FEM and Sparse Linear System Solving

Lecture 3, October 6, 2017: The finite element method in 2D.

http://people.inf.ethz.ch/arbenz/FEM17

Peter Arbenz
Computer Science Department, ETH Zürich
E-mail: arbenz@inf.ethz.ch

- The finite element method
  - Introduction, model problems.
  - 1D problems. Piecewise polynomials in 1D.
  - 2D problems. Triangulations. Piecewise polynomials in 2D.
  - Variational formulations. Galerkin finite element method.
  - Implementation aspects.
- Direct solvers for sparse systems
  - LU and Cholesky decomposition
  - Sparse matrices
  - Fill-reducing orderings
- Iterative solvers for sparse systems
  - Stationary iterative methods, preconditioning
  - Preconditioned conjugate gradient method (PCG)
  - Incomplete factorization preconditioning
  - Multigrid preconditioning
  - Nonsymmetric problems (GMRES, BiCGstab)
  - Indefinite problems (SYMMLQ, MINRES)

Goal: To develop finite element methods for numerical solution of partial differential equations in 2D.

- Variational formulation: Rewrite the equation in variational form
- Finite element approximation: Seek an approximate solution in the space of continuous piecewise linear functions
- Derivation of a linear system of equations
- Basic algorithm to compute the finite element solution
- Types of elements in 2D (triangles, bilinear, biquadratic, bilinear isoparametric, quadratic curved elements)

Weak formulation: motivation

- A sufficiently smooth function u satisfying

$$-\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$
$$u(\boldsymbol{x}) = g_D(\boldsymbol{x}),\ \boldsymbol{x} \in \Gamma_D, \quad \frac{\partial u(\boldsymbol{x})}{\partial n} = g_N(\boldsymbol{x}),\ \boldsymbol{x} \in \Gamma_N. \qquad (*)$$

  is known as *classical solution* of the boundary value problem.
- If boundary is not smooth or source function is discontinous then solution u may not be smooth enough to be considered classical.
- For such problems (that are perfectly reasonable) an alternative mathematical description is required.
- Since this alternative description is less stringent in terms of admissible data, it is called a weak formulation.

A tool: Green's formulae

Let $u(\boldsymbol{x})$ and $v(\boldsymbol{x})$ be a scalar differentiable function. Then

$$\int_\Omega \frac{\partial u(\boldsymbol{x})}{\partial x_i} v(\boldsymbol{x}) d\boldsymbol{x} + \int_\Omega u(\boldsymbol{x}) \frac{\partial v(\boldsymbol{x})}{\partial x_i} d\boldsymbol{x} = \int_{\partial\Omega} u(\boldsymbol{x}) v(\boldsymbol{x}) n_i(\boldsymbol{x}) d\boldsymbol{s}.$$

If $\boldsymbol{v}(\boldsymbol{x})$ is a vector function with differentiable components $v_i(\boldsymbol{x})$,

$$\int_\Omega \mathbf{grad}\, u(\boldsymbol{x}) \cdot \boldsymbol{v}(\boldsymbol{x}) d\boldsymbol{x} = -\int_\Omega u(\boldsymbol{x}) \,\mathrm{div}\, \boldsymbol{v}(\boldsymbol{x}) d\boldsymbol{x} + \int_{\partial\Omega} u(\boldsymbol{x})\, \boldsymbol{v}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) d\boldsymbol{s}$$

If $u(\boldsymbol{x}), v(\boldsymbol{x})$ are sufficiently smooth functions, then ($\boldsymbol{v} = \mathbf{grad}\, v$)

$$-\int_\Omega \Delta v\ u\ d\boldsymbol{x} = \int_\Omega \mathbf{grad}\, v \cdot \mathbf{grad}\, u\ d\boldsymbol{x} - \int_{\partial\Omega} u(\boldsymbol{x}) \underbrace{\mathbf{grad}\, v \cdot \boldsymbol{n}}_{\frac{\partial v}{\partial n}}\ d\boldsymbol{s}$$

Weak formulation: transformation

- We require that for an appropriate set of test functions v,

$$\int_{\Omega} (\Delta u + f) v \, d\boldsymbol{x} = 0, \quad \text{for all } v. \tag{1}$$

- If u is a classical solution, then (1) evidently holds.
- If v is sufficiently smooth then we can rewrite (1) using Green's formula (divergence theorem):

$$\int_{\Omega} \textbf{grad}\, u \cdot \textbf{grad}\, v \, d\boldsymbol{x} = \int_{\Omega} f v \, d\boldsymbol{x} + \int_{\partial\Omega} v \frac{\partial u}{\partial n} \, d\boldsymbol{s} \tag{2}$$

  for all suitable test functions v.
- Problem (2) may have a solution u called a weak solution that is not smooth enough to be a classical solution. If a classical solution exists then (2) is equivalent to (*) and the weak solution is classical.

Variational formulation: L_2 spaces

- What functions can we choose for u and v?
- There are a number of integrals that have to be evaluated.
- The space of square-integrable functions is denoted by

$$L_2(\Omega) = \left\{ u : \Omega \to \mathbb{R} \middle| \int_\Omega |u(\boldsymbol{x})|^2 < \infty \right\}.$$

- Since (by means of the Cauchy–Schwarz inequality)

$$\int_\Omega \mathbf{grad}\, u \cdot \mathbf{grad}\, v \, d\boldsymbol{x} \leq \left(\int_\Omega |\mathbf{grad}\, u|^2 \, d\boldsymbol{x} \right)^{\frac{1}{2}} \left(\int_\Omega |\mathbf{grad}\, v|^2 \, d\boldsymbol{x} \right)^{\frac{1}{2}}$$

  we need to consider functions whose first derivatives are in $L_2(\Omega)$. This leads to spaces called Sobolev spaces.

Variational formulation: Sobolev spaces

- The space of square-integrable functions that have also square-integrable first deriatives is denoted by

$$\mathcal{H}^1(\Omega) = \left\{ u : \Omega \to \mathbb{R} \,\middle|\, u, \frac{\partial u}{\partial x_i} \in L_2(\Omega) \right\}.$$

- We need the following modification of $\mathcal{H}^1(\Omega)$ that take into account the boundary conditions

$$\mathcal{H}^1_E(\Omega) = \left\{ u \in \mathcal{H}^1(\Omega) \,\middle|\, u = g \text{ on } \partial\Omega_D \right\}, \quad (\text{Solution space})$$

$$\mathcal{H}^1_{E_0}(\Omega) = \left\{ v \in \mathcal{H}^1(\Omega) \,\middle|\, v = 0 \text{ on } \partial\Omega_D \right\}. \quad (\text{Test space})$$

Variational formulation of the Poisson problem

We can now state the weak form of the Poisson problem:

Find $u \in \mathcal{H}^1_E(\Omega)$ such that

$$\int_\Omega \mathbf{grad}\, u \cdot \mathbf{grad}\, v \; d\boldsymbol{x} = \int_\Omega vf \; d\boldsymbol{x} + \int_{\partial\Omega_N} v\, g_N \; d\boldsymbol{s} \quad \text{for all } v \in \mathcal{H}^1_{E_0}(\Omega) \tag{3}$$

The function u that is to be found has to satisfy the *Dirichlet* ("essential") boundary conditions. The Neumann ("natural") boundary conditions are enforced by the last term in (3).
The *test functions* have to satisfy the homogeneous Dirichlet boundary conditions only.

The (conforming) Galerkin finite element method I

- The idea of Galerkin is to approximate the solution u of (3) from a finite-dimensional subspace of $\mathcal{H}^1_E(\Omega)$.
- We construct an n-dimensional vector space of *test functions* $S^h_0 = \text{span}\{\varphi_1, \ldots, \varphi_n\} \subset \mathcal{H}^1_{E_0}$.
- To ensure that the Dirichlet boundary conditions are satisfied, we extend this basis set by additional functions $\varphi_{n+1}, \ldots, \varphi_{n+n_\partial}$ and choose coefficients ξ_j such that $\sum_{j=n+1}^{n+n_\partial} \xi_j \varphi_j$ (*e.g.*) interpolates the boundary data g_D on Γ_D.
- The approximation $u_h \in S^h_E$ is then

$$u_h = \sum_{j=1}^{n} \xi_j \varphi_j + \sum_{j=n+1}^{n+n_\partial} \xi_j \varphi_j,$$

where the parameters ξ_j in the first sum can be chosen freely.

Galerkin finite element method II

- Galerkin approximation is a finite-dimensional version of the weak formulation:

  Find $u_h \in S_E^h$ such that

  $$\int_\Omega \mathbf{grad}\, u_h \cdot \mathbf{grad}\, v_h \, d\boldsymbol{x} = \int_\Omega v_h f \, d\boldsymbol{x} + \int_{\partial\Omega_N} v_h \, g_N \, d\boldsymbol{s} \quad \text{for all } v_h \in S_0^h. \tag{4}$$

- S_E^h is the space of continuous piecewise linear polynomials on K. $S_0^h \subset S_E^h$ is the subspace that satisfies the boundary conditions,

  $$S_0^h = \{v \in S^h : v = 0 \text{ on } \partial\,\Omega_D\}$$

- This gives us n equations for the n unknown parameters $\xi_1, \ldots, \xi_n$. (Note again that the parameters $\xi_{n+1}, \ldots, \xi_{n+n_\partial}$ are determined by the Dirichlet boundary conditions.)

Galerkin finite element method III

- Since $\{\varphi_1, \ldots, \varphi_n\}$ is a basis of S_0^h we can replace the term "for all $v_h \in S_0^h$" by "for $\varphi_1, \ldots, \varphi_n$".
- The equation (4) thus becomes:
  Find coefficients $\xi_1, \ldots, \xi_n$ such that

$$\sum_{j=1}^{n} \xi_j \int_\Omega \textbf{grad}\, \varphi_j \cdot \textbf{grad}\, \varphi_i \; d\boldsymbol{x} = \int_\Omega \varphi_i f \; d\boldsymbol{x} + \int_{\partial\Omega_N} \varphi_i \, g_N \; d\boldsymbol{s} - \sum_{j=n+1}^{n+n_\partial} \int_\Omega g_D(x_j) \textbf{grad}\, \varphi_j \cdot \textbf{grad}\, \varphi_i \; d\boldsymbol{x} \qquad (5)$$

for $i = 1, \ldots, n$.

Galerkin finite element method IV

We can write this as a matrix equation

$$\mathbf{A}\,\xi = \boldsymbol{b}. \tag{6}$$

where the elements of the matrix $\mathbf{A} = (a_{ij})$ are given by

$$a_{ij} = \int_\Omega \mathbf{grad}\,\varphi_j \cdot \mathbf{grad}\,\varphi_i$$

and the elements of the right-hand side (rhs) $\boldsymbol{b}$ are

$$b_i = \int_\Omega \varphi_i f \; d\boldsymbol{x} + \int_{\partial\Omega_N} \varphi_i\, g_N \; d\boldsymbol{s} - \sum_{j=n+1}^{n+n_\partial} \xi_j \int_\Omega \mathbf{grad}\,\varphi_j \cdot \mathbf{grad}\,\varphi_i \; d\boldsymbol{x}.$$

- There are three types of integrals that have to be computed.
- The elements of the so-called *stiffness matrix*

$$a_{ij} = \int_\Omega \mathbf{grad}\,\varphi_j \cdot \mathbf{grad}\,\varphi_i \,d\boldsymbol{x}. \tag{7}$$

  The Dirichlet boundary portion of the right-hand side requires the same type of integrals.
- Force term of the right side. Let us write $f \approx \sum_{j=1}^{n} f(\boldsymbol{x}_j)\varphi_j$. Then we need to compute integrals of the kind

$$\int_\Omega \varphi_i \varphi_j \,d\boldsymbol{x}. \tag{8}$$

- Neumann boundary portion of the right-hand side

$$\int_{\partial\Omega_N} \varphi_i\, g_N \,d\boldsymbol{s}. \tag{9}$$

Properties of the Stiffness Matrix

- The stiffness matrix **A** is symmetric and positive definite.
  A is symmetric: $A_{ij} = A_{ji}$
  A is positive definite: $\mathbf{v}^T \mathbf{A} \mathbf{v} > 0$ for all nonzero n-vectors **v**
- The condition number of the stiffness matrix **A** satisfies the estimate

$$\kappa(\mathbf{A}) := ||\mathbf{A}|| ||\mathbf{A}^{-1}|| \le Ch^{-2}.$$

Galerkin finite element method V

The matrix **A** is symmetric and positive-definite:

$$\begin{aligned}
\boldsymbol{v}^T \mathbf{A} \boldsymbol{v} &= \sum_{j=1}^{n} \sum_{i=1}^{n} v_j a_{ji} v_i \\
&= \sum_{j=1}^{n} \sum_{i=1}^{n} v_j \left(\int_\Omega \mathbf{grad}\, \varphi_j \cdot \mathbf{grad}\, \varphi_i \, d\boldsymbol{x} \right) v_i \\
&= \int_\Omega \left(\sum_{j=1}^{n} v_j \mathbf{grad}\, \varphi_j \right) \cdot \left(\sum_{i=1}^{n} v_i \mathbf{grad}\, \varphi_i \right) \, d\boldsymbol{x} \\
&= \int_\Omega \mathbf{grad}\, v_h \cdot \mathbf{grad}\, v_h \geq 0.
\end{aligned}$$

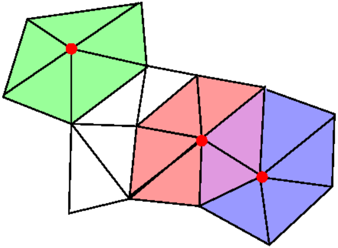
$\mathbf{grad}\, v_h = \mathbf{0}$ implies $v_h = 0$ since $\partial\Omega_D \neq \emptyset$.

Galerkin finite element method

- NOTATION: The inclusions $S_E^h \subset \mathcal{H}_E^1$ and $S_0^h \subset \mathcal{H}_{E_0}^1$ lead to *conforming* approximations.
- φ_i should be convenient to manipulate, easily adaptable to complicated geometries, and have good approximation properties, i.e. the error $\|u - u_h\|$ should reduce rapidly as the problem dimension n is increased.
- *Piecewise polynomials* satisfy these requirements.
- REMARK: Special domains like intervals, squares, cubes, etc. or, circles, balls, etc. admit much more powerful search spaces.

Support of the basis functions φ

- The *support* of a function is the domain where the function is nonzero (patch).
- For the piecewise linear basis functions φ_i this is the patch of triangles that have $\boldsymbol{x}_i$ as one of their vertices.
- The integrals in (7) and (8) are nonzero only if the supports of φ_i and φ_j overlap.



- *Most of the elements of the system matrix* $\mathbf{A}$ *are zero!*

Elementwise integration

- The key idea is to do the integrals *elementwise*:

$$\int_\Omega \varphi_j \cdot \varphi_i \, d\mathbf{x} = \sum_{\triangle_k \in \mathcal{K}} \int_{\triangle_k} \varphi_j \cdot \varphi_i \, d\mathbf{x},$$
$$\int_\Omega \mathbf{grad}\, \varphi_j \cdot \mathbf{grad}\, \varphi_i \, d\mathbf{x} = \sum_{\triangle_k \in \mathcal{K}} \int_{\triangle_k} \mathbf{grad}\, \varphi_j \cdot \mathbf{grad}\, \varphi_i \, d\mathbf{x} \tag{10}$$

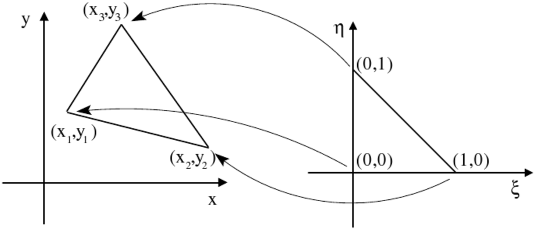
- The integral on the left is obtained by assembling the elementwise portions on the right.

- In a single *linear* element $\triangle_k$ there are three *element basis functions* $\varphi_{k,1}, \varphi_{k,2}, \varphi_{k,3}$. The function $\varphi_{k,i}$ is one at (local) node i and zero at the two other nodes.
  (We will later need a map of the local to the global node numbers.)
- Any global basis function φ_j is composed of element basis functions.
- Therefore, finally, we need to compute the integrals

$$\int_{\triangle_k} \varphi_{k,j} \cdot \varphi_{k,i} \, d\mathbf{x} \tag{11}$$

$$\int_{\triangle_k} \mathbf{grad}\, \varphi_{k,j} \cdot \mathbf{grad}\, \varphi_{k,i} \, d\mathbf{x} \tag{12}$$

- The first step in the computation of the element stiffness matrix $\mathbf{A}_k$ is to define a map from the *reference element* $\triangle_*$ onto the given element $\triangle_k$.



- The transformation is determined by the coordinates of the actual element $\triangle_k$. So, it is different for all elements.
- The transformation must be differentiable.

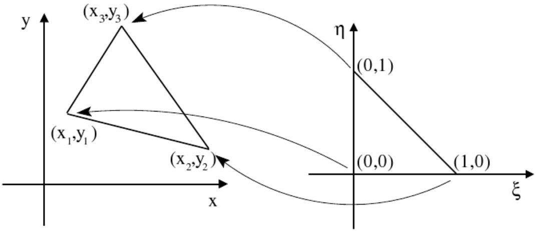
REMARK: This and some more pictures are taken from the book by Elman, Silvester, & Wathen.

For straight sided triangles the local-global mapping is defined for all points $(x, y) \in \triangle_k$. It is given by

$$x(\xi, \eta) = x_1 \chi_1(\xi, \eta) + x_2 \chi_2(\xi, \eta) + x_3 \chi_3(\xi, \eta)$$
$$y(\xi, \eta) = y_1 \chi_1(\xi, \eta) + y_2 \chi_2(\xi, \eta) + y_3 \chi_3(\xi, \eta)$$

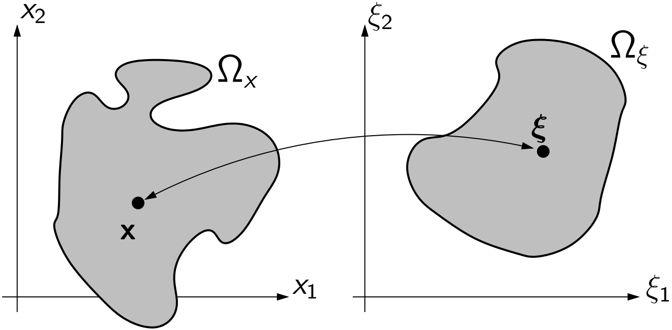
where

$$\chi_1(\xi, \eta) = 1 - \xi - \eta$$
$$\chi_2(\xi, \eta) = \xi$$
$$\chi_3(\xi, \eta) = \eta$$



Tools from analysis

For a given invertible map $\Omega_\xi \ni \boldsymbol{\xi} = (\xi, \eta) \longrightarrow (x, y) = \boldsymbol{x} \in \Omega_x$ from a domain Ω_ξ to a domain Ω_x.



Tools from analysis (cont.)

Let $\tilde{f}(\boldsymbol{\xi})$ be a function on Ω_ξ. We define a new function $f(\boldsymbol{x})$ by

$$f(\boldsymbol{x}) := \tilde{f}(\boldsymbol{\xi}(\boldsymbol{x})).$$

f and $\tilde{f}$ have the same value at corresponding points $\boldsymbol{\xi}$ and $\boldsymbol{x}$.
For the integrals we have

$$\boxed{\int_{\Omega_x} f(\boldsymbol{x})d\boldsymbol{x} = \int_{\Omega_\xi} \tilde{f}(\boldsymbol{\xi})\,|J_{\boldsymbol{x}}|(\boldsymbol{\xi})\,d\boldsymbol{\xi},} \tag{13}$$

where

$$J_{\boldsymbol{x}}(\boldsymbol{\xi}) = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}} = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

is the so-called functional or Jacobi matrix.

Tools from analysis (cont.)

For the transformation of the reference triangle $\triangle_* \subset \Omega_\xi$ to an arbitrary triangle $\triangle_k \subset \Omega_x$ the Jacobi matrix is

$$J_k := \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}.$$

Its determinant (the Jacobian) is given by

$$|J_k| = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2|\triangle_k|$$

The Jacobian is the quotient of the area of $\triangle_k$ and $\triangle_*$. It is a constant ($\neq 0$) since the mapping is linear (and $|\triangle_k| \neq 0$).

Tools from analysis (cont.)

- To integrate the gradients of the basis functions for the stiffness matrix **A** we transform derivatives from the original (x, y) coordinate system into the (ξ, η) coordinate system.

$$\begin{bmatrix} \frac{\partial\varphi}{\partial\xi} \\ \frac{\partial\varphi}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial\xi} & \frac{\partial y}{\partial\xi} \\ \frac{\partial x}{\partial\eta} & \frac{\partial y}{\partial\eta} \end{bmatrix} \begin{bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \end{bmatrix} = \frac{\partial(x,y)}{\partial(\xi,\eta)} \begin{bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \end{bmatrix}$$

- The inverse of this is

$$\begin{bmatrix} \frac{\partial\varphi}{\partial x} \\ \frac{\partial\varphi}{\partial y} \end{bmatrix} = \frac{\partial(\xi,\eta)}{\partial(x,y)} \begin{bmatrix} \frac{\partial\varphi}{\partial\xi} \\ \frac{\partial\varphi}{\partial\eta} \end{bmatrix} = \frac{1}{|J_k|} \begin{bmatrix} b_2 & b_3 \\ c_2 & c_3 \end{bmatrix} \begin{bmatrix} \frac{\partial\varphi}{\partial\xi} \\ \frac{\partial\varphi}{\partial\eta} \end{bmatrix}$$

with $b_2 = y_3 - y_1,\ b_3 = y_1 - y_2,\ c_2 = x_1 - x_3,\ c_3 = x_2 - x_1.$

Elementwise integration [continued]

The first integral (11) now becomes

$$\int_{\triangle_k} \varphi_{k,j}\,\varphi_{k,i}\; d\boldsymbol{x} = \int_{\triangle_k} \varphi_{k,j}\,\varphi_{k,i}\; dx\,dy$$
$$= \int_{\triangle_*} \varphi_{*,j}\,\varphi_{*,i}\,|J_k|\; d\xi\,d\eta$$

Putting all the nine integrals in one matrix gives the element mass matrix

$$M_k := \left(\int_{\triangle_k} \varphi_{k,j}\,\varphi_{k,i}\; dx\,dy\right)_{1\le i,j\le 3} = \frac{|J_k|}{24}\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

The second integral (12) becomes

$$
\begin{aligned}
&\int_{\triangle_k} \textbf{grad}\, \varphi_{k,j} \cdot \textbf{grad}\, \varphi_{k,i}\; dx\, dy \\
&= \int_{\triangle_k} \left(\frac{\partial \varphi_{k,j}}{\partial x} \frac{\partial \varphi_{k,i}}{\partial x} + \frac{\partial \varphi_{k,j}}{\partial y} \frac{\partial \varphi_{k,i}}{\partial y} \right)\; dx\, dy \\
&= \frac{1}{|J_k|^2} \int_{\triangle_*} \left(b_2 \frac{\partial \varphi_{*,j}}{\partial \xi} + b_3 \frac{\partial \varphi_{*,j}}{\partial \eta} \right) \left(b_2 \frac{\partial \varphi_{*,i}}{\partial \xi} + b_3 \frac{\partial \varphi_{*,i}}{\partial \eta} \right) |J_k|\; d\xi\, d\eta \\
&\quad + \frac{1}{|J_k|^2} \int_{\triangle_*} \left(c_2 \frac{\partial \varphi_{*,j}}{\partial \xi} + c_3 \frac{\partial \varphi_{*,j}}{\partial \eta} \right) \left(c_2 \frac{\partial \varphi_{*,i}}{\partial \xi} + c_3 \frac{\partial \varphi_{*,i}}{\partial \eta} \right) |J_k|\; d\xi\, d\eta
\end{aligned}
$$

Collecting terms gives

$$\frac{1}{|J_k|}\int_{\triangle_*}\left(b_2\frac{\partial\varphi_{*,j}}{\partial\xi}+b_3\frac{\partial\varphi_{*,j}}{\partial\eta}\right)\left(b_2\frac{\partial\varphi_{*,i}}{\partial\xi}+b_3\frac{\partial\varphi_{*,i}}{\partial\eta}\right)\,d\xi\,d\eta$$

$$+\frac{1}{|J_k|}\int_{\triangle_*}\left(c_2\frac{\partial\varphi_{*,j}}{\partial\xi}+c_3\frac{\partial\varphi_{*,j}}{\partial\eta}\right)\left(c_2\frac{\partial\varphi_{*,i}}{\partial\xi}+c_3\frac{\partial\varphi_{*,i}}{\partial\eta}\right)\,d\xi\,d\eta$$

$$=\frac{1}{|J_k|}\int_{\triangle_*}\left((b_2^2+c_2^2)\frac{\partial\varphi_{*,j}}{\partial\xi}\frac{\partial\varphi_{*,i}}{\partial\xi}+(b_2b_3+c_2c_3)\frac{\partial\varphi_{*,j}}{\partial\xi}\frac{\partial\varphi_{*,i}}{\partial\eta}\right.$$

$$\left.+(b_2b_3+c_2c_3)\frac{\partial\varphi_{*,j}}{\partial\eta}\frac{\partial\varphi_{*,i}}{\partial\xi}+(b_3^2+c_3^2)\frac{\partial\varphi_{*,j}}{\partial\eta}\frac{\partial\varphi_{*,i}}{\partial\eta}\right)\,d\xi\,d\eta$$

For *linear* elements we have $\varphi_{*,j} = \chi_j$. So,

$$\begin{bmatrix} \frac{\partial \varphi_{*,1}}{\partial \xi} \\ \frac{\partial \varphi_{*,2}}{\partial \xi} \\ \frac{\partial \varphi_{*,3}}{\partial \xi} \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} \frac{\partial \varphi_{*,1}}{\partial \eta} \\ \frac{\partial \varphi_{*,2}}{\partial \eta} \\ \frac{\partial \varphi_{*,3}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix},$$

and the element stiffness matrix becomes

$$A_k := \left(\int_{\triangle_k} \mathbf{grad}\, \varphi_{k,j} \cdot \mathbf{grad}\, \varphi_{k,i} \; d\boldsymbol{x} \right)_{1 \le j,i \le 3} = \frac{b_2^2 + c_2^2}{2|J_k|} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$+ \frac{b_2 b_3 + c_2 c_3}{2|J_k|} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} + \frac{b_3^2 + c_3^2}{2|J_k|} \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

REMARK: The factor 1/2 is from $\int_{\triangle_*} d\xi d\eta$.

Defining $b_1 = y_2 - y_3$ and $c_1 = x_3 - x_2$ and observing that (see slide 26) $b_1 + b_2 + b_3 = c_1 + c_2 + c_3 = 0$ we find that

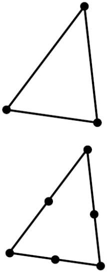
$$\begin{aligned} A_k &= \left(\int_{\triangle_k} \textbf{grad}\, \varphi_{k,j} \cdot \textbf{grad}\, \varphi_{k,i} \; d\boldsymbol{x} \right)_{1 \le j,i \le 3} \\ &= \frac{1}{2|J_k|} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix} \\ &= \frac{1}{2|J_k|} [\boldsymbol{b}\boldsymbol{b}^T + \boldsymbol{c}\boldsymbol{c}^T] \end{aligned}$$

where $\boldsymbol{b} = [b_1, b_2, b_3]^T$ and $\boldsymbol{c} = [c_1, c_2, c_3]^T$.

NOTE: This is the element matrix for the Poisson equation. We will consider the case with variable coefficients soon.

Other 2D elements: quadratic triangles

The underlying domain must be partitioned into elements, such as linear triangles quadratic triangles,



In contrast to the linear triangle (also called $\boldsymbol{P}_1$ element) on the top, the quadratic triangle (or $\boldsymbol{P}_2$ element) on the bottom has 6 degrees of freedom. The ansatz for bilinear functions in the reference triangle is

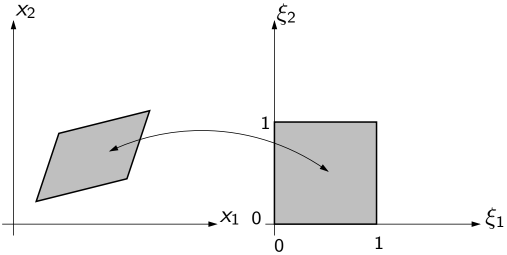
$$\varphi(\xi, \eta) = a + b\,\xi + c\,\eta + d\,\xi^2 + e\,\xi\eta + f\,\eta^2.$$

The 6 coefficients $a, \dots, f$ are determined (usually) by the values at the 3 vertices and the 3 midpoints of the edges. Continuity among $\boldsymbol{P}_2$ elements is ensured.

The transformation of the general to the reference element is linear.

Other 2D elements: bilinear elements

On bilinear elements (square / rectangle / parallelogram) or $\boldsymbol{Q}_1$ elements the ansatz for in the reference square is



$$\varphi(\xi, \eta) = a + b\,\xi + c\,\eta + d\,\xi\eta.$$

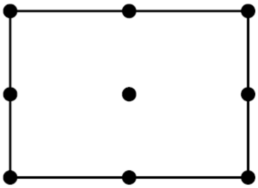
The 4 dofs $a, \ldots, d$ of φ are determined by the 4 values at the corners of the element.

The transformation of the general to the reference element is the same as with linear triangles.

Other 2D elements: biquadratic elements

The biquadratic or $\boldsymbol{Q}_2$ element has 9 degrees of freedom. The ansatz is

$$\varphi(\xi,\eta) = a_0 + a_1\xi + a_2\eta + a_3\xi^2 + a_4\xi\eta + a_5\eta^2 + a_6\xi^2\eta + a_7\xi\eta^2 + a_8\xi^2\eta^2$$

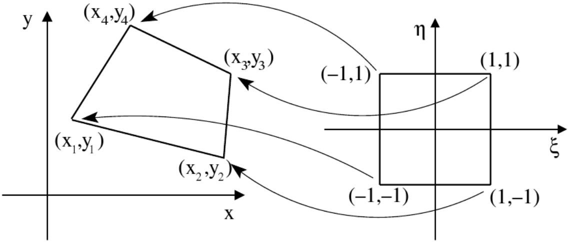


The last term together with the element midpoint can be omitted.

Note that linear triangles can be used together with bilinear elements since all functions are linear on the edges.
The same holds for $\boldsymbol{P}_2$ and $\boldsymbol{Q}_2$ elements.

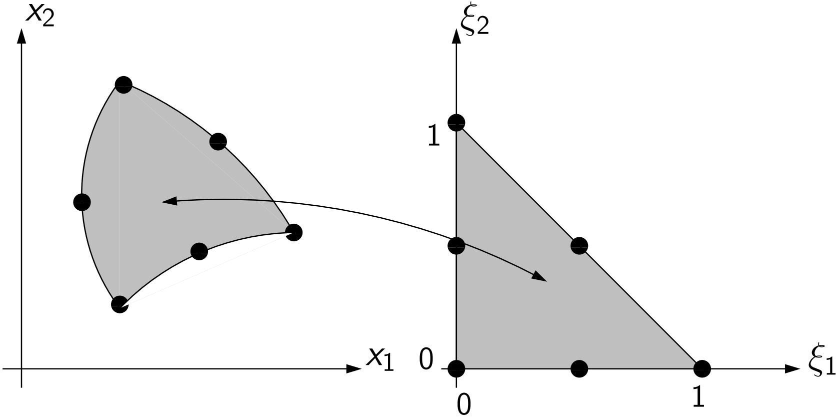
In 3D this is not the case!!

Other 2D elements: bilinear isoparametric elements



In bilinear isoparametric elements the ansatz for the basis functions is piecewise bilinear in the reference square. The map from the reference element to the actual element is bilinear as the ansatz for the functions (therefore the notion 'isoparametric').
Note that the Jacobian $|J_k|$ is not constant anymore! Therefore, the computation of the integrals is more complicated and Gaussian quadrature is not exact anymore.

Other types of elements: quadratic curved elements



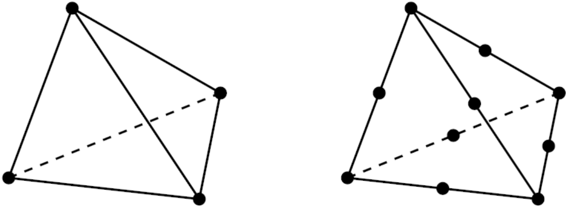
Other types of elements: quadratic curved elements (cont.)

Quadratic curved elements use a quadratic ansatz for the functions as well as for the mapping from the reference element to the actual element.

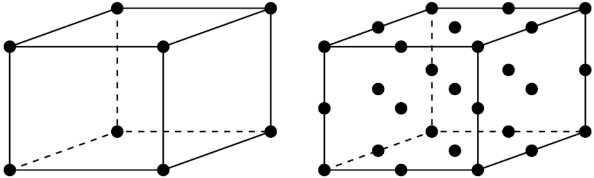
$$\varphi(\xi,\eta) = a_{00} + a_{10}\,\xi + a_{01}\,\eta + a_{20}\,\xi^2 + a_{11}\,\xi\eta + a_{02}\,\eta^2,$$
$$x(\xi,\eta) = x_{00} + x_{10}\,\xi + x_{01}\,\eta + x_{20}\,\xi^2 + x_{11}\,\xi\eta + x_{02}\,\eta^2,$$
$$y(\xi,\eta) = y_{00} + y_{10}\,\xi + y_{01}\,\eta + y_{20}\,\xi^2 + y_{11}\,\xi\eta + y_{02}\,\eta^2,$$

The midpoint of the edges can be used to better approximate curved boundaries.
The Jacobian is not constant. Therefore, care has to be taken to evaluate matrix elements accurately.

3D elements: tetrahedra, brick elements



Representations of $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ tetrahedral elements.



Representations of $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ brick elements.

Example: A model problem with variable coefficients

Consider the model problem, find u such that

$$-\text{div}\,(a(\boldsymbol{x})\,\textbf{grad}\,u(\boldsymbol{x})) = f(\boldsymbol{x}), \quad \text{in } \Omega,$$
$$-\boldsymbol{n}\cdot(a\,\textbf{grad}\,u) = \kappa(u - g_D) - g_N \quad \text{on } \partial\Omega$$

$a(\boldsymbol{x}) > 0, f(\boldsymbol{x}), \kappa > 0, g_D$ and g_N are given. A solution of this problem is in space $V = \mathcal{H}^1(\Omega)$. (We do not have a Dirichlet boundary, but think how you could enforce it in this setting.)

Step 1: Obtain the variational formulation.

$$\int_\Omega fv dx = \int_\Omega -\text{div}\,(a\,\textbf{grad}\,u)v\,dx$$
$$= \int_\Omega a\,\textbf{grad}\,u\cdot\textbf{grad}\,v\,dx - \int_{\partial\Omega}\boldsymbol{n}\cdot(a\,\textbf{grad}\,u)v\,ds$$
$$= \int_\Omega a\,\textbf{grad}\,u\cdot\textbf{grad}\,v\,dx + \int_{\partial\Omega}(\kappa(u - g_D) - g_N)v\,ds \quad \forall v \in V$$

Find $u \in V$ such that

$$\int_\Omega a\, \mathbf{grad}\, u \cdot \mathbf{grad}\, v\, dx + \int_{\partial\Omega} \kappa u v\, ds = \int_\Omega f v\, dx + \int_{\partial\Omega} (\kappa g_D + g_N) v\, ds$$

for all $v \in V$.

Step 2: Replace V with FE space V_h to obtain: Find $u_h \in V_h \subset V$

$$\int_\Omega a \nabla u_h \cdot \nabla v\, dx + \int_{\partial\Omega} \kappa u_h v\, ds = \int_\Omega f v\, dx + \int_{\partial\Omega} (\kappa g_D + g_N) v\, ds \tag{14}$$

holds $\forall v \in V_h$.

Remember: $V = \mathcal{H}^1(\Omega) = \left\{ v \in L^2(\Omega) \ : \ ||\mathbf{grad}\, v||_{L^2(\Omega)} < \infty \right\}$

Basic Algorithm to Compute the Finite Element Solution

The linear system for eq. (14) can be written as

$$(\mathbf{A} + \mathbf{R})\xi = \boldsymbol{b} + \boldsymbol{r}.$$

where the entries of matrices $\mathbf{A}$ and $\mathbf{R}$ and vectors $\boldsymbol{b}$ and $\boldsymbol{r}$ are:

$$A_{ij} = \int_\Omega a \,\mathbf{grad}\, \varphi_i \cdot \mathbf{grad}\, \varphi_j \, dx$$

$$R_{ij} = \int_{\partial\Omega} \kappa \, \varphi_i \varphi_j \, ds$$

$$b_i = \int_\Omega f \varphi_i \, dx$$

$$r_i = \int_{\partial\Omega} (\kappa g_D + g_N) \varphi_i \, ds$$

where $1 \leq i, j \leq n$. n is the number of nodes.

Assembly of the stiffness matrix

$$
\begin{aligned}
A_k &= \left(\int_{\triangle_k} a(\boldsymbol{x})\, \mathbf{grad}\, \varphi_{k,j} \cdot \mathbf{grad}\, \varphi_{k,i}\ d\boldsymbol{x} \right)_{1 \le j,i \le 3} \\
&\approx \left(\bar{a} \int_{\triangle_k} \mathbf{grad}\, \varphi_{k,j} \cdot \mathbf{grad}\, \varphi_{k,i};\, d\boldsymbol{x} \right)_{1 \le j,i \le 3} \\
&= \frac{\bar{a}}{2|J_k|} \begin{bmatrix} b_1^2 + c_1^2 & b_1 b_2 + c_1 c_2 & b_1 b_3 + c_1 c_3 \\ b_2 b_1 + c_2 c_1 & b_2^2 + c_2^2 & b_2 b_3 + c_2 c_3 \\ b_3 b_1 + c_3 c_1 & b_3 b_2 + c_3 c_2 & b_3^2 + c_3^2 \end{bmatrix} \\
&= \frac{\bar{a}}{2|J_k|} [\boldsymbol{b}\boldsymbol{b}^T + \boldsymbol{c}\boldsymbol{c}^T]
\end{aligned}
$$

where $\bar{a}$ is the value of $a(\boldsymbol{x})$ at the element's center of mass.

```
function A = StiffnessAssembler2D(p,t,a)
np = size(p,2);
nt = size(t,2);
A = sparse(np,np); % allocate stiffness matrix
for K = 1:nt
  loc2glb = t(1:3,K); % local-to-global map
  x = p(1,loc2glb); % node x-coordinates
  y = p(2,loc2glb); % node y-coordinates
  [area,b,c] = HatGradients(x,y);

  xc = mean(x); yc = mean(y); % element centroid
  abar = a(xc,yc); % value of a(x,y) at centroid

  AK = abar*(b*b' + c*c')*area; % element stiffness matrix

  A(loc2glb,loc2glb) = A(loc2glb,loc2glb) ...
                        + AK; % add element stiffnesses to A
end
```

Assembling the boundary conditions

R: The boundary matrix, $\boldsymbol{r}$: the boundary vector
N_i and N_j: two nodes on the boundary $\partial\Omega$; E: edge between them.
For simplicity, assume κ, g_D, g_N are constant on E.

$$R_{ij}^E = \int_E \kappa\varphi_i\varphi_j ds = \frac{1}{6}\kappa(1+\delta_{ij})|E|, \quad i,j=1,2$$

$$r_i^E = \int_E (\kappa g_D + g_N)\varphi_i ds = \frac{1}{2}(\kappa g_D + g_N)|E|$$

```
function [R,r] = RobinAssembler2D(p,e,kappa,gD,gN)
R = RobinMassMatrix2D(p,e,kappa);
r = RobinLoadVector2D(p,e,kappa,gD,gN);
```

where `[p,e,t]=initmesh(g)` and `g` is the geometry.

```
function R = RobinMassMatrix2D(p,e,kappa)
np = size(p,2); % number of nodes
ne = size(e,2); % number of boundary edges
R = sparse(np,np); % allocate boundary matrix

for E = 1:ne
  loc2glb = e(1:2,E); % boundary nodes
  x = p(1,loc2glb); % node x-coordinates
  y = p(2,loc2glb); % node y-
  len = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2); % edge length
  xc = mean(x); yc = mean(y); % edge mid-point
  k = kappa(xc,yc); % value of kappa at mid-point
  RE = k/6*[2 1; 1 2]*len; % edge boundary matrix
  R(loc2glb,loc2glb) = R(loc2glb,loc2glb) + RE;
end
```

```
function r = RobinLoadVector2D(p,e,kappa,gD,gN)
np = size(p,2);
ne = size(e,2);
r = zeros(np,1);

for E = 1:ne
  loc2glb = e(1:2,E);
  x = p(1,loc2glb);
  y = p(2,loc2glb);
  len = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2);
  xc = mean(x); yc = mean(y);
  tmp = kappa(xc,yc)*gD(xc,yc)+gN(xc,yc);
  rE = tmp*[1; 1]*len/2;
  r(loc2glb) = r(loc2glb) + rE;
end
```

Exercise 3

http://people.inf.ethz.ch/arbenz/FEM17/exercises/ex3.pdf