PDE Project Course

1. Theory of finite element methods

Johan Jansson

johanjan@math.chalmers.se

Department of Computational Mathematics

Goal of the lecture

- Refresh or give knowledge and notation of theoretical aspects of finite element methods which are relevant for carrying out the projects.

Lecture plan

- Introduction to FEM
- FEM for Poisson’s equation
- Adaptivity for Poisson’s equation

Introduction to FEM

A method for solving PDEs

The finite element method (FEM), also known as Galerkin's method, is a general method for solving PDEs (or ODEs) of the form

$$A(u(x)) = f(x), \; x \in \Omega$$

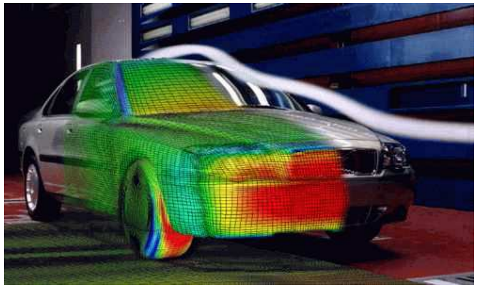
where A is a differential operator, f is a given force term, u is the solution and Ω is the domain.

Solving PDEs

- Analytic solutions can be obtained only for simple geometries in special cases:

$$-\Delta u = 0$$

- Using the computer, we can obtain solutions to general problems with complex geometries:



$$\begin{aligned} \dot{u} + u \cdot \nabla u - \nu \Delta u + \nabla p &= f \\ \nabla \cdot u &= 0 \end{aligned}$$

The finite element method

Find an approximate solution U of the form

$$U(x) = \sum_{j=1}^{N} \xi_j \varphi_j(x).$$

Here U is linear linear combination of (a finite number of) basis functions with local support:

$$\{\varphi_j\}_{j=1}^{N}.$$

Basis functions

- Typically piecewise linear nodal basis functions. Other bases also possible.
- We define a mesh $T_h = \{K\}$ “triangulating” the domain Ω into elements K (intervals, triangles, tetrahedrons, ...), where the vertices of the elements form nodes N. h is a function defined by $h(x) = h_K$, where h_K is the diameter of the ball circumscribing K.

Basis functions

- Nodal basis function defined by:

$$\phi_j(N_i) = \begin{cases} 1 & , i = j, \\ 0 & , i \neq j \end{cases}$$

Linear between nodes.

Basis functions 1D

- Local basis functions on the *reference element* (an interval): $K_r = [0, 1]$
- Local basis functions:

$$\begin{aligned}\phi_0(x) &= 1 - x \\ \phi_1(x) &= x\end{aligned}$$

- The global basis function $\phi_j(x)$ on any element K can be computed via a *mapping* $F : \Omega_0 \rightarrow \Omega$. $\phi_j(x)$ on Ω is the union of $\phi_j(x)$ on all K.

Basis functions 2D

- Local basis functions on the *reference element* (a triangle): $K_r = \{(0,0),(1,0),(0,1)\}$
- Local basis functions:

$$\begin{aligned} \phi_0(x) &= 1 - x - y \\ \phi_1(x) &= x \\ \phi_2(x) &= y \end{aligned}$$

Some notation from functional analysis

- Scalar product for functions v, w:

$$(v, w) = \int_\Omega v(x)w(x)\;dx$$

- $L_2(\Omega)$-norm of a function v:

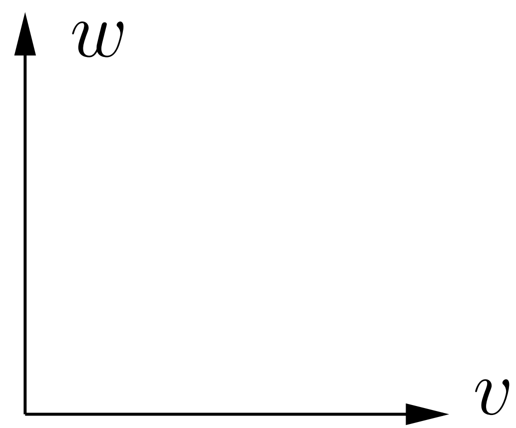
$$\|v\|_{L_2(\Omega)} = \left(\int_\Omega v^2\;dx\right)^{1/2} = \sqrt{(v, v)}$$

Some notation from functional analysis

- Cauchy's inequality:

$$|(v, w)| \leq \|v\| \, \|w\|$$

- v and w are *orthogonal* iff $(v, w) = 0$



Galerkin's method

It is often advisable to rewrite the differential equation $A(u) = f$ from *operator form* to *variational form*:

$$a(u, v) = (f, v) \quad \forall v \in V,$$

where $a(\cdot, \cdot) = (A(\cdot), \cdot)$ is a *bilinear form*, and V is a suitable function space.
Concretely:

$$\int_\Omega A(u) v \, dx = \int_\Omega f v \, dx \quad \forall v \in V$$

Galerkin's method

The finite element method is based on Galerkin's method:

- Let V_h denote a finite dimensional *trial space*.
- Let $\hat{V}_h$ denote a finite dimensional *test space*.
- Find $U \in V_h$ such that the residual $R(U) = A(U) - f$ is orthogonal to $\hat{V}_h$:

$$
\begin{aligned}
(R(U), v) &= 0 \quad \forall v \in \hat{V}_h \Leftrightarrow \\
a(U, v) - (f, v) &= 0 \quad \forall v \in \hat{V}_h
\end{aligned}
$$

Galerkin's method

Starting from the variational formulation with $V_h = \hat{V}_h = \mathrm{span}\{\varphi_j\}_{j=1}^N$ we have

$$\begin{aligned} a(U,v) - (f,v) &= 0, && \forall v \in V_h, \\ a(\textstyle\sum_{j=1}^N \xi_j \varphi_j, v) &= (f,v), && \forall v \in V_h, \\ \textstyle\sum_{j=1}^N \xi_j a(\varphi_j, \varphi_i) &= (f, \varphi_i), && i = 1, \ldots, N, \end{aligned}$$

or

$$A_h \xi = b,$$

where $A_h = a(\varphi_j, \varphi_i)$, $b = (f, \varphi_i)$.

FEM for Poisson's equation

Poisson in three different forms

- Equation:

$$-\Delta u = f$$

- Variational formulation:

$$\int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Omega f v \, dx \quad \forall v \in V$$

- Linear system:

$$A_h = \int_\Omega \nabla \varphi_j \cdot \nabla \hat{\varphi}_i \, dx, \quad b = \int_\Omega f \hat{\varphi}_i \, dx$$

Details

- Green's formula:

$$\int_\Omega \nabla u \cdot \nabla v \, dx = \int_\Gamma \partial_n u v \, ds - \int_\Omega \Delta u v \, dx$$

- Start from equation:

$$\begin{aligned}
-\Delta u &= f \Rightarrow \\
-\int_\Omega \Delta u v \, dx &= \int_\Omega f v \, dx \Rightarrow \\
\int_\Omega \nabla u \cdot \nabla v \, dx - \int_\Gamma \partial_n u v \, ds &= \int_\Omega f v \, dx \Rightarrow \\
\int_\Omega \nabla u \cdot \nabla v \, dx &= \int_\Omega f v \, dx
\end{aligned}$$

Details

- Continue

$$
\begin{aligned}
\int_\Omega \nabla u \cdot \nabla v \, dx &= \int_\Omega f v \, dx \\
&\Rightarrow \text{insert Ansatz} \\
\sum_{j=1}^N \xi_j \int_\Omega \nabla \varphi_j \cdot \nabla \varphi_i \, dx &= \int_\Omega f \varphi_i \, dx \Rightarrow \\
A_h \xi &= b
\end{aligned}
$$

Adaptivity for Poisson

How large is the error?

We expect the error $e = U - u$ to decrease if we increase the dimension N of V_h and $\hat{V}_h$. This can be done in different ways:

- h-adaptivity: decrease the mesh size h
- p-adaptivity: increase the polynomial order p
- hp-adaptivity: a combination of the h and p methods

We will only consider h-adaptivity.

An a posteriori error estimate

Let $\|\cdot\|_E$ denote the *energy-norm* given by $\|v\|_E = \|\nabla v\|$. Then the (piecewise linear) finite element solution $U = U(x)$ satisfies the error estimate

$$\|e\|_E = \|U - u\|_E \leq C\|h(R_1(U) + R_2(U))\|,$$

where $R_1(U) = |f + \Delta U| = |f|$ and

$$R_2(U) = \frac{1}{2} \max_{S\subset\partial K} h_K^{-1}|[\partial_S U]|.$$

Adaptive error control

Find V_h, given by a *triangulation* $\mathcal{T}_h$, such that

$$\|e\|_E \leq \mathrm{TOL},$$

where TOL is a given tolerance for the error.

This is satisfied if

$$C\|h(R_1(U) + R_2(U))\| \leq \mathrm{TOL}.$$

An adaptive algorithm

1. Choose an initial triangulation $\mathcal{T}_h^0$.
2. Compute the solution U on the current triangulation.
3. Compute the residuals R_1, R_2, and the error estimate.
4. If the error estimate is below the tolerance we stop. Otherwise, we refine the elements where $R_1 + R_2$ is large and start again at 2.