deal.II Users and Developers Training

March 21 – 24 2016

Timo Heister (heister@clemson.edu)

Luca Heltai (luca.heltai@sissa.it)

Basis of FEM!

Brief re-hash of the FEM, using the Poisson equation:

We start with the strong form:

$$
\begin{aligned}
-\Delta u &= f \qquad \text{in } \Omega \\
u &= 0 \qquad \text{on } \partial\Omega
\end{aligned}
$$

Basis of FEM!

Brief re-hash of the FEM, using the Poisson equation:

We start with the strong form:

$$-\Delta u = f$$

...and transform this into the weak form by multiplying *from the left* with a test function:

$$(\nabla \phi, \nabla u) = (\phi, f) \quad \forall \phi$$

The solution of this is a function *u(x)* from an infinite-dimensional function space.

Basis of FEM!

Since computers can't handle objects with infinitely many coefficients, we seek a finite dimensional function of the form

$$u_h = \sum_{j=1}^{N} U_j \phi_j(x)$$

To determine the *N* coefficients, test with the *N* basis functions:

$$(\nabla \phi_i, \nabla u_h) = (\phi_i, f) \quad \forall i = 1 \dots N$$

If basis functions are linearly independent, this yields *N* equations for *N* coefficients.
This is called the *Galerkin* method.

Basis of FEM!

Practical question 1: How to define the basis functions?

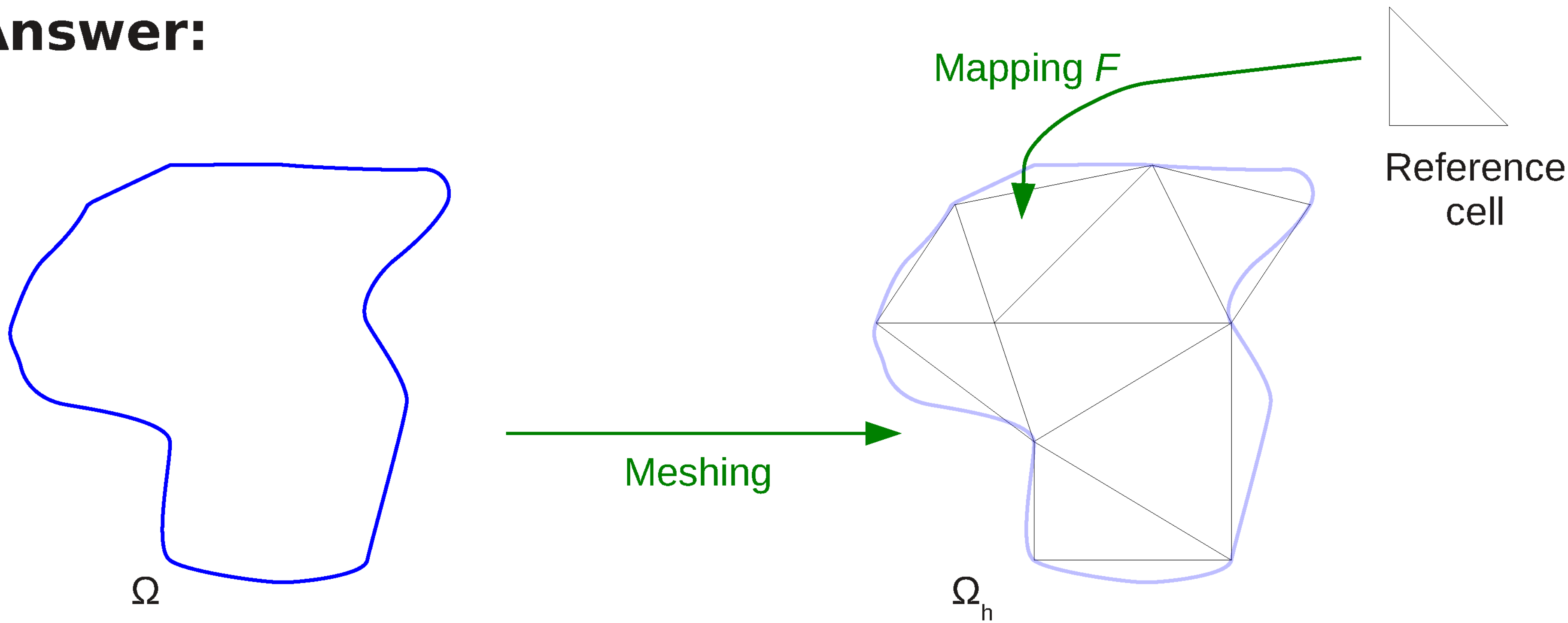
Answer: In the finite element method, this is done using the following concepts:

- Subdivision of the domain into a mesh
- Each cell of the mesh is a mapping of the reference cell
- Definition of basis functions on the reference cell
- Each shape function corresponds to a degree of freedom on the global mesh

Basis of FEM!

Practical question 1: How to define the basis functions?

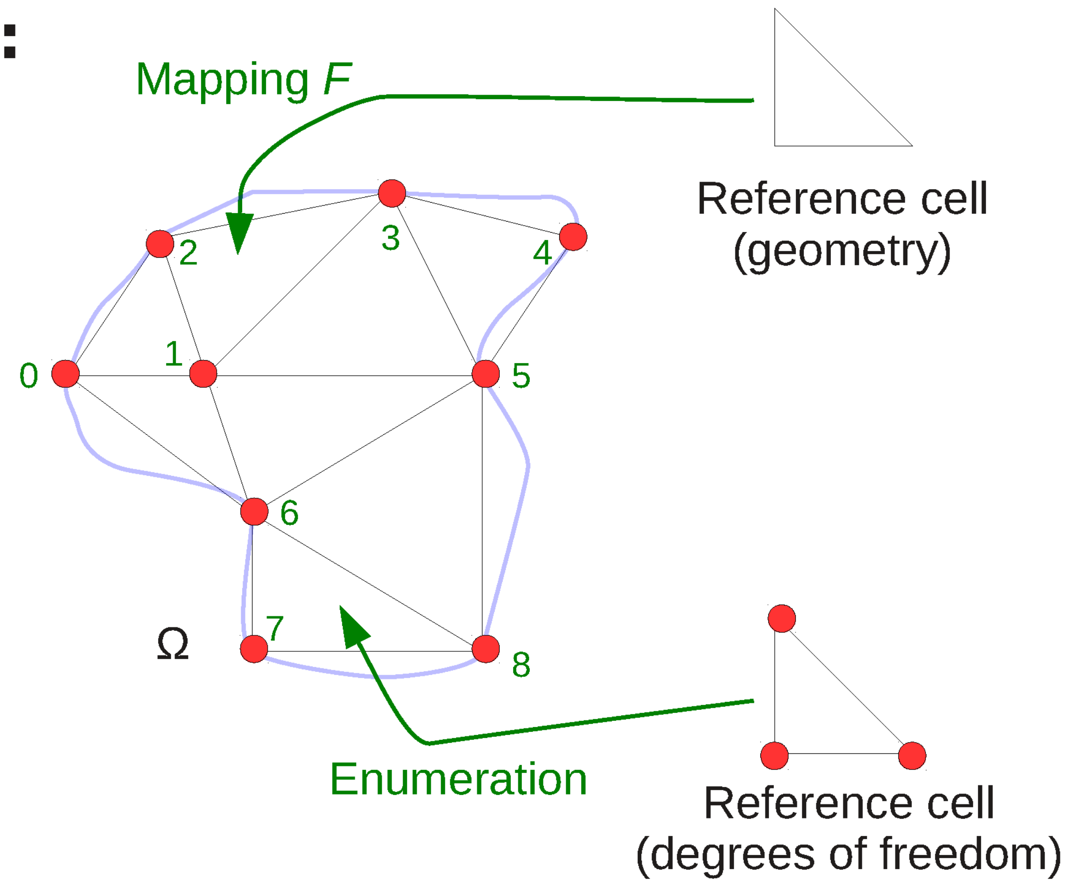
Answer:



Basis of FEM!

Practical question 1: How to define the basis functions?

Answer:



Basis of FEM!

Practical question 1: How to define the basis functions?

Answer: In the finite element method, this is done using the following concepts:

- Subdivision of the domain into a mesh
- Each cell of the mesh is a mapping of the reference cell
- Definition of basis functions on the reference cell
- Each shape function corresponds to a degree of freedom on the global mesh

Concepts in red will correspond to things we need to implement in software, explicitly or implicitly.

Basis of FEM!

Given the definition $u_h=\sum_{j=1}^{N} U_j \phi_j(x)$, we can expand the bilinear form

$$(\nabla \phi_i, \nabla u_h)=(\phi_i, f) \quad \forall i=1\ldots N$$

to obtain:

$$\sum_{j=1}^{N} (\nabla \phi_i, \nabla \phi_j) U_j=(\phi_i, f) \quad \forall i=1\ldots N$$

This is a linear system

$$AU=F$$

with

$$A_{ij}=(\nabla \phi_i, \nabla \phi_j) \qquad F_i=(\phi_i, f)$$

Basis of FEM!

Practical question 2: How to compute

$$A_{ij} = (\nabla \phi_i, \nabla \phi_j) \qquad\qquad F_i = (\phi_i, f)$$

Answer: By mapping back to the reference cell...

$$\begin{aligned} A_{ij} &= (\nabla \phi_i, \nabla \phi_j) \\ &= \sum_K \int_K \nabla \phi_i(x) \cdot \nabla \phi_j(x) \\ &= \sum_K \int_{\hat{K}} J_K^{-1}(\hat{x}) \hat{\nabla} \hat{\phi}_i(\hat{x}) \cdot J_K^{-1}(\hat{x}) \hat{\nabla} \hat{\phi}_j(\hat{x}) \; |\det J_K(\hat{x})| \end{aligned}$$

...and quadrature:

$$A_{ij} \approx \sum_K \sum_{q=1}^{Q} J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_i(\hat{x}_q) \cdot J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_j(\hat{x}_q) \; \underbrace{|\det J(\hat{x}_q)| \; w_q}_{=:\mathrm{JxW}}$$

Similarly for the right hand side *F*.

Basis of FEM!

Practical question 3: How to store the matrix and vectors of the linear system

$$AU = F$$

Answers:

- *A* is sparse, so store it in compressed row format
- *U,F* are just vectors, store them as arrays
- Implement efficient algorithms on them, e.g. matrix-vector products, preconditioners, etc.
- For large-scale computations, data structures and algorithms must be parallel

Basis of FEM!

Practical question 4: How to solve the linear system

$$AU = F$$

Answers: In practical computations, we need a variety of

- Direct solvers
- Iterative solvers
- Parallel solvers

Basis of FEM!

Practical question 5: What to do with the solution of the linear system

$$AU = F$$

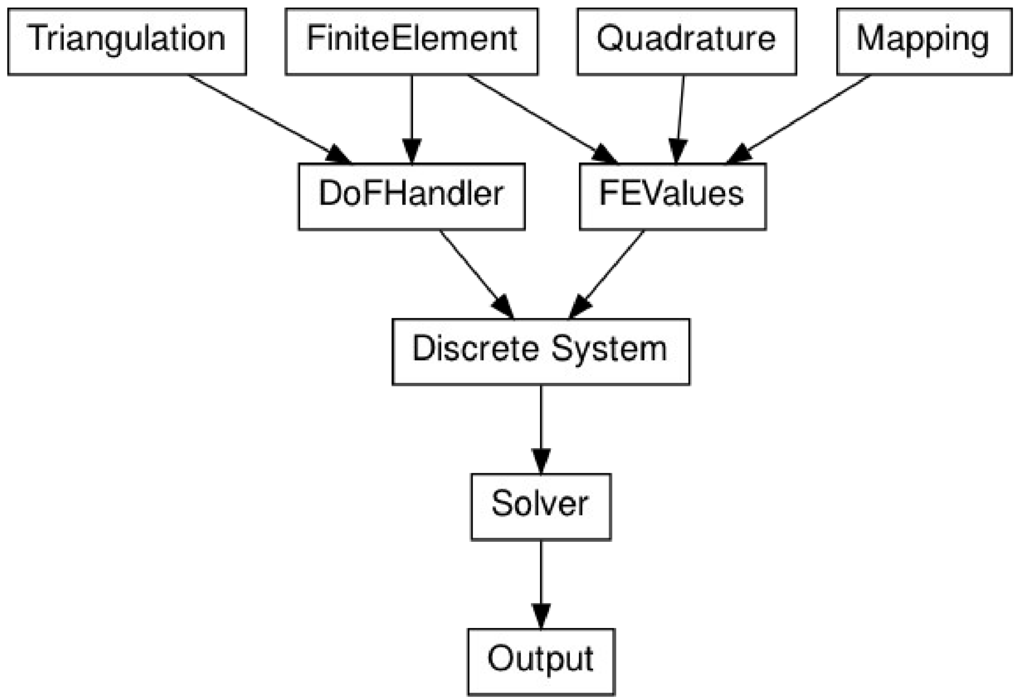
Answers: The goal is not to solve the linear system, but to do something with its solution:

- Visualize
- Evaluate for quantities of interest
- Estimate the error

These steps are often called *postprocessing the solution.*

Basis of FEM!

Together, the concepts we have identified lead to the following components that all appear (explicitly or implicitly) in finite element codes:



Start with Tutorial: step-3

Step-3 shows:

- How to set up a linear system
- How to assemble the linear system from the bilinear form:
  - The loop over all cells
  - The *FEValues* class
- Solving linear systems
- Visualizing the solution

Start with Tutorial: step-3

Recall:

- For the Laplace equation, the bilinear form is written as a sum over all cells:

$$\begin{aligned} A_{ij} &= (\nabla \phi_i, \nabla \phi_j) \\ &= \sum_K \int_K \nabla \phi_i(x) \cdot \nabla \phi_j(x) \end{aligned}$$

- But on each cell, only few shape functions are nonzero!
- For Q_1, only $16=4^2$ matrix entries are nonzero per cell
- Only compute this (dense) sub-matrix, then “distribute” it to the global A
- Similar for the right hand side vector.

Start with Tutorial: step-3

Recall:

- We use quadrature

$$
\begin{aligned}
A_{ij}^{K} &= \int_K \nabla \hat{\phi}_i(x) \cdot \nabla \hat{\phi}_j \, dx \\
&\approx \sum_{q=1}^{Q} J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_i(\hat{x}_q) \; \cdot \; J_K^{-1}(\hat{x}_q) \hat{\nabla} \hat{\phi}_j(\hat{x}_q) \; \underbrace{|\det J(\hat{x}_q)| \; w_q}_{=:\mathrm{JxW}}
\end{aligned}
$$

- We really only have to evaluate shape functions, Jacobians, etc., at quadrature points – not as functions
- All evaluations happen on the reference cells

Now … Exercise Time!