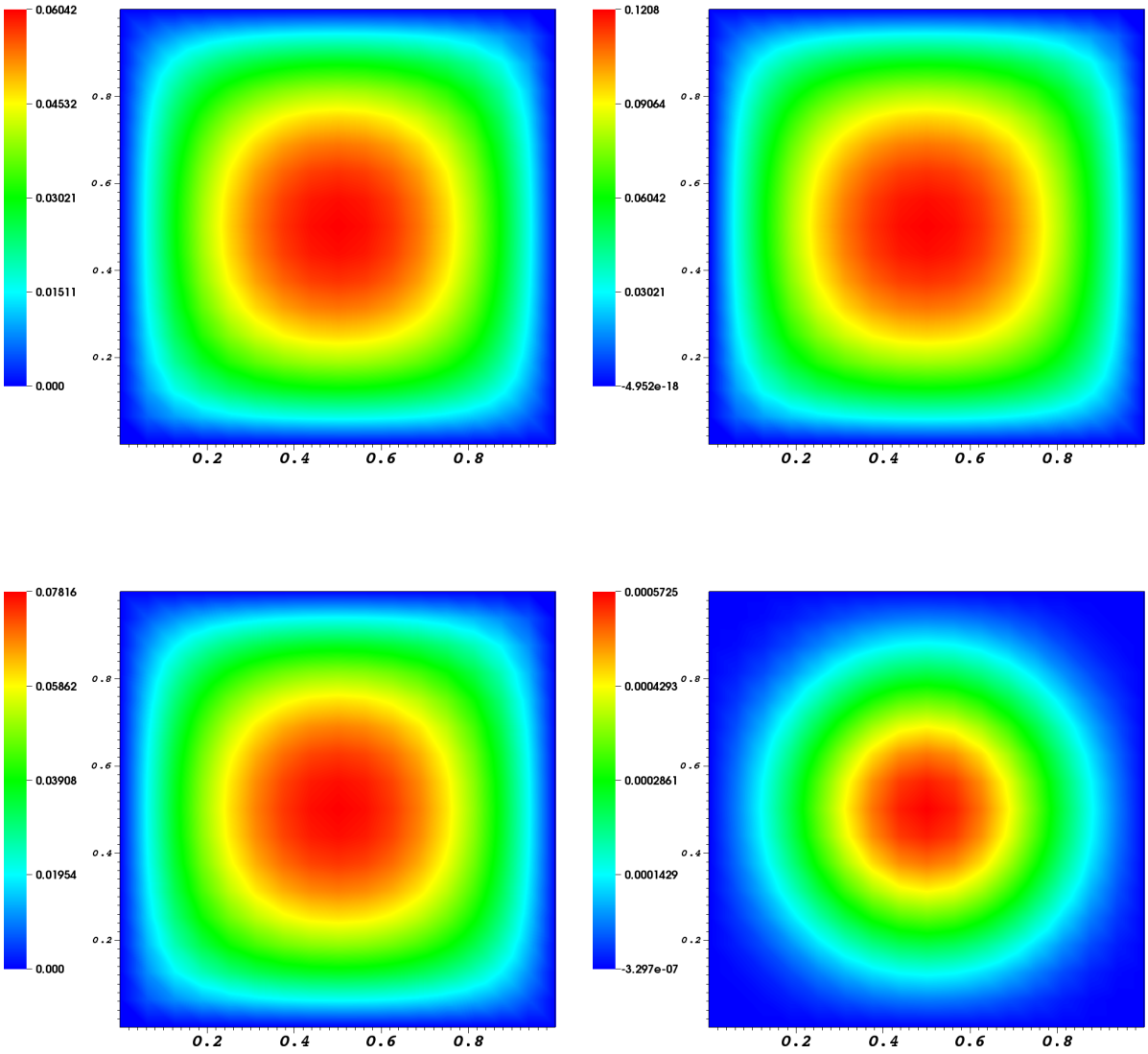

Figure 39: Scalar wave equation: Evolution of displacements (left) and velocities (right).

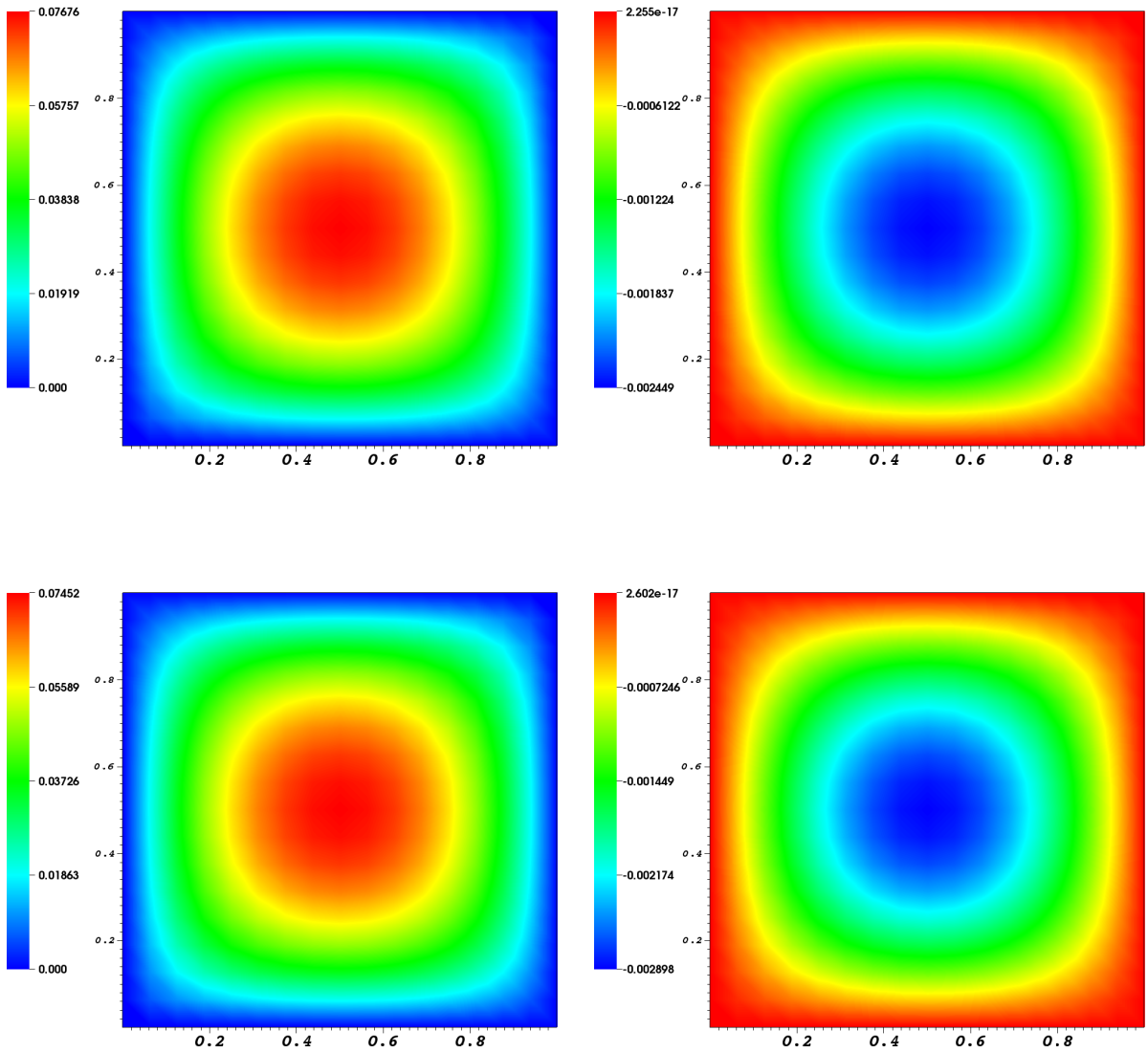

Figure 40: Scalar wave equation: Evolution of displacements (left) and velocities (right).

10.6 Numerical tests: elastic wave equation

We extend the scalar wave equation to the elastic wave equation. Specifically, the linearized elastic wave equation is augmented with the acceleration term:

Problem 10.49 (Discretized nonstationary linearized elasticity / Elastodynamics)**.** *Given the initial conditions u_0 and v_0 and a vector-valued right hand side $\hat{f} = (f_x, f_y)$. Find $(\hat{u}_h, \hat{v}_h) \in \hat{V}_h^0 \times \hat{W}_h$ such that*

$$(\hat{\rho}\partial_t \hat{v}_h, \hat{\varphi}_h) + (\widehat{\Sigma}_{h,lin}, \widehat{\nabla}\hat{\varphi}_h) = (\hat{f}, \hat{\varphi}_h) \quad \forall \hat{\varphi}_h \in \hat{V}_h^0,$$
$$(\partial_t \hat{u}_h, \hat{\psi}_h) - (\hat{v}_h, \hat{\psi}_h) = 0 \quad \forall \hat{\psi}_h \in \hat{W}_h,$$

where $\hat{V}_h^0 \subset V := \{u \in H^1 |\ u = 0 \text{ on } \Gamma_{left}\}$, here Γ_{left} is the left boundary of the specimen and and $\hat{W}_h \subset L^2$ and

$$\widehat{\Sigma}_{h,lin} = 2\mu \widehat{E}_{h,lin} + \lambda tr\widehat{E}_{h,lin}\hat{I},$$

and the identity matrix $\hat{I}$ and

$$\widehat{E}_{h,lin} = \frac{1}{2}(\widehat{\nabla}\hat{u}_h + \widehat{\nabla}\hat{u}_h^T).$$

10.6.1 Constant force - stationary limit

First, we redo the 2D test from Section 9.4 in which the left boundary is fixed and $\hat{\rho} = 1$ and $(f_x, f_y) = (0, -9.81)$. These findings are displayed in Figure 41. As time-stepping scheme, the Crank-Nicolson scheme, i.e., $\theta = 0.5$ is used. The time step size is $k = 1s$ and the total time is $T = 10s$. The initial solutions are $u_0 = 0$ and $v_0 = 0$.

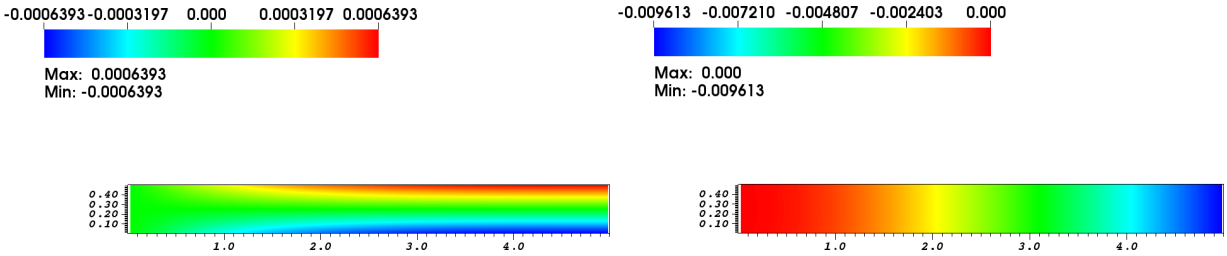

Figure 41: 2D nonstationary linearized elasticity with a constant right hand side $(f_x, f_y) = (0, -9.81)$. At left, the u_x solution is displayed. At right, the u_y solution is displayed.

10.6.2 Time-dependent force - Crank-Nicolson scheme

In the second test, we prescribe the right hand is as

$$(f_x^n, f_y^n) = (0, \sin(t))^T,$$

with the uniform time step size $k = 0.02s$ and $N = 314$ (that is 314 time step solutions). The total time is $T = k * N = 6.28s$. The results at $N = 2, 157, 314$ are displayed in Figure 42.

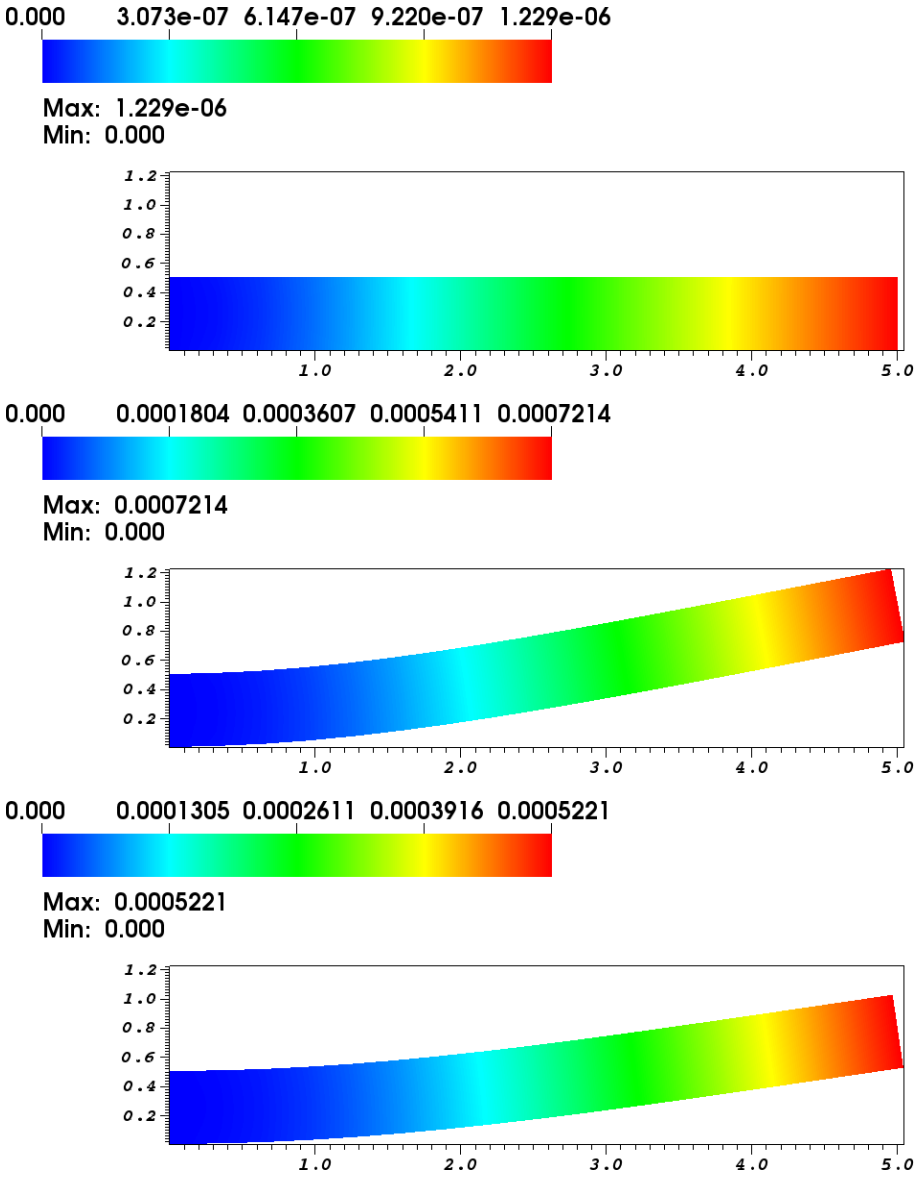

Figure 42: 2D nonstationary linearized elasticity with $(f_x^n, f_y^n) = (0, \sin(t))^T$. The u_y solution is displayed at $N = 2, 157, 314$. The displacements are amplified by a factor of 1000 such that a visible deformation of the elastic beam can be seen. Here, the Crank-Nicolson scheme was used.

10.6.3 Time-dependent force - backward Euler scheme

In the third test, we use the backward Euler scheme. From the theory we know that this scheme is too dissipative and introduces artificial numerical damping. Indeed, observing Figure 43, we see that the displacements at $N = 157$ and $N = 314$ are much smaller than in the corresponding Crank-Nicolson test case.

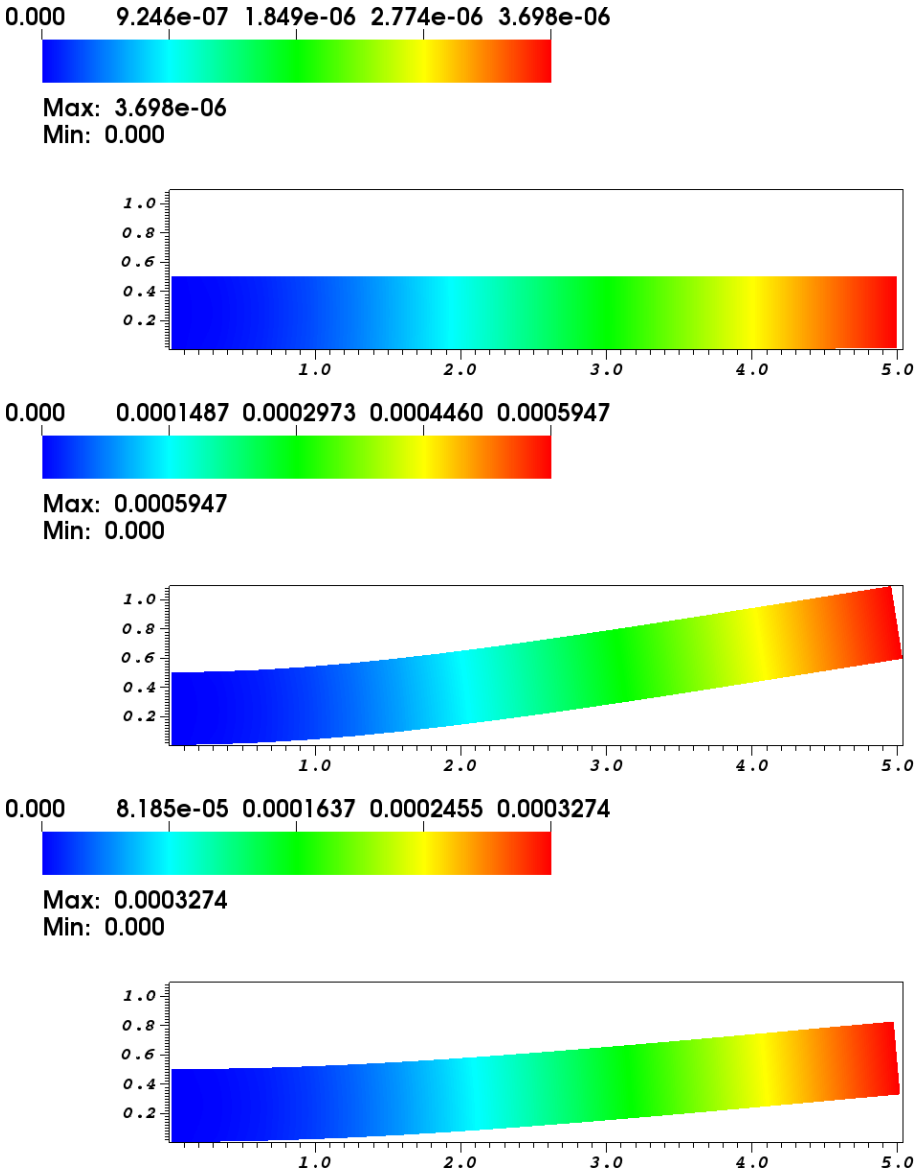

Figure 43: 2D nonstationary linearized elasticity with $(f_x^n, f_y^n) = (0, \sin(t))^T$. The u_y solution is displayed at $N = 2, 157, 314$. The displacements are amplified by a factor of 1000 such that a visible deformation of the elastic beam can be seen. Here, the backward Euler scheme was used.

10.7 Chapter summary and outlook

In this chapter, we have investigated time-dependent problems. In the next chapter we leave time-dependence and go back to stationary problems, but include now nonlinear phenomena. These lead to fascinating research topics for which theory and discretizations only have partially developed (since extremely challenging!), but allow for various future research directions.

11 Nonlinear problems

Most relevant problems are nonlinear and such settings are really fascinating and very important. For this reason, a first class on numerical methods for partial differential, with the risk of being dense, should at least strive how nonlinear problems can be approached.

We introduce nonlinear problems in terms of the so-called p-Laplace problem, which has important applications, but reduces to the Poisson problem for $p = 2$. This setting has no time dependence.

11.1 Differentiation in Banach spaces

We have learned about differentiation in Banach spaces in Section 7.3.4. Examples are:

$$T(u) = u^2$$

then

$$T'_u(u)(h) = 2u \cdot h.$$

Or for semi-linear forms:

$$a(u)(\phi) = (u^2, \phi)$$

we want to differentiate in the first argument:

$$a'_u(u)(h, \phi) = (2u \cdot h, \phi).$$

11.2 Linearization techniques

We discuss linearization techniques in the following subsections. The idea is to provide algorithmic frameworks that serve for the implementation. Concerning Newton's method for general problems there is not that much theory; see e.g., [22]. In general one can say that in many problems the theoretical assumptions are not met, but nevertheless Newton's method works well in practice.

11.2.1 Fixed-point iteration

In ODE computations, applying a fixed-point theorem, namely the Banach fixed point theorem, is called a **Picard iteration**. The basic idea is to introduce an iteration using an index k and to linearize the nonlinear terms by taking these terms from the previous iteration $k - 1$.

This is best illustrated in terms of an example. Let

$$-\Delta u + u^2 = f$$

The variational formulation reads:

$$(\nabla u, \nabla \phi) + (u^2, \phi) = (f, \phi) \quad \forall \phi \in V.$$

An iterative scheme is constructed as follows:

Algorithm 11.1 (Fixed-point iterative scheme)**.** *For $k = 1, 2, 3, \ldots$ we seek $u^{k+1} \in V$ such that*

$$(\nabla u^k, \nabla \phi) + (u^k u^{k-1}, \phi) = (f, \phi) \quad \forall \phi \in V,$$

until a stopping criterion is fulfilled (choice one out of four):

- *Error criterion:*

$$\|u^k - u^{k-1}\| < TOL, \quad \textit{(absolute)}$$
$$\|u^k - u^{k-1}\| < TOL\|u^k\|, \quad \textit{(relative)}$$

- *Residual criterion:*

$$\|(\nabla u^k, \nabla\phi_i) + ([u^2]^k, \phi_i) - (f, \phi_i)\| < TOL, \quad \textit{(absolute)}$$
$$\|(\nabla u^k, \nabla\phi_i) + ([u^2]^k, \phi_i) - (f, \phi_i)\| < TOL\|(f, \phi_i)\|, \quad \textit{(relative)}$$

for all $i = 1, \ldots, dim(V_h)$.

Remark 11.2. *For time-dependent PDEs, a common linearization can be*

$$(u^2, \phi) \to (uu^{n-1}, \phi)$$

where $u^{n-1} := u(t^{n-1})$ *is the previous time step solution. In this case, no additional fixed-point iteration needs to be constructed.*

11.2.2 Newton's method in $\mathbb{R}$ - the Newton-Raphson method

Let $f \in C^1[a, b]$ with at least one point $f(x) = 0$, and $x_0 \in [a, b]$ be a so-called initial guess. The task is to find $x \in \mathbb{R}$ such that

$$f(x) = 0.$$

In most cases it is impossible to calculate x explicitly. Rather we construct a sequence of iterates $(x_k)_{k\in\mathbb{R}}$ and hopefully reach at some point

$$|f(x_k)| < TOL, \quad \text{where } TOL \text{ is small, e.g., } TOL = 10^{-10}.$$

What is true for all Newton derivations in the literature is that one has to start with a Taylor expansion. In our lecture we do this as follows. Let us assume that we are at x_k and can evaluate $f(x_k)$. Now we want to compute this next iterate x_{k+1} with the unknown value $f(x_{k+1})$. Taylor gives us:

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + o(x_{k+1} - x_k)^2$$

We assume that $f(x_{k+1}) = 0$ (or very close to zero $f(x_{k+1}) \approx 0$). Then, x_{k+1} is the sought root and neglecting the higher-order terms we obtain:

$$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k).$$

Thus:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots. \tag{215}$$

This iteration is possible as long as $f'(x_k) \neq 0$.

Remark 11.3 (Relation to Section 8.3)**.** *We see that Newton's method can be written as*

$$x_{k+1} = x_k + d_k, \quad k = 0, 1, 2, \ldots,$$

where the search direction is

$$d_k = -\frac{f(x_k)}{f'(x_k)}.$$

The iteration (215) terminates if a stopping criterion

$$\frac{|x_{k+1} - x_k|}{|x_k|} < TOL, \quad \text{or} \quad |x_{k+1} - x_k| < TOL, \tag{216}$$

or

$$|f(x_{k+1})| < TOL \tag{217}$$

is fulfilled. All these TOL do not need to be the same, but sufficiently small and larger than machine precision.

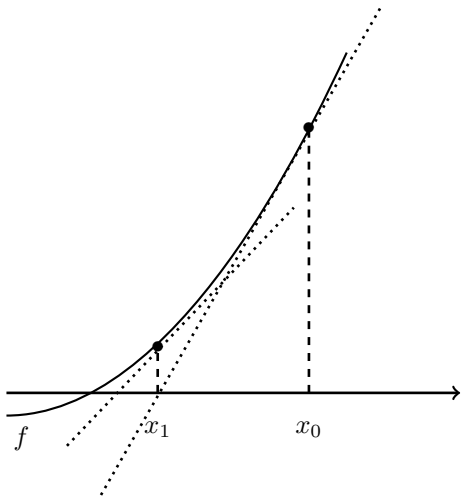

Figure 44: Geometrical interpretation of Newton's method.

Remark 11.4. *Newton's method belongs to fix-point iteration schemes with the iteraction function:*

$$F(x) := x - \frac{f(x)}{f'(x)}. \tag{218}$$

For a fixed point $\hat{x} = F(\hat{x})$ it holds: $f(\hat{x}) = 0$. Compare again to Section 8.3.

The main results is given by:

Theorem 11.5 (Newton's method)**.** *The function $f \in C^2[a,b]$ has a root $\hat{x}$ in the interval $[a,b]$ and*

$$m := \min_{a \le x \le b} |f'(x)| > 0, \quad M := \max_{a \le x \le b} |f''(x)|.$$

Let $\rho > 0$ such that

$$q := \frac{M}{2m}\rho < 1, \quad K_\rho(\hat{x}) := \{x \in \mathbb{R} : |x - \hat{x}| \le \rho\} \subset [a,b].$$

Then, for any starting point $x_0 \in K_\rho(\hat{x})$, the sequence of iterations $x_k \in K_\rho(\hat{x})$ converges to the root $\hat{x}$. Furthermore, we have the a priori estimate

$$|x_k - \hat{x}| \le \frac{2m}{M} q^{2^k}, \quad k \in \mathbb{N},$$

and a posteriori estimate

$$|x_k - \hat{x}| \le \frac{1}{m}|f(x_k)| \le \frac{M}{2m}|x_k - x_{k+1}|^2, \quad k \in \mathbb{N}.$$

Often, Newton's method is formulated in terms of a defect-correction scheme.

Definition 11.6 (Defect)**.** *Let $\tilde{x} \in \mathbb{R}$ an approximation of the solution $f(x) = y$. The defect (or similarly the residual) is defined as*

$$d(\tilde{x}) = y - f(\tilde{x}).$$

Definition 11.7 (Newton's method as defect-correction scheme)**.**

$$\begin{aligned} f'(x_k)\delta x = d_k, \quad d_k := y - f(x_k), \\ x_{k+1} = x_k + \delta x, \quad k = 0, 1, 2, \ldots. \end{aligned}$$

The iteration is finished with the same stopping criterion as for the classical scheme. To compute the update δx we need to invert $f'(x_k)$:

$$\delta x = (f'(x_k))^{-1} d_k.$$

This step seems trivial but is the most critical one if we deal with problems in $\mathbb{R}^n$ with $n > 1$ or in function spaces. Because here, the derivative becomes a matrix. Therefore, the problem results in solving a linear equation system of the type $A\delta x = b$ and computing the inverse matrix A^{-1} is an expensive operation. $\diamond$

Remark 11.8. *This previous forms of Newton's method are already very close to the schemes that are used in research. One simply extends from $\mathbb{R}^1$ to higher dimensional cases such as nonlinear PDEs or optimization. The 'only' aspects that are however big research topics are the choice of*

- *good initial Newton guesses;*
- *globalization techniques.*

Two very good books on these topics, including further materials as well, are [22, 48].

11.2.3 Newton's method: overview. Going from $\mathbb{R}$ to Banach spaces

Overview:

- Newton-Raphson (1D), find $x \in \mathbb{R}$ via iterating $k = 0, 1, 2, \ldots$ such that $x_k \approx x$ via:

$$\begin{aligned} \text{Find } \delta x \in \mathbb{R}: \quad & f'(x_k)\delta x = -f(x_k), \\ \text{Update:} \quad & x_{k+1} = x_k + \delta x. \end{aligned}$$

- Newton in $\mathbb{R}^n$, find $x \in \mathbb{R}^n$ via iterating $k = 0, 1, 2, \ldots$ such that $x_k \approx x$ via:

$$\begin{aligned} \text{Find } \delta x \in \mathbb{R}^n: \quad & F'(x_k)\delta x = -F(x_k), \\ \text{Update:} \quad & x_{k+1} = x_k + \delta x. \end{aligned}$$

  Here we need to solve a linear equation system to compute the update $\delta x \in \mathbb{R}^n$.

- Banach spaces, find $u \in V$, with $dim(V) = \infty$, via iterating $k = 0, 1, 2, \ldots$ such that $u_k \approx u$ via:

$$\begin{aligned} \text{Find } \delta u \in V: \quad & F'(u_k)\delta u = -F(u_k), \\ \text{Update:} \quad & u_{k+1} = u_k + \delta u. \end{aligned}$$

  Such a problem needs to be discretized and results again in solving a linear equation system in the defect step.

- Banach spaces, applied to variational formulations, find $u \in V$, with $dim(V) = \infty$, via iterating $k = 0, 1, 2, \ldots$ such that $u_k \approx u$ via:

$$\begin{aligned} \text{Find } \delta u \in V: \quad & a'(u_k)(\delta u, \phi) = -a(u_k)(\phi), \\ \text{Update:} \quad & u_{k+1} = u_k + \delta u. \end{aligned}$$

  As before, the infinite-dimensional problem is discretized resulting in solving a linear equation system in the defect step.

11.2.4 A basic algorithm for a residual-based Newton method

In this type of methods, the main criterion is a decrease of the residual in each step:

Algorithm 11.9 (Residual-based Newton's method). *Given an initial guess x_0. Iterate for $k = 0, 1, 2, \ldots$ such that*

$$\begin{aligned} \text{Find } \delta x \in \mathbb{R}^n: \quad & F'(x_k)\delta x_k = -F(x_k), \\ \text{Update:} \quad & x_{k+1} = x_k + \lambda_k \delta x_k, \end{aligned}$$

with $\lambda_k \in (0, 1]$ (see the next sections how λ_k can be determined). A full Newton step corresponds to $\lambda_k = 1$. The criterion for convergence is the contraction of the residuals measured in terms of a discrete vector norm:

$$\|F(x_{k+1})\| < \|F(x_k)\|.$$

In order to save some computational cost, close to the solution x^*, intermediate simplified Newton steps can be used. In the case of $\lambda_k = 1$ we observe

$$\theta_k = \frac{\|F(x_{k+1})\|}{\|F(x_k)\|} < 1.$$

If $\theta_k < \theta_{max}$, e.g., $\theta_{max} = 0.1$, then the old Jacobian $F'(x_k)$ is kept and used again in the next step $k + 1$. Otherwise, if $\theta_k > \theta_{max}$, the Jacobian will be assembled. Finally the stopping criterion is one of the following (relative preferred!):

$$\begin{aligned} &\|F(x_{k+1})\| \leq TOL_N \quad \textit{(absolute)} \\ &\|F(x_{k+1})\| \leq TOL_N \|F(x_0)\| \quad \textit{(relative)} \end{aligned}$$

If fulfilled, set $x^* := x_{k+1}$ and the (approximate) root x^* of the problem $F(x) = 0$ is found.

11.3 Newton's method for variational formulations

In this section, we describe in detail an Newton method for solving nonlinear nonstationary PDE problems in variational form. That is we have the following indices:

- h for spatial discretization.
- n for the current time step solution.
- j Newton iteration.
- l (optional): line search iterations.

Therefore, we deal with the following problem on the discrete level:

$$a(u_h^n)(\phi) = l(\phi) \quad \forall \phi \in V_h,$$

which is solved with a Newton-like method. We can express this relation in terms of the defect:

$$d := l(\phi) - a(u_h^n)(\phi) = 0 \quad \forall \phi \in V_h.$$

Algorithm 11.10 (Basic Newton method as defect-correction problem). *Given an initial Newton guess $u_h^{n,0} \in V_h$, find for $j = 0, 1, 2, \ldots$ the update $\delta u_h^n \in V_h$ of the linear defect-correction problem*

$$a'(u_h^{n,j})(\delta u_h^n, \phi) = -a(u_h^{n,j})(\phi) + l(\phi), \tag{219}$$

$$u_h^{n,j+1} = u_h^{n,j} + \lambda \delta u_h^n, \tag{220}$$

$$\textit{Check } \| - a(u_h^{n,j})(\phi) + l(\phi)\| < TOL_N, \tag{221}$$

$$\textit{or check } \| - a(u_h^{n,j})(\phi) + l(\phi)\| < TOL_N \| - a(u_h^{n,0})(\phi) + l(\phi)\|, \tag{222}$$

with a line search damping parameter $\lambda \in (0, 1]$ and a Newton tolerance TOL_N, e.g., $TOL_N = 10^{-10}$. For $\lambda = 1$, we deal with a full Newton step. Adapting λ and choosing $0 < \lambda < 1$ helps to achieve convergence of Newton's method for certain nonlinear problems when a full Newton step does not work.

Remark 11.11 (Initial Newton guess for time-dependent problems)**.** *In time-dependent problems the 'best' initial Newton guess is the previous time step solution; namely*

$$u_h^{n,0} := u_h^{n-1}.$$

Remark 11.12 (Initial Newton guess in general)**.** *When possible the solution should be obtained in nested iterations. We start on a coarse mesh with mesh size h_0 and here $u_{h_0}^0 = 0$ when nothing better is available. Then, we compute the solution u_{h_0}. After Newton converged, we project u_{h_0} on the next mesh and use $u_{h_1}^0 := u_{h_0}$ as initial Newton guess and repeat the entire procedure.*

Remark 11.13 (Dirichlet boundary conditions)**.** *In Newton's method, non-homogeneous Dirichlet boundary conditions are only prescribed on the initial guess $u_h^{n,0}$ and in all further updates non-homogeneous Dirichlet conditions are replaced by homoegeneous Dirichlet conditions. The reasons is that we only need to prescribe these conditions once per Newton iteration and not several times, which would lead to wrong solution.*

In the following we present a slightly more sophisticated algorithm with two additional features:

- backtracking line search to choose λ;
- simplified Newton steps in order to avoid too many assemblings of the Jacobian matrix $a'(u_h^{n,j})(\delta u_h^n, \phi)$.

The second point is a trade-off: the less assemblings we do, the more Newton's performance deteriorates and quadratic convergence gets lost resulting in a fixed-point-like scheme in which many more nonlinear iterations j are required to achieve the tolerance. On the other hand, when we do not construct the Jacobian, we save the assembling and can keep working with the previous matrix. For details, we refer for instance to [22].

Algorithm 11.14 (Backtracking line search)**.** *A simple strategy is to modify the update step in* (219) *as follows: For given $\lambda \in (0,1)$ determine the minimal $l^* \in \mathbb{N}$ via $l = 0, 1, \ldots, N_l$, such that*

$$\|R(u_{h,l}^{n,j+1})\|_\infty < \|R(u_{h,l}^{n,j})\|_\infty,$$
$$u_{h,l}^{n,j+1} = u_h^{n,j} + \lambda^l \delta u_h^n.$$

For the minimal l, we set

$$u_h^{n,j+1} := u_{h,l^*}^{n,j+1}.$$

In this context, the nonlinear residual $R(\cdot)$ is defined as

$$R(u_h^{n,j})(\phi_i) := a(u_h^{n,j})(\phi_i) - l(\phi_i) \quad \forall \phi_i \in V_h,$$

and

$$\|R(u_h^{n,j})\|_\infty := \max_{i \in [1, dim(V_h)]} \{R(u_h^{n,j})(\phi_i)\}$$

The final full Newton algorithm based on a contraction of the residuals reads:

Algorithm 11.15 (Residual-based Newton's method with backtracking line search and simplified Newton steps)**.** *Given an initial Newton guess $u_h^{n,0} \in V_h$. For the iteration steps $j = 0, 1, 2, 3, \ldots$:*

1. *Find $\delta u_h^n \in V_h$ such that*

$$a'(u_h^{n,j})(\delta u_h^n, \phi) = -a(u_h^{n,j})(\phi) + l(\phi) \quad \forall \phi \in V_h, \tag{223}$$
$$u_h^{n,j+1} = u_h^{n,j} + \lambda_j \delta u_h^n, \tag{224}$$

   for $\lambda_j = 1$.

2. *The criterion for convergence is the contraction of the residuals:*

$$\|R(u_h^{n,j+1})\|_\infty < \|R(u_h^{n,j})\|_\infty. \tag{225}$$

3. *If* (225) *is violated, re-compute in* (223) $u_{h,l}^{n,j+1}$ *by choosing* $\lambda_j^l = 0.5$*, and compute for* $l = 1, ..., l_M$ *(e.g.* $l_M = 5$*) a new solution*

$$u_h^{n,j+1} = u_h^{n,j} + \lambda_j^l \delta u_h^n,$$

until (225) *is fulfilled for a* $l^* < l_M$ *or* l_M *is reached. In the latter case, no convergence is obtained and the program aborts.*

4. *In case of* $l^* < l_M$ *we check next the stopping criterion:*

$$\|R(u_h^{n,j+1})\|_\infty \leq TOL_N, \quad \textit{(absolute)} \|R(u_h^{n,j+1})\|_\infty \qquad \leq TOL_N \|R(u_h^{n,0})\|_\infty, \quad \textit{(relative)}$$

If this is criterion is fulfilled, set $u_h^n := u_h^{n,j+1}$*. Else, we increment* $k \to k+1$ *and goto Step 1.*

Remark 11.16 (On using quasi-Newton steps)**.** *Usually, when the Newton reduction rate*

$$\theta_k = \frac{\|R(u_h^{n,j+1})\|}{\|R(u_h^{n,j})\|},$$

was sufficiently good, e.g., $\theta_k \leq \theta_{max} < 1$ *(where e.g.* $\theta_{max} \approx 0.1$*), a common strategy is to work with the 'old' Jacobian matrix, but with a new right hand side.*

11.3.1 Pseudo C++ code of a Newton implementation with line search

An example of a pseudo C++ code demonstrating this algorithm is [72]:

```
newton_iteration ()
{
  const double lower_bound_newton_residual = 1.0e-10;
  const unsigned int max_no_newton_steps  = 20;

  // Decision whether the system matrix should be build at each Newton step
  const double nonlinear_theta = 0.1;

  // Line search parameters
  unsigned int line_search_step;
  const unsigned int  max_no_line_search_steps = 10;
  const double line_search_damping = 0.6;
  double new_newton_residual;

  // Application of nonhomogeneous Dirichlet boundary conditions to the variational equations:
  set_nonhomo_Dirichlet_bc ();

  // Evaluate the right hand side residual
  assemble_system_rhs();

  double newton_residual = system_rhs.linfty_norm();
  double old_newton_residual = newton_residual;
  unsigned int newton_step = 1;

  if (newton_residual < lower_bound_newton_residual)
      std::cout << '\t' << std::scientific << newton_residuum << std::endl;

  while (newton_residual > lower_bound_newton_residual &&
         newton_step < max_no_newton_steps)
    {
      old_newton_residual = newton_residual;

      assemble_system_rhs();
      newton_residual = system_rhs.linfty_norm();

      if (newton_residuum < lower_bound_newton_residuum)
        {
          std::cout << '\t'
                    << std::scientific
                    << newton_residual << std::endl;
```

```
          break;
        }

      // Simplified Newton steps
      if (newton_residual/old_newton_residual > nonlinear_theta)
        assemble_system_matrix ();

      // Solve linear equation system Ax = b
      solve ();

      line_search_step = 0;
      for ( ; line_search_step < max_no_line_search_steps; ++line_search_step)
         {
          solution += newton_update;

          assemble_system_rhs ();
          new_newton_residual = system_rhs.linfty_norm();

          if (new_newton_residual < newton_residual)
            break;
          else
            solution -= newton_update;

          newton_update *= line_search_damping;
         }

      // Output to the terminal for the user
      std::cout << std::setprecision(5) <<newton_step << '\t' << std::scientific << newton_residual << '\t'
                << std::scientific << newton_residual/old_newton_residual  <<'\t' ;
      if (newton_residual/old_newton_residual > nonlinear_theta)
        std::cout << "r" << '\t' ;
      else
        std::cout << " " << '\t' ;
      std::cout << line_search_step  << '\t' << std::endl;

      // Goto next newton iteration, increment j->j+1
      newton_step++;
    }

}
```

11.4 An academic example of finite-difference-in-time, Galerkin-FEM-in-space-discretization and linearization in a Newton setting

We collect ingredients from most other sections to discuss a final PDE, which is time-dependent and nonlinear. This specific PDE itself has no use neither in theory nor in practice and is purely academic, but is inspired by the conservation of momentum of a nonstationary, nonlinear, fluid-structure interaction problem [71].

We are given the following example:

Problem 11.17. *Let $\Omega = (0,1)$ and $I = (0,T)$ with the end time value $T > 0$. Let the following PDE be given: Find $v : R \times I \to \mathbb{R}$ and $u : R \times I \to R$, for allmost all times, such that*

$$\partial_t^2 u - J\nabla \cdot \sigma(v)F^{-T} = f, \quad \text{in } \Omega, \quad \text{plus bc. and initial cond.},$$

and where $J := J(u), F := F(u)$ and $\sigma(v) = (\nabla v + \nabla v^T)$.

11.4.0.1 Time discretization We aim to apply a One-Step-θ scheme applied to the mixed problem:

$$\begin{aligned} \partial_t v - J\nabla \cdot \sigma(v)F^{-T} &= f, \\ \partial_t u - v &= 0. \end{aligned}$$

One-Step-θ discretization with time step size k, and $\theta \in [0,1]$, leads to

$$\frac{v - v^{n-1}}{k} - \theta J \nabla \cdot \sigma(v) F^{-T} - (1-\theta) J^{n-1} \nabla \cdot \sigma(v^{n-1})(F^{-T})^{n-1} = \theta f + (1-\theta) f^{n-1},$$
$$\frac{u - u^{n-1}}{k} - \theta v - (1-\theta) v^{n-1} = 0.$$

11.4.0.2 Spatial pre-discretization: weak form on the continuous level We multiply by the time step k, apply with test functions from suitable spaces V and W and obtain the weak formulations

$$\begin{aligned}
&(v - v^{n-1}, \varphi) + k\theta (J\sigma(v)F^{-T}, \nabla\varphi) + k(1-\theta)(J^{n-1}\sigma(v^{n-1})(F^{-T})^{n-1}, \varphi) \\
=&k\theta(f,\varphi) + k(1-\theta)(f^{n-1},\varphi) \quad \forall \varphi \in V, \\
&(u - u^{n-1}, \psi) + k\theta(v,\psi) + k(1-\theta)(v^{n-1},\psi) = 0 \quad \forall \psi \in W.
\end{aligned}$$

Sorting terms on left and right hand sides:

$$\begin{aligned}
&(v,\varphi) + k\theta(J\sigma(v)F^{-T}, \nabla\varphi) \\
=&(v^{n-1},\varphi) - k(1-\theta)(J^{n-1}\sigma(v^{n-1})(F^{-T})^{n-1},\varphi) \\
&+ k\theta(f,\varphi) + k(1-\theta)(f^{n-1},\varphi) \quad \forall \varphi \in V, \\
&(u,\psi) + k\theta(v,\psi) = (u^{n-1},\psi) - k(1-\theta)(v^{n-1},\psi) \quad \forall \psi \in W.
\end{aligned}$$

11.4.0.3 A single semi-linear form - first step towards Newton solver Now, we build a single semi-linear form $A(\cdot)(\cdot)$ and right hand side $F(\cdot)$. Let[10] $U := \{v, u\} \in V \times W$ and $\Psi := \{\varphi, \psi\} \in V \times W$: Find $U \in V \times W$ such that:

$$\begin{aligned}
A(U)(\Psi) &= (v,\varphi) + k\theta(J\sigma(v)F^{-T}, \nabla\varphi) + (u,\psi) + k\theta(v,\psi) \\
F(\Psi) &= (v^{n-1},\varphi) - k(1-\theta)(J^{n-1}\sigma(v^{n-1})(F^{-T})^{n-1},\varphi) + k\theta(f,\varphi) \\
&\quad + k(1-\theta)(f^{n-1},\varphi)(u^{n-1},\psi) - k(1-\theta)(v^{n-1},\psi)
\end{aligned}$$

for all $\Psi \in V \times W$.

11.4.0.4 Evaluation of directional derivatives - second step for Newton solver Let $\delta U := \{\delta v, \delta u\} \in V \times W$. Then the directional derivative of $A(U)(\Psi)$ is given by:

$$\begin{aligned}
A'(U)(\delta U, \Psi) = (\delta v,\varphi) &+ k\theta\Big(J'(\delta u)\sigma(v)F^{-T} + J\sigma'(\delta v)F^{-T} + J\sigma(v)(F^{-T})'(\delta u), \nabla\varphi\Big) \\
&+ (\delta u,\psi) + k\theta(\delta v,\psi),
\end{aligned}$$

where we applied the chain rule for the term $J\sigma(v)F^{-T}$. Here, in the 'non-prime' terms in the nonlinear part; namely J, F^{-T} and $\sigma(v)$, the previous Newton solution is inserted. We have now all ingredients to perform the Newton step (219).

11.4.0.5 Spatial discretization in finite-dimensional spaces In the final step, we assume conforming finite-dimensional subspaces $V_h \subset V$ and $W_h \subset W$ with $V_h := \{\varphi_1, \ldots, \varphi_N\}$ and $W_h := \{\psi_1, \ldots, \psi_M\}$. Then, the update solution variables in each Newton step are given by:

$$\delta v_h := \sum_{j=1}^{N} v_j \varphi_j, \quad \text{and} \quad \delta u_h := \sum_{j=1}^{M} u_j \psi_j.$$

For the linearized semi-linear form $A'(U)(\delta U, \Psi)$, we have:

$$A'(U_h)(\delta U_h, \Psi_h)$$

[10] Of course, the solution spaces for ansatz- and test functions might differ.

and with this (M_{ij} representing the entries of the Jacobian):

$$M = (M)_{ij} := A'(U_h)(\Psi_j, \Psi_i) = \sum_{j=1}^{N} v_j(\varphi_j, \varphi_i) + k\theta\Big(\sum_{j=1}^{M} u_j J(\psi_j)\sigma(v)F^{-T} + J(\sum_{j=1}^{N} v_j\sigma'(\varphi_j))F^{-T} + J\sigma(v)(\sum_{j=1}^{M} u_j(F^{-T})'(\psi_j)), \nabla\varphi_i\Big) + \sum_{j=1}^{M} u_j(\psi_j, \psi_i) + k\theta\sum_{j=1}^{N} v_j(\varphi_j, \psi_i)$$

for all test functions running through $i = 1, \ldots, N, N+1, \ldots, M$.

11.4.0.6 Newton's method and the resulting linear system We recall:

$$A'(U_h^{n,j})(\delta U_h^n, \phi) = -A(U_h^{n,j})(\phi) + F(\phi),$$
$$U_h^{n,j+1} = U_h^{n,j} + \lambda\delta U_h^n.$$

The linear equation system reads in matrix form:

$$M\delta U = B \tag{226}$$

where M has been defined before, B is the discretization of the residual:

$$B \sim A(u_h^{n,j})(\phi) + F(\phi)$$

and the solution vector δU is given by

$$\delta U = (\delta v_1, \ldots, \delta v_N, \delta u_1, \ldots, \delta u_M)^T.$$

Since we dealt originally with two PDEs (now somewhat hidden in the semilinear form), it is often desirable to write (226) in block form:

$$\begin{pmatrix} M_{vv} & M_{vu} \\ M_{uv} & M_{uu} \end{pmatrix}\begin{pmatrix} \delta v \\ \delta u \end{pmatrix} = \begin{pmatrix} B_v \\ B_u \end{pmatrix}$$

where B_v and B_u are the residual parts corresponding to the first and second PDEs, respectively. Since in general such matrix systems are solved with iterative solvers, the block form allows a better view on the structure and construction of preconditioners. For instance, starting again from (226), a preconditioner is a matrix P^{-1} such that

$$P^{-1}M\delta U = P^{-1}B$$

so that the condition number of $P^{-1}M$ is moderate. Obviously, the ideal preconditioner would be the inverse of A: $P^{-1} = A^{-1}$. In practice one tries to build P^{-1} in such a way that

$$P^{-1}M = \begin{pmatrix} I & * \\ 0 & I \end{pmatrix}$$

and where P^{-1} is a lower triangular block matrix:

$$P^{-1} = \begin{pmatrix} P_1^{-1} & 0 \\ P_3^{-1} & P_4^{-1} \end{pmatrix}$$

The procedure is as follows (see lectures for linear algebra in which the inverse is explicitely constructed):

$$\begin{aligned}
M &= \begin{pmatrix} M_{vv} & M_{vu} \\ M_{uv} & M_{uu} \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \\
&= \begin{pmatrix} I & M_{vv}^{-1} M_{vu} \\ M_{uv} & M_{uu} \end{pmatrix} \begin{pmatrix} M_{vv}^{-1} & 0 \\ 0 & I \end{pmatrix} \\
&= \begin{pmatrix} I & M_{vv}^{-1} M_{vu} \\ 0 & \underbrace{M_{uu} - M_{uv} M_{vv}^{-1} M_{vu}}_{=S} \end{pmatrix} \begin{pmatrix} M_{vv}^{-1} & 0 \\ -M_{uv} M_{vv}^{-1} & I \end{pmatrix} \\
&= \begin{pmatrix} I & M_{vv}^{-1} M_{vu} \\ 0 & I \end{pmatrix} \underbrace{\begin{pmatrix} M_{vv}^{-1} & 0 \\ -S^{-1} M_{uv} M_{vv}^{-1} & S^{-1} \end{pmatrix}}_{=P^{-1}}
\end{aligned}$$

where $S = M_{uu} - M_{uv} M_{vv}^{-1} M_{vu}$ is the so-called **Schur complement**. The matrix P^{-1} is used as (exact) preconditioner for M. Indeed we double-check:

$$\begin{pmatrix} M_{vv}^{-1} & 0 \\ -S^{-1} M_{uv} M_{vv}^{-1} & S^{-1} \end{pmatrix} \begin{pmatrix} M_{vv} & M_{vu} \\ M_{uv} & M_{uu} \end{pmatrix} = \begin{pmatrix} I & M_{vv}^{-1} M_{vu} \\ 0 & I \end{pmatrix}$$

Tacitely we assumed in the entire procedure that S and M_{vv} are invertible.

Using P^{-1} in a Krylov method, we only have to perform matrix-vector multiplications such as

$$\begin{pmatrix} X_{new} \\ Y_{new} \end{pmatrix} = \begin{pmatrix} P_1^{-1} & 0 \\ P_3^{-1} & P_4^{-1} \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix}$$

Now we obtain:

$$X_{new} = P_1^{-1} X \tag{227}$$

$$Y_{new} = P_3^{-1} X + P_4^{-1} Y \tag{228}$$

Remark 11.18. *Be careful, we deal with two iterative procedures in this example: Newton's method to compute iteratively the nonlinear solution. Inside Newton's method, we solve the linear equations systems with a Krylov space method, which is also an iterative method.*

11.5 Navier-Stokes - FEM discretization

In a similar way to Section 9.5, we briefly present the NSE FEM discretization. As we previously discusssed, rather than solving directly for v_h and p_h we solve now for the (Newton) updates δv_h and δp_h - later more.

The problem reads:

$$\text{Find } U_h \in X_h \text{ such that: } A(U_h)(\Psi_h) = F(\Psi_h) \quad \forall \Phi \in X_h,$$

where

$$\begin{aligned} A(U_h)(\Psi_h) &= (v_h \cdot \nabla v_h, \psi_h^v) + \frac{1}{Re}(\nabla v_h, \nabla \psi_h^v) - (p_h, \nabla \cdot \psi_h^v) + (\nabla \cdot v_h, \psi_h^p), \\ F(\Psi_h) &= (f_f, \psi_h^v). \end{aligned}$$

Here, we added the dimensionless Reynolds number Re in order to be aware how the equations change their type if Re is either small or large.

Remark 11.19 (Notations for bilinear and semilinear forms)**.** *If the PDE is linear, we use the notation for a bilinear form: $A(U_h, \Psi_h)$. If the PDE is nonlinear, this is indicated in the notation by using $A(U_h)(\Psi_h)$.*

In the following, we apply Newton's method. Given an initial guess $U_h^0 := \{v_h^0, p_h^0\}$, we must solve the problem:

$$\text{Find } \delta U_h := \{\delta v_h, \delta p_h\} \in X_h \text{ such that: } A'(U_h^l)(\delta U_h, \Psi_h) = -A(U_h^l)(\Psi_h) + F(\Psi_h), \quad U_h^{l+1} = U_h^l + \delta U_h.$$

Here,

$$\begin{aligned} A'(U_h^l)(\delta U_h, \Psi_h) &= (\delta v_h \cdot \nabla v_h^l + v_h^l \cdot \nabla \delta v_h, \psi_h^v) + \frac{1}{Re}(\nabla \delta v_h, \nabla \psi_h^v) - (\delta p_h, \nabla \cdot \psi_h^v) + (\nabla \cdot \delta v_h, \psi_h^p), \\ F(\Psi_h) &= (f_f, \psi_h^v), \\ A(U_h^l)(\Psi_h) &= (v_h^l \cdot \nabla v_h^l, \psi_h^v) + \frac{1}{Re}(\nabla v_h^l, \nabla \psi_h^v) - (p_h^l, \nabla \cdot \psi_h^v) + (\nabla \cdot v_h^l, \psi_h^p). \end{aligned}$$

As explained, we solve now for the updates and their respresentation with the help of the shape functions:

$$\delta v_h = \sum_{j=1}^{N_V} \delta v_j \psi_h^{v,j}, \quad \delta p_h = \sum_{j=1}^{N_P} \delta p_j \psi_h^{p,j}$$

With that the discrete, linearized convection operator reads:

$$L := (\psi_h^{v,j} \cdot \nabla v_h^l + v_h^l \cdot \nabla \psi_h^{v,j}, \psi_h^{v,i})_{ij=1}^{N_V, N_V}.$$

The block structure reads:

$$\begin{pmatrix} L + \frac{1}{Re}A & B \\ -B^T & 0 \end{pmatrix} \begin{pmatrix} \delta v \\ \delta p \end{pmatrix} = \begin{pmatrix} f - [L^l + \frac{1}{Re}A^l + B] \\ 0 - [-B^T]^l \end{pmatrix},$$

where we have added on the right hand side the vectors of Newton's residual.

Remark 11.20. *A numerical test is presented in Section 7.14.*

11.6 Chapter summary and outlook

In this section we provided a very short introduction to nonlinear problems. Most importantly we introduced numerical solver techniques such as fixed-point schemes and Newton methods.

12 Computational convergence analysis

We provide some tools to perform a computational convergence analysis. In these notes we faced two situations of 'convergence':

- **Discretization error:** Convergence of the discrete solution u_h towards the (unknown) exact solution u;
- **Iteration error:** Convergence of an iterative scheme to approximate the discrete solution u_h through a sequence of approximate solutions $u_h^{(k)}, k = 1, 2, \ldots$.

In the following we further illustrate the terminologies 'first order convergence', 'convergene of order two', 'quadratic convergence', 'linear convergence', etc.

12.1 Discretization error

Before we go into detail, we discuss the relationship between the degrees of freedom (DoFs) N and the mesh size parameter h. In most cases the discretization error is measured in terms of h and all a priori and a posteriori error estimates are stated in a form

$$\|u - u_h\| = O(h^\alpha), \quad \alpha > 0.$$

In some situations it is however better to create convergence plots in terms of DoFs vs. the error. One example is when adaptive schemes are employed with different h. Then it would be not clear to which h the convergence plot should be drawn. But simply counting the total numbers of DoFs is not a problem though.

12.1.1 Relationship between h and N (DoFs)

The relationship of h and N depends on the basis functions (linear, quadratic), whether a Lagrange method (only nodal points) or Hermite-type method (with derivative information) is employed. Moreover, the dimension of the problem plays a role.

We illustrate the relationship for a Lagrange method with linear basis functions in 1D,2D,3D:

Proposition 12.1. *Let d be the dimension of the problem: $d = 1, 2, 3$. It holds*

$$N = \left(\frac{1}{h} + 1\right)^d$$

where h is the mesh size parameter (lengh of an element or diameter in higher dimensions for instance), and N the number of DoFs.

Proof. Sketch. No strict mathematical proof. We initialize as follows:

- 1D: 2 values per line;
- 2D: 4 values per quadrilaterals;
- 3D: 8 values per hexahedra.

Of course, for triangles or primsms, we have different values in 2D and 3D. We work on the unit cell with $h = 1$. All other h can be realized by just normalizing h. By simple counting the nodal values, we have in 1D

```
h    N
=======
1    2
1/2  3
1/4  5
1/8  9
1/16 17
1/32 33
...
=======
```

We have in 2D

```
h    N
=======
1    4
1/2  9
1/4  25
1/8  36
1/16 49
1/32 64
...
=======
```

We have in 3D

```
h    N
=======
1    8
1/2  27
1/4  64
...
=======
```

□

12.1.2 Discretization error

With the previous considerations, we have now a relationship between h and N that we can use to display the discretization error.

Proposition 12.2. *In the approximate limit it holds:*

$$N \sim \Big(\frac{1}{h}\Big)^d$$

yielding

$$h \sim \frac{1}{\sqrt[d]{N}}$$

These relationships allow us to replace h in error estimates by N.

Proposition 12.3 (Linear and quadratic convergence in 1D)**.** *When we say a scheme has a linear or quadratic convergence in 1D, (i.e., $d = 1$) respectively, we mean:*

$$O(h) = O\Big(\frac{1}{N}\Big)$$

or

$$O(h^2) = O\Big(\frac{1}{N^2}\Big)$$

In a linear scheme, the error will be divided by a factor of 2 when the mesh size h is divided by 2 and having quadratic convergence the error will decrease by a factor of 4.

Proposition 12.4 (Linear and quadratic convergence in 2D)**.** *When we say a scheme has a linear or quadratic convergence in 2D, (i.e., $d = 2$) respectively, we mean:*

$$O(h) = O\Big(\frac{1}{\sqrt{N}}\Big)$$

or

$$O(h^2) = O\Big(\frac{1}{N}\Big)$$

12.1.3 Computationally-obtained convergence order

In order to calculate the convergence order α from numerical results, we make the following derivation. Let $P(k) \to P$ for $k \to 0$ be a converging process and assume that

$$P(k) - \tilde{P} = O(k^\alpha).$$

Here $\tilde{P}$ is either the exact limit P (in case it is known) or some 'good' approximation to it. Let us assume that three numerical solutions are known (this is the minimum number if the limit P is not known). That is

$$P(k), \quad P(k/2), \quad P(k/4).$$

Then, the convergence order can be calculated via the formal approach $P(k) - \tilde{P} = ck^\alpha$ with the following formula:

Proposition 12.5 (Computationally-obtained convergence order)**.** *Given three numerically-obtained values $P(k), P(k/2)$ and $P(k/4)$, the convergence order can be estimated as:*

$$\alpha = \frac{1}{log(2)} log\Big(\Big|\frac{P(k) - P(k/2)}{P(k/2) - P(k/4)}\Big|\Big). \tag{229}$$

The order α is an estimate and heuristic because we assumed a priori a given order, which strictly speaking we have to proof first.

Proof. We assume:

$$\begin{aligned} P(k) - P(k/2) &= O(k^\alpha), \\ P(k/2) - P(k/4) &= O((k/2)^\alpha). \end{aligned}$$

First, we have

$$P(k/2) - P(k/4) = O((k/2)^\alpha) = \frac{1}{2^\alpha} O(k^\alpha)$$

We simply re-arrange:

$$\begin{aligned} P(k/2) - P(k/4) &= \frac{1}{2^\alpha}\Big(P(k) - P(k/2)\Big) \\ \Rightarrow \quad 2^\alpha &= \frac{P(k) - P(k/2)}{P(k/2) - P(k/4)} \\ \Rightarrow \quad \alpha &= \frac{1}{log(\alpha)} \frac{P(k) - P(k/2)}{P(k/2) - P(k/4)} \end{aligned}$$

□

In the following we present results (for all details, we refer the reader to [74]) for the (absolute) end time error of an ODE problem (but it could be any other PDE problem as well) on three mesh levels (different time step sizes k) with three schemes (FE - forward Euler, BE - backward Euler, CN - Crank-Nicolson):

```
Scheme         #steps  k     Absolute error
=========================================
FE err.:       8       0.36  +0.13786
BE err.:       8       0.36  -0.16188
CN err.:       8       0.36  -0.0023295
FE err.:       16      0.18  +0.071567
BE err.:       16      0.18  -0.077538
CN err.:       16      0.18  -0.00058168
FE err.:       32      0.09  +0.036483
BE err.:       32      0.09  -0.037974
CN err.:       32      0.09  -0.00014538
=========================================
```

We monitor that doubling the number of intervals (i.e., halving the step size k) reduces the error in the forward and backward Euler scheme by a factor of 2. This is (almost) linear convergence, which is confirmed by using Formula (229) yielding $\alpha = 0.91804$. The CN scheme is much more accurate (for instance using $n = 8$ the error is 0.2% rather than $13 - 16\%$) and we observe that the error is reduced by a factor of 4. Thus quadratic convergence is detected. Here the 'exact' order on these three mesh levels is $\alpha = 1.9967$.

Remark 12.6. *Another example where the previous formula is used can be found in Section 6.13.4.*

12.2 Iteration error

Iterative schemes are used to approximate the discrete solution u_h. This has a priori nothing to do with the discretization error. The main interest is how fast can we get a good approximation of the discrete solution u_h. One example can be found for solving implicit methods for ODEs in which Newton's method is used to compute the discrete solutions of the backward Euler scheme.

To speak about convergence, we compare two subsequent iterations:

Proposition 12.7. *Let us assume that we have an iterative scheme to compute a root z. The iteration converges with order p when*

$$\|x_k - z\| \leq c \, \|x_{k-1} - z\|^p, \quad k = 1, 2, 3, \ldots$$

with $p \geq 1$ and $c = const$. In more detail:

- *Linear convergence: $c \in (0, 1)$ and $p = 1$;*
- *Superlinear convergence: $c := c_k \to 0$, $(k \to \infty)$ and $p = 1$;*
- *Quadratic convergence $c \in \mathbb{R}$ and $p = 2$.*

Cupic and higher convergence are defined as quadratic convergence with the respectice p.

Remark 12.8 (Other characterizations of superlinear and quadratic convergence)**.** *Other (but equivalent) formulations for superlinear and quadratic convergence, respectively, in the case $z \neq x_k$ for all k, are:*

$$\lim_{k \to \infty} \frac{\|x_k - z\|}{\|x_{k-1} - z\|} = 0,$$
$$\limsup_{k \to \infty} \frac{\|x_k - z\|}{\|x_{k-1} - z\|^2} < \infty.$$

Corollary 12.9 (Rule of thumb)**.** *A rule of thumb for quadratic convergence is: the number of correct digits doubles at each step. For instance, a Newton scheme to compute $f(x) = \sqrt{x} - 2 = 0$ yields the following results:*

```
Iter x            f(x)
==============================
0    3.000000e+00 7.000000e+00
1    1.833333e+00 1.361111e+00
2    1.462121e+00 1.377984e-01
3    1.414998e+00 2.220557e-03
4    1.414214e+00 6.156754e-07
5    1.414214e+00 4.751755e-14
==============================
```

12.3 Chapter summary and outlook

We finish this booklet in the next chapter by posing some characteristic questions that help to recapitulate the important topics of this lecture.

13 Wrap-up

13.1 Quiz

In this final section, we consolidate what we have learned through some questions:

1. State the Poisson problem with homogeneous Dirichlet conditions. Write this problem in a variational form using the correct function spaces.
2. Coming from the variational form, give the discrete form and construct a finite element scheme of the Poisson problem until the final final linear equation system.
3. What is the maximum principle?
4. What is an M matrix?
5. How can we characterize elliptic, parabolic, and hyperbolic problems?
6. What is the difference between a priori and a posteriori error estimates?
7. What is useful when employing goal-oriented error estimation? What is a goal functional?
8. How is well-posedness of problems defined?
9. What is the Poisson problem? How do we obtain existence, uniqueness, and continuous dependence on the right hand side?
10. Give a physical interpretation of the Poisson problem.
11. Define the Laplace operator.
12. Why are all integrals evaluated on a master element in FEM?
13. Write down the loops to assemble a finite element system matrix.
14. Define the H^1 space.
15. Why is it sufficient to use the Riesz lemma to prove existence of the Poisson problem rather than using the Lax-Milgram lemma?
16. What is a best approximation?
17. What is Galerkin orthogonality? And what is its geometrical interpretation?
18. How do we get an improved estimate for the L^2 norm for the Poisson problem using finite elements?
19. Given a differential equation from physics: what is the procedure to obtain a variational formulation and to proof existence for both problems?
20. How do we discretize a given variational problem?
21. Formulate the heat / wave equation and discretize in space and time.
22. How can we solve the linear equation systems $Au = b$?
23. What are the differences between FD and FEM?
24. Give examples of different time-stepping schemes. What schemes exist (three classes!) and what are their principles differences?
25. Formulate Newton's method in $\mathbb{R}$.
26. What is the Lax-Milgram lemma? What are the assumptions?

27. Give a stopping criterion for an iterative scheme!
28. Why is the condition number important?
29. What is the condition number of the system matrix of the Poisson problem?
30. Why do we need a posteriori error estimation?
31. Why do we need mesh adaptivity?
32. Define the directional derivative!
33. What is the Aubin-Nitsche trick?
34. What does the Céa lemma say?
35. What are the three characterizing properties of an finite element?
36. Write down the formula of partial integration in higher dimensions!
37. What are possible boundary conditions for second-order boundary value problems?
38. Why do we need numerical quadrature in finite element calculations?
39. Formulate Newton's method for variational problems
40. What are examples of vector-valued problems? Id est: what is a vector-valued problem?
41. How are linear and quadratic convergence characterized in practice?
42. Explain the key ideas to obtain stability estimates for parabolic problems.
43. What is the principle idea to proof energy conservation of the wave equation on the time-discrete level?
44. Give an example of a nonlinear equation.
45. What are numerical possiblities to solve nonlinear equations?
46. Explain a fixed-point iteration for solving nonlinear problems.

13.2 The end

Gottfried Wilhelm Leibniz (1646 - 1716):
“Es ist unwürdig, die Zeit von hervorragenden Leuten mit knechtischen Rechenarbeiten zu verschwenden, weil bei Einsatz einer Maschine auch der Einfältigste die Ergebnisse sicher hinschreiben kann.”
Gottfried Wilhelm Leibniz erfand das Staffelwalzenprinzip, welches eine große Errungenschaft beim Bau von mechanischen Rechenmaschinen darstellte, und gewissermaßen wegbereitend für unsere heutigen Rechnenmaschinen war: die Computer, welche einen Schwerpunkt in der modernen numerischen Mathematik und des Wissenschaftlichen Rechnens einnehmen.

References

[1] G. Allaire. *An Introduction to Mathematical Modelling and Numerical Simulation.* Oxford University Press, 2007.

[2] G. Allaire and F. Alouges. Analyse variationnelle des équations aux dérivées partielles. MAP 431: Lecture notes at Ecole Polytechnique, 2016.

[3] H. W. Alt. *Lineare Funktionalanalysis.* Springer, Berlin-Heidelberg, 5nd edition, 2006.

[4] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The `deal.II` library, version 8.5. *Journal of Numerical Mathematics*, 2017.

[5] W. Bangerth, M. Geiger, and R. Rannacher. Adaptive Galerkin finite element methods for the wave equation. *Comput. Methods Appl. Math.*, 10:3–48, 2010.

[6] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[7] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations.* Birkhäuser, Lectures in Mathematics, ETH Zürich, 2003.

[8] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[9] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica, Cambridge University Press*, pages 1–102, 2001.

[10] C. Bernardi and E. Süli. Time and space adaptivity for the second-order wave equation. *Math. Models Methods Appl. Sci.*, 15(2):199–225, 2005.

[11] M. Braack and A. Ern. A posteriori control of modeling errors and discretization errors. *Multiscale Model. Simul.*, 1(2):221–238, 2003.

[12] D. Braess. *Finite Elemente.* Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, vierte, überarbeitete und erweiterte edition, 2007.

[13] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods.* Number 15 in Texts in applied mathematics ; 15 ; Texts in applied mathematics. Springer, New York, NY, 3. ed. edition, 2008.

[14] H. Brezis. *Analyse Fonctionnelle, Theorie et Applications.* Masson Paris, 1983.

[15] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations.* Springer New York, Dordrecht, Heidelberg, London, 2011.

[16] G. F. Carey and J. T. Oden. *Finite Elements. Volume III. Compuational Aspects.* The Texas Finite Element Series, Prentice-Hall, Inc., Englewood Cliffs, 1984.

[17] C. Carstensen, M. Feischl, M. Page, and D. Praetorius. Axioms of adaptivity. *Computers and Mathematics with Applications*, 67(6):1195 – 1253, 2014.

[18] C. Carstensen and R. Verfürth. Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM J. Numer. Anal.*, 36(5):1571–1587, 1999.

[19] P. G. Ciarlet. *Mathematical Elasticity. Volume 1: Three Dimensional Elasticity.* North-Holland, 1984.

[20] P. G. Ciarlet. *The finite element method for elliptic problems.* North-Holland, Amsterdam [u.a.], 2. pr. edition, 1987.

[21] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology*, volume 5. Springer-Verlag, Berlin-Heidelberg, 2000.

[22] P. Deuflhard. *Newton Methods for Nonlinear Problems*, volume 35 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, 2011.

[23] W. Dörfler. A convergent adaptive algorithm for poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[24] L. C. Evans. *Partial differential equations*. American Mathematical Society, 2000.

[25] L. C. Evans. *Partial differential equations*. American Mathematical Society, 2010.

[26] T. Fliessbach. *Mechanik*. Spektrum Akademischer Verlag, 2007.

[27] V. Girault and P.-A. Raviart. *Finite Element method for the Navier-Stokes equations*. Number 5 in Computer Series in Computational Mathematics. Springer-Verlag, 1986.

[28] R. Glowinski and J. Periaux. Numerical methods for nonlinear problems in fluid dynamics. In *Proc. Intern. Seminar on Scientific Supercomputers*. North Holland, Feb. 2-6 1987.

[29] H. Goldstein, C. P. P. Jr., and J. L. S. Sr. *Klassische Mechanik*. Wiley-VCH, 3. auflage edition, 2006.

[30] C. Goll, T. Wick, and W. Wollner. DOpElib: Differential equations and optimization environment; A goal oriented software library for solving pdes and optimization problems with pdes. *Archive of Numerical Software*, 5(2):1–14, 2017.

[31] P. Grisvard. *Elliptic Problems in Nonsmooth Domains*, volume 24. Pitman Advanced Publishing Program, Boston, 1985.

[32] C. Groß mann and H.-G. Roos. *Numerische Behandlung partieller Differentialgleichungen*. Teubner-Studienbücher Mathematik ; Lehrbuch Mathematik. Teubner, Wiesbaden, 3., völlig überarb. und erw. aufl. edition, 2005.

[33] C. Großmann, H.-G. Roos, and M. Stynes. *Numerical Treatment of Partial Differential Equations*. Springer, 2007.

[34] W. Hackbusch. *Theorie und Numerik elliptischer Differentialgleichungen, edition=, address=, publisher=Vieweg+Teubner Verlag, year=1986, series=, number=*.

[35] G. H"ammerlin and K.-H. Hoffmann. *Numerische Mathematik*. Springer Verlag, 1992.

[36] M. Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des Wissenschaftlichen Rechnens*. Vieweg-Teubner Verlag, 2009.

[37] J. G. Heywood and R. Rannacher. Finite-element approximation of the nonstationary Navier-Stokes problem part iv: Error analysis for second-order time discretization. *SIAM Journal on Numerical Analysis*, 27(2):353–384, 1990.

[38] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *International Journal of Numerical Methods in Fluids*, 22:325–352, 1996.

[39] G. Holzapfel. *Nonlinear Solid Mechanics: A continuum approach for engineering*. John Wiley and Sons, LTD, 2000.

[40] T. Hughes. *The finite element method*. Dover Publications, 2000.

[41] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135 – 4195, 2005.

[42] S. H. John W. Eaton, David Bateman and R. Wehbring. *GNU Octave version 3.8.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2014. ISBN 1441413006.

[43] C. Johnson. *Numerical solution of partial differential equations by the finite element method.* Cambridge University Press, Campridge, 1987.

[44] K. Koenigsberger. *Analysis 1.* Springer Lehrbuch. Springer, Berlin – Heidelberg – New York, 6. auflage edition, 2004.

[45] K. Koenigsberger. *Analysis 2.* Springer Lehrbuch. Springer, Berlin – Heidelberg – New York, 5. auflage edition, 2004.

[46] E. Kreyszig. *Introductory functional analysis with applications.* Wiley, 1989.

[47] O. Ladyzhenskaya. *The boundary value problems of mathematical physics.* Springer, New York, 1985.

[48] J. Nocedal and S. J. Wright. *Numerical optimization.* Springer Ser. Oper. Res. Financial Engrg., 2006.

[49] R. Rannacher. Finite element solution of diffusion problems with irregular data. *Numer. Math.*, 43:309–327, 1984.

[50] R. Rannacher. On the stabilization of the Crank-Nicolson scheme for long time calculations. Preprint, August 1986.

[51] R. Rannacher. Finite element methods for the incompressible Navier-Stokes equations. Lecture notes, August 1999.

[52] R. Rannacher. Numerische methoden der Kontinuumsmechanik (Numerische Mathematik 3). Vorlesungsskriptum, 2001.

[53] R. Rannacher. Numerische methoden für gewöhnliche Differentialgleichungen (Numerische Mathematik 1). Vorlesungsskriptum, 2002.

[54] R. Rannacher. Numerische methoden für partielle Differentialgleichungen (Numerische Mathematik 2). Vorlesungsskriptum, 2006.

[55] R. Rannacher. Analysis 2. Vorlesungsskriptum, 2010.

[56] R. Rannacher. *Numerik partieller Differentialgleichungen.* Heidelberg University Publishing, 2017.

[57] H.-J. Reinhardt. Die methode der finiten elemente. Vorlesungsskriptum, Universität Siegen, 1996.

[58] T. Richter. Numerische methoden für gewöhnliche und partielle differentialgleichungen. Lecture notes, Heidelberg University, 2011.

[59] T. Richter. *Fluid-structure interactions: models, analysis, and finite elements.* Springer, 2017.

[60] T. Richter and T. Wick. Variational localizations of the dual weighted residual estimator. *Journal of Computational and Applied Mathematics*, 279(0):192 – 208, 2015.

[61] T. Richter and T. Wick. *Einfuehrung in die numerische Mathematik - Begriffe, Konzepte und zahlreiche Anwendungsbeispiele.* Springer, 2017.

[62] M. Schäfer and S. Turek. *Flow Simulation with High-Performance Computer II*, volume 52 of *Notes on Numerical Fluid Mechanics*, chapter Benchmark Computations of laminar flow around a cylinder. Vieweg, Braunschweig Wiesbaden, 1996.

[63] H. R. Schwarz. *Methode der finiten Elemente.* Teubner Studienbuecher Mathematik, 1989.

[64] F. T. Suttmeier and T. Wick. Numerische Methoden für partielle Differentialgleichungen (in german). Notes (Vorlesungsmitschrift) at University of Siegen, 2006.

[65] R. Temam. *Navier-Stokes Equations: Theory and Numerical Analysis.* AMS Chelsea Publishing, Providence, Rhode Island, 2001.

[66] T. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8(2):83–130, 2001.

[67] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen - Theorie, Verfahren und Anwendungen*. Vieweg und Teubner, Wiesbaden, 2nd edition, 2009.

[68] S. Turek, L. Rivkind, J. Hron, and R. Glowinski. Numerical analysis of a new time-stepping θ-scheme for incompressible flow simulations. Technical report, TU Dortmund and University of Houston, 2005. Dedicated to David Gottlieb on the occasion of his 60th anniversary.

[69] D. Werner. *Funktionalanalysis*. Springer, 2004.

[70] T. Wick. *Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve Dynamics*. PhD thesis, University of Heidelberg, 2011.

[71] T. Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89(13-14):1456–1467, 2011.

[72] T. Wick. Solving monolithic fluid-structure interaction problems in arbitrary Lagrangian Eulerian coordinates with the deal.II library. *Archive of Numerical Software*, 1:1–19, 2013.

[73] T. Wick. Modeling, discretization, optimization, and simulation of fluid-structure interaction. Lecture notes at Heidelberg University, TU Munich, and JKU Linz available on https://www-m17.ma.tum.de/Lehrstuhl/LehreSoSe15NMFSIEn, 2015.

[74] T. Wick. Introduction to numerical modeling. MAP 502: Lecture notes at Ecole Polytechnique, 2017.

[75] J. Wloka. *Partielle Differentialgleichungen*. B. G. Teubner Verlag, Stuttgart, 1982.

[76] J. Wloka. *Partial differential equations*. Cambridge University Press, 1987.

[77] E. Zeidler. *Nonlinear functional analysis and its applications I-IV*. 1985.

[78] E. Zeidler. *Nonlinear functional analysis - nonlinear monotone operators*. Springer-Verlag, 1990.

Index

H^1, 95, 98
H^1 norm, 139
H^1_0, 95, 99
L^2, 96
L^2 norm, 139
2D-1 benchmark, 164

A norm, 48
A posteriori error estimation, 141
A priori error estimation, 141
A-stability, 202
a.e., 96
Adaptive FEM, 142
Adaptive finite elements, 142
Adjoint methods, 146
Adjoint problem, 149
Adjoint variable, 143, 146
AFEM, 142
    Basic algorithm, 142
Almost everywhere, 96
almost everywhere, 96
Ansatz function, 58
Approximation theory, 76
Assembling in FEM, 67
Assembling integrals, 67
Aubin-Nitsche trick, 119

Backtracking line search, 228
Backward Euler scheme, 221
Banach space, 93
Benchmark, 164
Big O, 19
Bilinear form, 58
    Norm, 58
    Scalar product, 58
Bochner integral, 193
Boundary conditions, 32
Boundary value problem, 23
Bounded operator, 19
Brachistochrone, 144
BVP, 23

Céa lemma, 114
    First version, 84
Cauchy sequence, 78, 93
Cauchy's inequality, 100
Cauchy-Schwarz inequality, 100
Change of variables, 20, 21
Compact support, 56
Compatiblity condition, 88
Complete space, 93
Completeness, 93
Computational convergence, 139, 235
Computational convergence analysis, 135, 139, 235
Computing error norms, 139
Condition number, 52
Conditional stability, 207
Conforming FEM, 111
Consistency, 46, 200, 203
Continuous mapping, 19
Convergence, 203
    Definition, 93
Convergence order, 139
    Computationally obtained, 237
Coupled problems, 88
Crank-Nicolson scheme, 219
Cross product, 17
Current configuration, 183

Damping parameter, 174
Data errors, 12
Defect, 225
Deformation gradient, 21
Deformed configuration, 183
Degrees of freedom, 72
Dense subspace, 97
Density, 97
descent method, 174
Differential equation
    Classification, 33
    General definition, 32
    Linear vs. nonlinear, 33
Differential problem, 54
Differentiation in Banach spaces, 147
Differentiation under the integral sign, 25
Diffusion, 23
Dirac goal functional, 162
Dirac rhs, 162
Dirichlet boundary conditions
    in Newton's method, 228
Dirichlet condition, 32
Discretization
    Finite differences, 42
Discretization error, 12, 236
Discretization parameter, 11
Divergence, 16
Divergence theorem, 21
DoFs, 72
Dual-weighted residual method, 142
Duality arguments, 142
DWR, 142

Efficient error estimator, 142
Eigenvalues, 17
Elastic wave equation, 219
Elasticity, 26, 183
Elastodynamics, 219
Element residual, 154
Elements, 111
Elliptic operator, 29
Elliptic PDE, 28
Energy conservation
  Wave equation, 213
Energy norm, 98
Energy space, 98
Error
  a posteriori in FEM, 141
  a priori in FEM, 114
  Data, 12
  Discretization, 12
  Goal-oriented, 141
  Heat equation, 205
  Model, 12
  ODE, 200
  Round-off, 12
  Systematic, 12
Error estimation
  Goal-oriented, 142
  Residual-based, 157
Error localization, 153
Error of consistence, 121
Errors in numerical mathematics, 12
Essential boundary conditions, 32
Euler
  Backward, 199
  Implicit, 199
Euler method, 199
Euler-Lagrange equations, 54, 144
Existence
  Finite differences, 44
  Linearized elasticity, 185
Explicit schemes, 199

Face residuals, 154
FD, 39
FEM, 53
  2D simulations, 135
  3D simulations, 136
  Computational convergence analysis, 136
  Quadratic, 74
  See also index Finite Elements, 53
Finite differences, 39
Finite element
  Definition, 64
Finite element method, 53
Finite elements, 53, 59
  Adaptivity, 142
  Computational convergence analysis, 136
  Construction of manufactured solution, 53
  Construction of right hand side, 136
  Error analysis, 114, 136
  Error analysis in simulations, 136
  Manufactured solution, 136
  See also index FEM, 53
  Conforming, 60
  Evaluation of integrals, 63
  Linear, 59
  Mesh, 59
  Weak form, 62
First Strang lemma, 121
Fixed point methods, 171
Fixed-point iteration, 223
Fluid flow equations, 27
Fluid mechanics, 164
Forward Euler method, 199
Frobenius scalar product, 58
Functional of interest, 141
Fundamental lemma of calculus of variations, 56, 97

Galerkin equations, 62
Galerkin method, 62
Galerkin orthogonality, 83
Galerkin orthogonalty, 175
Gauß-Seidel method, 173
generalized minimal residual, 181
Global error nroms, 139
GMRES, *see* generalized minimal residual
Goal functional, 141
Goal-oriented error estimation, 141
Gottfried Wilhelm Leibniz, 241
Gradient, 16
Gradient descent, 174
Green's function, 40
Green-Lagrange strain tensor, 183

Hölder's inequality, 100
Hanging nodes, 158
Hat function, 59
Heat equation, 28, 30
Higher-order FEM, 74
Hilbert space, 94, 98
Homogeneous Dirichlet condition, 28
Hyperbolic PDE, 28
Hyperbolic problems, 210

IBVP, 30, 194
Implicit schemes, 199
Inequalities
  Cauchy, 100
  Cauchy-Schwarz, 100

Hölder, 100
Minkowski, 100
Young, 100
Inner product, 94
Integration by parts, 22, 55, 91
Iteration error, 238
Iterative solvers, 171

Jacobi method, 173
Jacobian, 16
Jumping coefficients, 88

Korn's inequality
1st, 185
2nd, 185
Krylov space, 175

L-shaped domain, 162
Lagrange multiplier, 143, 145
Lagrange multipliers, 145
Lagrangian, 143
Lamé parameters, 183
Landau symbols, 19
Laplace equation, 23, 28
Lax-Milgram lemma, 100, 184
Leibniz, 241
Lemma
Lax-Milgram, 100
Line search, 228
Linear form, 57
Linear functional, 19
Linear operator, 18
Linear PDE, 33
Linearization by Newton, 224
Linearized elasticity, 183
Little o, 19
Local degrees of freedom, 112
Local truncation error, 46

M matrix, 45
Mass matrix, 197
Master element, 70, 72
Mathematical modeling, 11
Maximum norm, 18
Maximum principle
Continuous level, 42
Discrete, 44
Mesh adaptivity, 141
Mesh elements, 111
Minimization problem, 54
Minimum angle condition, 120
Minkowski's inequality, 100
Model errors, 12
Multiindex notation, 16

Nabla operator, 16
Natural boundary conditions, 32
Navier-Stokes, 164
Navier-Stokes equations, 27
Neumann condition, 32
Newton's method, 224, 227
Defect-correction, 225
overview, 226
Nitsche, 119
Nonlinear PDE, 33
Nonlinear problems, 223
Nonstationary linearized elasticity, 219
Norm, 18, 139
Normal vector, 15
Normed space, 18
Numerical analysis
Finite differences, 46
Heat equation, 205
ODE, 200
Numerical integration, 69
Numerical methods, 11
Numerical quadrature, 69, 121

octave, 49, 139
ODE, 11
One-Step-Theta scheme, 195
One-step-theta scheme, 195
Ordinary differential equation, 11
Orthogonal projection, 78
Orthogonality, 94

Parabolic PDE, 28
Parabolic problems, 193, 194
Partial differential equation, 11
Partial integration, 22, 91
PDE, 11
Elliptic, 28
Hyperbolic, 28
Parabolic, 28
Petrov-Galerkin method, 62
Physical configuration, 183
Picard iteration, 223
Poisson
1D simulations, 49, 65, 75
2D simulations, 135
3D simulations, 136
Poisson equation, 28
Poisson problem, 23, 39
Pre-Hilbert space, 76, 94
Primal variable, 143
Principal invariants of a matrix, 17
Principle of minimum potential energy, 54
Principle of virtual work, 54
Projection, 82

Quadratic FEM, 74

Questions to these lecture notes, 239
Quiz, 239

Rannacher time stepping, 214
Reference configuration, 183
Reference element, 70, 72
Regularity
    Poisson 1D, 42
Relationship mesh size and DoFs, 235
Relaxation parameter, 174
Reliable error estimator, 142
Reynolds' transport theorem, 25
Riesz representation theorem, 184
Ritz method, 62
Ritz-Galerkin method, 62
Robin condition, 32
Rotation, 17
Round-off errors, 12

Saddle point, 143
Scalar product, 94
Scalar product, 58
Schur complement, 233
Scientific computing, 11
Search direction, 174
Semi-norm, 18
Shape functions, 59
Shape functions, 62
Simplified Newton steps, 228
Singularities of H^1 functions, 99
Smooth functions, 56
Sobolev spaces, 95
Software, 11
Solid mechanics equation, 26
Stability, 47, 200, 201
stability analysis, 205
Stiff PDEs, 195
stiff problem, 205
Stiffness, 204
Stiffness matrix, 197
Stokes problem, 165
Strang
    First lemma, 121
Substitution rule, 20
Systematic errors, 12

Taylor expansion, 46, 203
Test function, 54, 58
Time stepping scheme
    Fractional-Step-Theta, 214
Time stepping schemes
    A-Stability, 213
    Crank-Nicolson, 213
    Explicit Euler, 213
    Fractional-Step-θ, 214
    Shifted Crank-Nicolson, 214
Total potential energy, 54
Trace, 16, 104
Transformation of integrals, 20
Transmission problems, 88
Trial function, 58
Truncation error, 46

Unconditionally stable, 206
Uniqueness
    Finite differences, 44

V-ellipticity, 187
Variational problem, 54
Vector space, 17
Vector-valued problems, 164
Volume ratio, 21

Wave equation, 28, 31
Weak derivative, 97
Well-posedness, 13
    Finite differences, 43
    Poisson in 1D, 40

Young's inequality, 100