Advanced Computational Techniques

Part 1
Stationary problems

Krzysztof Banaś

Department of Applied Computer Science and Modelling
AGH University of Science and Technology, Kraków, Poland

Notation and notational conventions

- $\Omega \subset \boldsymbol{R}^N$, *N=1,2 or 3* – domain in 1D, 2D or 3D space
  - all domains are assumed to have smooth boundaries $\partial\Omega$
- standard font - scalar, bold - vector
  - $\boldsymbol{x} = [x, y, z] = [x_1, x_2, x_3]$ - point in 3D space
  - t - time instant
  - $f(\mathbf{x}, t)[\mathbf{f}(\mathbf{x}, t)]$ – scalar [vector] function of space and time
    - the function will usually denote some **description of state** of domain points
    - the dependence on space and time is often omitted in notation
- when indices i, j, k, l refer to cartesian space coordinates the summation convention for repeated indices is used
  - $u_i n_i = \sum_i u_i n_i$
- ”,” denotes differentiation (for indices i, j, k, l of cartesian space coordinates and partial derivatives with respect to time)
  - $u_{i,i} = \frac{\partial u_i}{\partial x_i} = \nabla \cdot \mathbf{u} = \mathrm{div}\mathbf{u} \qquad u_{,t} = \frac{\partial u}{\partial t}$
- standard mathematical notation, operators, etc.
  - e.g. indices of matrix entries: A_{ij} element i, j of matrix $\mathbf{A}$

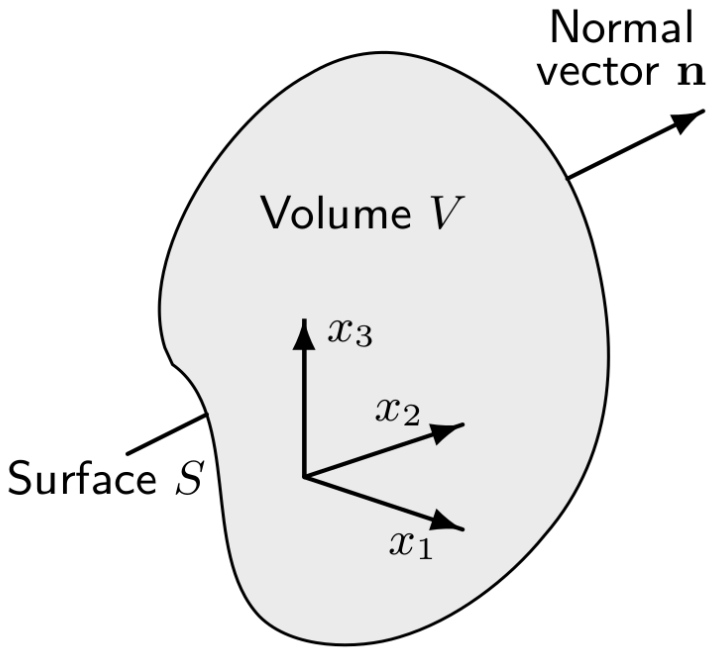
Description of state

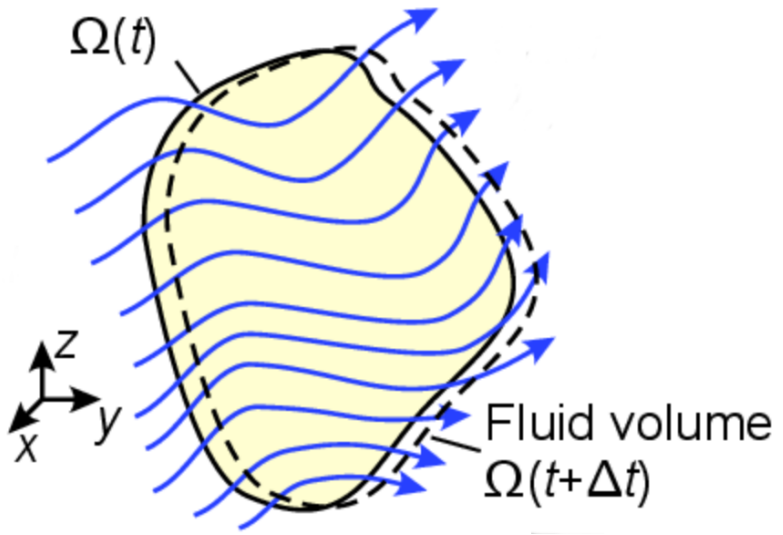
Examples of state description:

- material point
  - position (spatial coordinates)
    - **Cartesian** – $\boldsymbol{x}$, polar, spherical, cylindrical
  - velocity
    - $\boldsymbol{v} = \dfrac{d\boldsymbol{x}}{dt}$ (a single component: $v_i = \dfrac{dx_i}{dt} = x_{i,t}$)
  - acceleration
    - $\boldsymbol{a} = \dfrac{d\boldsymbol{v}}{dt}$ (a single component: $a_i = \dfrac{dv_i}{dt} = v_{i,t}$)
  - displacement
    - $\boldsymbol{l} = \boldsymbol{x} - \boldsymbol{x}_0 \quad (\dfrac{d\boldsymbol{l}}{dt} = \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{v})$
- continuous object (1D, 2D and 3D domains)
  - scalar fields: energy – $e(\boldsymbol{x},t)$), temperature – $T(\boldsymbol{x},t)$
  - vector fields: displacement – $\boldsymbol{l}(\boldsymbol{x},t)$), velocity – $\boldsymbol{v}(\boldsymbol{x},t)$
  - tensor fields: strain – $\boldsymbol{\epsilon}(\boldsymbol{x},t)$ ($\epsilon_{ij} = l_{i,j} + l_{j,i}$), stress – $\boldsymbol{\sigma}(\boldsymbol{x},t)$
- discretization – a process of transferring a description in terms of infinite number of values into a description that uses only a finite number of values

Derivation of fundamental equations in mechanics and thermodynamics

The setting for the derivation of fundamental equations in mechanics and thermodynamics in 2D and 3D





The conservation of mass principle – mass balance

Mass is neither created nor destroyed

- the rate of change of mass inside any domain must be equal to the mass flux through the boundary of the domain:

$$\frac{d}{dt}\int_{\Omega}\rho dV = -\int_{\partial\Omega}\rho v_i n_i dS$$

where:

- $\rho(\mathbf{x},t)$ – density at time t and point $\mathbf{x}\in\Omega$,
- $\mathbf{v}(\mathbf{x},t)$ – velocity,
- $\mathbf{n}$ – the unit outward normal to $\partial\Omega$

After applying the divergence theorem and taking into account that the equation holds for any domain Ω we arrive at the differential equation

The continuity equation (mass balance)

$$\frac{\partial\rho}{\partial t} + (\rho v_i)_{,i} = 0$$

The conservation of momentum principle

Newton's second law of mechanics

- the rate of change of momentum of a group of particles is equal to the sum of all forces exerted on this group

$$\frac{d}{dt}\int_{\Omega_t}\rho v_j dV = \int_{\partial\Omega_t}\sigma_{ji}n_i dS + \int_{\Omega_t} b_j dV$$

where

- Ω_t – portion of space occupied by the group of particles
  - Ω_t may vary in time
- $\boldsymbol{\sigma}$ – stress tensor, b_j – body forces

The transport theorem

$$\frac{d}{dt}\int_{\Omega_t} f dV = \int_{\Omega_t}\left(\frac{\partial f}{\partial t} + (fv_i)_{,i}\right) dV$$

valid for any smooth function $f(\mathbf{x}, t)$

The transport theorem and the momentum balance

Material assumptions (constitutive equations):

- stress tensor σ_{ji} symmetric due to the conservation of angular momentum
- stresses split according to the formula (δ_{ji} – Kronecker's delta):

$$\sigma_{ji} = -p\delta_{ji} + \tau_{ji}$$

  - p – thermodynamic pressure as isotropic normal stresses $p = -\frac{1}{3}\sigma_{ii}$
  - τ_{ji} – viscous stresses
    - isotropic Newtonian fluid, viscous stresses proportional to the rate of change of deformation tensor (gradient of velocity tensor), $\tau_{ji} = \mu(v_{i,j} + v_{j,i}) + \lambda\delta_{ji}v_{k,k}$
    - no volume viscosity (Stokes hypothesis, $\tau_{ii} = 0$), $\tau_{ji} \approx \mu(v_{i,j} + v_{j,i} - \frac{2}{3}\delta_{ji}v_{k,k})$

Assuming the above constitutive equations and applying the transport theorem together with the divergence theorem to the momentum balance leads to:

Momentum equation

$$\frac{\partial(\rho v_j)}{\partial t} + (\rho v_j v_i)_{,i} + p_{,i} - \tau_{ji,i} = b_j$$

Thermodynamic considerations

Basic principles

- the principle of energy conservation, energy balance:
  - specific total energy (energy per unit volume): $e = e_I + e_K + e_p$
  - specific internal energy, e_I, expressed in specific forms for different materials and processes
  - specific kinetic energy, $e_K = \frac{1}{2} v_i v_i$
  - specific potential energy, e_P, possible for external force fields (further neglected)
- the first law of thermodynamics (general statement): $\Delta U = Q - W$
  - the change in internal energy ΔU of a system
  - the amount of heat supplied to the system Q
  - the work done by the system W
- the first law of thermodynamics (in practical calculations):
  - the expression for specific internal energy: $de_I = Tds - pdV$
    - T – temperature, s – entropy, p – pressure, $V = \frac{1}{\rho}$ – volume

Thermodynamic considerations

Constitutive equations

- ideal gas law: $\frac{pV}{T} = const$
  - for practical calculations: $e_I = \frac{p}{(\gamma-1)\rho} = c_V T$
  - heat capacity ratio $\gamma = \frac{c_p}{c_V}$
  - specific heat capacities:
    - at constant volume, $c_V = \left(\frac{\partial Q}{\partial T}\right)_{V=const}$
    - at constant pressure, $c_p = \left(\frac{\partial Q}{\partial T}\right)_{p=const}$
  - the speed of sound c, $c^2 = \frac{\gamma p}{\rho}$
- heat flux q_i
  - Fourier's law: $q_i = -\kappa T_{,i}$
  - κ - the coefficient of thermal conductivity
- $\mu = \frac{1.45\ T^{\frac{3}{2}}}{T+110} \cdot 10^{-6}$ – Sutherland's law for viscosity as function of temperature

The conservation of energy principle

The energy balance

- the rate of change of the total energy for a group of particles is equal to the rate at which work is done by the external forces plus an explicit inflow of energy through the boundary

$$\frac{d}{dt}\int_{\Omega_t}(\rho e)dV = -\int_{\partial\Omega_t} v_j(p\delta_{ji}-\tau_{ji})n_i dS - \int_{\partial\Omega_t} q_i n_i dS$$

Additional terms possible for e.g.

- heat sources
- body forces

The standard procedure comprised of applying the transport and the divergence theorems leads to :

Energy balance equation

$$\frac{\partial(\rho e)}{\partial t} + ((\rho e + p)v_i - \tau_{ji}v_j + q_i)_{,i} = 0$$

Classification of partial differential equations (PDEs)

- The most common PDEs in scientific and technical applications are second order PDEs (PDEs that involve up to the second order derivatives)
  - all second order linear PDEs can be classified as elliptic, parabolic or hyperbolic
- Stationary problems correspond usually to elliptic PDEs, with the standard form:

$$-\frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right)=f(\boldsymbol{x}) \qquad -\left(a_{ij}u_{,j}\right)_{,i}=f$$

- Non-stationary problems, similar to heat equation, correspond to parabolic PDEs, with the standard form

$$\frac{\partial u}{\partial t}-\frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right)=f(\boldsymbol{x}) \qquad u_{,t}-\left(a_{ij}u_{,j}\right)_{,i}=f$$

- Non-stationary problems, of the type similar to elastodynamics equations, correspond to hyperbolic PDEs, with the typical form for scalar unknowns:

$$\frac{\partial^2 u}{\partial t^2}-\frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right)=f(\boldsymbol{x}) \qquad u_{,tt}-\left(a_{ij}u_{,j}\right)_{,i}=f$$

Classification of partial differential equations (PDEs)

Elliptic partial differential equations:

- prototypical example – Poisson problem (Laplace problem for f=0):

$$\Delta u = f \qquad (\ \Delta u = u_{,ii}\ \text{– Laplacian operator}\ )$$

- elliptic PDEs with the appropriate boundary conditions are prototypical boundary value problems (BVPs)
- the solutions to elliptic problems (BVPs) are smooth in typical situations
- the solutions to elliptic BVPs satisfy the maximum principle
  - when certain conditions are fulfilled the maximum is obtained on the boundary of the domain
- standard formulations of the finite difference and the finite element methods work well for elliptic problems
  - there are usually no problems with stability of solutions (they do not tend to infinity)
  - the systems of linear equations associated with elliptic problems are often (for symmetric coefficient arrays and some other conditions) symmetric and positive definite

Classification of partial differential equations (PDEs)

Parabolic partial differential equations:

- parabolic equations require initial and boundary conditions for the existence and uniqueness of solutions
  - parabolic equations lead to initial-boundary value problems (IBVPs)
  - the boundary conditions and initial condition(s) must agree
- the solutions are smooth in typical situations (due to the elliptic, second order in space, terms)
  - even for non-smooth initial conditions the solution rapidly smooths out
  - with increasing time the solution further smooths out (the spatial derivatives tend to zero for problems with no sources)
- the systems of linear equations associated with parabolic problems are often (for symmetric coefficient arrays and some other conditions) symmetric and positive definite

Classification of partial differential equations (PDEs)

Hyperbolic partial differential equations:

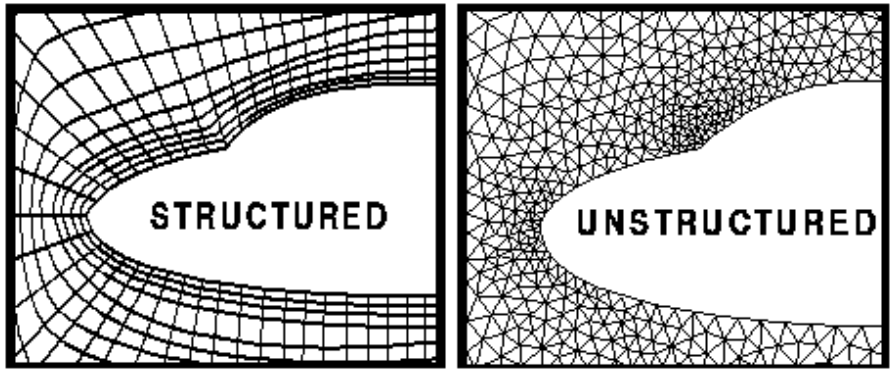
- hyperbolic equations require initial and boundary conditions for the existence and uniqueness of solutions
  - hyperbolic equations lead to initial-boundary value problems (IBVPs)
  - the boundary conditions and initial condition(s) must agree
- there exist curves, called characteristics, along which the solution becomes the solution of an ODE
  - for specific cases the solution along characteristics does not change
  - e.g. for discontinuous initial condition(s) the solution remain discontinuous
  - for advection problems in steady velocity fields the characteristics coincide with the streamlines of the velocity field
- hyperbolic problems correspond to wave and transport (convection, advection) phenomena
  - because of that, boundaries are often classified as inflow or outflow boundaries

Discretization

- We will consider several types of discretization
  - domain discretization – using a finite number of points and parameters, instead of infinite number of points to describe geometric domains
  - function discretization – describing a function (usually continuous) using a finite number of parameters and values at a finite number of points
  - equation discretization – transforming a differential equation for a function (usually continuous) into an equation for a discretized function
- Discretization usually introduces an error – the original domain, original function and the solution to the original equation differ at certain points from their discretized counterparts
  - the discretization error can be measured in a number of different ways
  - with the increasing number of parameters and points the discretized domains, discretized functions and solutions to discretized equations usually tend to their original counterparts (the discretization error goes to 0)
  - discretization is a form of approximation, we will often use the two terms interchangeably
- Only discretized equations, functions and domains are amenable to computer processing

Domain discretization

- Computational domain
  - $\Omega \subset \boldsymbol{R}^N$, *N=1,2 or 3* – domain in 1D, 2D or 3D space
- Domain discretization – using a finite number of points and parameters, instead of infinite number of points to describe the domain
  - the simplest discretization uses a finite number of points and straight line segments to join them
  - more complex discretizations employ a finite number of points and curvilinear segments that join them
  - in 2D the most popular are linear or curvilinear triangles and quadrilaterals
  - in 3D the most popular are linear or curvilinear tetrahedra and hexahedra, as well as prisms and pyramids
- The set of points and segments joining them is called a grid (or mesh)
- There are two basic types of grids (with different neighbourhood relations)
  - structured (regular) grids
  - unstructured grids



The finite element method

The finite element method is a method for approximating the solutions to boundary value problems

- The two fundamental ingredients of the FEM are:
  - the use of weak variational statements of the problems
  - the discretization of the computational domains into small parts, called elements, within which the solution is approximated using simple polynomials
- The FEM is especially efficient for solving elliptic problems (stationary with no time variable) in complex 3D domains
- The FEM can also be used for solving initial boundary value problems (with time variable), usually in combination with other discretization methods such as the finite difference method or the discontinuous Galerkin method

The finite element method

Discretization of the computational domain:

- The sum of all elements must completely fill the computational domain
- Elements cannot overlap
- Elements should have sufficient quality
  - the ratio of the sizes of edges should be limited
  - the internal angles between the edges should not be too small
- The ratio of the sizes of neighbouring elements should be limited
- Types of meshes:
  - 1D – division into small intervals
  - 2D – popular elements: triangles, quadrilaterals
  - 3D – popular elements: tetrahedra, hexahedra, prisms (less frequent: pyramids)
  - apart from elements with straight edges (and plane faces in 3D) there are elements with curved boundaries

The finite element method

Finite element function spaces:

- elements $\rightarrow$ shape functions, ϕ_i
- computational domain $\rightarrow$ basis functions constructed from shape functions, ψ_j
  - in the standard FEM basis functions are continuous
  - basis functions have as small support (the domain of non-zero values) as possible
- finite element solutions as linear combinations of basis functions

$$u^h(x) = \mathrm{U}_1^h\psi_1 + \mathrm{U}_2^h\psi_2 + \mathrm{U}_3^h\psi_3 + \ldots + \mathrm{U}_N^h\psi_N = \sum_j^N \mathrm{U}_j^h\psi_j$$

$$u^h(x) \in V^h(x) = \mathrm{span}\,\{\psi_1, \psi_2, \psi_3, \ldots, \psi_N\}$$

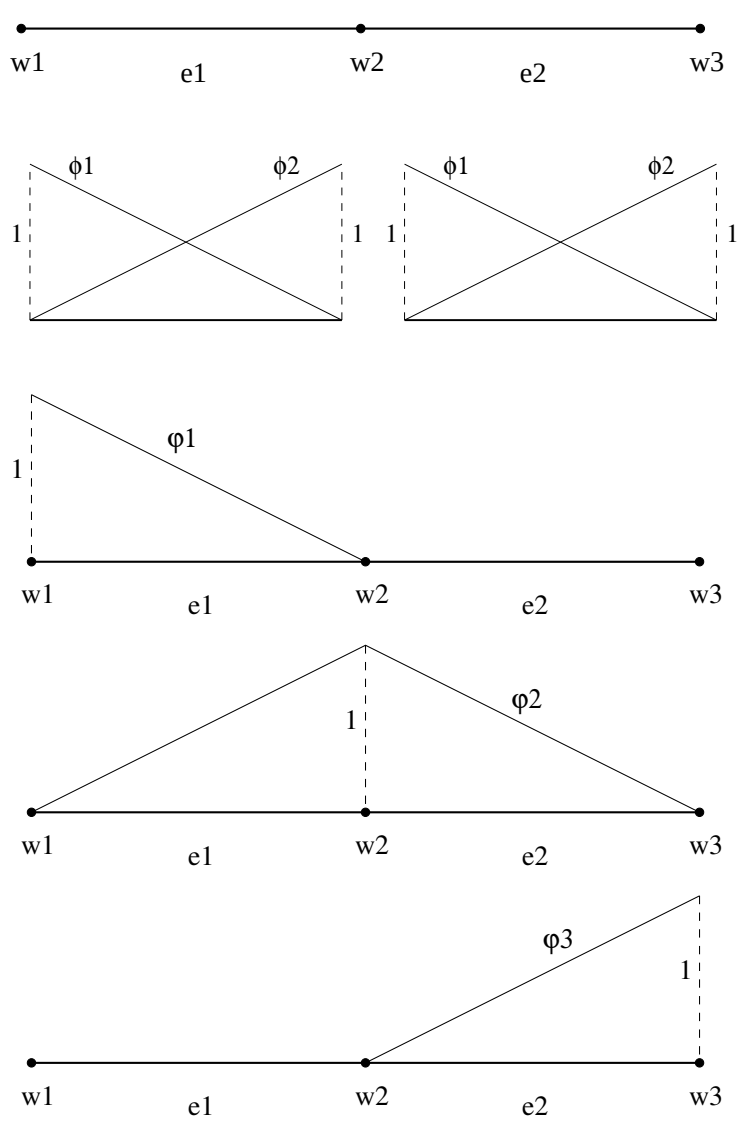
  - coefficients U_j^h of linear combination (degrees of freedom) form a discrete FEM solution to the approximation problem
  - N – the size of vector $\mathbf{U}^h$, i.e. the number of degrees of freedom, is the size of a particular FEM problem

Finite element formulation for the model 1D problem

- The division of computational domain into elements
- Shape functions ϕ_i inside elements
- Basis functions ψ_j for the whole computational domain

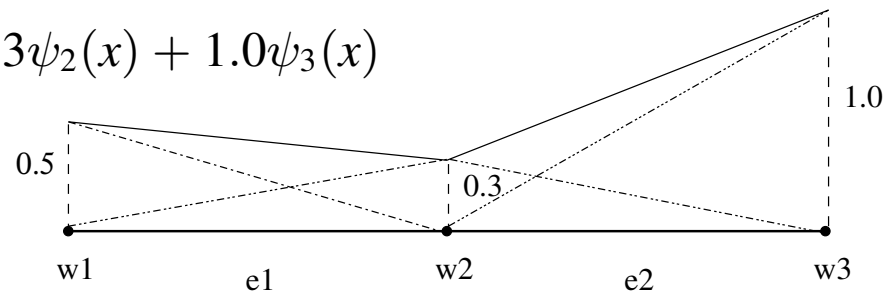
Example:

- domain: $(0, 1)$
- elements: $e_1 - (0, 0.5)$ and $e_2 - (0.5, 0)$
- element vertices (finite element nodes): $\{w_1, w_2, w_3\} - \{0, 0.5, 1.0\}$



Finite element interpolation

- Using finite element spaces it is possible to construct not only approximate solutions but also interpolants (functions that agree with a set of discrete values)
- Finite element interpolation is especially easy for the spaces where finite element degrees of freedom correspond to the values at specific points (warning: there are spaces where it is not true!)
  - for typical finite element spaces with linear basis (shape) functions the values of degrees of freedom are the values of finite element solutions at element vertices
  - Example:
    - interpolation for the set of points: $\{(w_1, 0.5), (w_2, 0.3), (w_3, 1.0)\}$
    - $\mathbf{U}^h = \{0.5, 0.3, 1.0\}$
    - $u^h(x) = 0.5\psi_1(x) + 0.3\psi_2(x) + 1.0\psi_3(x)$



Finite element formulation for the model 1D problem

- Weak formulation:
  Find a function $u^h \in V^h \subset V$ such that the following holds:
  $$\left(\frac{du^h}{dx}, \frac{dw^h}{dx}\right) = (f, w) + u_1' \cdot w^h(1) \qquad \forall w^h \in V_0^h \subset V_0$$
- Domain discretization:
  Partition of $(0, 1)$ into subintervals (x_{j-1}, x_j) of length $h_j = x_j - x_{j-1}$ with $h = \max h_j$
- Finite element discretization (approximation):
  $$u^h = \sum_j^N \mathrm{U}_j^h \psi_j \qquad w^h = \sum_i^N \mathrm{W}_i^h \psi_i$$

Hence:

$$\left(\frac{d\sum_j^N \mathrm{U}_j^h \psi_j}{dx}, \frac{d\sum_i^N \mathrm{W}_i^h \psi_i}{dx}\right) = \sum_i^N \mathrm{W}_i^h \sum_j^N \mathrm{U}_j^h \left(\frac{d\psi_j}{dx}, \frac{d\psi_i}{dx}\right)$$

Finite element approximation for the model 1D problem

- General finite element solution procedure consists of two steps:
  - creation of the system of linear equations:

$$\sum_i^N \mathrm{W}_i^h \left(\sum_j^N \mathrm{U}_j^h \left(\frac{d\psi_j}{dx}, \frac{d\psi_i}{dx} \right) - (f, \psi_i) - u_1' \cdot \psi_i(1) \right) = 0 \qquad \forall \mathbf{W}^h = \{\mathrm{W}_1^h, \mathrm{W}_2^h, ..., \mathrm{W}_N^h\}$$

Hence:

$$\sum_j^N \mathrm{U}_j^h \left(\frac{d\psi_j}{dx}, \frac{d\psi_i}{dx} \right) = (f, \psi_i) + u_1' \cdot \psi_i(1) \qquad i = 1, 2, ..., N$$

i.e.

$$\sum_j^N \mathrm{A}_{i,j} \mathrm{U}_j^h = \mathrm{b}_i \qquad i = 1, 2, ..., N$$

with: $\mathrm{A}_{i,j} = \left(\frac{d\psi_j}{dx}, \frac{d\psi_i}{dx} \right)$ and $\mathrm{b}_i = (f, \psi_i) + u_1' \cdot \psi_i(1)$

  - solution of the system of linear equations

Finite element approximation

Finite element approximations to elliptic problems have several important properties:

- For many problems it is relatively easy to prove the existence and uniqueness of exact and approximate finite element solutions using the corresponding weak formulations
  - this concerns in particular the model 1D problem considered
- FEM approximate solutions satisfy the best approximation property:
  - for the model 1D problem:

$$\|(u^h - u)'\| < \|(w^h - u)'\| \qquad \forall w^h \in V_0^h$$

- Using the interpolant of the exact solution as the function w^h in the formula above and the interpolation error estimate it is possible to estimate the error of the finite element solution as:
  - for the model 1D problem:

$$\|e^h\| = \|u^h - u\| < Ch^2 \cdot \max |u''|$$

Finite element approximation

The properties of FEM approximations have important consequences:

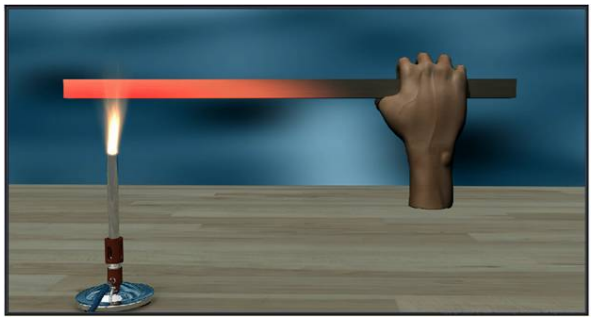
- one can control the error of finite element solutions by a suitable choice of element sizes
- with the maximal element size going to zero finite element solutions converge to the exact solution

Additional observations:

- the error depends on the second order derivative of the exact solution, not the gradient (as is often incorrectly stated)
- with the element size going to zero, the number of elements in the computational domain and the number of degrees of freedom in the system of linear equations associated with the problem go to infinity
    - however: the computational cost does not grow quadratically with the number of degrees of freedom, since the matrices of linear systems are very sparse
        - for really large problems the number of zeros in the system matrices can easily exceed 99,99%

1D stationary heat transfer problem

- Reality (experiment)



- Physical model

Energy conservation - the rate of change of heat flux is equal to heat source

$$dq/dx = s(x)$$

Fourier's law - heat flux is proportional to the temperature gradient

$$q = -k \cdot dT/dx$$

- k - heat conduction coefficient

- Mathematical model

Ordinary differential equation

$$-\frac{d}{dx}\left(k\frac{dT}{dx}\right) = s(x)$$

+ Boundary conditions

(for both ends, possible types:

- temperature, e.g.

$$T(0) = T_0$$

- heat flux, e.g.

$$q(L) = -k \cdot dT/dx = q_0$$

- other (convection, radiation)

= Boundary value problem

$\rightarrow$ **existence and uniqueness of results**

Finite element formulation for stationary heat transfer problems

- For differential formulation of the form (with zero Dirichlet BC only, for simplicity):

$$-\nabla \cdot (k(T,\boldsymbol{x})\nabla T) = s$$

- The following weak statement can be derived:

Find approximate function $T^h \in V_T^h$, such that the following statement:

$$\int_\Omega k(T^h,\boldsymbol{x})T^h_{,i}w^h_{,i}d\Omega = \int_\Omega sw^h d\Omega$$

holds for every test function $w^h \in V_w^h$.

- For material properties being the function of $\boldsymbol{x}$ only, the problem is (quasi-)linear
- For material properties being the function of T as well, the problem has material non-linearity

Finite element formulation for stationary heat transfer problems

- Adding Neumann and Robin boundary conditions:

$$-k(T^h,\boldsymbol{x})\frac{dT}{d\boldsymbol{n}} = -k(T^h,\boldsymbol{x})T_{,i}n_i = -q_N \qquad \text{on} \qquad \Gamma_N$$

$$-k(T^h,\boldsymbol{x})\frac{dT}{d\boldsymbol{n}} = -k(T^h,\boldsymbol{x})T_{,i}n_i = c(T^h,\boldsymbol{x})(T-T_{ext}) \qquad \text{on} \qquad \Gamma_R$$

- Lead to the formulation with additional terms:

Find approximate function $T^h \in V_T^h$, such that the following statement:

$$\int_\Omega k(T^h,\boldsymbol{x})T^h_{,i}w^h_{,i}d\Omega = \int_\Omega sw^h d\Omega + \int_{\Gamma_N} q_N w^h d\Gamma - \int_{\Gamma_R} c(T-T_{ext})w^h d\Gamma$$

holds for every test function $w^h \in V_w^h$

Finite element formulation for stationary heat transfer problems

- The final formulation for linear stationary heat transfer problems:

Find approximate function $T^h \in V_T^h$, such that the following statement:

$$\int_\Omega kT^h_{,i} w^h_{,i} d\Omega + \int_{\Gamma_R} cTw^h d\Gamma = \int_\Omega sw^h d\Omega + \int_{\Gamma_N} q_N w^h d\Gamma + \int_{\Gamma_R} cT_{ext} w^h d\Gamma$$

holds for every test function $w^h \in V_w^h$

- ... leads to the following formulae for the entries of the global stiffness matrix and the global load vector

$$\mathrm{A}_{i,j} = \int_\Omega k \frac{d\psi_j}{dx_l} \frac{d\psi_i}{dx_l} d\Omega + \int_{\Gamma_R} c\psi_j \psi_i d\Gamma$$

$$\mathrm{b}_i = \int_\Omega s\psi_i d\Omega + \int_{\Gamma_N} q_N \psi_i d\Gamma + \int_{\Gamma_R} cT_{ext} \psi_i d\Gamma$$

Finite element systems of linear equations

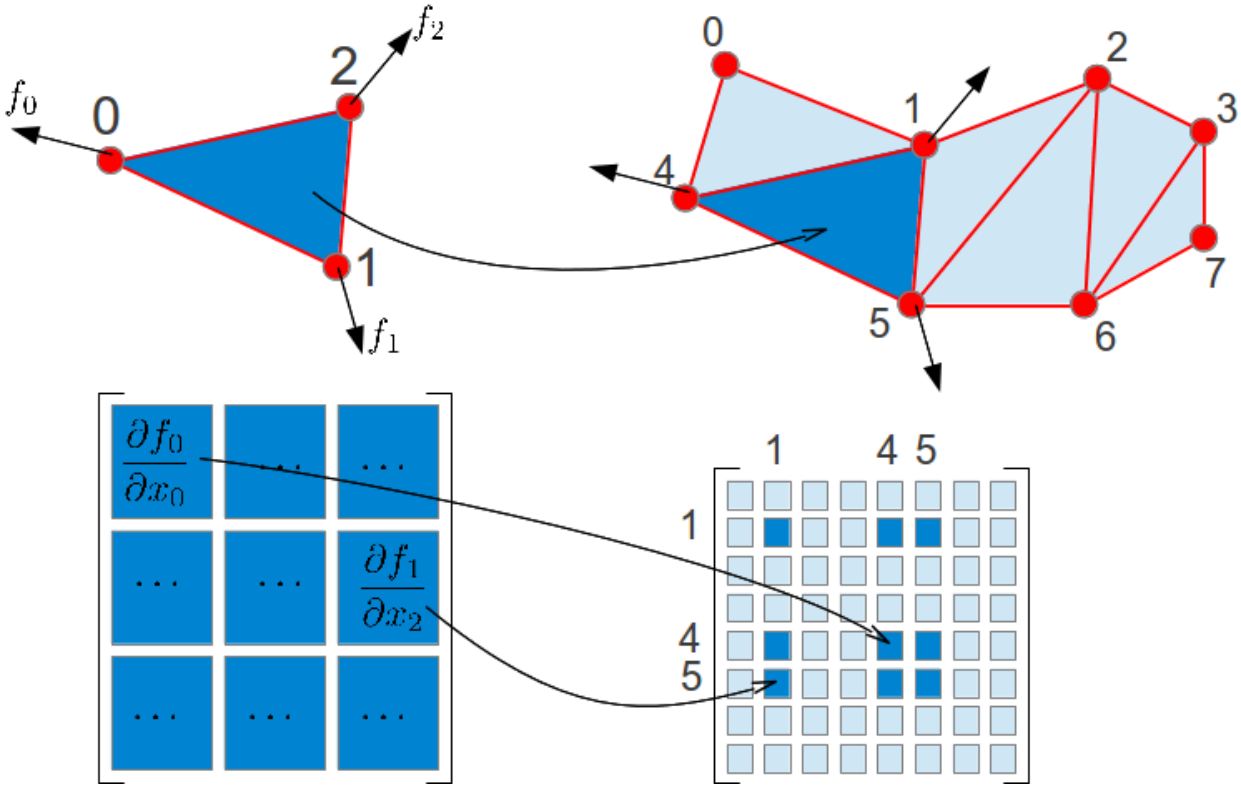
- Standard discretizations for linear stationary problems require the solution of a system of linear equations

$$\sum_{j}^{N} \mathrm{A}_{i,j}\mathrm{U}_{j}^{h} = \mathrm{b}_{i} \qquad i = 1, 2, ..., N \qquad \equiv \qquad \mathbf{A}\mathbf{U}^{h} = \mathbf{b}$$

  - for non-stationary problems and implicit time integration a system of linear equations is solved at every time step
  - for non-linear problems a system of linear equations is solved for every iteration of the solution method
- The procedures for solving a linear system include
  - the creation of the system of linear equations that includes the integration of the terms from the weak statement for suitable pairs of basis functions
    - the integrals are calculated separately for each element, forming local, element system matrices and right hand side vectors
    - the local matrices and vectors are than assembled into the global system matrix and the global right hand side vector
  - the solution of the system, that takes into account its special form

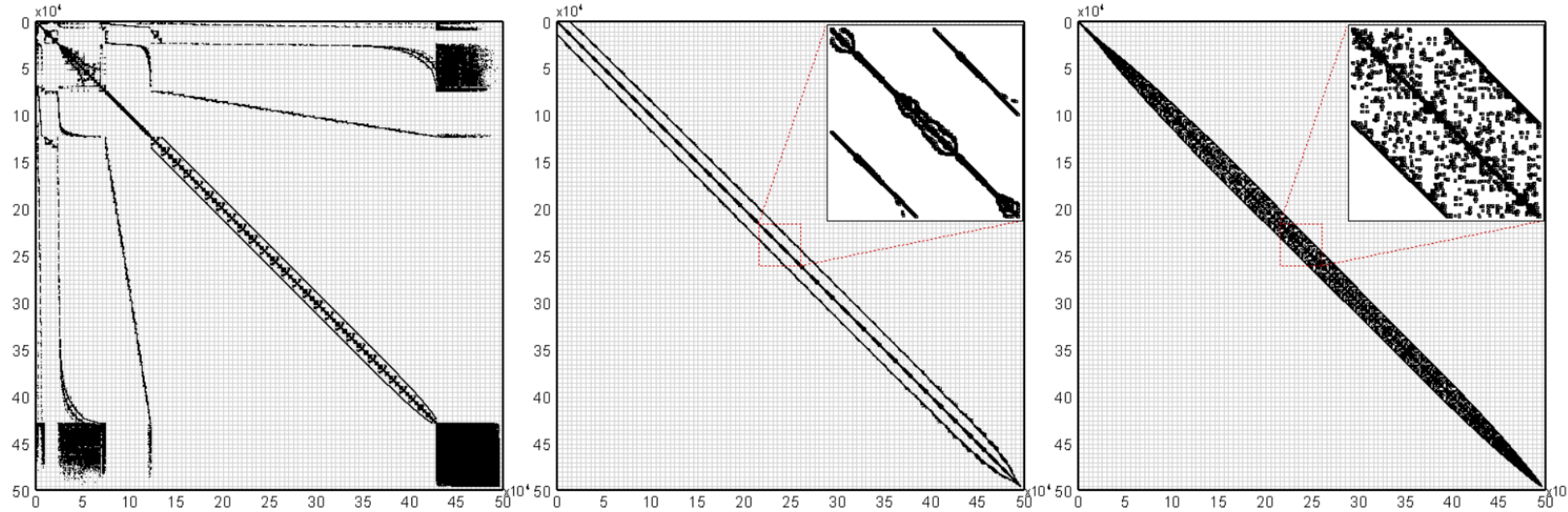
Finite element systems of linear equations

- The assembly of global finite element systems of linear equations
  - local element matrices computed using numerical integration
  - local numbering of degrees of freedom
  - global numbering of degrees of freedom



Finite element systems of linear equations

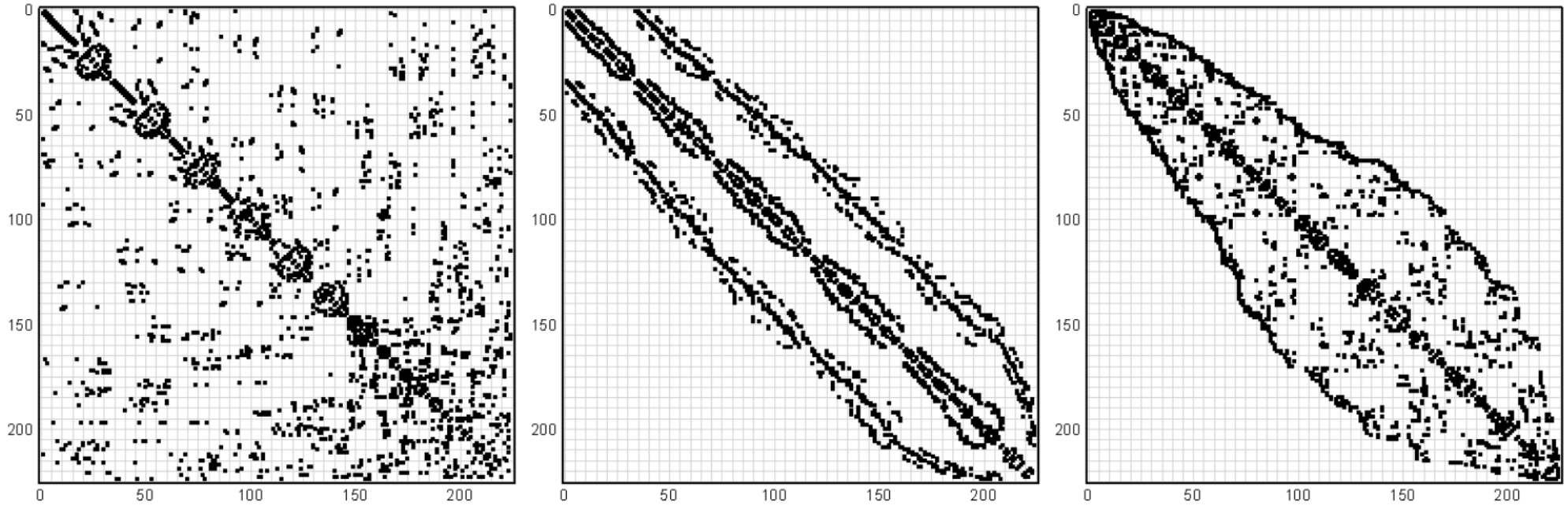
- The solved equations are
  - usually large (up to billions of unknowns)
  - sparse (for large systems more than 99.99% entries in the system matrix are zero)
  - often ill conditioned – with large condition number and slow convergence of iterative methods



Finite element systems of linear equations

Practical solutions for solving FEM systems of linear equations

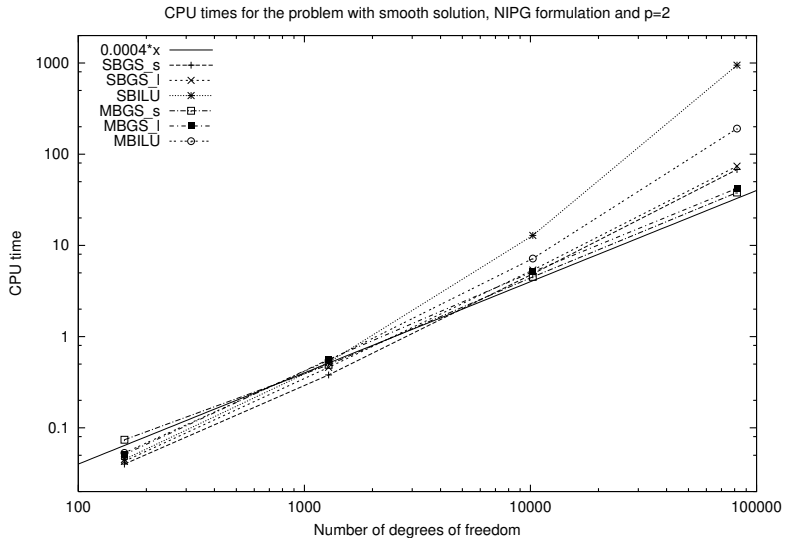
- Direct methods for solving large sparse systems of linear equations
  - the variants of Gaussian elimination
  - the problem of fill-in
    - renumbering
    - frontal methods

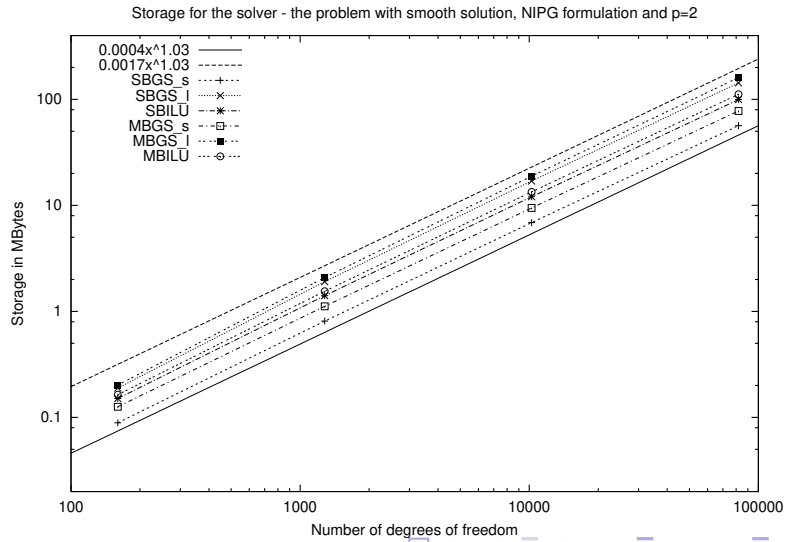


Finite element systems of linear equations

Practical solutions for solving FEM systems of linear equations

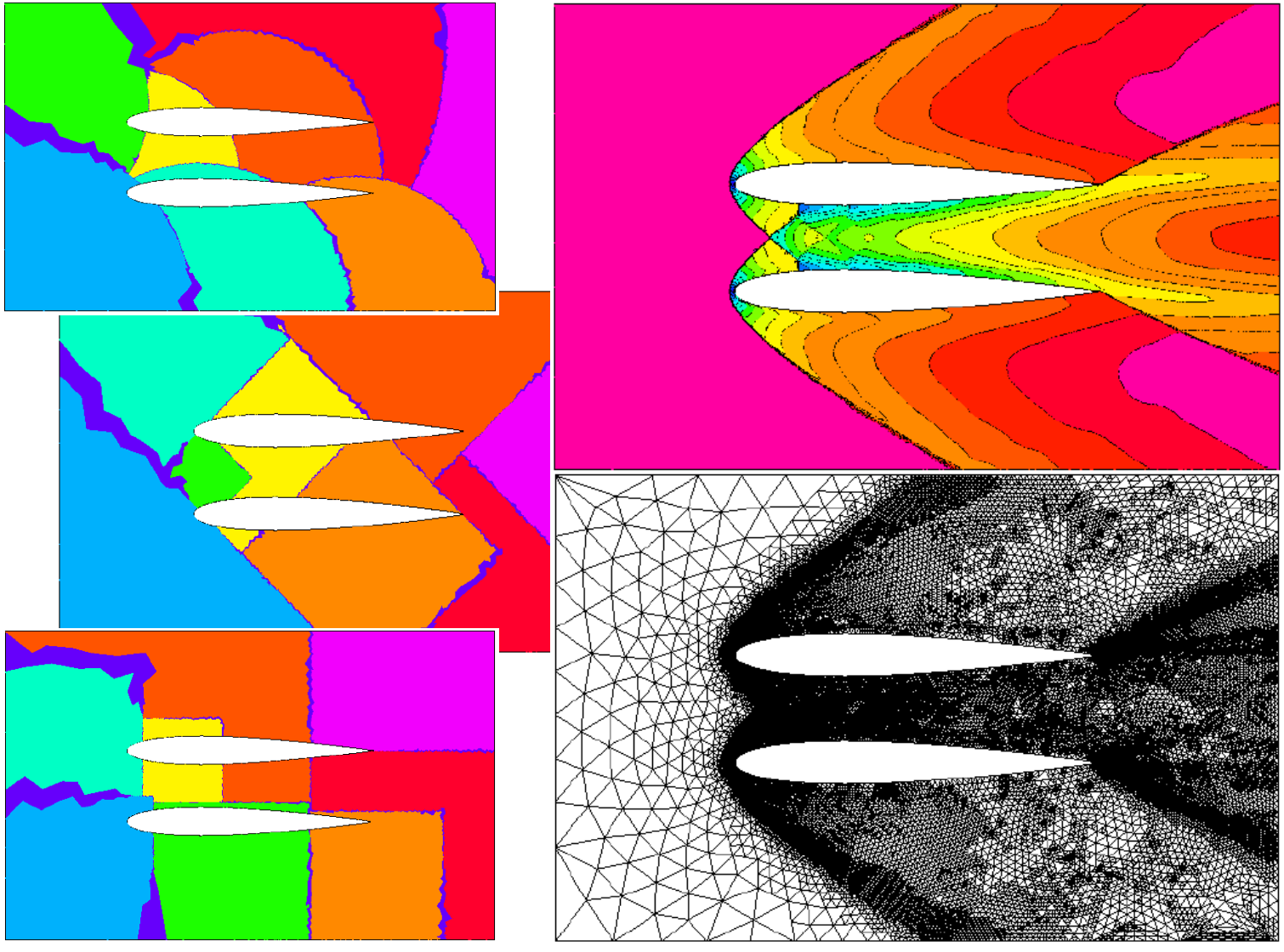
- Iterative methods for solving large sparse systems of linear equations
  - slow convergence of standard iterative methods
  - simple preconditioners: Jacobi (diagonal scaling), Gauss-Seidel, incomplete LU factorization
  - complex preconditioners: multigrid, special preconditioners for specific problems
  - the best iterative solvers can have linear complexity, both in terms of solution time and storage requirements





Finite element solution procedures

Parallel solution based on domain decomposition



Non-linear problem solution

- Finite element space discretization of non-linear problems leads to the set of non-linear algebraic equations for the vector of degrees of freedom $\mathbf{U}^h$, that can be shortly written as:

$$\mathbf{A}(\mathbf{U}^h)\mathbf{U}^h = \mathbf{b}$$

- The general methods for solving multidimensional systems of the form

$$\mathbf{F}(\mathbf{U}) = 0$$

usually refer to the Newton's iterative method, that finds the subsequent approximations

$$\mathbf{U}_{k+1} = \mathbf{U}_k + \boldsymbol{\Delta}\mathbf{U}_k$$

where $\boldsymbol{\Delta}\mathbf{U}_k$ is the solution to the equation

$$\mathbf{J}(\mathbf{U}_k) \cdot \boldsymbol{\Delta}\mathbf{U}_k = -\mathbf{F}(\mathbf{U}_k)$$

with the Jacobian matrix $\mathbf{J}$ representing the gradient of the function $\mathbf{F}$

$$\mathbf{J} = \partial\mathbf{F}/\partial\mathbf{U}$$

Non-linear problem solution

- Applying the Newton's method to the system:

$$\mathbf{A}(\mathbf{U}^h)\mathbf{U}^h = \mathbf{b}$$

leads to the equation

$$\left(\frac{\partial \mathbf{A}}{\partial \mathbf{U}^h}(\mathbf{U}_k^h)\mathbf{U}_k^h + \mathbf{A}(\mathbf{U}_k^h)\right) \cdot \boldsymbol{\Delta}\mathbf{U}_k^h = -\mathbf{A}(\mathbf{U}_k^h)\mathbf{U}_k^h + \mathbf{b}$$

- When the derivative $\frac{\partial \mathbf{A}}{\partial \mathbf{U}^h}$ is assumed to vanish, the system reduces to the form

$$\mathbf{A}(\mathbf{U}_k^h) \cdot \mathbf{U}_{k+1}^h = \mathbf{b}$$

that can be interpreted as using fixed point (Picard's) iterations

$$\mathbf{U}_{k+1}^h = \mathbf{A}(\mathbf{U}_k^h)^{-1} \cdot \mathbf{b}$$

for the original nonlinear problem

Non-linear problem solution

- In general (for 1D case) Picard's (fixed point) iterations are defined as subsequent computations

$$x_{k+1} = g(x_k)$$

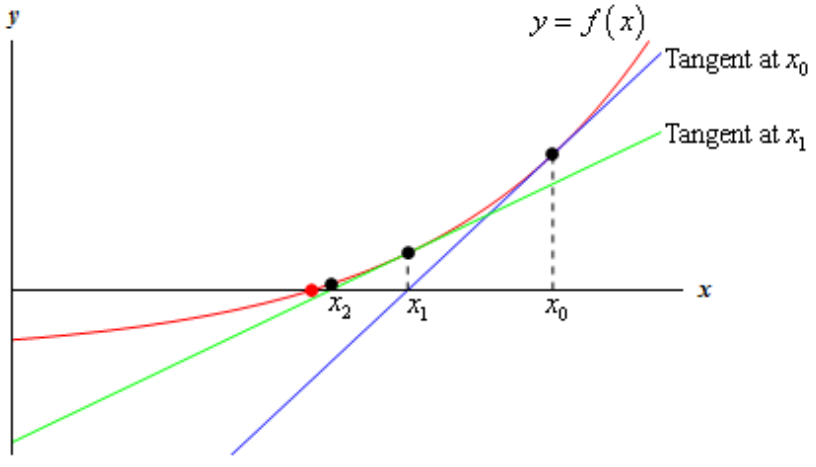
that after convergence lead to the satisfaction of the nonlinear problem

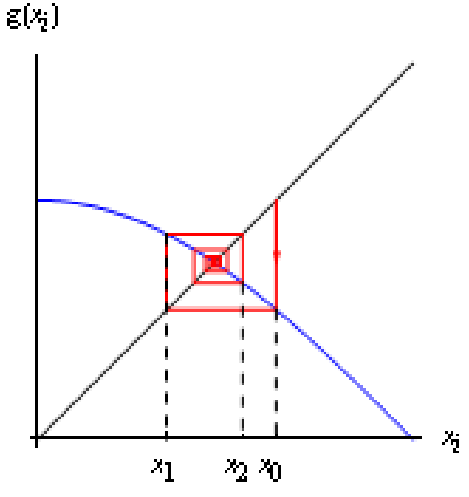
$$x = g(x)$$

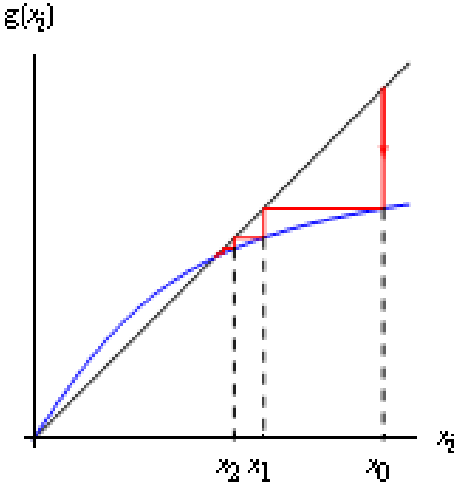
- Newton's method iterations for the problem $f(x) = 0$:

$$x_{k+1} = x(k) - f'(x_k)^{-1} \cdot f(x_k) \quad [= g(x_k)]$$

can be interpreted as a special case of fixed point iterations







Finite element approximation of elliptic problems

Norms for measuring error (and other functions defined over Ω)

- L_2

$$\|e\|^2_{L_2(\Omega)} = \int_\Omega (e \cdot e) d\Omega$$

- H^1

$$\|e\|^2_{H^1(\Omega)} = \int_\Omega \left(e_{,i} \cdot e_{,i} + e \cdot e\right) d\Omega$$

- H^1 seminorm

$$|e|^2_{H^1(\Omega)} = \int_\Omega (e_{,i} \cdot e_{,i}) d\Omega$$

- energy norm
  - for many problems their bilinear forms satisfy the requirements for scalar products and, hence, can be used to define a norm:

$$\|e\|^2_a = a(e,e)$$

Finite element approximation of elliptic problems

- Fundamental error estimate for finite element approximation of elliptic problems is

$$\|u - u_h\|_V \leq C \cdot \|u - w_h\|_V \qquad \forall w_h \in \tilde{V}_h$$

  where $\|.\|_V$ is a norm induced by the scalar product defined for the space V
- The standard method of obtaining absolute error estimates for finite element approximation is to select a particular suitable function w_h (usually interpolant of u in V_h) and then obtaining error estimates for w_h
- Interpolation theory gives error estimates for interpolants in different finite element spaces – for linear, quadratic, etc. shape functions
- For standard continuous polynomials of order p one can finally get:

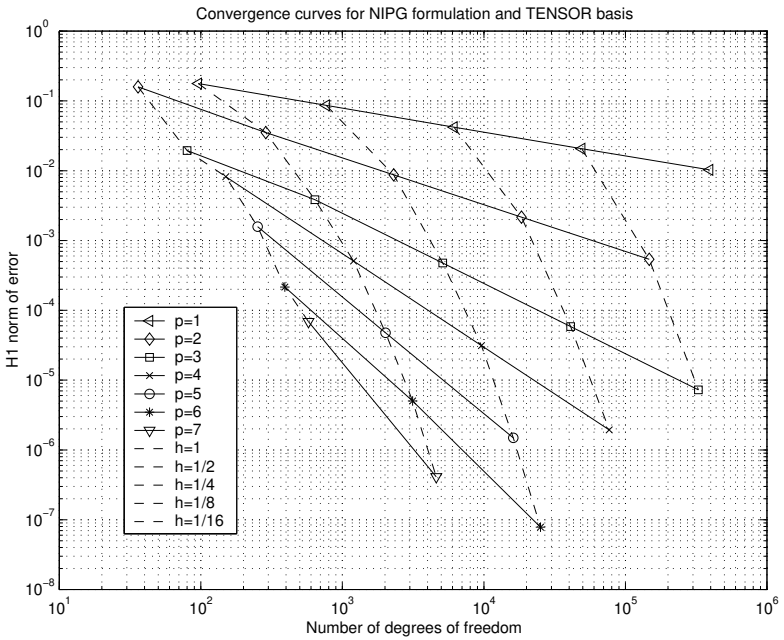
$$\|u - u_h\|_{H^1(\Omega)} \leq Ch^p |u|_{H^{p+1}(\Omega)}$$

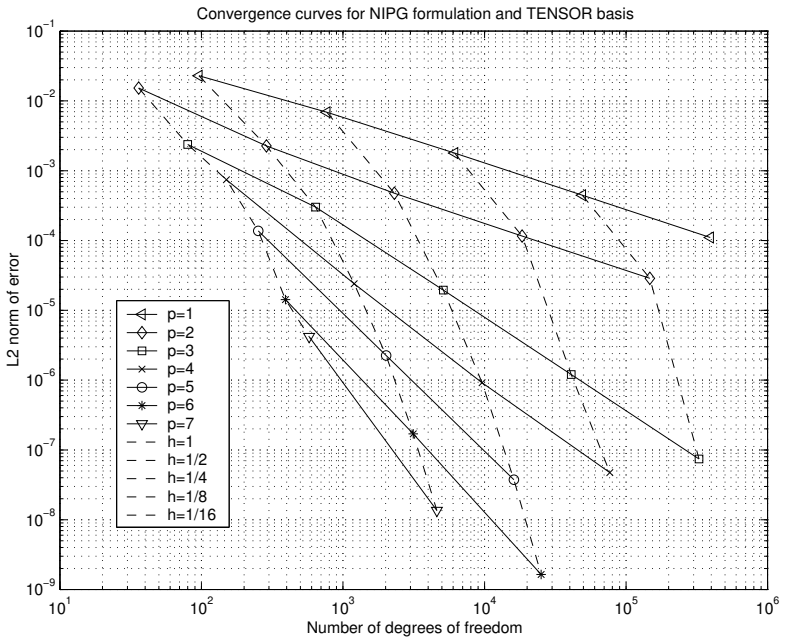
- The estimate requires the exact solution u to be sufficiently smooth, that depends on the problem and the shape of the computational domain Ω

Finite element approximation of elliptic problems

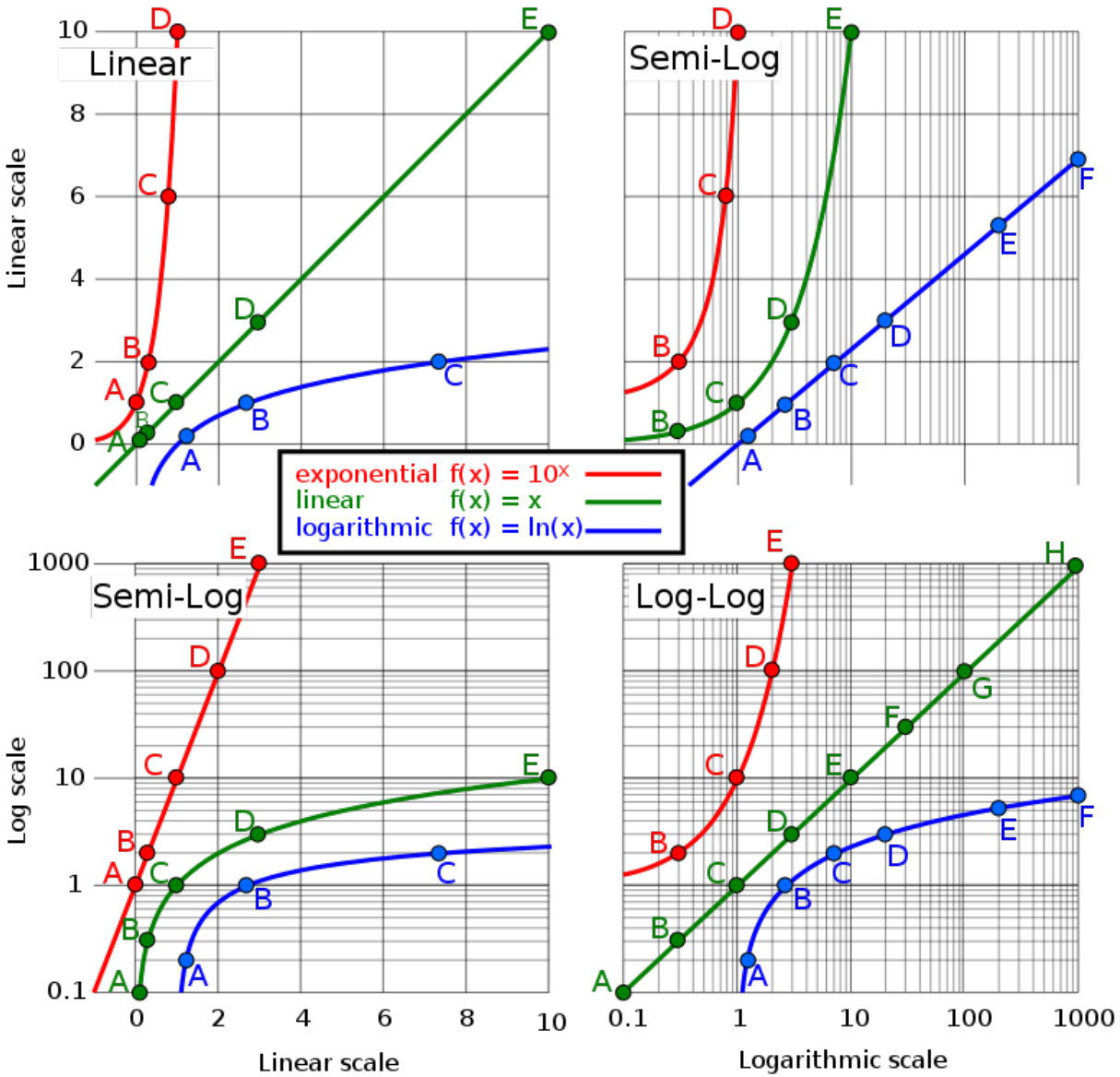
Typical convergence curves for finite element approximation of elliptic problems, measured in L_2 and H^1 norms

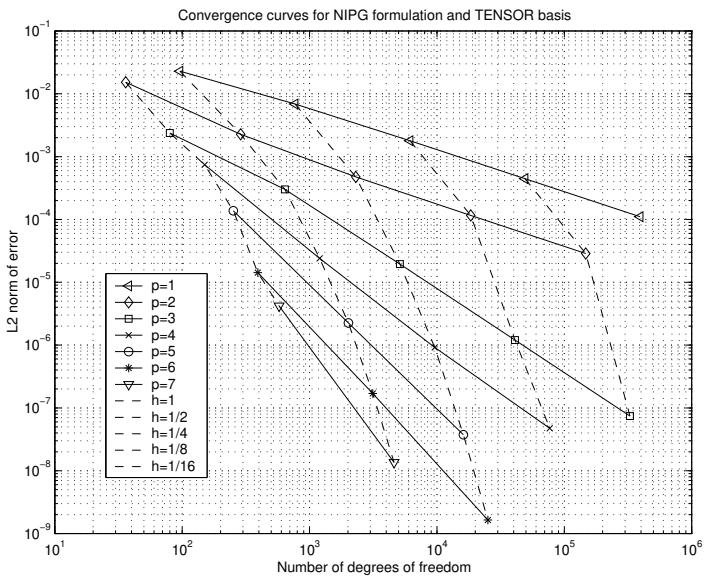
- log-log scale to explicitly show convergence rates
- the solution usually converges in L_2 norm with the rate h^{p+1}
- higher order approximations have better accuracy, but require more computational resources, for the same number of degrees of freedom





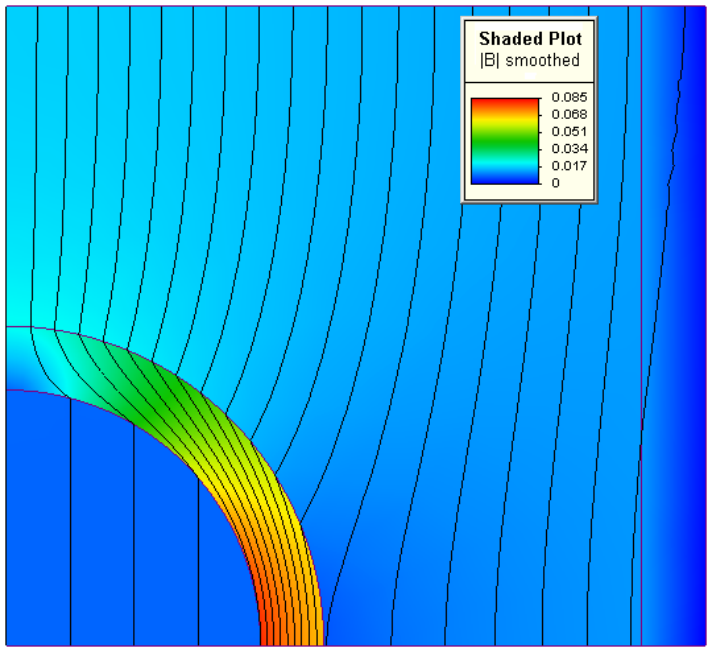
Intermezzo - how to read graphs

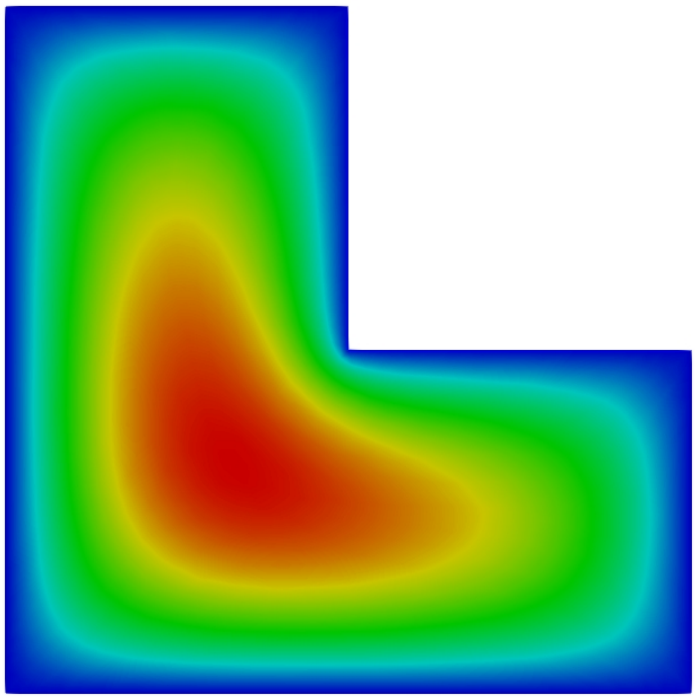




Finite element approximation of elliptic problems

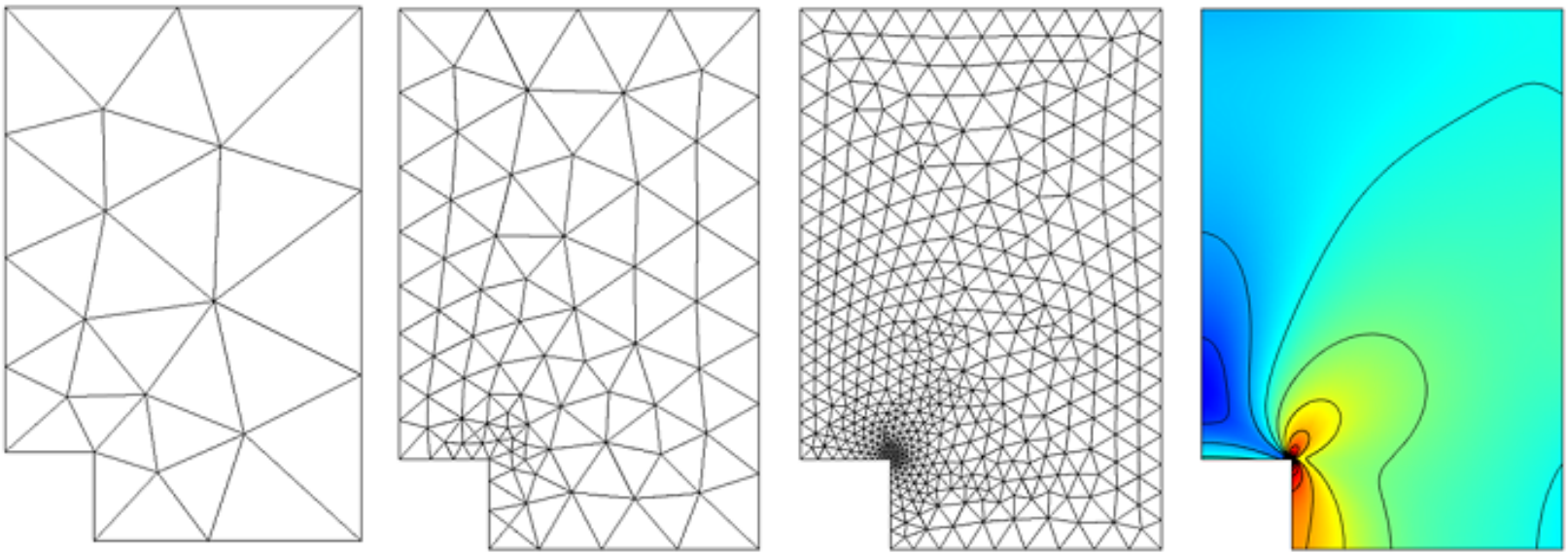
- The error of the finite element solution is related to the smoothness of the exact solution
  - for certain problems (e.g. with discontinuous coefficients - left) and for certain computational domains (e.g. with corners - right) the exact solution has large higher order derivatives
  - the convergence rates for such problems and uniform mesh refinements are slow





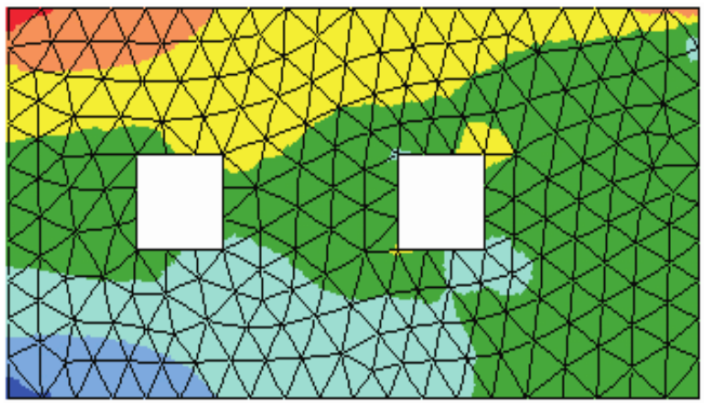
Finite element approximation of elliptic problems

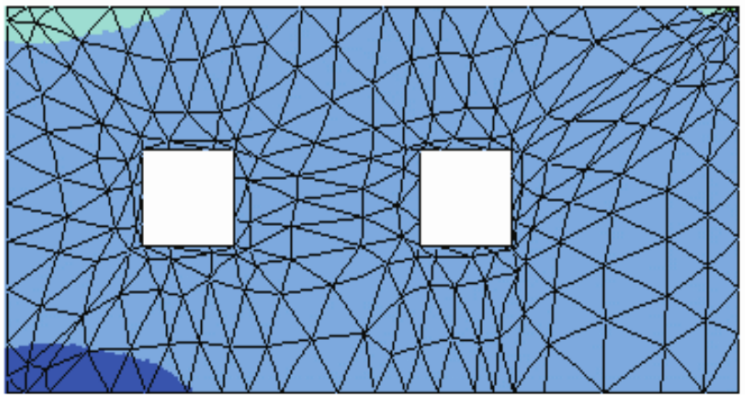
- The nature of the finite element error estimates suggests that it is possible to decrease approximation error, especially for the problems with singularities, by local changes to approximation properties
- This observation gives rise to the adaptive finite element method, where in the places with higher approximation error the approximation is locally improved



Finite element approximation of elliptic problems

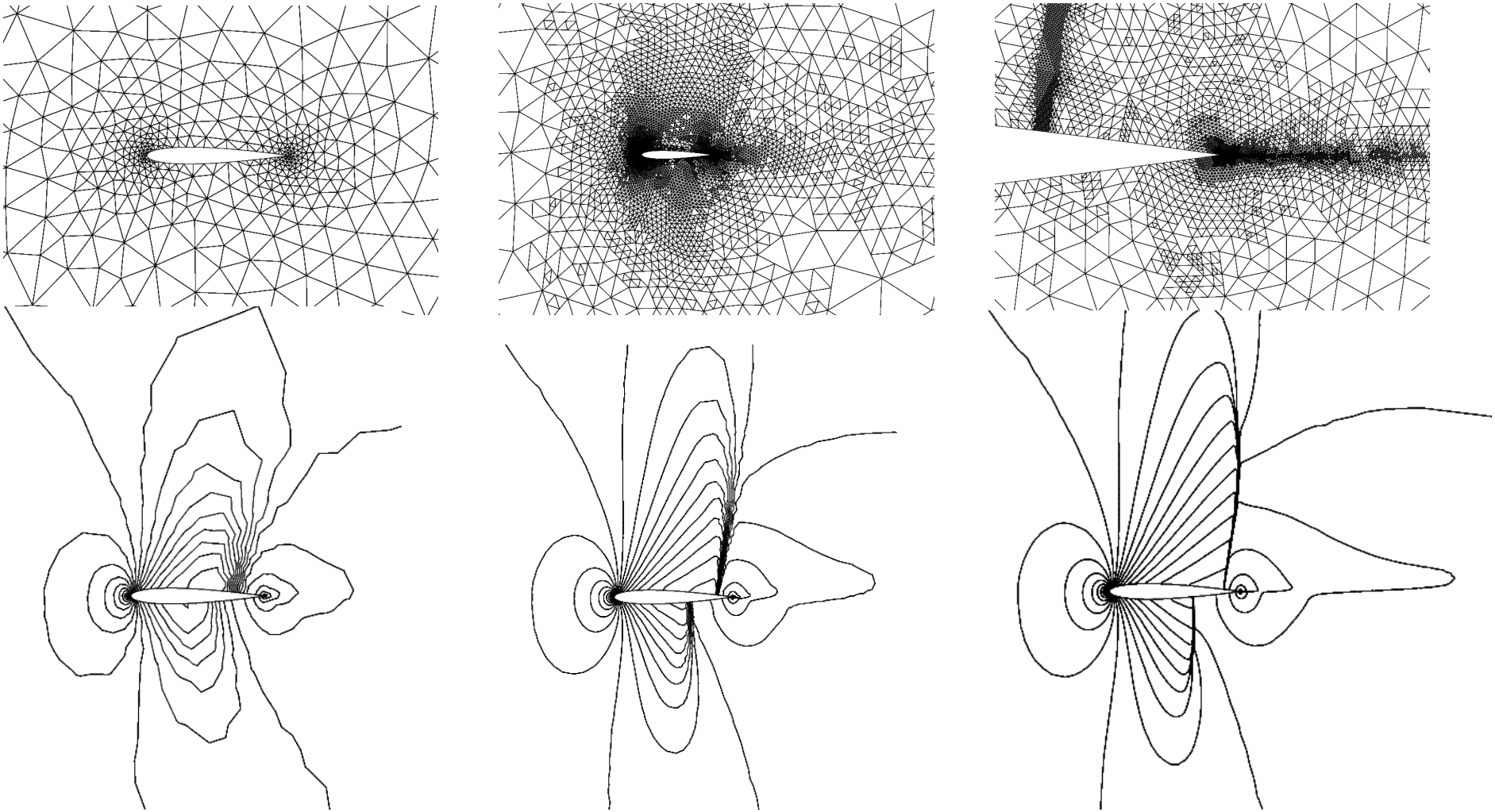
- There are several main types of adaptivity:
  - *h*-adaptivity – the size of elements is reduced by dividing elements (the number of degrees of freedom grows)
  - *p* adaptivity – the local order of approximation is increased (this requires special techniques to maintain the continuity of the solution)
  - *hp*-adaptivity – the combination of the two above
  - *r*-adaptivity – the finite element nodes are moved, in order to create parts of the domain with smaller elements (the total number of degree of freedom may remain the same)
  - remeshing – creating a new, finer grid for the selected parts of the domain (or a new mesh with variable ”density” of finite element nodes)





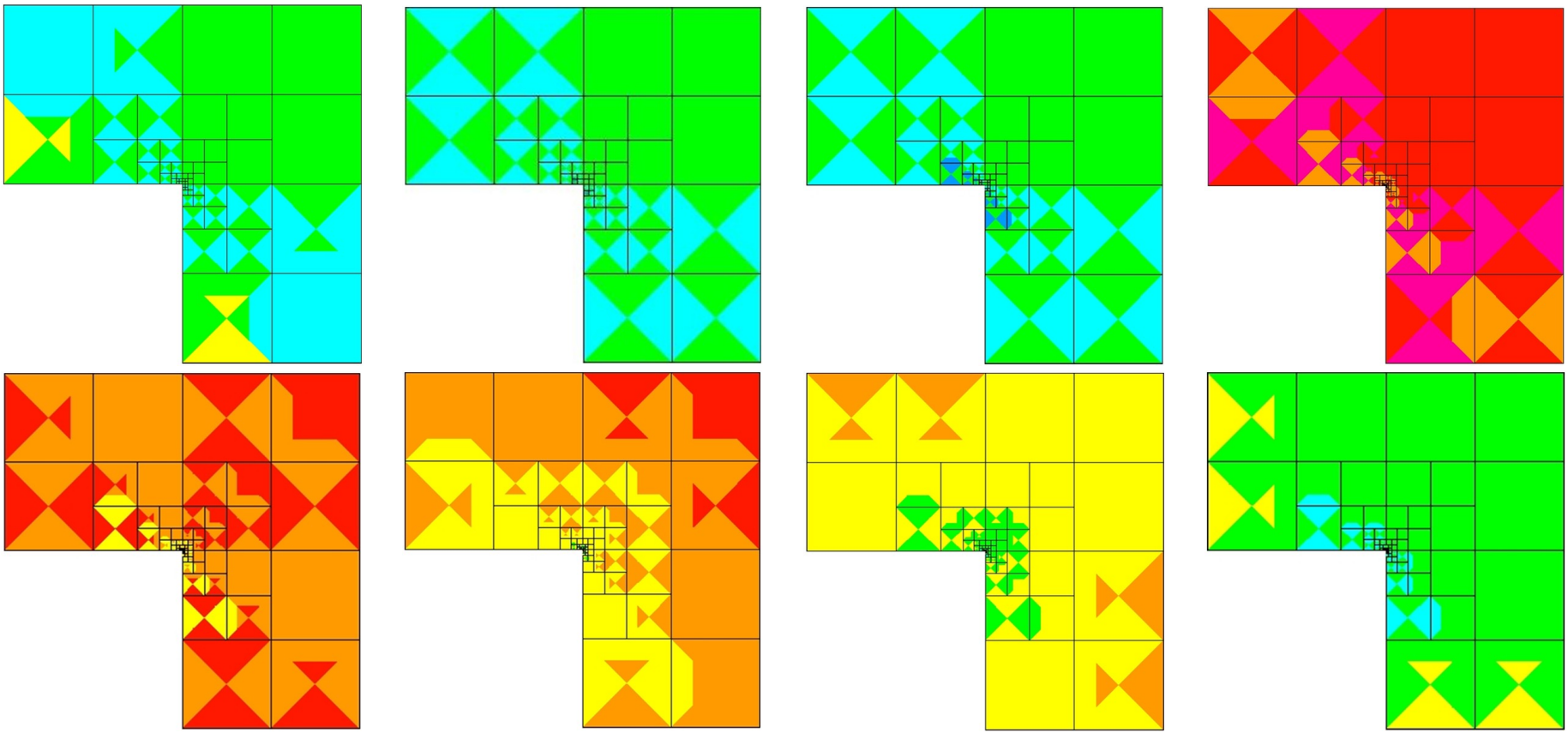
Finite element approximation of elliptic problems

- *h* adaptivity at work



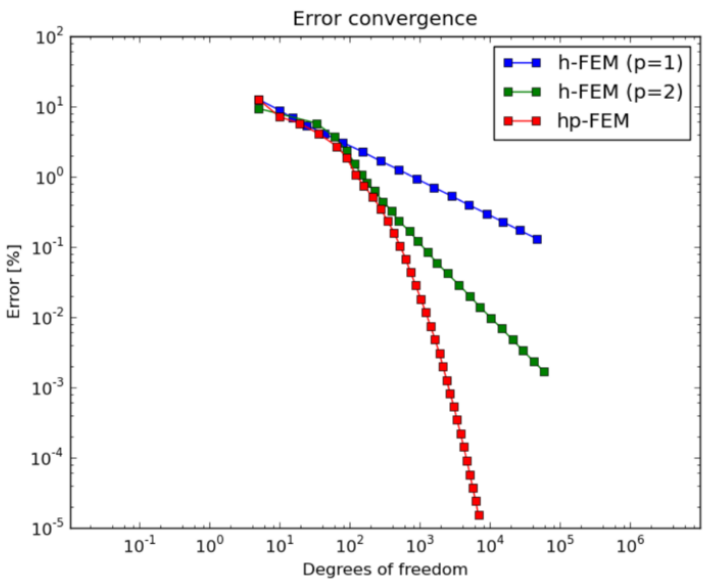
Finite element approximation of elliptic problems

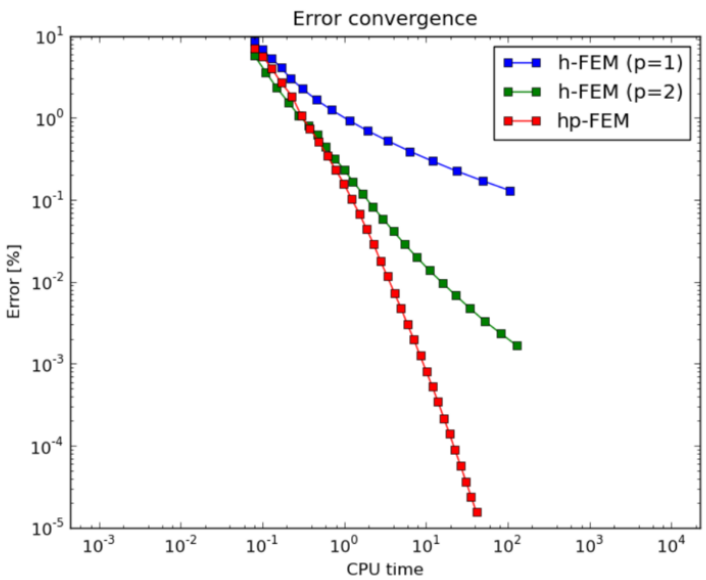
- *hp*-adaptive approximation for an elliptic problem in the L-shape domain
- subsequent figures show adapted meshes with increasing magnification, up to 100000000, colours represent the order of approximation p, from 1 (blue) to 6 (pink)



Finite element approximation of elliptic problems

- The justification for using *hp*-adaptivity is its best convergence rate
  - while standard (even higher order) *h*-adaptive FEM converges algebraically, *hp*-adaptive FEM has exponential convergence rates
- The main problems of *hp*-adaptivity are:
  - adaptive strategies – the selection which of the two options apply for a given element
  - complex coding
  - the limited number of problems for which *hp*-adaptivity can show its full potential





Finite element approximation of elliptic problems

Advanced topics:

- Error estimates - different techniques
  - Zienkiewicz-Zhu and derivative recovery
- Adaptive strategies:
  - equidistribution of errors
  - *hp* - adaptivity
- Special strategies for special types of problems
  - anisotropic mesh refinements