Adaptive Finite Element Methods

R. Verfürth

Fakultät für Mathematik
Ruhr-Universität Bochum

www.ruhr-uni-bochum.de/num1

Lecture Series / Bochum / Winterterm 2018/19

Contents

Introduction

A Posteriori Error Estimates

Implementation

Solution of Discrete Problems

References

Introduction

- Motivation
- Sobolev spaces
- Finite element spaces

Basic Steps of Finite Element Discretization

- Derive a variational formulation of the differential equation.
- Replace the infinite dimensional test and trial spaces of the variational problem by finite dimensional subspaces consisting of functions which are piece-wise polynomials on a partition into non-overlapping subdomains.
- Build the stiffness matrix and the load vector.
- Solve the resulting linear or nonlinear system of equations.

Basic Steps of A Priori Error Estimation

- General results imply that the discrete problem admits a unique solution.
- General results also imply that the error of the discrete solution is proportional to the error of the best approximation with a constant depending on properties of the variational problem.
- Bound the error of the best approximation by the error of a suitable interpolation.
- A priori error estimates do not require the solution of the discrete problem.

Drawbacks of A Priori Error Estimates

- They only yield information on the asymptotic behaviour of the error.
- They give no information on the actual size of the error nor on its spatial and temporal distribution.
- The error estimate is globally deteriorated by local singularities arising from, e.g., re-entrant corners or interior or boundary layers.

Goals of A Posteriori Error Estimation and Adaptivity

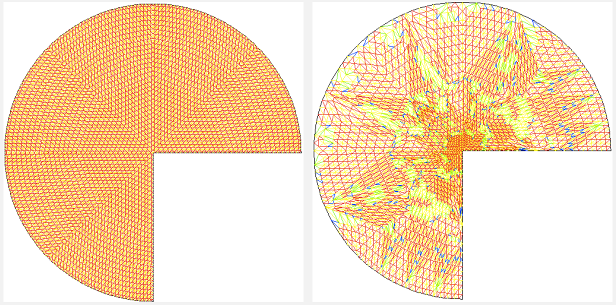
- Extract an easy-to-compute and precise information on the actual size of the error and its spatial and temporal distribution using the data of the differential equation and the computed solution of the discrete problem.
- Obtain an approximation for the solution of the differential equation with a given tolerance using a (nearly) minimal amount of unknowns.

Basic Ingredients of A Posteriori Error Estimates and Adaptivity

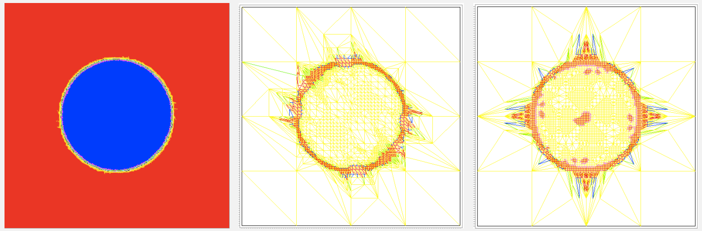
- Stability of the variational problem, i.e. a one-to-one correspondence of load and displacement
- A suitable representation of the error in terms of the residual associated with the discrete solution and the strong form of the differential equation
- Element-wise error estimates for suitable interpolation operators and cut-off functions
- Suitable marking, refinement and coarsening strategies
- The a posteriori error estimates require the explicit knowledge of the solution of the discrete problem.
- The optimal discrete solution and the associated mesh are determined in an iterative way by successively improving the solution and the mesh.

Example: Poisson Equation with Singular Solution

Refinement	Elements	Unknowns	Error
uniform	24576	12033	0.5%
adaptive	11242	5529	0.5%



Example: Reaction-Diffusion Equation with an Interior Layer



	Triangles		Quadrilaterals	
	uniform	adaptive	uniform	adaptive
Unknowns	16129	2923	16129	4722
Triangles	32768	5860	0	3830
Quadrilaterals	0	0	16384	2814
Error	3.8%	3.5%	6.1%	4.4%

Variational Formulation

- Divergence theorem
- Weak derivatives
- Sobolev spaces
- Properties of Sobolev spaces

Discretization

- Finite element partitions
- Finite element spaces
- Basis functions
- Interpolation error estimates
- Building the stiffness matrix and load vector

A Posteriori Error Estimates for a Model Problem

- Residual estimates
  - Equivalence of error and residual
  - L^2-representation of the residual
  - Local error estimates for suitable interpolation operators
  - Local inverse estimates for suitable cut-off functions
- Estimates based on the solution of auxiliary local discrete problems
- Estimates based on an averaging of the gradient
- Estimates based on a hierarchical splitting of finite element spaces
- Estimates based on a suitable $H(\text{div})$-lifting of the residual

A Posteriori Error Estimates for Elliptic Problems

- General elliptic equations of 2nd order
- Saddle point problems arising from mixed formulations in elasticity and fluid mechanics
- Non-linear problems

A Posteriori Error Estimates for Parabolic Problems

- Discretization
- Space-time finite elements
- Method of characteristics
- Adaptivity

Finite Volume Methods

- Systems in divergence form
- Finite volume discretization
- Finite volume meshes
- Numerical fluxes
- Relation to finite element methods
- Discontinuous Galerkin methods

Mesh Adaption

- Basic adaptive algorithm
- Marking strategies
- Mesh refinement
- Mesh coarsening
- Mesh smoothing
- Data structures

Solution of Discrete Problems

- Properties of direct and iterative solvers
- Nested algorithm
- Classical iterative solvers
- Conjugate Gradient methods
- Multigrid methods
- Non-linear and indefinite problems

Sobolev Spaces

- Basic idea
- Integration by parts
- Weak derivatives
- Sobolev spaces
- Properties of Sobolev spaces

Reaction-Diffusion Equation

$$\begin{aligned} -\operatorname{div}(A\nabla u) + \alpha u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma \end{aligned}$$

- Ω a polyhedron in $\mathbb{R}^d$ with $d = 2$ or $d = 3$
- $A(x)$ a symmetric positive definite, $d \times d$ matrix for every x in Ω
- $\alpha(x)$ a non-negative number for every x in Ω

Divergence Theorem

- Divergence:

$$\operatorname{div}\mathbf{w} = \sum_{i=1}^{d} \frac{\partial w_i}{\partial x_i}$$

- Divergence Theorem:

$$\int_\Omega \operatorname{div}\mathbf{w}dx = \int_\Gamma \mathbf{w}\cdot\mathbf{n}dS$$

Integration by Parts in Several Dimensions I

- The divergence theorem applied to $\mathbf{w} = v(A\nabla u)$ yields

$$\int_\Omega v\operatorname{div}(A\nabla u)dx + \int_\Omega \nabla v\cdot A\nabla u dx$$
$$= \int_\Omega \operatorname{div}(vA\nabla u)dx = \int_\Omega \operatorname{div}\mathbf{w}dx = \int_\Gamma \mathbf{w}\cdot\mathbf{n}dS$$
$$= \int_\Gamma v\mathbf{n}\cdot A\nabla u dS.$$

- If $v = 0$ on Γ, this implies

$$\int_\Omega \nabla v\cdot A\nabla u dx = -\int_\Omega v\operatorname{div}(A\nabla u)dx.$$

Idea of the Variational Formulation

- Multiply the differential equation with a continuously differentiable function v with $v = 0$ on Γ

$$-\operatorname{div}(A\nabla u)(x)v(x) + \alpha(x)u(x)v(x) = f(x)v(x) \text{ für } x\in\Omega.$$

- Integrate the result over Ω

$$\int_\Omega \left[-\operatorname{div}(A\nabla u)v + \alpha uv\right]dx = \int_\Omega fv dx.$$

- Use integration by parts for the term containing derivatives

$$-\int_\Omega \operatorname{div}(A\nabla u)v dx = \int_\Omega \nabla v\cdot A\nabla u dx.$$

Problems

- The properties of the functions u and v must be stated more precisely to obtain a well-posed variational problem.
- Classical properties such as ‘continuously differentiable’ are too restrictive.
- The notion ‘derivative’ must be generalised.
- In view of the discrete problems, piecewise differentiable functions should be differentiable in the new weaker sense.

Integration by Parts in Several Dimensions II

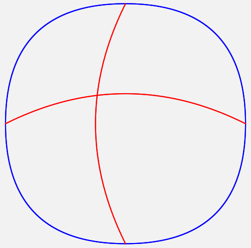
- The divergence theorem applied to $\mathbf{w} = uv\mathbf{e}_i$ ($\mathbf{e}_i$ i-th unit vector with i-th component 1 and vanishing remaining components) yields
$$\int_\Omega \frac{\partial u}{\partial x_i} v dx + \int_\Omega u \frac{\partial v}{\partial x_i} dx$$
$$= \int_\Omega \frac{\partial (uv)}{\partial x_i} dx = \int_\Omega \operatorname{div} \mathbf{w} dx = \int_\Gamma \mathbf{w} \cdot \mathbf{n} dS$$
$$= \int_\Gamma uv\mathbf{n}_i dS.$$
- If $u = 0$ or $v = 0$ on Γ, this implies
$$\int_\Omega \frac{\partial u}{\partial x_i} v dx = - \int_\Omega u \frac{\partial v}{\partial x_i} dx.$$

Weak Derivative

- The function u is said to be weakly differentiable w.r.t. x_i with weak derivative w_i, if every continuously differentiable function v with $v = 0$ on Γ satisfies
$$\int_\Omega w_i v dx = - \int_\Omega u \frac{\partial v}{\partial x_i} dx.$$
- If u is weakly differentiable w.r.t. to all variables $x_1, \ldots, x_d$, we call u weakly differentiable and write ∇u for the vector $(w_1, \ldots, w_d)$ of the weak derivatives.

Examples

- Every function which is continuously differentiable in the classical sense is weakly differentiable and its classical derivative coincides with the weak derivative.
- Every continuous piecewise differentiable function is weakly differentiable and its weak derivative is the piecewise classical derivative.



Sobolev Spaces

- $\|v\| = \left\{ \int_\Omega |v|^2 dx \right\}^{\frac{1}{2}}$ denotes the L^2-norm.
- $L^2(\Omega)$ is the Lebesgue space of all functions v with finite L^2-norm $\|v\|$.
- $H^1(\Omega)$ is the Sobolev space of all functios v in $L^2(\Omega)$, which are weakly differentiable and for which $|\nabla v|$, the Euclidean norm of ∇v, is in $L^2(\Omega)$.
- $H^1_0(\Omega)$ is the Sobolev space of all functions v in $H^1(\Omega)$ with $v = 0$ on Γ.
- $H^1_D(\Omega)$ is the Sobolev space of all functions v in $H^1(\Omega)$ with $v = 0$ on a subset Γ_D of Γ.

Examples

- Every bounded function is in $L^2(\Omega)$.
- $v(x) = \frac{1}{\sqrt{x^2+y^2}}$ is not in $L^2(B(0,1))$ ($B(0,1)$ circle with radius 1 centred at the origin), since $\int_{B(0,1)} |v(x)|^2 dx = 2\pi \int_0^1 \frac{1}{r} dr$ is not finite.
- Every continuously differentiable function is in $H^1(\Omega)$.
- A piecewise differentiable function is in $H^1(\Omega)$, if and only if it is globally continuous.
- Functions in $H^1(\Omega)$ must not admit point values. $v(x) = \ln(|\ln(\sqrt{x^2+y^2})|)$ is in $H^1(B(0,1))$ but has no finite value at the origin.

Variational Problem

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

$$\int_\Omega \left[\nabla v \cdot A \nabla u + \alpha u v\right] dx = \int_\Omega f v dx.$$

Properties of the Variational Problem

- The variational problem admits a unique solution.
- The solution of the variational problem is the unique minimum in $H_0^1(\Omega)$ of the energy function $\frac{1}{2}\int_\Omega \left[\nabla u \cdot A \nabla u + \alpha u^2\right] dx - \int_\Omega f u dx.$

Finite Element Spaces

- Partitions
- Finite element spaces
- Local and global degrees of freedom
- Nodal basis functions
- Evaluation of the nodal basis functions

Reaction-Diffusion Equation

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

$$\int_\Omega [\nabla v \cdot A \nabla u + \alpha u v] dx = \int_\Omega f v dx.$$

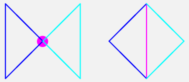
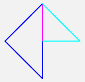
Basic Idea

- Subdivide Ω into non-overlapping simple sub-domains called elements such as triangles, parallelograms, tetrahedra of parallelepipeds, ... (partition).
- In the variational problem replace the space $H_0^1(\Omega)$ by a finite dimensional subspace consisting of continuous functions which are element-wise polynomials (finite element space).
- This gives rise to a linear system of equations for the approximation $u_{\mathcal{T}}$ of the solution u of the differential equation.

Partition

$\mathcal{T} = \{K_i : 1 \le i \le N_{\mathcal{T}}\}$ denotes a partition of Ω with the following properties:

- Ω is the union of all elements K in $\mathcal{T}$.
- Admissibility: Any two elements K and K' in $\mathcal{T}$ are either disjoint or share a vertex or a complete edge or, if $d = 3$, a complete face.

  admissible  not admissible 
- Affine equivalence: Every element K is a triangle or parallelogram, if $d = 2$, or a tetrahedron or parallelepiped, if $d = 3$.

Remarks

- Curved boundaries can be approximated by piecewise straight lines or planes.
- The admissibility is necessary to ensure the continuity of the finite element functions and thus the inclusion of the finite element spaces in $H_0^1(\Omega)$.
- If the admissibility is violated, the continuity of the finite element functions must be enforced which leads to a more complicated implementation.
- Partitions can also consist of general quadrilaterals or hexahedra which leads to a more complicated implementation.

Finite Element Spaces

- $$R_k(\widehat{K}) = \begin{cases} \operatorname{span}\{x_1^{\alpha_1}\cdot\ldots\cdot x_d^{\alpha_d} : \alpha_1+\ldots+\alpha_d \le k\} \\ \widehat{K} \text{ reference simplex} \\ \operatorname{span}\{x_1^{\alpha_1}\cdot\ldots\cdot x_d^{\alpha_d} : \max\{\alpha_1,\ldots,\alpha_d\} \le k\} \\ \widehat{K} \text{ reference cube} \end{cases}$$
- $R_k(K) = \{\widehat{p}\circ F_K^{-1} : \widehat{p}\in \widehat{R}_k\}$
- $S^{k,-1}(\mathcal{T}) = \{v:\Omega\to\mathbb{R} : v\big|_K \in R_k(K) \text{ for all } K\in\mathcal{T}\}$
- $S^{k,0}(\mathcal{T}) = S^{k,-1}(\mathcal{T})\cap C(\overline{\Omega})$
- $S_0^{k,0}(\mathcal{T}) = S^{k,0}(\mathcal{T})\cap H_0^1(\Omega)$
  $= \{v\in S^{k,0}(\mathcal{T}) : v=0 \text{ on } \Gamma\}$
- $S_D^{k,0}(\mathcal{T}) = S^{k,0}(\mathcal{T})\cap H_D^1(\Omega)$
  $= \{v\in S^{k,0}(\mathcal{T}) : v=0 \text{ on } \Gamma_D\}$

Remarks

- The global continuity ensures that $S^{k,0}(\mathcal{T})\subset H^1(\Omega)$.
- The polynomial degree k may vary from element to element; this leads to a more complicated implementation.

Discrete Problem

Find $u_{\mathcal{T}}\in S_0^{k,0}(\mathcal{T})$ (trial function) such that for all $v_{\mathcal{T}}\in S_0^{k,0}(\mathcal{T})$ (test function)

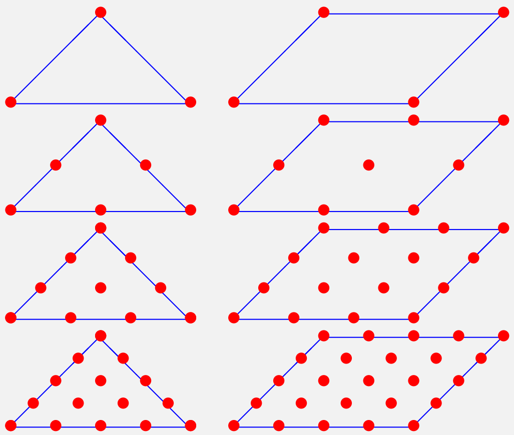
$$\int_\Omega \left[\nabla v_{\mathcal{T}}\cdot A\nabla u_{\mathcal{T}} + \alpha u_{\mathcal{T}} v_{\mathcal{T}}\right]dx = \int_\Omega f v_{\mathcal{T}} dx.$$

Properties of the Discrete Problem

- The discrete problem admits a unique solution.
- The solution of the discrete problem is the unique minimum in $S_0^{k,0}(\mathcal{T})$ of the energy function
  $$\frac{1}{2}\int_\Omega\left[\nabla u\cdot A\nabla u+\alpha u^2\right]dx-\int_\Omega f u dx.$$
- After choosing a basis for $S_0^{k,0}(\mathcal{T})$ the discrete problem amounts to a linear system of equations with $\approx k^d N_{\mathcal{T}}$ ($N_{\mathcal{T}} = \sharp\mathcal{T}$) equations and unknowns.

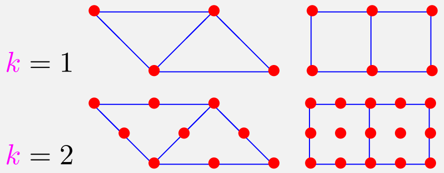
Element-wise Degrees of Freedom $\mathcal{N}_{K,k}$

- $k = 1$
- $k = 2$
- $k = 3$
- $k = 4$



Global Degrees of Freedom $\mathcal{N}_{\mathcal{T},k}$

- $\mathcal{N}_{\mathcal{T},k} = \bigcup_{K \in \mathcal{T}} \mathcal{N}_{K,k}$

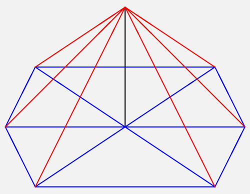


- The functions in $S^{k,0}(\mathcal{T})$ are uniquely defined by their values in $\mathcal{N}_{\mathcal{T},k}$ thanks to the admissibility of $\mathcal{T}$.

Nodal Basis Functions

The nodal basis function associated with a vertex $z \in \mathcal{N}_{\mathcal{T},k}$ is uniquely defined by he conditions

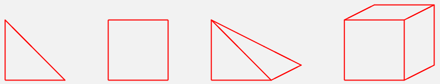
- $\lambda_{z,k} \in S^{k,0}(\mathcal{T})$,
- $\lambda_{z,k}(z) = 1$,
- $\lambda_{z,k}(y) = 0$ for all $y \in \mathcal{N}_{\mathcal{T},k} \setminus \{z\}$.



Properties

- $\{\lambda_{z,k} : z \in \mathcal{N}_{\mathcal{T},k}\}$ is a basis for $S^{k,0}(\mathcal{T})$.
- $\{\lambda_{z,k} : z \in \mathcal{N}_{\mathcal{T},k} \setminus \Gamma\}$ is a basis for $S_0^{k,0}(\mathcal{T})$.
  (Degrees of freedom on the boundary Γ are suppressed.)
- $\lambda_{z,k}$ vanishes outside the union of all elements that share the vertex z.
- The stiffness matrix is sparse.

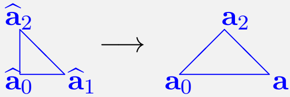
Evaluation of the Nodal Basis Functions by Transformation to a Reference Element

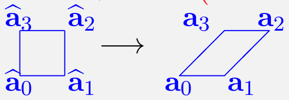
- Reference elements $\widehat{K}$ 
- Determine the nodal basis functions $\widehat{\lambda}_{\widehat{z},k}$ for the reference element $\widehat{K}$.
- Determine an affine transformation of the reference element $\widehat{K}$onto the current element K
  $\widehat{K} \ni \widehat{x} \mapsto x= b_K + B_K\widehat{x} \in K$.
- Express $\lambda_{z,k}$ in terms of $\widehat{\lambda}_{\widehat{z},k}$ using the affine transformation
  $\lambda_{z,k}(x) = \widehat{\lambda}_{\widehat{z},k}(\widehat{x})$.

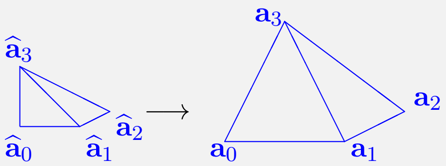
Examples for $\widehat{\lambda}_{\widehat{z},k}$

- Reference triangle ◺
  - $k = 1$ Vertices $1-x-y$, x, y
  - $k = 2$
    Vertices $(1-x-y)(1-2x-2y)$, $x(2x-1)$, $y(2y-1)$
    Midpoints of edges $4x(1-x-y)$, $4xy$, $4y(1-x-y)$
- Reference square □
  - $k = 1$ Vertices $(1-x)(1-y)$, $x(1-y)$, xy, $(1-x)y$
  - $k = 2$
    Vertices $(1-2x)(1-x)(1-2y)(1-y)$,
    $x(2x-1)(1-2y)(1-y)$, $x(2x-1)y(2y-1)$,
    $(1-2x)(1-x)y(2y-1)$
    Midpoints of edges $4x(1-x)(1-y)(1-2y)$,
    $4x(2x-1)y(1-y)$,
    $4x(1-x)y(2y-1)$, $4y(1-y)(1-2x)(1-x)$
    Barycentre $16x(1-x)y(1-y)$

Examples for Affine Transformations

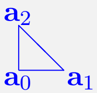
- 

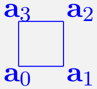
  $b_K = \mathbf{a}_0$, $B_K = \left(\mathbf{a}_1 - \mathbf{a}_0 \,,\, \mathbf{a}_2 - \mathbf{a}_0\right)$
- 

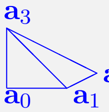
  $b_K = \mathbf{a}_0$, $B_K = \left(\mathbf{a}_1 - \mathbf{a}_0 \,,\, \mathbf{a}_3 - \mathbf{a}_0\right)$
- 

  $b_K = \mathbf{a}_0$, $B_K = \left(\mathbf{a}_1 - \mathbf{a}_0 \,,\, \mathbf{a}_2 - \mathbf{a}_0 \,,\, \mathbf{a}_3 - \mathbf{a}_0\right)$
- Similar formulae hold for parallelepipeds.

Evaluation Using the Element Geometry $(k = 1)$

- 

$$\lambda_{\mathbf{a}_i,1}(x) = \frac{\det(x - \mathbf{a}_{i+1} \,,\, \mathbf{a}_{i+2} - \mathbf{a}_{i+1})}{\det(\mathbf{a}_i - \mathbf{a}_{i+1} \,,\, \mathbf{a}_{i+2} - \mathbf{a}_{i+1})}$$
- 

$$\lambda_{\mathbf{a}_i,1}(x) = \frac{\det(x - \mathbf{a}_{i+2} \,,\, \mathbf{a}_{i+3} - \mathbf{a}_{i+2})}{\det(\mathbf{a}_i - \mathbf{a}_{i+2} \,,\, \mathbf{a}_{i+3} - \mathbf{a}_{i+2})} \cdot \frac{\det(x - \mathbf{a}_{i+2} \,,\, \mathbf{a}_{i+1} - \mathbf{a}_{i+2})}{\det(\mathbf{a}_i - \mathbf{a}_{i+2} \,,\, \mathbf{a}_{i+1} - \mathbf{a}_{i+2})}$$
- 

$$\lambda_{\mathbf{a}_i,1}(x) = \frac{\det(x - \mathbf{a}_{i+1}, \mathbf{a}_{i+2} - \mathbf{a}_{i+1}, \mathbf{a}_{i+3} - \mathbf{a}_{i+1})}{\det(\mathbf{a}_i - \mathbf{a}_{i+1}, \mathbf{a}_{i+2} - \mathbf{a}_{i+1}, \mathbf{a}_{i+3} - \mathbf{a}_{i+1})}$$
- Parallelepipeds similarly with 3 factors corresponding to 3 tetrahedra
- All indices must be taken modulo the number of element vertices.

Evaluation Using the Element Geometry $(k \geq 2)$

- Every $\lambda_{z,k}$ can be represented as a suitable product of first order nodal basis functions $\lambda_{\mathbf{a}_i,1}$ associated with the element vertices.
- Example: triangle, $k = 2$
  - Vertex $\mathbf{a}_i$
    $\lambda_{\mathbf{a}_i,2} = \lambda_{\mathbf{a}_i}[\lambda_{\mathbf{a}_i} - \lambda_{\mathbf{a}_{i+1}} - \lambda_{\mathbf{a}_{i+2}}]$
  - Midpoint z of the edge with endpoints $\mathbf{a}_i$ und $\mathbf{a}_{i+1}$
    $\lambda_{z,2} = 4\lambda_{\mathbf{a}_i}\lambda_{\mathbf{a}_{i+1}}$
- Example: parallelogramm, $k = 2$
  - Vertex $\mathbf{a}_i$
    $\lambda_{\mathbf{a}_i,2} = \lambda_{\mathbf{a}_i}[\lambda_{\mathbf{a}_i} - \lambda_{\mathbf{a}_{i+1}} + \lambda_{\mathbf{a}_{i+2}} - \lambda_{\mathbf{a}_{i+3}}]$
  - Midpoint z of the edge with endpoints $\mathbf{a}_i$ und $\mathbf{a}_{i+1}$
    $\lambda_{z,2} = 4\lambda_{\mathbf{a}_i}[\lambda_{\mathbf{a}_{i+1}} - \lambda_{\mathbf{a}_{i+2}}]$
  - Barycentre z
    $\lambda_{z,2} = 16\lambda_{\mathbf{a}_0}\lambda_{\mathbf{a}_2}$

Neumann Boundary Conditions

- The Neumann boundary condition $\mathbf{n} \cdot A\nabla u = g$ on $\Gamma_N \subset \Gamma$ gives rise to
  - an additional term $\int_{\Gamma_N} gvdS$ on the right-hand side of the variational problem,
  - an additional term $\int_{\Gamma_N} gv_{\mathcal{T}}dS$ on the right-hand side of the discrete problem.
- The additional entries of the load vector are taken into account when sweeping through the elements.
- Degrees of freedom associated with points on the Neumann boundary Γ_N are additional unknowns.

A Posteriori Error Estimates

- A residual error estimator for the model problem
- Other error estimators for the model problem
- Elliptic problems
- Parabolic problems
- Finite volume methods

Model Problem: Poisson Equation with Mixed Dirichlet and Neumann Boundary Conditions

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= g \quad \text{on } \Gamma_N \end{aligned}$$

Variational Problem

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

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\Gamma_N} g v.$$

Discrete Problem

Find $u_\mathcal{T} \in S^{1,0}_D(\mathcal{T})$ such that for all $v_\mathcal{T} \in S^{1,0}_D(\mathcal{T})$

$$\int_\Omega \nabla u_\mathcal{T} \cdot \nabla v_\mathcal{T} = \int_\Omega f v_\mathcal{T} + \int_{\Gamma_N} g v_\mathcal{T}.$$

Residual

The residual R is implicitely defined as the mapping

$$R : H^1_D(\Omega) \ni v \mapsto \underbrace{\int_\Omega f v + \int_{\Gamma_N} g v - \int_\Omega \nabla u_\mathcal{T} \cdot \nabla v}_{=\langle R, v\rangle} \in \mathbb{R}.$$

Equivalence of Error and Residual

- Every $v \in H^1_D(\Omega)$ satifies
  $\langle R, v\rangle = \int_\Omega \nabla(u - u_\mathcal{T}) \cdot \nabla v.$
- This implies
  $\|\nabla(u - u_\mathcal{T})\| = \sup_{v \in H^1_D(\Omega); \|\nabla v\| = 1} \langle R, v\rangle.$
- The energy norm of the error (displacement) is the same as the dual norm of the residual (load).

Consequences

- It suffices to estimate the dual norm of the residual.
- The residual only incorporates the known data of the differential equation and the computed solution of the discrete problem.
- The exact evaluation of the dual norm is as costly as the solution of the variational problem.
- All error estimators strive to approximate as well as possible the dual norm by an easy-to-compute quantity.

Basic Ingredients for Deriving a Residual Error Estimator

- Galerkin orthogonality of the error
- L^2-representation of the residual
- Local error estimates for a quasi-interpolation operator
- Local inverse estimates for cut-off functions

Galerkin Orthogonality

- Since $S_D^{1,0}(\mathcal{T}) \subset H_D^1(\Omega)$ inserting discrete test functions in the variational problem and subtracting the result from the discrete problem yields for every $v_{\mathcal{T}} \in S_D^{1,0}(\mathcal{T})$
$$\int_\Omega \nabla(u - u_{\mathcal{T}}) \cdot \nabla v_{\mathcal{T}} = 0.$$
- This implies for every $v_{\mathcal{T}} \in S_D^{1,0}(\mathcal{T})$
$$\langle R, v_{\mathcal{T}} \rangle = 0.$$

L^2-Representation

Integration by parts element-wise yields for every $v \in H_D^1(\Omega)$

$$\begin{aligned}
\langle R, v \rangle &= \int_\Omega fv + \int_{\Gamma_N} gv - \sum_{K \in \mathcal{T}} \int_K \nabla u_{\mathcal{T}} \cdot \nabla v \\
&= \int_\Omega fv + \int_{\Gamma_N} gv + \sum_{K \in \mathcal{T}} \Big\{ \int_K \Delta u_{\mathcal{T}} v - \int_{\partial K} \mathbf{n}_K \cdot \nabla u_{\mathcal{T}} v \Big\} \\
&= \sum_{K \in \mathcal{T}} \int_K \underbrace{(f + \Delta u_{\mathcal{T}})}_{=R_K(u_{\mathcal{T}})} v + \sum_{E \in \mathcal{E}_N} \int_E \underbrace{(g - \mathbf{n}_E \cdot \nabla u_{\mathcal{T}})}_{=R_E(u_{\mathcal{T}})} v \\
&\quad + \sum_{E \in \mathcal{E}_\Omega} \int_E \underbrace{-\mathbb{J}_E(\mathbf{n}_E \cdot \nabla u_{\mathcal{T}})}_{=R_E(u_{\mathcal{T}})} v.
\end{aligned}$$

A Quasi-Interpolation Operator

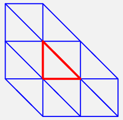
- Define the quasi-interpolation operator $I_{\mathcal{T}} : L^2(\Omega) \to S_D^{1,0}(\mathcal{T})$ by

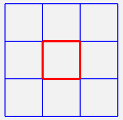
$$I_{\mathcal{T}} v = \sum_{z \in \mathcal{N}_\Omega \cup \mathcal{N}_N} \lambda_z \overline{v}_z \quad \text{with} \quad \overline{v}_z = \frac{\int_{\omega_z} v dx}{\int_{\omega_z} dx}.$$

- It has the following local approximation properties for all $v \in H_D^1(\Omega)$

$$\|v - I_{\mathcal{T}} v\|_K \le c_{A1} h_K |v|_{1,\widetilde{\omega}_K}$$

$$\|v - I_{\mathcal{T}} v\|_{\partial K} \le c_{A2} h_K^{\frac{1}{2}} |v|_{1,\widetilde{\omega}_K}.$$





- $\widetilde{\omega}_K$ is the union of all elements that share a point with K.

Upper Bounds

The Galerkin orthogonality, the L^2-representation, the error estimates for the quasi-interpolation operator and the Cauchy-Schwarz inequality for integrals and sums yield for every $v \in H_D^1(\Omega)$ the upper bound

$$\langle R, v \rangle \le c^* \Big\{ \sum_{K \in \mathcal{T}} h_K^2 \|R_K(u_{\mathcal{T}})\|_K^2 + \sum_{E \in \mathcal{E}_{\mathcal{T}}} h_E \|R_E(u_{\mathcal{T}})\|_E^2 \Big\}^{\frac{1}{2}} \|\nabla v\|.$$

Bubble Functions

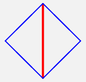
- Define element and face bubble functions by

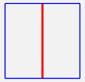
$$\psi_K = \alpha_K \prod_{z \in \mathcal{N}_K} \lambda_z, \quad \psi_E = \alpha_E \prod_{z \in \mathcal{N}_E} \lambda_z.$$

- The weights α_K and α_E are determined by the conditions

$$\max_{x \in K} \psi_K(x) = 1, \quad \max_{x \in E} \psi_E(x) = 1.$$

- K is the support of ψ_K; ω_E is the support of ψ_E.





Inverse Estimates for the Bubble Functions

For all elements K, all faces E and all polynomials v the following inverse estimates are valid

$$c_{I1,k} \|v\|_K \le \|\psi_K^{\frac{1}{2}} v\|_K,$$

$$\|\nabla(\psi_K v)\|_K \le c_{I2,k} h_K^{-1} \|v\|_K,$$

$$c_{I3,k} \|v\|_E \le \|\psi_E^{\frac{1}{2}} v\|_E,$$

$$\|\nabla(\psi_E v)\|_{\omega_E} \le c_{I4,k} h_E^{-\frac{1}{2}} \|v\|_E,$$

$$\|\psi_E v\|_{\omega_E} \le c_{I5,k} h_E^{\frac{1}{2}} \|v\|_E.$$

Lower Bounds

Inserting $\psi_K R_K(u_{\mathcal{T}})$ and $\psi_E R_E(u_{\mathcal{T}})$ as test functions in the L^2-representation and taking into account the inverse estimates for the bubble functions yields the lower bounds

$$\begin{aligned} h_K \|R_K(u_{\mathcal{T}})\|_K &\le c_1 \|\nabla(u - u_{\mathcal{T}})\|_K + c_2 h_K \|f - f_K\|_K \\ h_E^{\frac{1}{2}} \|R_E(u_{\mathcal{T}})\|_E &\le c_3 \|\nabla(u - u_{\mathcal{T}})\|_{\omega_E} + c_4 h_K \|f - f_K\|_{\omega_E} \\ &\quad + c_5 h_E^{\frac{1}{2}} \|g - g_E\|_{E \cap \Gamma_N} \end{aligned}$$

with $f_K = \dfrac{1}{|K|} \displaystyle\int_K f$ and $g_E = \dfrac{1}{|E|} \displaystyle\int_E g$.

Residual Error Estimator

$$\begin{aligned} \eta_{R,K} = \Big\{ & h_K^2 \|f_K + \Delta u_{\mathcal{T}}\|_K^2 \\ & + \frac{1}{2} \sum_{E \in \mathcal{E}_{K,\Omega}} h_E \|\mathbb{J}_E(\mathbf{n}_E \cdot \nabla u_{\mathcal{T}})\|_E^2 \\ & + \sum_{E \in \mathcal{E}_{K,N}} h_E \|g_E - \mathbf{n}_E \cdot \nabla u_{\mathcal{T}}\|_E^2 \Big\}^{\frac{1}{2}} \end{aligned}$$

with $f_K = \dfrac{1}{|K|} \displaystyle\int_K f$ and $g_E = \dfrac{1}{|E|} \displaystyle\int_E g$

Structure of the Error Estimator

- The approximations f_K and g_E of the data f and g are needed for establishing a lower bound for the error. They also facilitate the evaluation of $\eta_{R,K}$.
- $f_K + \Delta u_{\mathcal{T}}$ is the residual of the discrete solution w.r.t. the strong form of the differential operator.
- $\mathbb{J}_E(\mathbf{n}_E \cdot \nabla u_{\mathcal{T}})$ and $g_E - \mathbf{n}_E \cdot \nabla u_{\mathcal{T}}$ are the boundary terms which appear in the integration by parts relating the weak and strong form of the differential equation.

Residual A Posteriori Error Estimates

$$\begin{aligned} \|\nabla(u - u_{\mathcal{T}})\| \le c^* \Big\{ & \sum_{K \in \mathcal{T}} \eta_{R,K}^2 + \sum_{K \in \mathcal{T}} h_K^2 \|f - f_K\|_K^2 \\ & + \sum_{E \in \mathcal{E}_N} h_E \|g - g_E\|_E^2 \Big\}^{\frac{1}{2}} \\ \eta_{R,K} \le c_* \Big\{ & \|\nabla(u - u_{\mathcal{T}})\|_{\omega_K}^2 \\ & + \sum_{K' \in \mathcal{T}; \mathcal{E}_{K'} \cap \mathcal{E}_K \neq \emptyset} h_{K'}^2 \|f - f_{K'}\|_{K'}^2 \\ & + \sum_{E \in \mathcal{E}_{K,N}} h_E \|g - g_E\|_E^2 \Big\}^{\frac{1}{2}} \end{aligned}$$

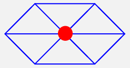
Structure of the Error Estimates

- The data oscillations $f - f_K$ and $g - g_E$ can be computed explicitly and are often of higher order.
- The constants c_* and c^* depend on the shape parameter of the partition.
- The upper bound is a global one since it is related to the inverse of the differential operator which is a global one (local load → global displacement).
- The lower bound is a local one since it is related to the differential operator which is a local one (local displacement → local load).

A Catalogue of Error Estimators

- Auxiliary local discrete problems associated with elements of higher order on patches consisting of
  - elements sharing a given vertex,
  - elements adjacent to a given element,
  - a single element
- Hierarchical estimates
- Averaging of the gradient (ZZ-estimator)
- $H(\mathrm{div})$-lifting of the residual

Auxiliary Dirichlet Problems on ω_z

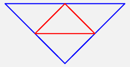


- Set $V_z = \operatorname{span}\{\varphi\psi_K, \rho\psi_E, \sigma\psi_{E'} : K \in \mathcal{T}, z \in \mathcal{N}_K, E \in \mathcal{E}_\Omega, z \in \mathcal{N}_E, E' \in \mathcal{E}_N, E' \subset \partial\omega_z, \varphi, \rho, \sigma \in \mathbb{P}_1\}$.
- Find $v_z \in V_z$ such that for all $w \in V_z$

$$\int_{\omega_z} \nabla v_z \cdot \nabla w = \sum_{K \subset \omega_z} \int_K f_K w + \sum_{E \subset \Gamma_N \cap \partial\omega_z} \int_E g_E w - \int_{\omega_z} \nabla u_{\mathcal{T}} \cdot \nabla w.$$

- Set $\eta_{D,z} = \|\nabla v_z\|_{\omega_z}$.

Auxiliary Dirichlet Problems on ω_K



- Set $\widetilde{V}_K = \operatorname{span}\{\varphi\psi_{K'}, \rho\psi_E, \sigma\psi_{E'} : K' \in \mathcal{T}, \mathcal{E}_K \cap \mathcal{E}_{K'} \neq \emptyset, E \in \mathcal{E}_K, E' \in \mathcal{E}_N, E' \subset \partial\omega_K, \varphi, \rho, \sigma \in \mathbb{P}_1\}$.
- Find $\widetilde{v}_K \in \widetilde{V}_K$ such that for all $w \in \widetilde{V}_K$

$$\int_{\omega_K} \nabla \widetilde{v}_K \cdot \nabla w = \sum_{K' \subset \omega_K} \int_{K'} f_{K'} w + \sum_{E \subset \Gamma_N \cap \partial\omega_K} \int_E g_E w - \int_{\omega_K} \nabla u_{\mathcal{T}} \cdot \nabla w.$$

- Set $\eta_{D,K} = \|\nabla \widetilde{v}_K\|_{\omega_K}$.

Auxiliary Neumann Problems on K

- Set $V_K = \operatorname{span}\{\varphi\psi_K, \rho\psi_E : E \in \mathcal{E}_K \backslash \mathcal{E}_D, \varphi, \rho \in \mathbb{P}_1\}$.
- Find $v_K \in V_K$ such that for all $w \in V_K$

$$\begin{aligned}\int_K \nabla v_K \cdot \nabla w = \int_K (f_K + \Delta u_{\mathcal{T}})w \\ -\frac{1}{2} \sum_{E\in\mathcal{E}_K\cap\mathcal{E}_\Omega} \int_E \mathbb{J}_E(\mathbf{n}_E \cdot \nabla u_{\mathcal{T}})w \\ + \sum_{E\in\mathcal{E}_K\cap\mathcal{E}_N} \int_E (g_E - \mathbf{n}_E \cdot \nabla u_{\mathcal{T}})w.\end{aligned}$$

- Set $\eta_{N,K} = \|\nabla v_K\|_K$.

Hierarchical Estimates

Basic Idea

- Approximately solve the discrete problem using a more accurate finite element space and compare this solution with the original discrete solution.
- To reduce the computational cost of the new problem, the new finite element space is decomposed into the original one and a nearly orthogonal higher order complement; only the contribution corresponding to the complement is computed.
- To further reduce the computational cost, the original bilinear form is replaced by an equivalent one which leads to a diagonal stiffness matrix.

Hierarchical Estimates

Estimator

- For every element or edge S set

$$\eta_S = \frac{1}{\|\nabla\psi_S\|}\Big\{\int_\Omega f\psi_S + \int_{\Gamma_N} g\psi_S - \int_\Omega \nabla u_{\mathcal{T}} \cdot \nabla\psi_S\Big\}.$$

- A hierarchical error estimator is then given by

$$\eta_{H,z} = \Big\{\sum_{K\subset\omega_z} \eta_K^2 + \sum_{E\subset\omega_z} \eta_E^2\Big\}^{\frac{1}{2}}.$$

Averaging of the Gradient (ZZ-Estimator)

- Compare the piece-wise constant gradient $\nabla u_{\mathcal{T}}$ with a continuous piece-wise linear gradient $Gu_{\mathcal{T}}$ obtained by taking a suitable average of the constant values around a given vertex.
- The averaged gradient is given by

$$Gu_{\mathcal{T}}(z) = \sum_{K\subset\omega_z} \frac{|K|}{|\omega_z|} \nabla u_{\mathcal{T}}\big|_K.$$

- The resulting ZZ-error estimator takes the form

$$\eta_{Z,K} = \|Gu_{\mathcal{T}} - \nabla u_{\mathcal{T}}\|_K.$$

$H(\mathrm{div})$-Lifting of the Residual

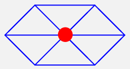
- There is a vector field $\rho_\mathcal{T}$ which admits a divergence element-wise and which satisfies for all $v \in H^1_D(\Omega)$
$$\int_\Omega \rho_\mathcal{T} \cdot \nabla v = \int_\Omega fv + \int_{\Gamma_N} gv - \int_\Omega \nabla u_\mathcal{T} \cdot \nabla v.$$
- $\rho_\mathcal{T}$ can be constructed by sweeping through the elements and may be chosen from suitable finite dimensional spaces.
- The resulting error estimator is given by
$$\eta_\mathcal{T} = \|\rho_\mathcal{T}\|.$$

Construction of $\rho_\mathcal{T}$

Single Element

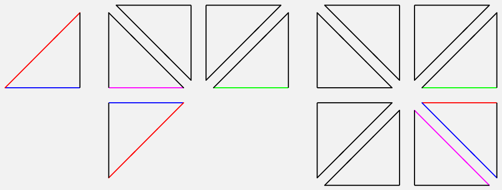
- Assume that $F \in L^2(K)$ and $G \in L^2(\partial K)$ satisfy
$$\int_K F + \int_{\partial K} G = 0.$$
- Then there is a vector field ρ_K with
  - $-\operatorname{div} \rho_K = F$ on K,
  - $\rho_K \cdot \mathbf{n}_K = G$ on ∂K.
- If F and G are polynomials, ρ_K is a polynomial too and can be computed by solving an appropriate linear system of equations.

Construction of $\rho_\mathcal{T}$



Patch of Elements

- Sweep through the elements K sharing a given vertex z.
- Apply the previous result to
$F = \lambda_z R_K(u_\mathcal{T})$ and
$$G = \begin{cases} \lambda_z R_{\partial K}(u_\mathcal{T}) & \text{on } (\partial K \cap \sigma_z) \setminus (E \cup E'), \\ \alpha_{E'} & \text{on } E', \\ \lambda_z R_{\partial K}(u_\mathcal{T}) - \alpha_E & \text{on } E, \\ 0 & \text{on } \partial K \setminus \sigma_z \end{cases}$$



Construction of $\rho_\mathcal{T}$

Global Assembly

- The previous step yields vector fields ρ_z.
- Set
$$\rho_\mathcal{T} = \sum_z \rho_z.$$
- Then
$$\int_\Omega \rho_\mathcal{T} \cdot \nabla v = \int_\Omega fv + \int_{\Gamma_N} gv - \int_\Omega \nabla u_\mathcal{T} \cdot \nabla v.$$

Equivalence of the Error Estimators

- All estimators are equivalent to the residual estimator in that they can be bounded from above and from below by constant multiples of the latter.
- The evaluation of the residual estimator is less costly.
- The residual estimator is well suited for controlling the mesh adaptation process.
- The estimators based on the solution of auxiliary problems often yield more accurate numerical values for the error.
- Contrary to the other estimators, the ZZ-estimator and the estimator based on the $H(\mathrm{div})$-lifting are not robust w.r.t. to dominant low order terms in the differential equation.

Elliptic Problems

- Linear elliptic problems of 2nd order
- Saddle-point problems
- Non-linear problems

Linear Elliptic Equations of 2nd Order

The Setting

- Differential equation:

$$\begin{aligned} -\mathrm{div}(A\nabla u) + \mathbf{a}\cdot\nabla u + \alpha u &= f && \text{in } \Omega \\ u &= 0 && \text{on } \Gamma_D \\ \mathbf{n}\cdot A\nabla u &= g && \text{on } \Gamma_N \end{aligned}$$

- Key parameters:
  - ε a positive lower bound for the smallest eigenvalue of the diffusion matrix A,
  - β a non-negative lower bound for $\alpha - \frac{1}{2}\,\mathrm{div}\,\mathbf{a}$
- An error estimator is called robust if it gives upper and lower bounds for the error which are uniform w.r.t. the parameters ε and β.

Linear Elliptic Equations of 2nd Order

A Robust Residual Error Estimator

- Define element residuals, edge residuals and weighting factors by
  - $R_K(u_{\mathcal{T}}) = f_K + \mathrm{div}(A\nabla u_{\mathcal{T}}) - \mathbf{a}\cdot\nabla u_{\mathcal{T}} - \alpha u_{\mathcal{T}}$,
  - $R_E(u_{\mathcal{T}}) = \begin{cases} -\mathbb{J}_E(\mathbf{n}_E\cdot A\nabla u_{\mathcal{T}}) & \text{if } E\in\mathcal{E}_\Omega, \\ g - \mathbf{n}_E\cdot A\nabla u_{\mathcal{T}} & \text{if } E\in\mathcal{E}_N, \\ 0 & \text{if } E\in\mathcal{E}_D, \end{cases}$
  - $\alpha_S = \min\{\varepsilon^{-\frac{1}{2}}h_S, \beta^{-\frac{1}{2}}\}$ for $S\in\mathcal{T}\cup\mathcal{E}$.
- A robust residual error estimator is then given by

$$\eta_{R,K} = \Big\{\alpha_K^2\|R_K(u_{\mathcal{T}})\|_K^2 + \sum_{E\in\mathcal{E}_K}\varepsilon^{-\frac{1}{2}}\alpha_E\|R_E(u_{\mathcal{T}})\|_E^2\Big\}^{\frac{1}{2}}.$$

Mixed Formulation of the Poisson Equation

- Differential equation Mixed formulation

$$-\Delta u = f \quad \text{in } \Omega \qquad\qquad \sigma = \nabla u \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Gamma \qquad\qquad \operatorname{div} \sigma = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \Omega$$

- Error estimator

$$\eta_K = \Big\{ \|\operatorname{div} \sigma_{\mathcal{T}} + f\|_K^2 + h_K^2 \|\sigma_{\mathcal{T}} - \nabla u_{\mathcal{T}}\|_K^2 + \sum_{E \subset \partial K} h_E \|\mathbb{J}_E(\sigma_{\mathcal{T}} \cdot \mathbf{t}_E)\|_E^2 \Big\}^{\frac{1}{2}}$$

Equations of Linearized Elasticity

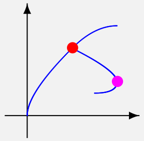
- The displacement formulation can be handled in the same way as the Poisson equation.
- The displacement formulation breaks down for nearly incompressible materials which is reflected by the so-called locking phenomenon.
- The locking phenomenon can be overcome by using a mixed formulation similar to the one of the Poisson equation.
- The corresponding residual error estimator is similar to the one for the mixed formulation of the Poisson equation.
- Special attention must be paid to the robustness of the error estimator w.r.t. to the Lamé parameters.

Fluid Mechanics

- A residual error estimator consists of the element residuals of the momentum and the continuity equation and of the edge residuals of the momentum equation.
- The terms corresponding to the momentum equation ($-\Delta \mathbf{u} + Re(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f}$) have the same weighting factors as for the model problem.
- The term corresponding to the continuity equation ($\operatorname{div} \mathbf{u} = 0$) has the weighting factor 1.
- The correct scaling of the weighting factors can be deduced by a dimensional analysis.
- It is not possible to separate the errors of the velocity and pressure approximation.

Non-Linear Problems

- Residual error estimators are constructed in the same way as for linear problems.
- Auxiliary local problems are based on a linearization of the non-linear problem; the non-linearity only enters into the right-hand side of the auxiliary problem.
- A posteriori error estimates only make sense for well-posed problems which exhibit a continuous dependence of the solution (displacement) on the right-hand side (load).
- Bifurcation points and turning points require special techniques.



Parabolic Problems

- Discretizations
- Space-time finite elements
- Method of characteristics
- Adaptivity

Linear Parabolic Equation of 2nd Order

$$\frac{\partial u}{\partial t} - \operatorname{div}(A\nabla u) + \mathbf{a}\cdot\nabla u + \alpha u = f \quad \text{in } \Omega\times(0,T]$$

$$u = 0 \quad \text{on } \Gamma\times(0,T]$$

$$u(\cdot,0) = u_0 \quad \text{in } \Omega$$

- Ω polyhedron in $\mathbb{R}^d$ with $d = 2$ or $d = 3$
- $A(x,t)$ symmetric positive definite matrix for every x in Ω, t in $(0,T]$
- $\mathbf{a}(x,t)$ vector in $\mathbb{R}^d$ for every x in Ω, t in $(0,T]$
- $\alpha(x,t)$ non negative number for every x in Ω, t in $(0,T]$
- $\alpha(x,t) - \frac{1}{2}\operatorname{div}\mathbf{a}(x,t) \geq 0$ for every x in Ω, t in $(0,T]$

Standard Discretizations

- Three major approaches:
  - method of lines,
  - Rothe's method,
  - space-time finite elements.
- All approaches yield the same result for classical uniform partitions.
- The method of lines is very inflexible and not suited for adaptivity.
- The analysis of Rothe's method is rather intricate since it requires differentiability properties w.r.t. time which often are not available.
- Space-time finite element methods allow a posteriori error estimates and are well suited for space-time adaptivity.

Method of Lines

- Choose a fixed partition $\mathcal{T}$ of Ω and an associated finite element space $X(\mathcal{T})$ (space discretization); denote by $A_{\mathcal{T}}$ and $f_{\mathcal{T}}$ the corresponding stiffness matrix and load vector.
- The space discretization then yields the following ODE system:
  $$\frac{du_{\mathcal{T}}}{dt} = f_{\mathcal{T}} - A_{\mathcal{T}}u_{\mathcal{T}}.$$
- Apply to this system a standard ODE solver (implicit Euler, Crank-Nicolson, Runge-Kutta, ...) (time discretization).
- The Crank-Nicolson method, e.g., yields
  $$\frac{u_{\mathcal{T}}^n - u_{\mathcal{T}}^{n-1}}{\tau} = \frac{1}{2}\big(f_{\mathcal{T}}^n - A_{\mathcal{T}}u_{\mathcal{T}}^n + f_{\mathcal{T}}^{n-1} - A_{\mathcal{T}}u_{\mathcal{T}}^{n-1}\big).$$

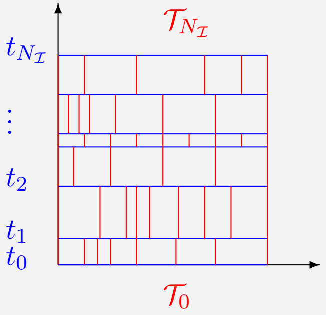
Rothe's Method

- Interpret the parabolic problem as an ODE in infinite dimension and apply to this a standard ODE solver (implicit Euler, Crank-Nicolson, Runge-Kutta, ...) (time discretization).
- Each time step then requires the solution of a stationary elliptic equation which is discretized using a standard finite element method (space discretization).
- The Crank-Nicolson method, e.g., yields the elliptic equations

$$\frac{u^n - u^{n-1}}{\tau} + \frac{1}{2}\big(-\operatorname{div}(A\nabla u^n) + \mathbf{a}\cdot\nabla u^n + \alpha u^n - \operatorname{div}(A\nabla u^{n-1}) + \mathbf{a}\cdot\nabla u^{n-1} + \alpha u^{n-1}\big) = \frac{1}{2}\big(f^n + f^{n-1}\big).$$

Space-Time Finite Elements

Meshes



- $\mathcal{I} = \{[t_{n-1}, t_n] : 1 \le n \le N_\mathcal{I}\}$: partition of $[0,T]$ with $0 = t_0 < \ldots < t_{N_\mathcal{I}} = T$ (The t_n are determined successively while advancing in time.)
- $\tau_n = t_n - t_{n-1}$
- $\mathcal{T}_n$: partitions of Ω
- $X_n = X(\mathcal{T}_n)$: associated finite element spaces

Space-Time Finite Elements

Discretization

Compute an approximation $u^0_{\mathcal{T}_0} \in X_0$ of u_0 and for $n = 1, 2, \ldots$ successively find $u^n_{\mathcal{T}_n} \in X_n$ (trial function) such that with $u^{n\theta} = \theta u^n_{\mathcal{T}_n} + (1-\theta)u^{n-1}_{\mathcal{T}_{n-1}}$ every $v^n_{\mathcal{T}_n} \in X_n$ (test function) satisfies:

$$\begin{aligned}
&\int_\Omega \frac{1}{\tau_n}(u^n_{\mathcal{T}_n} - u^{n-1}_{\mathcal{T}_{n-1}})v_{\mathcal{T}_n}\,dx + \int_\Omega \nabla u^{n\theta}\cdot A\nabla v_{\mathcal{T}_n}\,dx\\
&+ \int_\Omega \mathbf{a}\cdot\nabla u^{n\theta} v_{\mathcal{T}_n}\,dx + \int_\Omega \alpha u^{n\theta} v_{\mathcal{T}_n}\,dx\\
=& \int_\Omega f v_{\mathcal{T}_n}\,dx
\end{aligned}$$

Space-Time Finite Elements

Choice of θ

- $\theta = \frac{1}{2}$ corresponds to the Crank-Nicolson method.
- $\theta = 1$ corresponds to the implicit Euler scheme.
- $\theta = 0$ corresponds to the explicit Euler scheme.
- To ensure stability one should choose $\theta \ge \frac{1}{2}$.

Space-Time Finite Elements

Properties

- If $\theta > 0$, every time step requires the solution of a linear system of equations which corresponds to a finite element discretization of a stationary elliptic equation.
- If $\mathbf{a} \neq 0$, the stiffness matrix is not symmetric and indefinite.
- When using an iterative solver, $u_{\mathcal{T}_{n-1}}^{n-1}$ is a good initial guess for the computation of $u_{\mathcal{T}_n}^{n}$.
- The discretization error is of the order $h^2 + \tau^\gamma$ with $\gamma = 2$ for $\theta = \frac{1}{2}$ and $\gamma = 1$ for $\theta \neq \frac{1}{2}$ (h maximal mesh size in space, τ maximal mesh size in time).

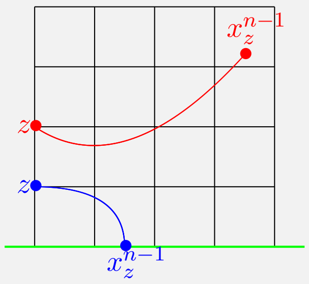
Method of Characteristics

Idea

- For every point $(x^*, t^*) \in \Omega \times (0, T]$ the following ODE (characteristic equation) admits a unique solution on the interval $(0, t^*)$
$$\frac{d}{dt} x(t; x^*, t^*) = \mathbf{a}(x(t; x^*, t^*), t), \quad x(t^*; x^*, t^*) = x^*.$$
- $U(x^*, t) = u(x(t; x^*, t^*), t)$ satisfies
$$\frac{dU}{dt} = \frac{\partial u}{\partial t} + \mathbf{a} \cdot \nabla u.$$
- Hence the differential equation can be put in the form
$$\frac{dU}{dt} - \operatorname{div}(A \nabla u) + \alpha u = f.$$
- The reaction-diffusion equation and the material derivative are discretized separately.

Method of Characteristics

Re-Interpolation



- Denote by $\mathcal{N}_n$ the degrees of freedom associated with X_n.
- For every n and every $z \in \mathcal{N}_n$ apply a classical ODE solver (implicit Euler, Crank-Nicolson, Runge-Kutta, ...) to the characteristic equation with $(x^*, t^*) = (z, t_n)$ and denote by x_z^{n-1} the resulting approximation for $x(t_{n-1}; z, t_n)$.

Method of Characteristics

Discrete Problem

- Find $\widetilde{u}_{\mathcal{T}_n}^{n-1} \in X_n$ such that $\widetilde{u}_{\mathcal{T}_n}^{n-1}(z) = u_{\mathcal{T}_{n-1}}^{n-1}(x_z^{n-1})$ holds for all $z \in \mathcal{N}_n$.
- Find $u_{\mathcal{T}_n}^n \in X_n$ such that every $v_{\mathcal{T}_n}^n \in X_n$ satisifes
$$\frac{1}{\tau_n} \int_\Omega u_{\mathcal{T}_n}^n v_{\mathcal{T}_n}^n \, dx + \int_\Omega \nabla u_{\mathcal{T}_n}^n \cdot A \nabla v_{\mathcal{T}_n}^n \, dx + \int_\Omega \alpha u_{\mathcal{T}_n}^n v_{\mathcal{T}_n}^n \, dx = \frac{1}{\tau_n} \int_\Omega \widetilde{u}_{\mathcal{T}_n}^{n-1} v_{\mathcal{T}_n}^n \, dx + \int_\Omega f v_{\mathcal{T}_n}^n \, dx$$

Method of Characteristics

Properties

- The method of characteristics, alias transport-diffusion algorithm, is well suited for the solution of parabolic equations with a large convection term.
- It decouples the discretization of the temporal and convective derivatives from the discretization of the other derivatives.
- It requires the solution of ODEs and of reaction-diffusion problems with a symmetric positive definite stiffness matrix.

A Residual Error Estimator

- $\mathcal{E}_n$: Edges ($d = 2$) or faces ($d = 3$) of elements in $\mathcal{T}_n$
- space indicator

$$\eta_h^n = \Big\{ \sum_{K\in\mathcal{T}_n} h_K^2 \int_K \Big| f(x,t_n) - \frac{1}{\tau_n}(u_{\mathcal{T}_n}^n - u_{\mathcal{T}_{n-1}}^{n-1}) + \operatorname{div}(A\nabla u_{\mathcal{T}_n}^n) - \mathbf{a}\cdot\nabla u_{\mathcal{T}_n}^n - \alpha u_{\mathcal{T}_n}^n \Big|^2 dx + \sum_{E\in\mathcal{E}_n} h_E \int_E \big| [\mathbf{n}_E \cdot A\nabla u_{\mathcal{T}_n}^n]_E \big|^2 dS \Big\}^{\frac{1}{2}}$$

- time indicator

$$\eta_\tau^n = \Big\{ \int_\Omega |\nabla u_{\mathcal{T}_n}^n - \nabla u_{\mathcal{T}_{n-1}}^{n-1}|^2 dx \Big\}^{\frac{1}{2}}$$

Remarks

- η_h^n consists of element residuals and inter-element jumps.
- The element residuals correspond to the strong form of the differential equation.
- The jump terms are the same as for the corresponding elliptic problem (time derivative suppressed).
- η_τ^n is a jump term w.r.t. time.

A Posteriori Error Estimates

- Denote by $u_{\mathcal{I}}$ the continuous piece-wise linear function w.r.t. time which coincides with $u_{\mathcal{T}_n}^n$ at time t_n.
- Then the error satisfies

$$\Big\{ \max_{0\le t\le T} \int_\Omega |u - u_{\mathcal{I}}|^2 dx + \int_0^T \int_\Omega |\nabla u - \nabla u_{\mathcal{I}}|^2 dx dt \Big\}^{\frac{1}{2}}$$
$$\approx \Big\{ \int_\Omega |u_{\mathcal{T}_0}^0 - u_0|^2 dx + \sum_{n=1}^{N_{\mathcal{I}}} \tau_n \big[(\eta_h^n)^2 + (\eta_\tau^n)^2 \big] \Big\}^{\frac{1}{2}}$$

Remarks

- $\approx$ denotes upper and lower bounds up to multiplicative constants.
- These constants depend on the polynomial degree and the shape parameters of the partitions.
- The upper bound is global w.r.t. space and time.
- The lower bound is global w.r.t. space and local w.r.t. time.
- η_h^n controls the spacial error.
- η_τ^n controls the temporal error.

Space-Time Adaptivity

0. Given: tolerance ε, partition $\mathcal{T}_0$, time step τ_1

1. Adapt $\mathcal{T}_0$ such that $\int_\Omega |u^0_{\mathcal{T}_0} - u_0|^2 dx \le \frac{1}{4}\varepsilon^2$.
Set $n = 1$, $t_1 = \tau_1$.

2. Solve the discrete problem for time t_n and compute the indicators η_h^n and η_τ^n.

3. If $\eta_\tau^n > \frac{\varepsilon}{2\sqrt{T}}$, replace t_n by $\frac{1}{2}(t_{n-1} + t_n)$ and return to step **2** (reducing τ_n).

4. Adapt $\mathcal{T}_n$ such that $\eta_h^n \le \frac{\varepsilon}{2\sqrt{T}}$.
If $\eta_\tau^n < \frac{\varepsilon}{4\sqrt{T}}$, double τ_n.

5. If $t_n = T$ ist, stopp.
Otherwise set $t_{n+1} = \min\{T, t_n + \tau_n\}$, augment n by 1 and return to step **2**.

Properties

- The algorithm yields a solution with
$$\left\{\int_\Omega |u^0_{\mathcal{T}_0} - u_0|^2 dx + \sum_{n=1}^{N_\mathcal{I}} \tau_n \left[\left(\eta_h^n\right)^2 + \left(\eta_\tau^n\right)^2\right]\right\}^{\frac{1}{2}} \le \varepsilon.$$
- When adapting $\mathcal{T}_n$ the quantities t_n, τ_n and η_τ^n are kept fixed.
- The adaptation of $\mathcal{T}_n$ eventually requires the repeated solution of discrete problems and the computation of η_h^n.

Finite Volume Methods

- Systems in divergence form
- Finite volume discretization
- Finite volume meshes
- Numerical fluxes
- Relation to finite element methods
- Discontinuous Galerkin methods

Systems in Divergence Form

- Domain: $\Omega \subset \mathbb{R}^d$
- Source: $\mathbf{g} : \mathbb{R}^m \times \Omega \times (0, \infty) \to \mathbb{R}^m$
- Mass: $\mathbf{M} : \mathbb{R}^m \to \mathbb{R}^m$
- Flux: $\underline{\mathbf{F}} : \mathbb{R}^m \to \mathbb{R}^{m \times d}$
- Initial value: $\mathbf{U}_0 : \Omega \to \mathbb{R}^m$
- Problem: Find $\mathbf{U} : \Omega \times (0, \infty) \to \mathbb{R}^m$ such that under suitable boundary conditions

$$\begin{aligned} \frac{\partial \mathbf{M}(\mathbf{U})}{\partial t} + \operatorname{div} \underline{\mathbf{F}}(\mathbf{U}) = \mathbf{g}(\mathbf{U}, x, t) \qquad & \text{in } \Omega \times (0, \infty) \\ \mathbf{U}(\cdot, 0) = \mathbf{U}_0 \qquad & \text{in } \Omega \end{aligned}$$

- $\operatorname{div} \underline{\mathbf{F}}(\mathbf{U}) = \left(\sum_{j=1}^{d} \frac{\partial \underline{\mathbf{F}}(\mathbf{U})_{i,j}}{\partial x_j} \right)_{1 \le i \le m}$

Advective and Viscous Fluxes

- The flux $\underline{\mathbf{F}}$ splits into two components:

  $\underline{\mathbf{F}} = \underline{\mathbf{F}}_{\text{adv}} + \underline{\mathbf{F}}_{\text{visc}}$.
- $\underline{\mathbf{F}}_{\text{adv}}$ is called advective flux and does not contain any derivatives.
- $\underline{\mathbf{F}}_{\text{visc}}$ is called viscous flux and contains spacial derivatives.
- The advective flux models transport or convection phenomena.
- The viscous flux models diffusion phenomena.

Examples

- Linear parabolic equations of 2nd order:
  - $\dfrac{\partial u}{\partial t} - \operatorname{div}(A \nabla u) + \mathbf{a} \cdot \nabla u + \alpha u = f$
  - $m = 1$
  - $\mathbf{U} = u$
  - $\mathbf{M}(\mathbf{U}) = u$
  - $\underline{\mathbf{F}}_{\text{adv}}(\mathbf{U}) = \mathbf{a} u$
  - $\underline{\mathbf{F}}_{\text{visc}}(\mathbf{U}) = -A \nabla u$
  - $\mathbf{g}(\mathbf{U}) = f - \alpha u + (\operatorname{div} \mathbf{a}) u$
- Euler equations
- Compressible Navier-Stokes equations
- Burger's equation

Finite Volume Discretization

First Step

- Choose a time step $\tau > 0$.
- Choose a partition $\mathcal{T}$ of Ω into arbitrary non-overlapping polyhedra.
- Fix $n \in \mathbb{N}^*$ and $K \in \mathcal{T}$.
- Integrate the system over $K \times [(n-1)\tau, n\tau]$:

$$\begin{aligned} & \int_{(n-1)\tau}^{n\tau} \int_K \frac{\partial \mathbf{M}(\mathbf{U})}{\partial t} dx dt + \int_{(n-1)\tau}^{n\tau} \int_K \operatorname{div} \underline{\mathbf{F}}(\mathbf{U}) dx dt \\ = & \int_{(n-1)\tau}^{n\tau} \int_K \mathbf{g}(\mathbf{U}, x, t) dx dt \end{aligned}$$

Finite Volume Discretization

Second Step

Apply integration by parts to the terms on the left-hand side:

$$\int_{(n-1)\tau}^{n\tau}\int_K \frac{\partial \mathbf{M}(\mathbf{U})}{\partial t}dxdt = \int_K \mathbf{M}(\mathbf{U}(x,n\tau))dx - \int_K \mathbf{M}(\mathbf{U}(x,(n-1)\tau))dx$$

$$\int_{(n-1)\tau}^{n\tau}\int_K \operatorname{div}\underline{\mathbf{F}}(\mathbf{U})dxdt = \int_{(n-1)\tau}^{n\tau}\int_{\partial K}\underline{\mathbf{F}}(\mathbf{U})\cdot\mathbf{n}_K dSdt$$

Finite Volume Discretization

Third Step

- Assume that $\mathbf{U}$ is piecewise constant w.r.t space and time.
- Denote by $\mathbf{U}_K^n$ and $\mathbf{U}_K^{n-1}$ the value of $\mathbf{U}$ on K at times $n\tau$ and $(n-1)\tau$:

$$\int_K \mathbf{M}(\mathbf{U}(x,n\tau))dx \approx |K|\mathbf{M}(\mathbf{U}_K^n)$$

$$\int_K \mathbf{M}(\mathbf{U}(x,(n-1)\tau))dx \approx |K|\mathbf{M}(\mathbf{U}_K^{n-1})$$

$$\int_{(n-1)\tau}^{n\tau}\int_{\partial K}\underline{\mathbf{F}}(\mathbf{U})\cdot\mathbf{n}_K dSdt \approx \tau\int_{\partial K}\underline{\mathbf{F}}(\mathbf{U}_K^{n-1})\cdot\mathbf{n}_K dS$$

$$\int_{(n-1)\tau}^{n\tau}\int_K \mathbf{g}(\mathbf{U},x,t)dxdt \approx \tau|K|\mathbf{g}(\mathbf{U}_K^{n-1},x_K,(n-1)\tau)$$

Finite Volume Discretization

Fourth Step

Approximate the boundary integral for the flux by a numerical flux:

$$\tau\int_{\partial K}\underline{\mathbf{F}}(\mathbf{U}_K^{n-1})\cdot\mathbf{n}_K dS \approx \tau\sum_{\substack{K'\in\mathcal{T}\\ \partial K\cap\partial K'\in\mathcal{E}}}|\partial K\cap\partial K'|\mathbf{F}_{\mathcal{T}}(\mathbf{U}_K^{n-1},\mathbf{U}_{K'}^{n-1})$$

Resulting Finite Volume Method

- For every element $K\in\mathcal{T}$ compute

$$\mathbf{U}_K^0 = \frac{1}{|K|}\int_K \mathbf{U}_0(x).$$

- For $n = 1, 2, \ldots$ successively compute for every element $K\in\mathcal{T}$

$$\mathbf{M}(\mathbf{U}_K^n) = \mathbf{M}(\mathbf{U}_K^{n-1}) - \tau\sum_{\substack{K'\in\mathcal{T}\\ \partial K\cap\partial K'\in\mathcal{E}}}\frac{|\partial K\cap\partial K'|}{|K|}\mathbf{F}_{\mathcal{T}}(\mathbf{U}_K^{n-1},\mathbf{U}_{K'}^{n-1}) + \tau\mathbf{g}(\mathbf{U}_K^{n-1},x_K,(n-1)\tau).$$

Possible Modifications

- The time step may be variable.
- The partition of Ω may differ from time step to time step.
- The approximation of $\mathbf{U}_K^n$ may not be constant.

Open Tasks

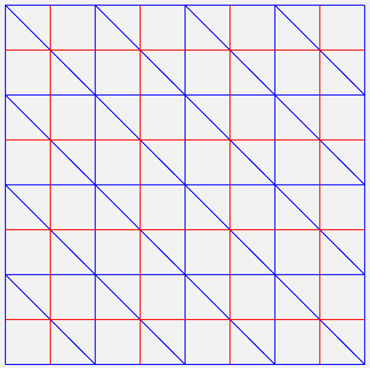
- Construct the partition $\mathcal{T}$.
- Construct the numerical flux $\underline{\mathbf{F}}_{\mathcal{T}}$.
- Take boundary conditions into account.

Construction of the Partition

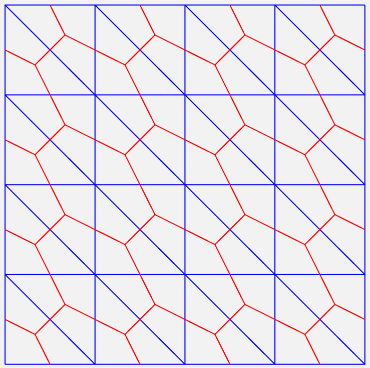
- Often the partition $\mathcal{T}$ is constructed as a dual mesh corresponding to an admissible primal finite element mesh $\widetilde{\mathcal{T}}$.
- In two space dimensions $(d = 2)$ there are two major approaches for the construction of dual meshes:
  - For every element $\widetilde{K} \in \widetilde{\mathcal{T}}$ draw the perpendicular bisectors.
  - Connect the barycentre of every element $\widetilde{K} \in \widetilde{\mathcal{T}}$ with the midpoints of its edges.

Perpendicular Bisectors and Barycentres

Perpendicular Bisectors

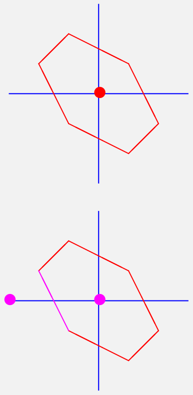


Barycentres



Properties of Dual Meshes

- Every element in $K \in \mathcal{T}$ corresponds to an element vertex x_K of $\widetilde{\mathcal{T}}$ and vice versa.
- For every edge E of $\mathcal{T}$ there are two element vertices $x_{E,1}$, $x_{E,2}$ of $\widetilde{\mathcal{T}}$ such that the line segment $\overline{x_{E,1}\, x_{E,2}}$ intersects E.



Advantages and Disadvantages of Perpendicular Bisectors

- The line segment $\overline{x_{E,1}\, x_{E,2}}$ and the edge E are perpendicular.
- The perpendicular bisectors of a triangle may intersect in a point outside of the triangle. The intersection of the perpendicular bisectors is inside the triangle, if and only if the triangle is acute.
- The perpendicular bisectors of a quadrilateral may not intersect at all. The perpendicular bisectors of a quadrilateral intersect in a common point, if and only if the quadrilateral is a rectangle.
- The construction with perpendicular bisectors is restricted to two space dimensions.

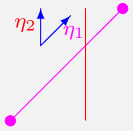
Construction of the Numerical Fluxes

Notations and Assumptions

- Assume that $\mathcal{T}$ is a dual mesh corresponding to a primal finite element mesh $\widetilde{\mathcal{T}}$.
- For every edge or face E of $\mathcal{T}$ denote by
  - K_1 and K_2 the adjacent volumes,
  - $\mathbf{U}_1$, $\mathbf{U}_2$ the values $\mathbf{U}_{K_1}^{n-1}$ and $\mathbf{U}_{K_2}^{n-1}$,
  - x_1, x_2 the element vertices in $\widetilde{\mathcal{T}}$ such that the line segment $\overline{x_1\, x_2}$ intersects E.
- Split the numerical flux $\underline{\mathbf{F}}_{\mathcal{T}}(\mathbf{U}_1, \mathbf{U}_2)$ into a viscous numerical flux $\underline{\mathbf{F}}_{\mathcal{T},\mathrm{visc}}(\mathbf{U}_1, \mathbf{U}_2)$ and an advective numerical flux $\underline{\mathbf{F}}_{\mathcal{T},\mathrm{adv}}(\mathbf{U}_1, \mathbf{U}_2)$.

Approximation of Viscous Fluxes

- Introduce a local coordinate system $\eta_1, \ldots, \eta_d$ such that η_1 is parallel to $\overline{x_1\, x_2}$ and such that the remaining coordinates are tangential to E.



- Express all derivatives in $\underline{\mathbf{F}}_{\mathrm{visc}}$ in terms of derivatives corresponding to the new coordinate system.
- Suppress all derivatives except those corresponding to η_1.
- Replace derivatives corresponding to η_1 by difference quotients of the form $\frac{\varphi_1 - \varphi_2}{|x_1 - x_2|}$.

Spectral Decomposition of Advective Fluxes

- Denote by $C(\mathbf{V}) = D(\underline{\mathbf{F}}_{\mathrm{adv}}(\mathbf{V}) \cdot \mathbf{n}_{K_1}) \in \mathbb{R}^{m\times m}$ the derivative of $\underline{\mathbf{F}}_{\mathrm{adv}}(\mathbf{V}) \cdot \mathbf{n}_{K_1}$ w.r.t. $\mathbf{V}$.
- Assume that this matrix can be diagonalized (Euler and Navier-Stokes equations fulfil this assumption.)

$$Q(\mathbf{V})^{-1} C(\mathbf{V}) Q(\mathbf{V}) = \Delta(\mathbf{V})$$

with an invertible matrix $Q(\mathbf{V}) \in \mathbb{R}^{m\times m}$ and a diagonal matrix $\Delta(\mathbf{V}) \in \mathbb{R}^{m\times m}$.
- Set $z^+ = \max\{z, 0\}$, $z^- = \min\{z, 0\}$ and

$$\begin{aligned}\Delta(\mathbf{V})^{\pm} &= \operatorname{diag}\big(\Delta(\mathbf{V})^{\pm}_{11}, \ldots, \Delta(\mathbf{V})^{\pm}_{mm}\big),\\ C(\mathbf{V})^{\pm} &= Q(\mathbf{V}) \Delta(\mathbf{V})^{\pm} Q(\mathbf{V})^{-1}.\end{aligned}$$

Approximation of Advective Fluxes

- Steger-Warming

$$\mathbf{F}_{\mathcal{T},\mathrm{adv}}(\mathbf{U}_1, \mathbf{U}_2) = C(\mathbf{U}_1)^+ \mathbf{U}_1 + C(\mathbf{U}_2)^- \mathbf{U}_2$$

- van Leer

$$\begin{aligned}&\mathbf{F}_{\mathcal{T},\mathrm{adv}}(\mathbf{U}_1, \mathbf{U}_2)\\ &= \Big[\frac{1}{2} C(\mathbf{U}_1) + C(\frac{1}{2}(\mathbf{U}_1 + \mathbf{U}_2))^+ - C(\frac{1}{2}(\mathbf{U}_1 + \mathbf{U}_2))^-\Big] \mathbf{U}_1\\ &\quad + \Big[\frac{1}{2} C(\mathbf{U}_2) - C(\frac{1}{2}(\mathbf{U}_1 + \mathbf{U}_2))^+ + C(\frac{1}{2}(\mathbf{U}_1 + \mathbf{U}_2))^-\Big] \mathbf{U}_2\end{aligned}$$

Properties

- Both approximations require the computation of $D\underline{\mathbf{F}}_{\mathrm{adv}}(\mathbf{V}) \cdot \mathbf{n}_{K_1}$ together with its eigenvalues and eigenvectors for suitable values of $\mathbf{V}$.
- The approach of van Leer usually is more costly than the one of Steger-Warming since it requires three evaluations of $C(\mathbf{V})$ instead of two.
- This extra cost can be avoided for the Euler and Navier-Stokes equations since these have the particular structure $\underline{\mathbf{F}}_{\mathrm{adv}}(\mathbf{V}) \cdot \mathbf{n}_{K_1} = C(\mathbf{V})\mathbf{V}$.

A One-Dimensional Example

- Burger's equation: $\dfrac{\partial u}{\partial t} + u \dfrac{\partial u}{\partial x} = 0$
- $\underline{\mathbf{F}}_{\mathrm{adv}}(u) = \frac{1}{2} u^2$, $C(u) = u$, $C(u)^{\pm} = u^{\pm}$
- Steger-Warming:

$$\underline{\mathbf{F}}_{\mathcal{T},\mathrm{adv}}(u_1, u_2) = \begin{cases} u_1^2 & \text{if } u_1 \ge 0, u_2 \ge 0\\ u_1^2 + u_2^2 & \text{if } u_1 \ge 0, u_2 \le 0\\ u_2^2 & \text{if } u_1 \le 0, u_2 \le 0\\ 0 & \text{if } u_1 \le 0, u_2 \ge 0 \end{cases}$$

- van Leer:

$$\underline{\mathbf{F}}_{\mathcal{T},\mathrm{adv}}(u_1, u_2) = \begin{cases} u_1^2 & \text{if } u_1 \ge -u_2\\ u_2^2 & \text{if } u_1 \le -u_2 \end{cases}$$

Relation to Finite Element Methods

- Suppose that $\mathcal{T}$ is a dual mesh corresponding to a primal finite element mesh $\widetilde{\mathcal{T}}$.
- Then there is a one-to-one correspondence between piecewise constant functions associated with $\mathcal{T}$ and continuous piecewise linear functions associated with $\widetilde{\mathcal{T}}$:

$$\begin{aligned} S^{0,-1}(\mathcal{T})^m \ni \mathbf{U}_{\mathcal{T}} &\leftrightarrow \widetilde{\mathbf{U}}_{\widetilde{\mathcal{T}}} \in S^{1,0}(\widetilde{\mathcal{T}})^m \\ \mathbf{U}_{\mathcal{T}}|_K &= \widetilde{\mathbf{U}}_{\widetilde{\mathcal{T}}}(x_K) \qquad \text{for all } K \in \mathcal{T}. \end{aligned}$$

A Simple Adaptive Algorithm

- Given a finite volume discretization and its solution $\mathbf{U}_{\mathcal{T}}$, compute the corresponding finite element function $\widetilde{\mathbf{U}}_{\widetilde{\mathcal{T}}}$.
- Apply a standard error estimator to $\widetilde{\mathbf{U}}_{\widetilde{\mathcal{T}}}$.
- Based on this error estimator apply a standard mesh adaptation process to $\widetilde{\mathcal{T}}$ and thus obtain a modified partition $\widehat{\mathcal{T}}$.
- Take $\widehat{\mathcal{T}}$ as primal mesh for the construction of a new dual mesh $\mathcal{T}'$. This is the refinement / coarsening of $\mathcal{T}$.

Idea of Discontinuous Galerkin Methods

- Approximate $\mathbf{U}$ by discontinuous functions which are polynomials w.r.t. space and time on small space-time cylinders of the form $K \times [(n-1)\tau, n\tau]$ with $K \in \mathcal{T}$.
- For every such cylinder multiply the differential equation by a corresponding test-polynomial and integrate the result over the cylinder.
- Use integration by parts for the flux term.
- Accumulate the contributions of all elements in $\mathcal{T}$.
- Compensate for the illegal partial integration by adding appropriate jump-terms across the element boundaries.
- Stabilize the scheme in a Petrov-Galerkin way by adding suitable element residuals.

A Simple Discontinuous Galerkin Scheme

- Compute $\mathbf{U}^0_{\mathcal{T}}$, the L^2-projection of $\mathbf{U}_0$ onto $S^{k,-1}(\mathcal{T})$.
- For $n \geq 1$ find $\mathbf{U}^n_{\mathcal{T}} \in S^{k,-1}(\mathcal{T})$ such that for all $\mathbf{V}_{\mathcal{T}}$

$$\begin{aligned} &\sum_{K\in\mathcal{T}} \frac{1}{\tau} \int_K M(\mathbf{U}^n_{\mathcal{T}}) \cdot \mathbf{V}_{\mathcal{T}} - \sum_{K\in\mathcal{T}} \int_K \underline{\mathbf{F}}(\mathbf{U}^n_{\mathcal{T}}) : \nabla \mathbf{V}_{\mathcal{T}} \\ &+ \sum_{E\in\mathcal{E}} \delta_E h_E \int_E \mathbb{J}_E(\mathbf{n}_E \cdot \underline{\mathbf{F}}(\mathbf{U}^n_{\mathcal{T}}) \mathbf{V}_{\mathcal{T}}) \\ &+ \sum_{K\in\mathcal{T}} \delta_K h_K^2 \int_K \operatorname{div} \underline{\mathbf{F}}(\mathbf{U}^n_{\mathcal{T}}) \cdot \operatorname{div} \underline{\mathbf{F}}(\mathbf{V}_{\mathcal{T}}) \\ = &\sum_{K\in\mathcal{T}} \frac{1}{\tau} \int_K M(\mathbf{U}^{n-1}_{\mathcal{T}}) \cdot \mathbf{V}_{\mathcal{T}} + \sum_{K\in\mathcal{T}} \int_K \mathbf{g}(\cdot, n\tau) \cdot \mathbf{V}_{\mathcal{T}} \\ &+ \sum_{K\in\mathcal{T}} \delta_K h_K^2 \int_K \mathbf{g}(\cdot, n\tau) \cdot \operatorname{div} \underline{\mathbf{F}}(\mathbf{V}_{\mathcal{T}}) \end{aligned}$$

Possible Modifications

- The jump and stabilization terms can be chosen more judiciously.
- The time-step may not be constant.
- The spatial mesh may depend on time.
- The functions $\mathbf{U}_\mathcal{T}$ and $\mathbf{V}_\mathcal{T}$ may be piece-wise polynomials of higher order w.r.t. to time. Then the term $\sum_{K\in\mathcal{T}} \int_{(n-1)\tau}^{n\tau} \int_K \frac{\partial M(\mathbf{U}_\mathcal{T})}{\partial t} \cdot \mathbf{V}_\mathcal{T}$ must be added on the left-hand side and terms of the form $\frac{\partial M(\mathbf{U}_\mathcal{T})}{\partial t} \cdot \mathbf{V}_\mathcal{T}$ must be added to the element residuals.

Implementation

- Mesh adaptation
- Data structures

Mesh Adaptation

- The general adaptive algorithm
- Mesh refinement
- Mesh coarsening
- Mesh smoothing

The General Adaptive Algorithm

0. Given: The data of a partial differential equation and a tolerance ε.
Sought: A numerical solution with an error less than ε.

1. Construct an initial coarse mesh $\mathcal{T}_0$ representing sufficiently well the geometry and data of the problem; set $k = 0$.

2. Solve the discrete problem on $\mathcal{T}_k$.

3. For every element K in $\mathcal{T}_k$ compute an a posteriori error indicator.

4. If the estimated global error is less than ε then stop. Otherwise decide which elements have to be refined or coarsened and construct the next mesh $\mathcal{T}_{k+1}$. Replace k by $k+1$ and return to step **2**.

Basic Ingredients

- An error indicator which furnishes the a posteriori error estimate.
- A refinement strategy which determines which elements have to be refined or coarsened and how this has to be done.

Basic Ingredients for Mesh Refinement

- The mesh refinement requires two key-ingredients:
  - a marking strategy that decides which elements should be refined,
  - refinement rules which determine the actual subdivision of a single element.
- To maintain the admissibility of the partitions, i.e. to avoid hanging nodes, the refinement process proceeds in two stages:
  - Firstly refine all those elements that are marked due to a too large value of η_K (regular refinement).
  - Secondly refine additional elements in order to eliminate the hanging nodes which are possibly created during the first stage (irregular refinement).
- The mesh refinement may possibly be combined with mesh coarsening and mesh smoothing.

Maximum Strategy for Marking

0. Given: A partition $\mathcal{T}$, error estimates η_K for the elements $K \in \mathcal{T}$, and a threshold $\theta \in (0,1)$.
Sought: A subset $\widetilde{\mathcal{T}}$ of marked elements that should be refined.

1. Compute $\eta_{\mathcal{T},\max} = \max_{K \in \mathcal{T}} \eta_K$.

2. If $\eta_K \geq \theta \eta_{\mathcal{T},\max}$ mark K by putting it into $\widetilde{\mathcal{T}}$.

Equilibration Strategy for Marking (Bulk Chasing or Dörfler Marking)

0. Given: A partition $\mathcal{T}$, error estimates η_K for the elements $K \in \mathcal{T}$, and a threshold $\theta \in (0,1)$.
Sought: A subset $\widetilde{\mathcal{T}}$ of marked elements that should be refined.

1. Compute $\Theta_{\mathcal{T}} = \sum_{K \in \mathcal{T}} \eta_K^2$. Set $\Sigma_{\mathcal{T}} = 0$ and $\widetilde{\mathcal{T}} = \emptyset$.

2. If $\Sigma_{\mathcal{T}} \geq \theta \Theta_{\mathcal{T}}$ return $\widetilde{\mathcal{T}}$; stop. Otherwise go to step **3**.

3. Compute $\widetilde{\eta}_{\mathcal{T},\max} = \max_{K \in \mathcal{T} \setminus \widetilde{\mathcal{T}}} \eta_K$.

4. For all elements $K \in \mathcal{T} \setminus \widetilde{\mathcal{T}}$ check whether $\eta_K = \widetilde{\eta}_{\mathcal{T},\max}$. If this is the case, mark K by putting it into $\widetilde{\mathcal{T}}$ and add η_K^2 to $\Sigma_{\mathcal{T}}$. Otherwise skip K. When all elements have been checked, return to step **2**.

Comparison of the Marking Strategies

- The maximum strategy is cheaper.
- At the end of the equilibration strategy the set $\widetilde{\mathcal{T}}$ satisfies

$$\sum_{K \in \widetilde{\mathcal{T}}} \eta_K^2 \geq \theta \sum_{K \in \mathcal{T}} \eta_K^2 .$$

- Convergence proofs for adaptive finite element methods are often based on this property.

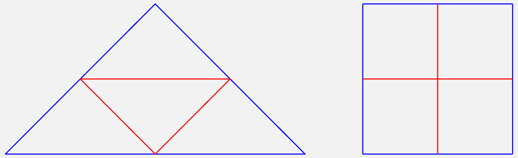
Ensuring a Sufficient Refinement

- Sometimes very few elements have an extremely large estimated error, whereas the remaining ones split into the vast majority with an extremely small estimated error and a third group of medium size consisting of elements with an estimated error of medium size.
- Then the marking strategies only refine the elements of the first group.
- This deteriorates the performance of the adaptive algorithm.
- This can be avoided he following modification:

  Given a small percentage ε, first mark the ε% elements with largest estimated error for refinement and then apply the marking strategies to the remaining elements.

Regular Refinement

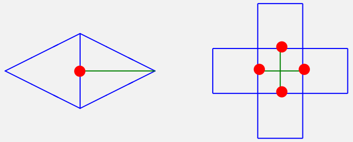
- Elements are subdivided by joining the midpoints of their edges.



- This preserves the shape parameter.

Hanging Nodes

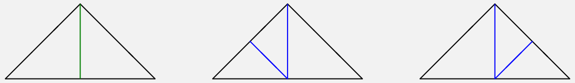
- Hanging nodes destroy the admissibility of the partition.



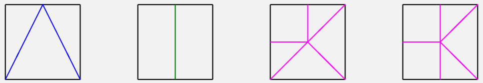
- Therefore
  - either the continuity of the finite element spaces must be enforced at hanging nodes
  - or an additional irregular refinement must be performed.
- Enforcing the continuity at hanging nodes may counteract the refinement.

Irregular Refinement

- Triangles

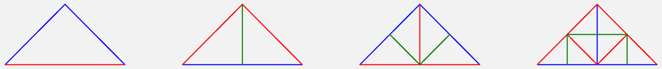


- Quadrilaterals



Marked Edge Bisection

- The first mesh is constructed such that the longest edge of an element is also the longest edge of its neighbour.
- The longest edges in the first mesh are marked.
- An element is refined by joining the midpoint of its marked edge with the vertex opposite to this edge (bisection).
- When bisecting the edge of an element, its two remaining edges become the marked edges of the resulting triangles.



Mesh Coarsening

- The coarsening of meshes is needed
  - to ensure the optimality of the adaptive process, i.e. to obtain a given accuracy with a minimal amount of unknowns,
  - to resolve moving singularities.
- The basic idea is to cluster elements with too small an error.
- This is achieved by
  - either going back in the grid hierarchy
  - or removing resolvable vertices.

Going Back in the Grid Hierarchy

0. Given: A hierarchy $\mathcal{T}_0, \ldots, \mathcal{T}_k$ of adaptively refined partitions, error indicators η_K for the elements K of $\mathcal{T}_k$, and parameters $1 \le m \le k$ and $n > m$.
Sought: A new partition $\mathcal{T}_{k-m+n}$.

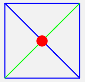
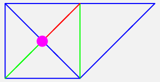
1. For every element $K \in \mathcal{T}_{k-m}$ set $\widetilde{\eta}_K = 0$.

2. For every element $K \in \mathcal{T}_k$ determine its ancestor $K' \in \mathcal{T}_{k-m}$ and add η_K^2 to $\widetilde{\eta}_{K'}^2$.

3. Successively apply the maximum or equilibration strategy n times with $\widetilde{\eta}$ as error indicator. In this process, equally distribute $\widetilde{\eta}_K$ over the descendants of K once an element K is subdivided.

Resolvable Vertices

- An element K of the current partition $\mathcal{T}$ has refinement level ℓ if it is obtained by subdividing ℓ times an element of the coarsest partition.
- Given a triangle K of the current partition $\mathcal{T}$ which is obtained by bisecting a parent triangle K', the vertex of K which is not a vertex of K' is called the refinement vertex of K.
- A vertex $z \in \mathcal{N}$ of the current partition $\mathcal{T}$ and the corresponding patch ω_z are called resolvable if
  - z is the refinement vertex of all elements contained in ω_z,
  - all elements contained in ω_z have the same refinement level.

resovable vertex  non-resovable vertex 

Removing Resolvable Vertices

0. Given: A partition $\mathcal{T}$, error indicators η_K for all elements K of $\mathcal{T}$, and parameters $0 < \theta_1 < \theta_2 < 1$.
Sought: Subsets $\mathcal{T}_c$ and $\mathcal{T}_r$ of elements that should be coarsened and refined, respectively.

1. Set $\mathcal{T}_c = \emptyset$, $\mathcal{T}_r = \emptyset$ and compute $\eta_{\mathcal{T},\max} = \max_{K \in \mathcal{T}} \eta_K$.

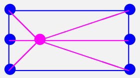
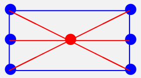
2. For all $K \in \mathcal{T}$ check whether $\eta_K \geq \theta_2 \eta_{\mathcal{T},\max}$. If this is the case, put K into $\mathcal{T}_r$.

3. For all vertices $z \in \mathcal{N}$ check whether z is resolvable. If this is the case and if $\max_{K \subset \omega_z} \eta_K \leq \theta_1 \eta_{\mathcal{T},\max}$, put all elements contained in ω_z into $\mathcal{T}_c$.

Mesh Smoothing

- Improve the quality of a given partition $\mathcal{T}$ by moving its vertices while retaining the adjacency of the elements.
- The quality is measured by a a quality function q such that a larger value of q indicates a better quality.
- The quality is improved by sweeping through the vertices with a Gauß-Seidel type smoothing procedure:

  For every vertex z in $\mathcal{T}$, fix the vertices of $\partial\omega_z$ and find a new vertex $\widetilde{z}$ inside ω_z such that $\min_{\widetilde{z} \in \widetilde{K}} q(\widetilde{K}) > \min_{z \in K} q(K)$.

Quality Functions

- Based on geometric criteria:

$$q(K) = \frac{4\sqrt{3}\mu_2(K)}{\mu_1(E_0)^2 + \mu_1(E_1)^2 + \mu_1(E_2)^2}$$

- Based on interpolation:

$$q(K) = \|\nabla(u_Q - u_L)\|_K^2$$

  with linear and quadratic interpolants of u
- Based on an error indicator:

$$q(K) = \int_K \Big| \sum_{i=0}^{2} e_i \nabla \psi_{E_i} \Big|^2$$

  with $e_i = h_{E_i}^{\frac{1}{2}} \mathbb{J}_{E_i}(\mathbf{n}_{E_i} \cdot \nabla u_{\mathcal{T}})$

Classes of ALF

- NODE
- ELEMENT
- LEVEL

NODE

- NODE realizes the concept of a node, i.e. of a vertex of a grid.
- It has three members c, t, and d.
- c stores the co-ordinates in Euclidean 2-space.
- t stores the type of the node and equals:
  - 0, if the node is an interior point of the computational domain,
  - k, $k > 0$, if the node belongs to the k-th component of the Dirichlet boundary,
  - $-k$, $k > 0$, if the node is belongs to the k-th component of the Neumann boundary.
- d gives the address of the corresponding degree of freedom and equals -1 if the corresponding node is not a degree of freedom.

ELEMENT

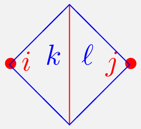
- ELEMENT realizes the concept of an element.
- It has six members nv, v, e, p, c, and t.
- nv determines the element type, i.e. triangle or quadrilateral.
- v realizes the vertex informations.
- e provides the edge informations.
- p gives the number of the parent element.
- c provides the number of the first child, the remaining children being enumerated consecutively.
- t determines the refinement type.

ELEMENT.v

- The vertices are enumerated consecutively in counter-clockwise order.
- v$[i]$ gives the global number of the i-th vertex of the element.
- v$[3] = -1$ if nv$= 3$, i.e., if the element is a triangle.

ELEMENT.e

- The edges are enumerated consecutively in counter-clockwise order such that the i-th edge has the vertices $i+1 \bmod \texttt{nv}$ and $i+2 \bmod \texttt{nv}$ as its endpoints.
- $\texttt{e}[i] = -1$ indicates that the corresponding edge is on a straight part of the boundary.
- $\texttt{e}[i] = -k-2$, $k \geq 0$, indicates that the endpoints of the corresponding edge are on the k-th curved part of the boundary.
- $\texttt{e}[i] = \ell \geq 0$ indicates that edge i of the current element is adjacent to element number ℓ.



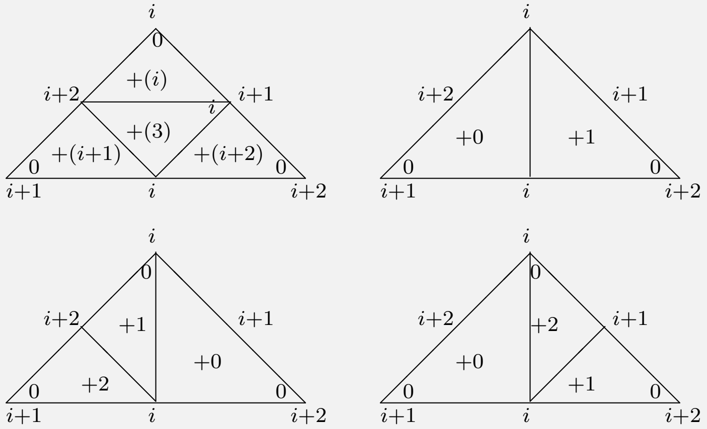
$k.\texttt{e}[i] = \ell$, $\ell.\texttt{e}[j] = k$

ELEMENT.t

- $\texttt{t}$ describes which edges are bisected.
- It is in
  - $\{0\}$, if the element is not refined,
  - $\{1,\dots,4\}$, if the element is refined green,
  - $\{5\}$, if the element is refined red,
  - $\{6,\dots,24\}$, if the element is refined blue,
  - $\{25,\dots,100\}$, if the element is refined purple.

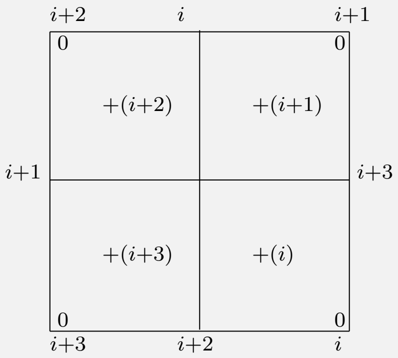
Enumeration of Descendants

Triangles



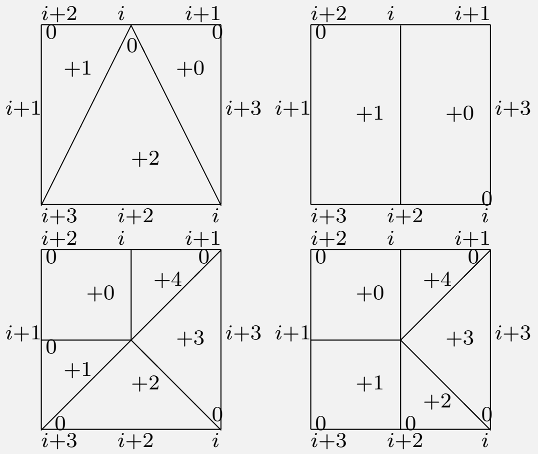
Enumeration of Descendants

Regular Quadrilaterals



Enumeration of Descendants

Irregular Quadrilaterals



Reasons for Keeping the Node and Edge Informations Separately

- It minimizes the storage requirement.
- The co-ordinates of a node must be stored only once.
- If nodes and elements are represented by a common structure, these co-ordinates are stored $4 - 6$ times.
- The elements represent the topology of the grid which is independent of the particular position of the nodes.
- If nodes and elements are represented by different structures it is much easier to implement mesh smoothing algorithms which affect the position of the nodes but do not change the mesh topology.

Grid Hierarchy

- Nodes are completely hierarchical, i.e. a node of grid $\mathcal{T}_i$ is also a node of any grid $\mathcal{T}_j$ with $j > i$.
- Since in general the grids are only partly refined, the elements are not completely hierarchical.
- Therefore, all elements of all grids are stored.
- The corresponding information is stored in the class `LEVEL`.

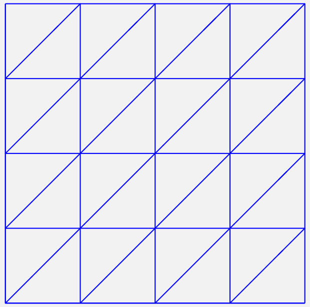
`LEVEL`

- `LEVEL` has seven members `nn`, `nt`, `nq`, `ne`, `first`, `last` and `dof`.
- `nn` gives the number of nodes.
- `nt` is the number of triangles.
- `nq` gives the number of quadrilaterals.
- `ne` is the number of edges.
- `first` provides the address of the first element of the current grid.
- `last` gives the address of the first element of the next grid.
- `dof` yields the number of degrees of freedom of the corresponding discrete finite element problem.

Solution of Discrete Problems

- Properties of direct and iterative solvers
- Classical iterative solvers
- Conjugate Gradient methods
- Multigrid methods
- Non-linear and indefinite problems

A Typical Model Problem



- Poisson equation $-\Delta u = f$ in Ω, $u = 0$ on Γ
- $\Omega = (0,1)^2$
- Courant triangulation consisting of $2n^2$ isosceles right-angled triangles with short sides of length $h = n^{-1}$
- Linear finite elements
- Number N of unknowns is of order $n^2 = h^{-2}$.

Properties of the Stiffness Matrix

- It is symmetric positive definite.
- It has 5 non-zero elements per row.
- It has bandwidth $h^{-1} \approx N^{\frac{1}{2}}$.
- Gaussian elimination requires N^2 operations.
- A matrix-vector multiplication requires $5N$ operations.
- Its smallest eigenvalue is of order 1.
- Its largest eigenvalue is of order $h^{-2} \approx N$.

Typical Properties of Direct Solvers

- They require $O(N^{2-\frac{1}{d}})$ storage for a discrete problem with N unknowns in d space dimensions.
- They require $O(N^{3-\frac{2}{d}})$ operations.
- They yield the exact solution of the discrete problem up to rounding errors.
- They yield an approximation for the differential equation with an $O(h^\alpha) = O(N^{-\frac{\alpha}{d}})$ error (typically: $\alpha \in \{1,2\}$).

Typical Properties of Classical Iterative Solvers

- They require $O(N)$ storage.
- They require $O(N)$ operations per iteration.
- Their convergence rate deteriorates with an increasing condition number of the discrete problem which usually is $O(h^{-2}) = O(N^{\frac{2}{d}})$.
- In order to reduce an initial error by a factor 0.1 one usually needs the following numbers of operations:
  - $O(N^{1+\frac{2}{d}})$ with the Gauß-Seidel algorithm,
  - $O(N^{1+\frac{1}{d}})$ with the conjugate gradient (CG-) algorithm,
  - $O(N^{1+\frac{1}{2d}})$ with the CG-algorithm with Gauß-Seidel preconditioning.

Comparison of Solvers

Arithmetic Operations

Example: Linear finite elements on a Courant triangulierung for the Poisson equation in the unit square; initial error is reduced by the factor 0.05

h	Gaussian el.	GS	CG	PCG	MG
$\frac{1}{16}$	$7.6 \cdot 10^5$	$2.6 \cdot 10^5$	$2.7 \cdot 10^4$	$1.6 \cdot 10^4$	$1.2 \cdot 10^4$
$\frac{1}{32}$	$2.8 \cdot 10^7$	$4.5 \cdot 10^6$	$2.2 \cdot 10^5$	$8.6 \cdot 10^4$	$4.9 \cdot 10^4$
$\frac{1}{64}$	$9.9 \cdot 10^8$	$7.6 \cdot 10^7$	$1.9 \cdot 10^6$	$5.0 \cdot 10^5$	$2.1 \cdot 10^5$
$\frac{1}{128}$	$3.3 \cdot 10^{10}$	$1.2 \cdot 10^9$	$1.5 \cdot 10^7$	$3.2 \cdot 10^6$	$8.4 \cdot 10^5$

Comparison of Solvers

Iterations

Example: Linear finite elements on a Courant triangulierung for the Poisson equation in the unit square; initial error is reduced by the factor 0.05

h	GS	CG	PCG	MG
$\frac{1}{16}$	236	12	4	1
$\frac{1}{32}$	954	23	5	2
$\frac{1}{64}$	3820	47	7	2
$\frac{1}{128}$	15287	94	11	1

Comparison of Solvers

Iterations and Convergence Rates

Example: Adaptively refined linear finite element discretization of a reaction-diffusion equation in the unit square with an interior layer; initial error is reduced by the factor 0.05

	CG		PCG		MG	
DOF	It.	κ	It.	κ	It.	κ
9	4	0.10	3	0.2	4	0.3
47	10	0.60	7	0.5	3	0.3
185	24	0.80	12	0.7	5	0.2
749	49	0.90	21	0.8	5	0.4
2615	94	0.95	37	0.9	6	0.4
5247	130	0.96	55	0.9	5	0.4

Conclusion

- Direct solvers need too much storage and and computer time.
- It suffices to compute an approximate solution of the discrete problem which, compared to the solution of the differential equation, has an error similar in size to the one of the exact solution of the discrete problem.
- Iterative solvers are superior if one arrives at improving their convergence rate and at finding good initial guesses.

Nested Grids

- Often one has to solve a sequence of discrete problems $L_k u_k = f_k$ corresponding to increasingly more accurate discretizations.
- Usually there is a natural interpolation operator $I_{k-1,k}$ which maps functions associated with the $(k-1)$-st discrete problem into those corresponding to the k-th discrete problem.
- Then the interpolate of any reasonable approximate solution of the $(k-1)$-st discrete problem is a good initial guess for any iterative solver applied to the k-th discrete problem.
- Often it suffices to reduce the initial error by a factor 0.1.

Nested Iteration

- Compute
$$\widetilde{u}_0 = u_0 = L_0^{-1} f_0.$$
- For $k = 1, \ldots$ compute an approximate solution $\widetilde{u}_k$ for $u_k = L_k^{-1} f_k$ by applying m_k iterations of an iterative solver for the problem
$$L_k u_k = f_k$$
with starting value $I_{k-1,k}\widetilde{u}_{k-1}$.
- m_k is implicitly determined by the stopping criterion
$$\|f_k - L_k \widetilde{u}_k\| \le \varepsilon \|f_k - L_k(I_{k-1,k}\widetilde{u}_{k-1})\|.$$

The Setting

- We have to solve a linear system $Lu = f$ with N unknowns.
- L is symmetric positive definite.
- κ denotes the condition number of L, i.e. the ratio of the largest over the smallest eigenvalue of L.
- $\kappa \approx N^{\frac{2}{d}}$

Idea of the Gradient Algorithm

- The solution of $Lu = f$ is equivalent to the minimization of the quadratic functional $J(u) = \frac{1}{2} u \cdot (Lu) - f \cdot u$.
- The negative gradient $-\nabla J(v) = f - Lv$ of J at v gives the direction of the steepest descent.
- Given an approximation v and a search direction $d \neq 0$, J attains its minimum on the line $t \mapsto v + td$ at the point $t^* = \frac{d \cdot (f - Lv)}{d \cdot (Ld)}$.

Gradient Algorithm

- Iteration step: Given the actual iterate u
  - compute the residual $r = f - Lu$,
  - replace u by $u + \frac{r \cdot r}{r \cdot Lr} r$.
- The gradient algorithm corresponds to a Richardson iteration with an automatic and optimal choice of the relaxation parameter.
- The convergence rate is $\frac{\kappa - 1}{\kappa + 1} \approx 1 - N^{-\frac{2}{d}}$.

Idea of the CG-Algorithm

- The gradient algorithm slows down since the search directions become nearly parallel.
- The algorithm speeds up when choosing the successive search directions L-orthogonal, i.e. $d_i \cdot (Ld_{i-1}) = 0$.
- L-orthogonal search directions can be computed during the algorithm by a suitable three-term recursion.

The CG-Algorithm

0. Given: an initial guess u_0 for the solution, and a tolerance $\varepsilon > 0$.

1. Compute $r_0 = f - Lu_0$, $d_0 = r_0$, $\gamma_0 = r_0 \cdot r_0$. Set $i = 0$.

2. If $\gamma_i < \varepsilon^2$ return u_i as approximate solution; stop. Otherwise go to step **3**.

3. Compute $s_i = Ld_i$, $\alpha_i = \frac{\gamma_i}{d_i \cdot s_i}$, $u_{i+1} = u_i + \alpha_i d_i$, $r_{i+1} = r_i - \alpha_i s_i$, $\gamma_{i+1} = r_{i+1} \cdot r_{i+1}$, $\beta_i = \frac{\gamma_{i+1}}{\gamma_i}$, $d_{i+1} = r_{i+1} + \beta_i d_i$. Increase i by 1 and go to step **2**.

Properties

- The CG-algorithm only requires matrix-vector multiplications and inner products.
- The convergence rate is $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \approx 1 - N^{-\frac{1}{d}}$.
- The CG-algorithm can only be applied to symmetric positive definite matrices, it breaks-down for non-symmetric or indefinite matrices.

The Idea of Pre-Conditioning

- Instead of the original system $Lu = f$ solve the equivalent system $\widehat{L}\widehat{u} = \widehat{f}$ with $\widehat{L} = H^{-1}LH^{-t}$, $\widehat{f} = H^{-1}f$, $\widehat{u} = H^t u$ and an invertible square matrix H.
- Choose the matrix H such that:
  - The condition number of $\widehat{L}$ is much smaller than the one of L.
  - Systems of the form $Cv = d$ with $C = HH^t$ are much easier to solve than the original system $Lu = f$.
- Apply the conjugate gradient algorithm to the new system $\widehat{L}\widehat{u} = \widehat{f}$ and express everything in terms of the original quantities L, f, and u.

The PCG-Algorithm

0. Given: an initial guess u_0 for the solution, and a tolerance $\varepsilon > 0$.

1. Compute $r_0 = f - Lu_0$, solve $Cz_0 = r_0$ and compute $d_0 = z_0$, $\gamma_0 = r_0 \cdot z_0$. Set $i = 0$.

2. If $\gamma_i < \varepsilon^2$ return u_i as approximate solution; stop. Otherwise go to step **3**.

3. Compute $s_i = Ld_i$, $\alpha_i = \frac{\gamma_i}{d_i \cdot s_i}$, $u_{i+1} = u_i + \alpha_i d_i$, $r_{i+1} = r_i - \alpha_i s_i$, solve $Cz_{i+1} = r_{i+1}$ and compute $\gamma_{i+1} = r_{i+1} \cdot z_{i+1}$, $\beta_i = \frac{\gamma_{i+1}}{\gamma_i}$, $d_{i+1} = z_{i+1} + \beta_i d_i$. Increase i by 1 and go to step **2**.

Properties

- The convergence rate of the PCG-algorithm is $\frac{\sqrt{\widehat{\kappa}}-1}{\sqrt{\widehat{\kappa}}+1}$ where $\widehat{\kappa}$ is the condition number of $\widehat{L}$.
- Good choices of C, e.g. SSOR-preconditioning, yield $\widehat{\kappa} = N^{\frac{1}{d}}$ and corresponding convergence rates of $1 - N^{-\frac{1}{2d}}$.

SSOR-Preconditioning

0. Given: r and a relaxation parameter $\omega \in (0, 2)$.
Sought: $z = C^{-1} r$.

1. Set $z = 0$.

2. For $i = 1, \ldots, N$ compute

$$z_i = z_i + \omega L_{ii}^{-1} \{ r_i - \sum_{j=1}^{N} L_{ij} z_j \}.$$

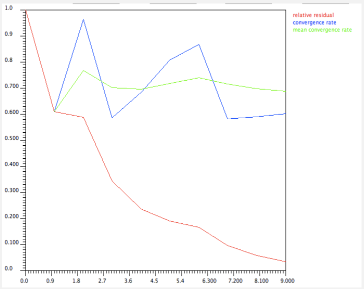
3. For $i = N, \ldots, 1$ compute

$$z_i = z_i + \omega L_{ii}^{-1} \{ r_i - \sum_{j=1}^{N} L_{ij} z_j \}.$$

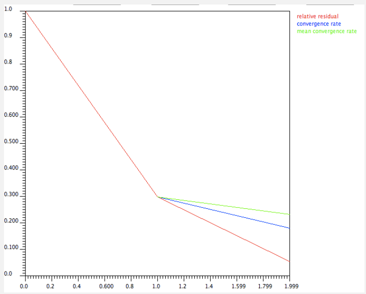
Comparison of CG and PCG Algorithms

Poisson equation on the unit square,
linear finite elements on Courant triangulation with $h = \frac{1}{64}$

CG
convergence rate 0.712



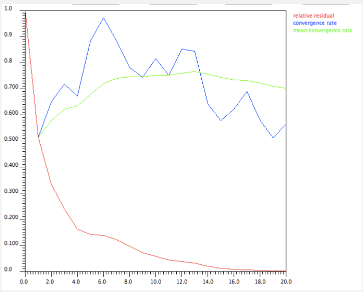
SSOR-PCG
convergence rate 0.376



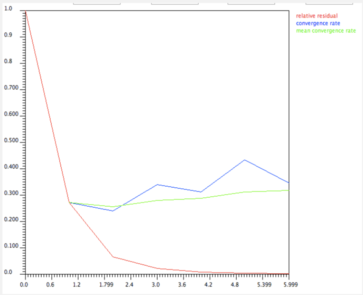
Comparison of CG and PCG Algorithms

Poisson equation on the unit square,
linear finite elements on Courant triangulation with $h = \frac{1}{128}$

CG
convergence rate 0.723



SSOR-PCG
convergence rate 0.377



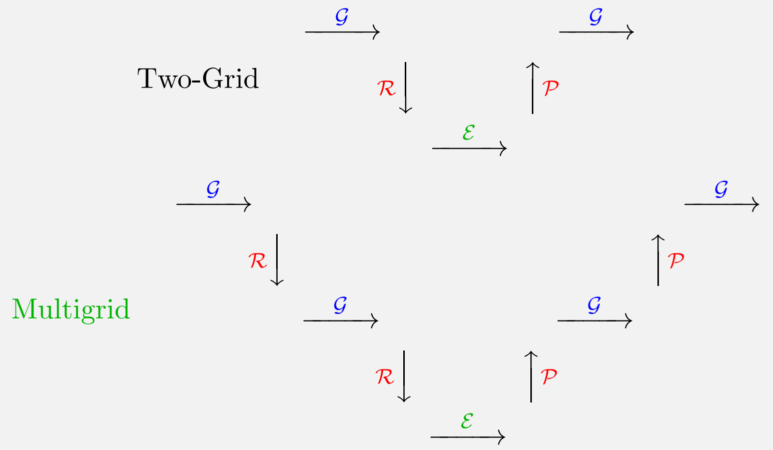
The Basic Idea of the Multigrid Algorithm

- Classical iterative methods such as the Gauß-Seidel algorithm quickly reduce highly oscillatory error components.
- Classical iterative methods such as the Gauß-Seidel algorithm are very poor in reducing slowly oscillatory error components.
- Slowly oscillating error components can well be resolved on coarser meshes with fewer unknowns.

The Basic Two-Grid Algorithm

- Perform several steps of a classical iterative method on the current grid.
- Correct the current approximation as follows:
  - Compute the current residual.
  - Restrict the residual to the next coarser grid.
  - Exactly solve the resulting problem on the coarse grid.
  - Prolongate the coarse-grid solution to the next finer grid.
- Perform several steps of a classical iterative method on the current grid.

Schematic Form



Basic Ingredients

- A sequence $\mathcal{T}_k$ of increasingly refined meshes with associated discrete problems $L_k u_k = f_k$.
- A smoothing operator M_k, which should be easy to evaluate and which at the same time should give a reasonable approximation to L_k^{-1}.
- A restriction operator $R_{k,k-1}$, which maps functions on a fine mesh $\mathcal{T}_k$ to the next coarser mesh $\mathcal{T}_{k-1}$.
- A prolongation operator $I_{k-1,k}$, which maps functions from a coarse mesh $\mathcal{T}_{k-1}$ to the next finer mesh $\mathcal{T}_k$.

The Multigrid Algorithm

0. Given: the actual level k, parameters μ, ν_1, and ν_2, the matrix L_k, the right-hand side f_k, an inital guess u_k.
Sought: improved approximate solution u_k.

1. If $k = 0$ compute $u_0 = L_0^{-1} f_0$; stop.

2. (Pre-smoothing) Perform ν_1 steps of the iterative procedure $u_k = u_k + M_k(f_k - L_k u_k)$.

3. (Coarse grid correction)

3.1 Compute $f_{k-1} = R_{k,k-1}(f_k - L_k u_k)$ and set $u_{k-1} = 0$.

3.2 Perform μ iterations of the MG-algorithm with parameters $k-1$, μ, ν_1, ν_2, L_{k-1}, f_{k-1}, u_{k-1} and denote the result by u_{k-1}.

3.3 Compute $u_k = u_k + I_{k-1,k} u_{k-1}$.

4. (Post-smoothing) Perform ν_2 steps of the iterative procedure $u_k = u_k + M_k(f_k - L_k u_k)$.

Typical Choices of Parameters

- $\mu = 1$ V-cycle or

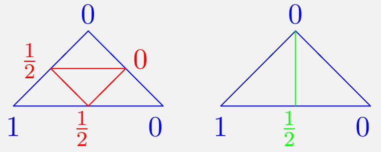
  $\mu = 2$ W-cycle
- $\nu_1 = \nu_2 = \nu$ or

  $\nu_1 = \nu$, $\nu_2 = 0$ or

  $\nu_1 = 0$, $\nu_2 = \nu$
- $1 \leq \nu \leq 4$.

Prolongation and Restriction

- The prolongation is typically determined by the natural inclusion of the finite element spaces, i.e. a finite element function corresponding to a coarse mesh is expressed in terms of the finite element basis functions corresponding to the fine mesh.



- The restriction is typically determined by inserting finite element basis functions corresponding to the coarse mesh in the variational form of the discrete problem corresponding to the fine mesh.

Smoothing

- Gauß-Seidel iteration
- SSOR iteration:
  - Perform a forward Gauß-Seidel sweep with over-relaxation as pre-smoothing.
  - Perform a backward Gauß-Seidel sweep with over-relaxation as post-smoothing.
- ILU smoothing:
  - Perform an incomplete lower upper decomposition of L_k by suppressing all fill-in.
  - The result is an approximate decomposition $\mathcal{L}_k\mathcal{U}_k \approx L_k$.
  - Compute $v_k = M_k u_k$ by solving the system $\mathcal{L}_k\mathcal{U}_k v_k = u_k$.

Number of Operations

- Assume that
  - one smoothing step requires $O(N_k)$ operations,
  - the prolongation requires $O(N_k)$ operations,
  - the restriction requires $O(N_k)$ operations,
  - $\mu \leq 2$,
  - $N_k > \mu N_{k-1}$,
- then one iteration of the multigrid algorithm requires $O(N_k)$ operations.

Convergence Rate

- The convergence rate is uniformly less than 1 for all meshes.
- The convergence rate is bounded by $\frac{c}{c+\nu_1+\nu_2}$ with a constant which only depends on the shape parameter of the meshes.
- Numerical experiments yield convergence rates less than 0.1.

CG Algorithm for Non-Symmetric or Indefinite Problems

- The CG algorithm typically breaks down when applied to non-symmetric or indefinite problems (stiffness matrix has eigenvalues with positive as well as negative real part).
- A naive solution is to apply the CG algorithm to the symmetric positive definite system of normal equations $L^T Lu = L^T f$.
- This doubles the number of iterations since the passage to the normal equations squares the condition number.
- A preferable solution are specialised variants of the CG algorithm such as the stabilised bi-conjugate gradient algorithm (Bi-CG-Stab algorithm).

Bi-CG-Stab Algorithm

0. Given: an initial guess u_0 and a tolerance $\varepsilon > 0$.

1. Compute $r_0 = b - Lu_0$ and set $\overline{r}_0 = r_0$, $v_{-1} = 0$, $p_{-1} = 0$, $\alpha_{-1} = 1$, $\rho_{-1} = 1$, $\omega_{-1} = 1$, and $i = 0$.

2. If $r_i \cdot r_i < \varepsilon^2$ return u_i as approximate solution; stop. Otherwise go to step **3**.

3. Compute $\rho_i = \overline{r}_i \cdot r_i$, $\beta_{i-1} = \frac{\rho_i \alpha_{i-1}}{\rho_{i-1}\omega_{i-1}}$. If $|\beta_{i-1}| < \varepsilon$ there may be a break-down; stop. Otherwise compute $p_i = r_i + \beta_{i-1}\{p_{i-1} - \omega_{i-1}v_{i-1}\}$, $v_i = Lp_i$, $\alpha_i = \frac{\rho_i}{\overline{r}_0 \cdot v_i}$. If $|\alpha_i| < \varepsilon$ there may be a break-down; stop. Otherwise compute $s_i = r_i - \alpha_i v_i$, $t_i = Ls_i$, $\omega_i = \frac{t_i \cdot s_i}{t_i \cdot t_i}$, $u_{i+1} = u_i + \alpha_i p_i + \omega_i s_i$, $r_{i+1} = s_i - \omega_i t_i$. Augment i by 1 and go to step **2**.

Properties

- The Bi-CG-Stab algorithm aims at a simultaneous solution of the original problem $Lu = f$ as well of the adjoint problem $L^T v = f$.
- The algorithm only needs the stiffness matrix L of the original problem.
- It only requires inner products and matrix vector multiplications.
- The Bi-CG-Stab algorithm may be preconditioned; possible methods for preconditioning are the SSOR iteration or the ILU decomposition applied to the symmetric part $\frac{1}{2}(L + L^T)$ of L.

Multigrid Algorithms for Non-Symmetric or Indefinite Problems

- Multigrid algorithms can directly be applied to non-symmetric or indefinite problems.
- Possibly one as to resort to a specialised smoother.
- The Richardson iteration applied to the normal equations is a robust smoother which however yields convergence rates of about 0.8.
- The ILU decomposition is a robust smoother too, but more costly and yields convergence rates of about 0.5.

Non-Linear Problems

- Non-linear problems are typically solved with a (damped) Newton iteration.
- Every step of the Newton iteration requires the solution of a linear problem.
- The latter can be solved with an iterative solver; the solution of the previous Newton step then is a good initial guess for the inner iteration.
- In multigrid methods one may reverse the roles of inner and outer iteration; then a few steps of the Newton method with a moderately accurate solution of the linear problems act as a smoother.

References

D. Braess
Finite Elements. Theory, Fast Solvers, and Applications in Solid Mechanics
Cambridge University Press, 2001

M. Ainsworth and J. T. Oden
A Posteriori Error Estimation in Finite Element Analysis
Wiley, 2000

R. Verfürth
A Posteriori Error Estimation Techniques for Finite Element Methods
Oxford University Press, 2013