



Solving Poisson's equation

Denis Davydov (denis.davydov@fau.de)

Luca Heltai (luca.heltai@sissa.it)

25 February - 1 March 2019



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT



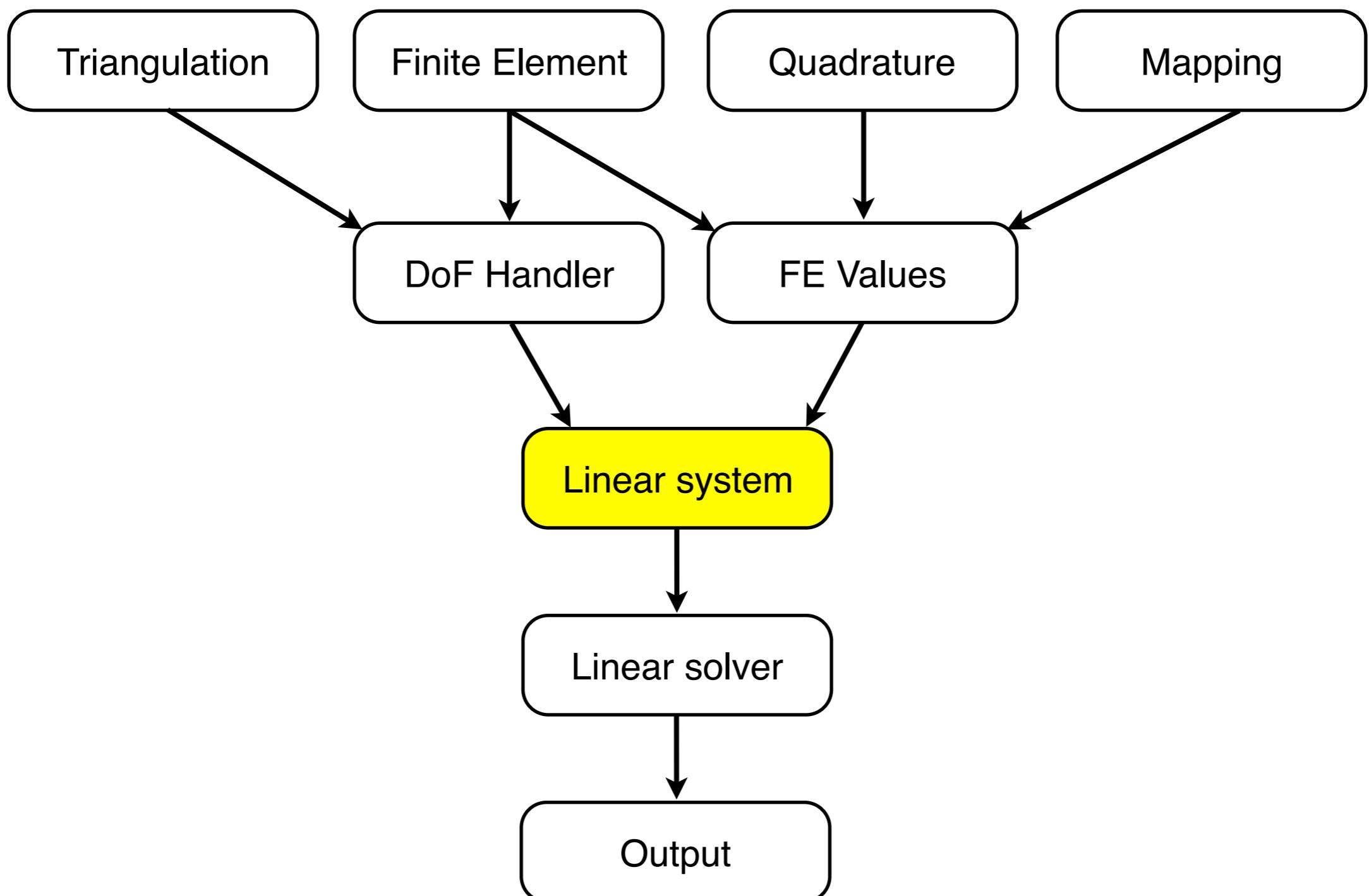
Aims for this module

- First introduction into assembly of sparse linear systems
 - Translation of weak form to assembly loops
 - Applying boundary conditions
- Using linear solvers
- Post-processing and visualization

Reference material

- Tutorials
 - Step-3
https://dealii.org/9.0.0/doxygen/deal.II/step_3.html
- Documentation
 - https://www.dealii.org/9.0.0/doxygen/deal.II/group_FE_vs_Mapping_vs_FEValues.html
 - https://www.dealii.org/9.0.0/doxygen/deal.II/group_UpdateFlags.html

Structure of a prototypical FE problem



Sparse linear systems

- Minimize data storage
 - Evaluate grid connectivity
- Functions to help set up
 - Sparsity pattern
 - Constraints
- Minimal access times
 - Direct manipulation of (non-zero) entries
 - Matrix-vector operations (skip over zero-entries)
- Types
 - Unity (monolithic, contiguous)
 - Block sparse structures
- Sub-organisation (e.g. component-wise)

$$[K] \{d\} = \{F\}$$

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

$$(K_{11} - K_{12}K_{22}^{-1}K_{21}) d_1$$

$$= F_1 - K_{12}K_{22}^{-1}F_2$$

$$d_2 = K_{22}^{-1} (F_2 - K_{21}d_1)$$

Constraints on sparse linear systems

- Strong Dirichlet boundary conditions
 - Apply user-defined spatially-dependent functions to specific boundaries
 - Can restrict to components of a multi-dimensional field
 - Limited to interpolatory FEM
- Neumann boundary conditions
 - Implementation dependent
- Other constraints need special consideration
 - Periodic boundary conditions
 - Refinement with hanging nodes
 - Some time-dependent formulations

$$[K] \{d\} = \{F\}$$

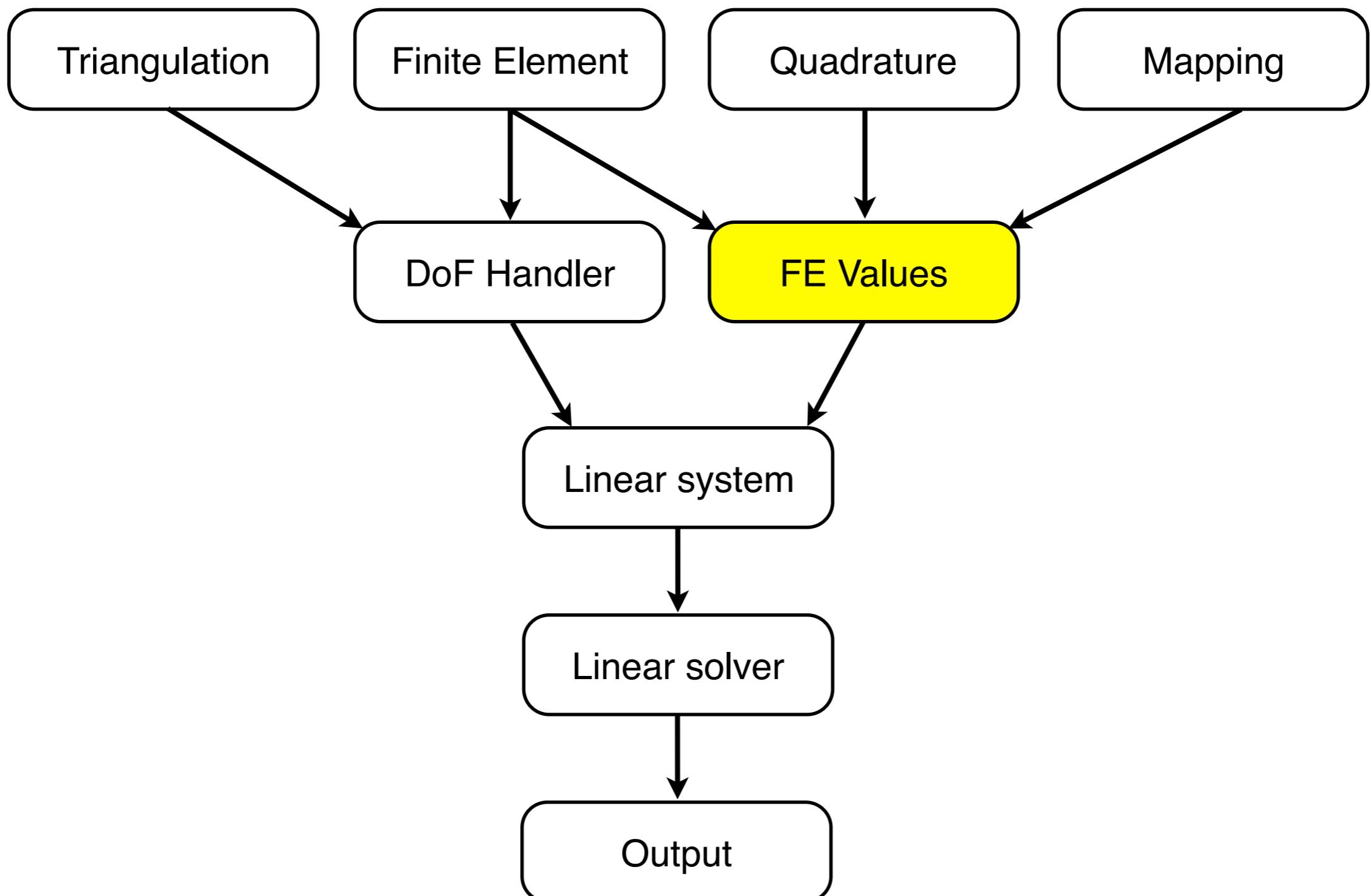
$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix}$$

$$(K_{11} - K_{12}K_{22}^{-1}K_{21}) d_1$$

$$= F_1 - K_{12}K_{22}^{-1}F_2$$

$$d_2 = K_{22}^{-1} (F_2 - K_{21}d_1)$$

Structure of a prototypical FE problem



Integration on a cell: the FEValues class

$$K = \int_{\Omega} \nabla \delta \phi(\mathbf{x}) \cdot k \nabla \phi(\mathbf{x}) dV$$

$$\approx \delta \phi^I \sum_K \left(\int_{\Omega_K^h} \nabla N^I(\mathbf{x}) \cdot k \nabla N^J(\mathbf{x}) dV^h \right) \phi^J$$

$$\approx \delta \phi^I \sum_K \left(\sum_q \nabla N^I(\mathbf{x}_q) \cdot k_q \nabla N^J(\mathbf{x}_q) w_q \right) \phi^J$$

$\overbrace{\hspace{10em}}$

$$K_{IJ} = (\nabla N^I, k \nabla N^J)$$

$$\approx \delta \phi^I \sum_K \left[\sum_q J_K^{-1}(\hat{\mathbf{x}}_q) \hat{\nabla} \hat{N}^I(\hat{\mathbf{x}}_q) \cdot k_q J_K^{-1}(\hat{\mathbf{x}}_q) \hat{\nabla} \hat{N}^J(\hat{\mathbf{x}}_q) |\det J_K(\hat{\mathbf{x}}_q)| w_q \right] \phi^J$$

$\overbrace{\hspace{10em}}$

$$K_{IJ}$$

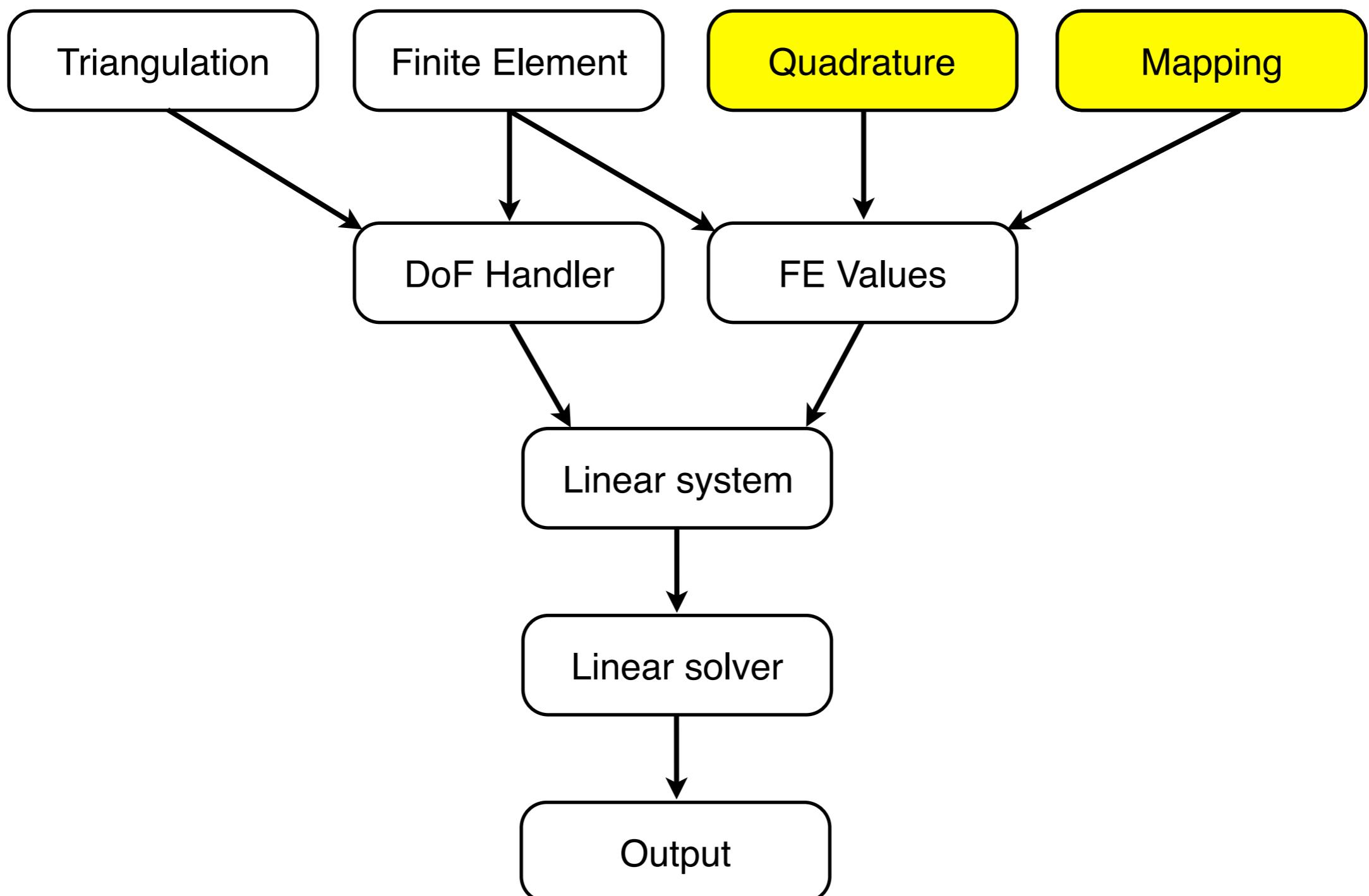
Integration on a cell: the FEValues class

- Object that helps perform integration
- Combines information of:
 - Cell geometry
 - Finite-element system
 - Quadrature rule
 - Mappings
- Can provide:
 - Shape function data
 - Quadrature weights and mapping jacobian at a point
 - Normal on face surface
 - Covariant/contravariant basis vectors
- More ways it can help:
 - Object to extract shape function data for individual fields
 - Natural expressions when coding
- Low level optimizations

$$K_{IJ} = \sum_K \left(\sum_q J_K^{-1}(\hat{\mathbf{x}}_q) \hat{\nabla} \hat{N}^I(\hat{\mathbf{x}}_q) \cdot J_K^{-1}(\hat{\mathbf{x}}_q) \hat{\nabla} \hat{N}^J(\hat{\mathbf{x}}_q) |\det J_K(\hat{\mathbf{x}}_q)| w_q \right)$$

```
cell_matrix(I,J) += k
  * fe_values.shape_grad (I, q_point)
  * fe_values.shape_grad (J, q_point)
  * fe_values.JxW (q_point);
```

Structure of a prototypical FE problem



Matrix form

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{F}$$

$$K_{ij} := a(N_i, N_j) \quad i, j \in \mathcal{N}_U$$

$$F_i := (N_i, f) + (N_i, h)_{\partial\Omega} - \sum_{j \in \mathcal{N}_D} a(N_i, N_j) q(\mathbf{x}_j)$$

$$(S) = (W) \approx (W^h) = (D)$$

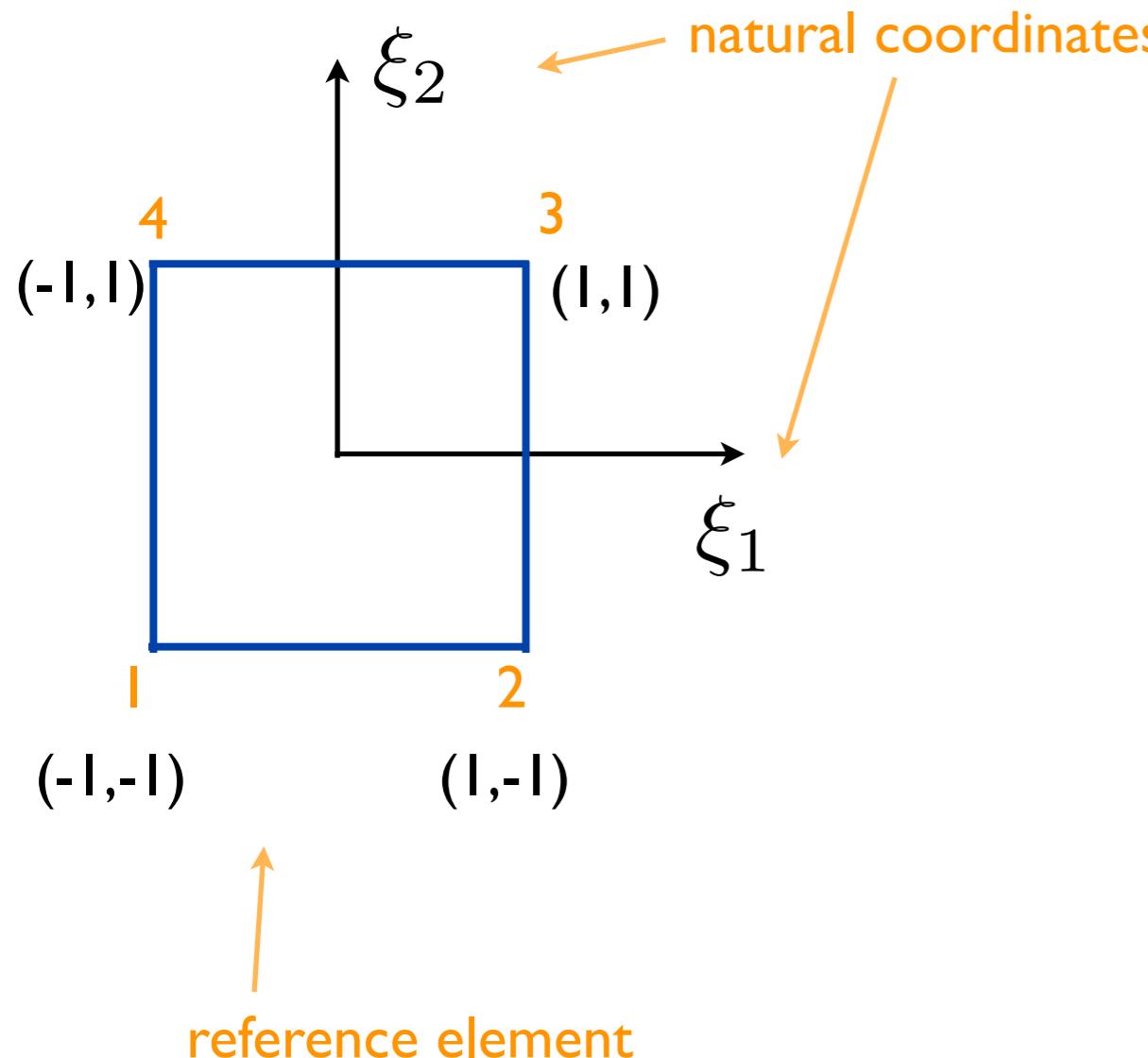
need to evaluate
integrals numerically

$$a(N_i, N_j) := \sum_K \int_{\Omega_K} \nabla N_i \cdot \mathbf{k} \cdot \nabla N_j d\mathbf{v}$$

$$(N_i, f) := \sum_K \int_{\Omega_K} N_i f(\mathbf{x}) d\mathbf{v}$$

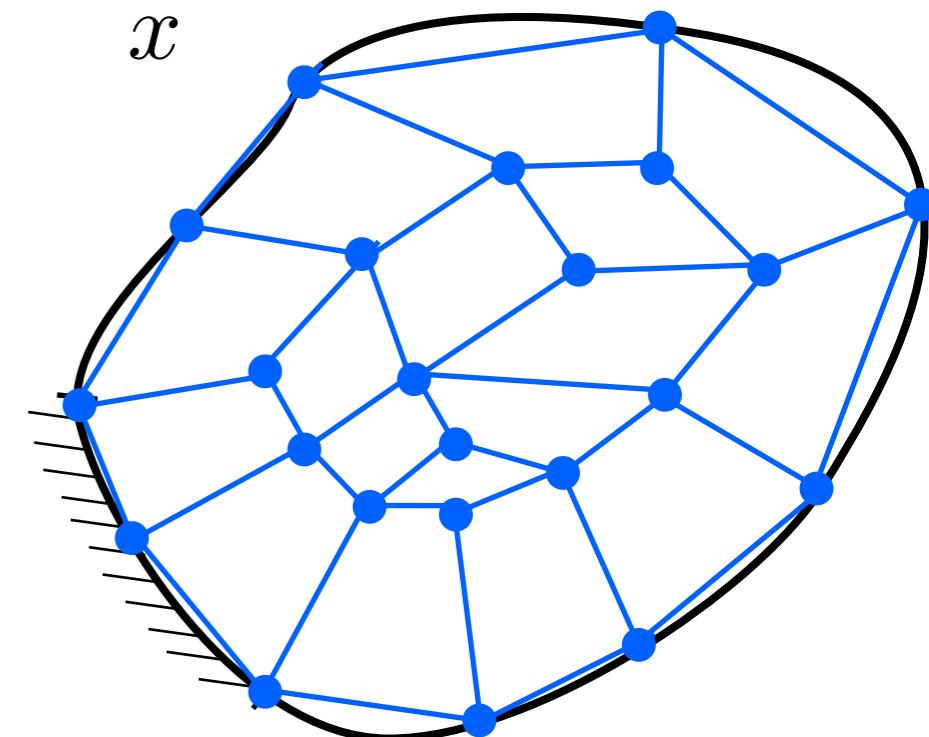
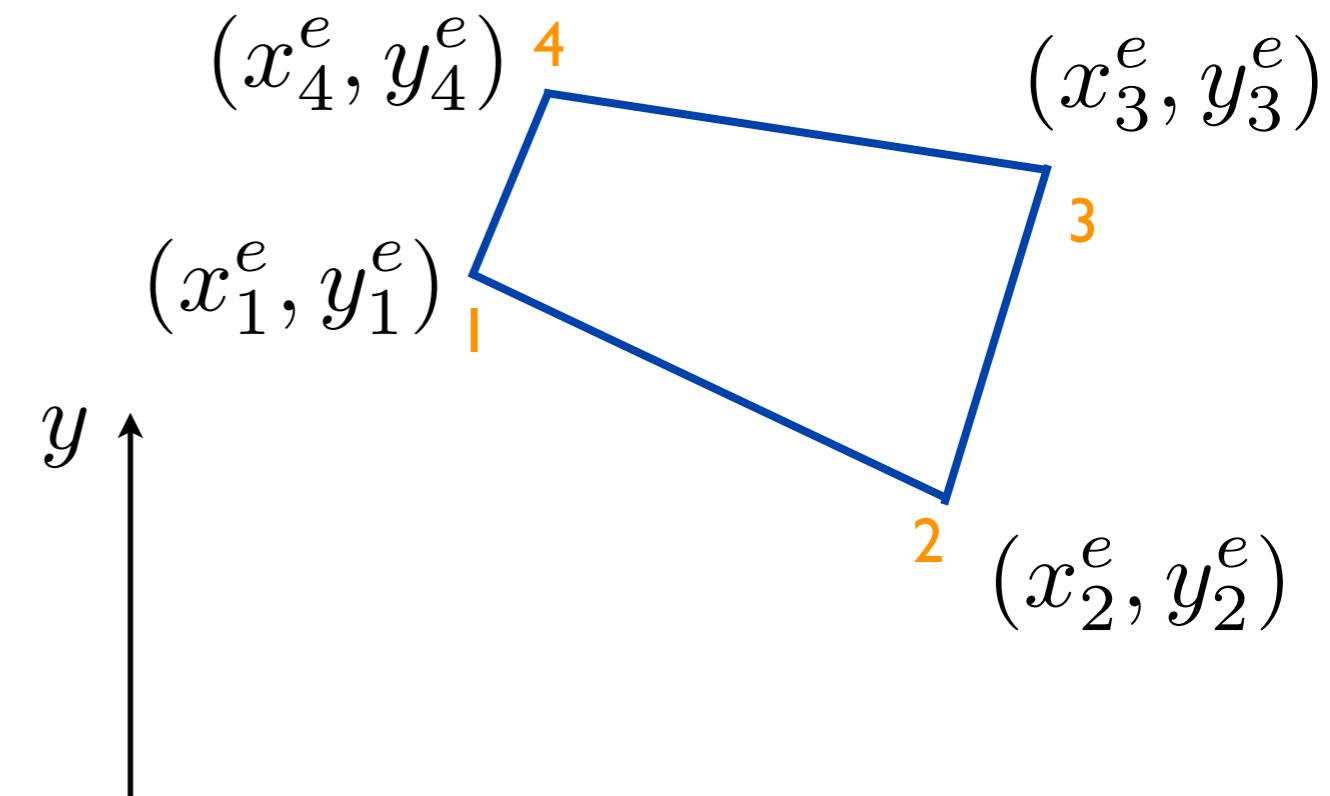
$$(w, h)_{\partial\Omega} := \sum_K \int_{\partial\Omega_K^N} w h ds$$

Q1 mapping



we can construct the mapping
between the two elements

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$$



Bilinear Quadrilateral Element

Bilinear expansion

$$x(\xi_1, \xi_2) =: \alpha_0 + \alpha_1 \xi_1 + \alpha_2 \xi_2 + \alpha_3 \xi_1 \xi_2$$

$$y(\xi_1, \xi_2) =: \beta_0 + \beta_1 \xi_1 + \beta_2 \xi_2 + \beta_3 \xi_1 \xi_2$$

+

$$x(\xi_1^a, \xi_2^a) = x_a^e \quad a = \overline{1, 4}$$

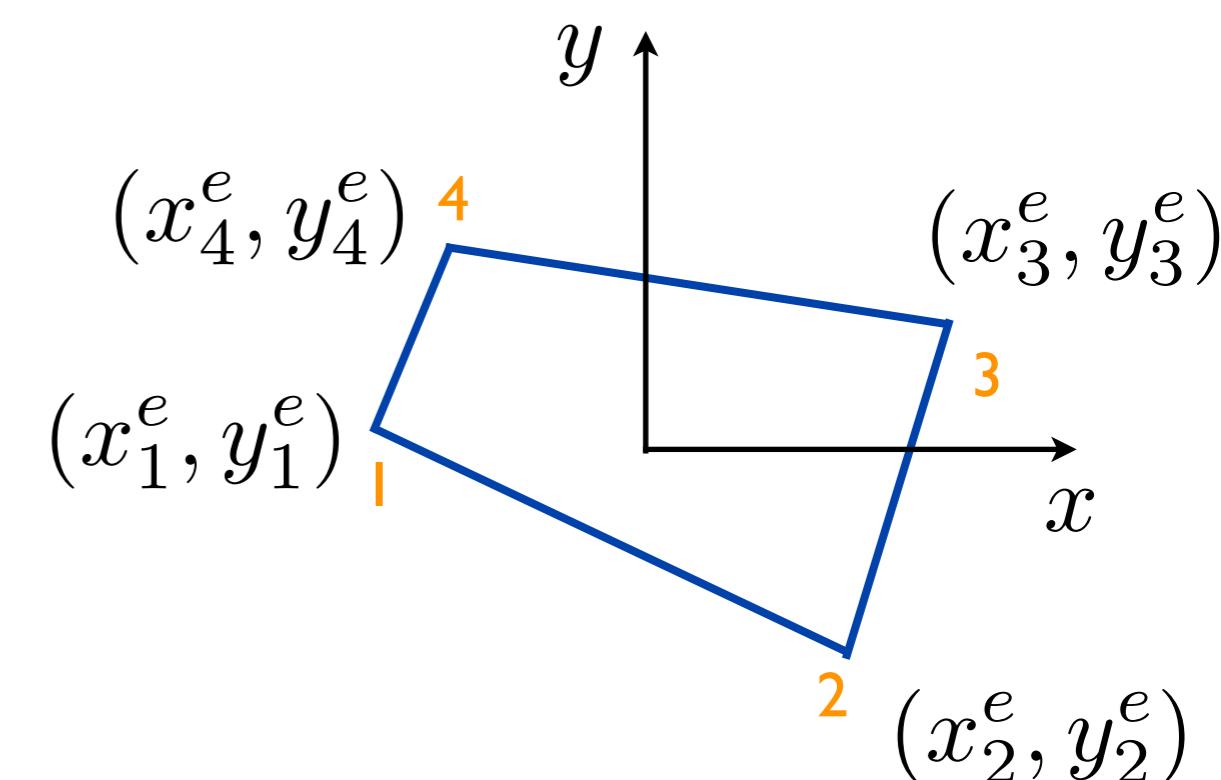
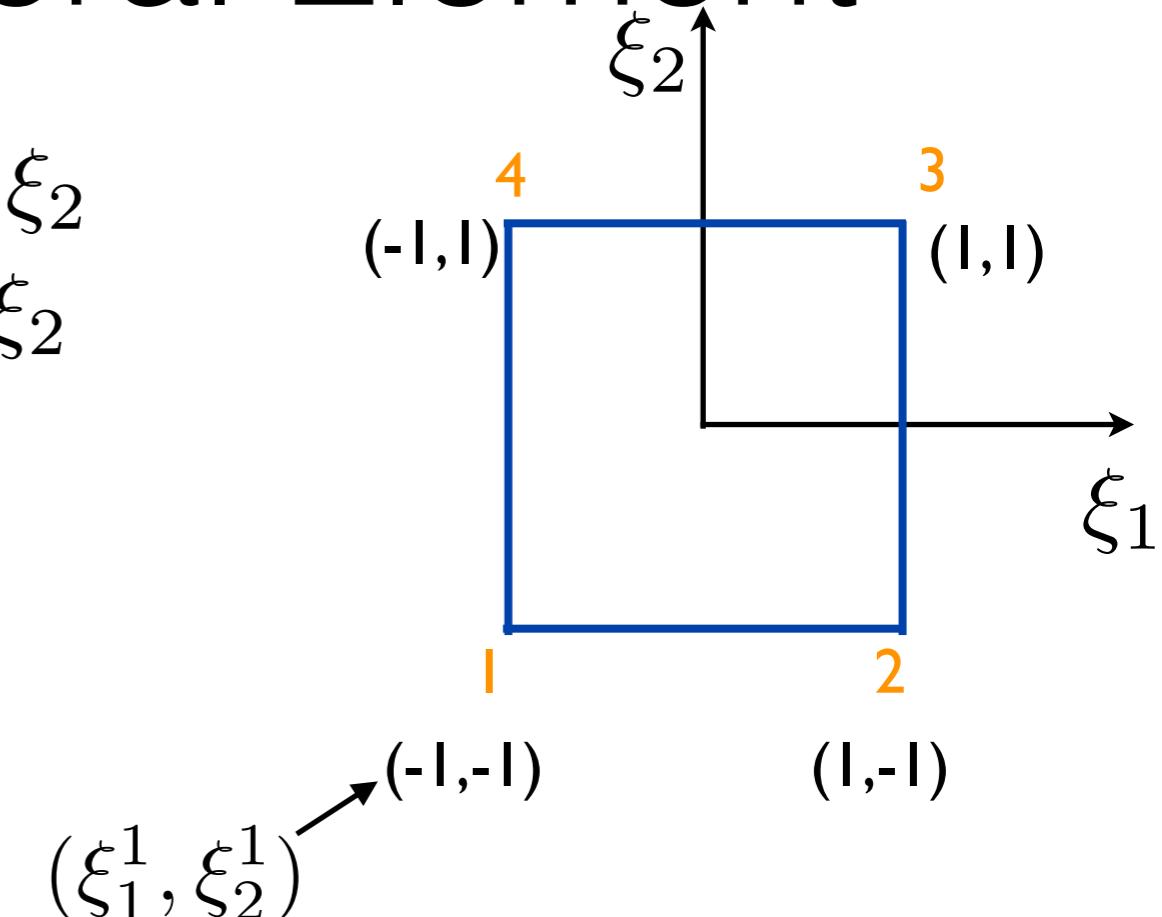
$$y(\xi_1^a, \xi_2^a) = y_a^e$$

=

$$\boldsymbol{x}(\xi) = \sum_{a=1}^4 N_a(\xi) \boldsymbol{x}_a^e$$

maps any point
in the reference
element to the
actual element

$$N_a(\xi) = \frac{1}{4} [1 + \xi_1^a \xi_1][1 + \xi_2^a \xi_2]$$



Mapping to the reference element:

$$\mathbf{J} := \frac{\partial \mathbf{x}}{\partial \xi}$$

$$\nabla = \frac{\partial}{\partial x_i} \mathbf{e}_i$$

$$\text{grad}(\bullet) = (\bullet) \nabla = \frac{\partial(\bullet)}{\partial x_i} \mathbf{e}_i = \frac{\partial(\bullet)}{\partial \xi_j} \frac{\partial \xi_j}{\partial x_i} \mathbf{e}_i = \widehat{\text{grad}}(\bullet) \cdot \mathbf{J}_K^{-1}$$

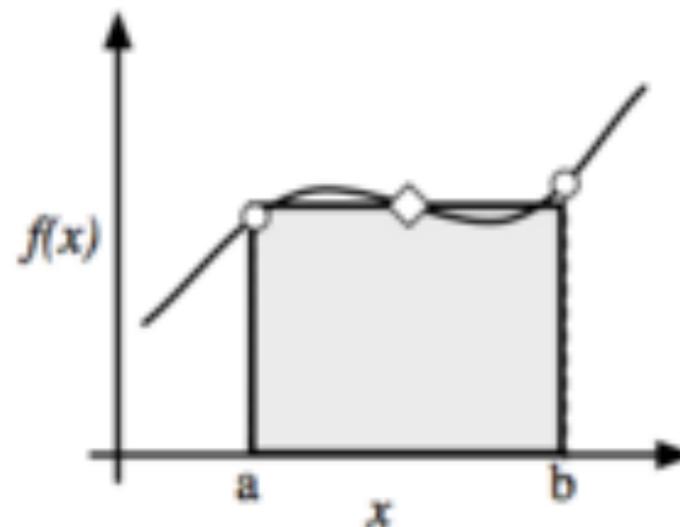
$$(S) = (W) \approx (W^h) = (D) \approx (D^q)$$

$$\begin{aligned}
 a(N_i, N_j) &= \sum_K \int_{\Omega_K} \text{grad } N_i(\mathbf{x}) \cdot \text{grad } N_j(\mathbf{x}) dv \\
 &= \sum_K \int_{\hat{\Omega}_K} [\widehat{\text{grad }} \widehat{N}_i(\xi) \cdot \mathbf{J}_K^{-1}(\xi)] \cdot [\widehat{\text{grad }} \widehat{N}_j(\xi) \cdot \mathbf{J}_K^{-1}(\xi)] \det(\mathbf{J}_K(\xi)) d\hat{v} \\
 &\approx \sum_K \sum_q [\widehat{\text{grad }} \widehat{N}_i(\xi_q) \cdot \mathbf{J}_K^{-1}(\xi_q)] \cdot [\widehat{\text{grad }} \widehat{N}_j(\xi_q) \cdot \mathbf{J}_K^{-1}(\xi_q)] \det(\mathbf{J}_K(\xi_q)) w_q
 \end{aligned}$$

do not depend on a particular cell
 

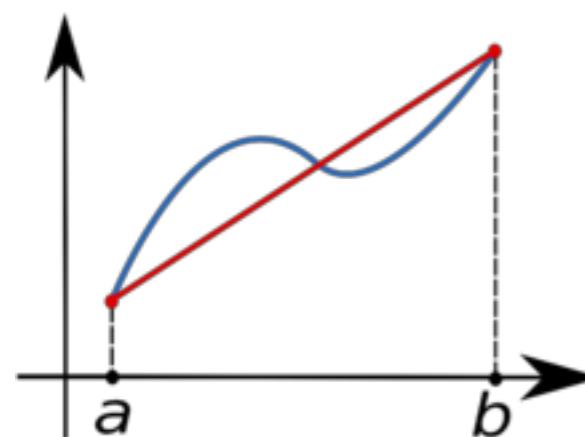
Integration rules:

1. midpoint



$$\int_a^b f(x)dx \approx f\left(\frac{a+b}{2}\right) [b-a]$$

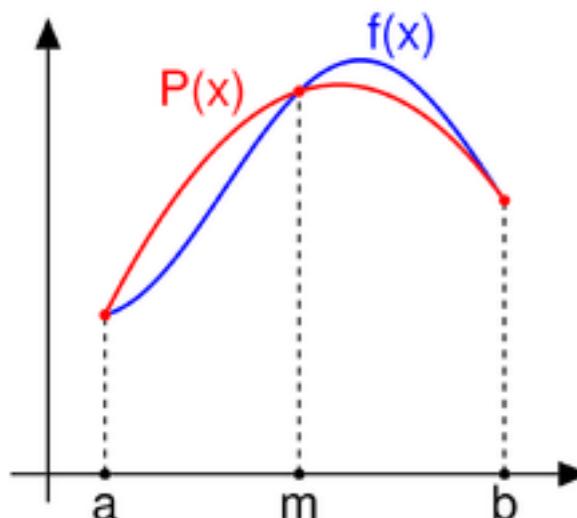
2. trapezoidal



$$\int_a^b f(x)dx \approx \left[\frac{f(a) + f(b)}{2} \right] [b-a]$$

Integration rules:

3. Simpson



$$\int_a^b f(x)dx \approx \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \frac{b-a}{6}$$

4. Gauss quadrature rule

Constructed to be exact for polynomials of degree $2n-1$

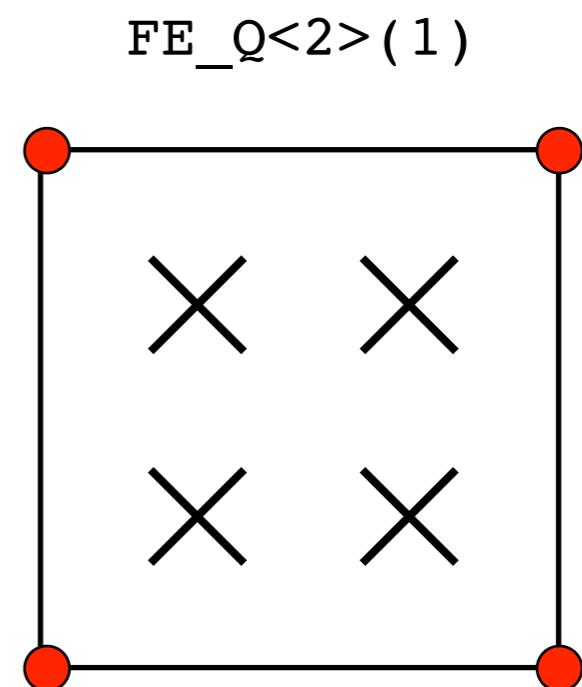
$$\int_{-1}^1 f(x)dx \approx \sum_q f(x_q)w_q$$

n_q	x_1	x_2	x_3	w_1	w_2	w_3
1	0			2		
2	$-1/\sqrt{3}$	$1/\sqrt{3}$		1	1	
3	$-\sqrt{3/5}$	0	$\sqrt{3/5}$	$5/9$	$8/9$	$5/9$

there are other integration rules: Monte Carlo, Newton-Cotes, Runge-Kutta,...

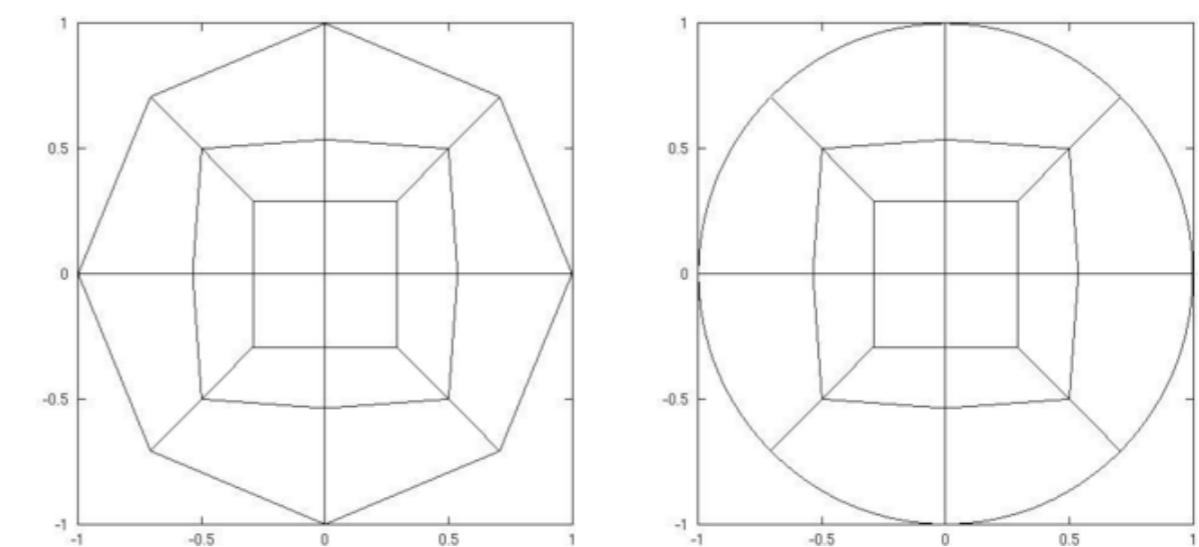
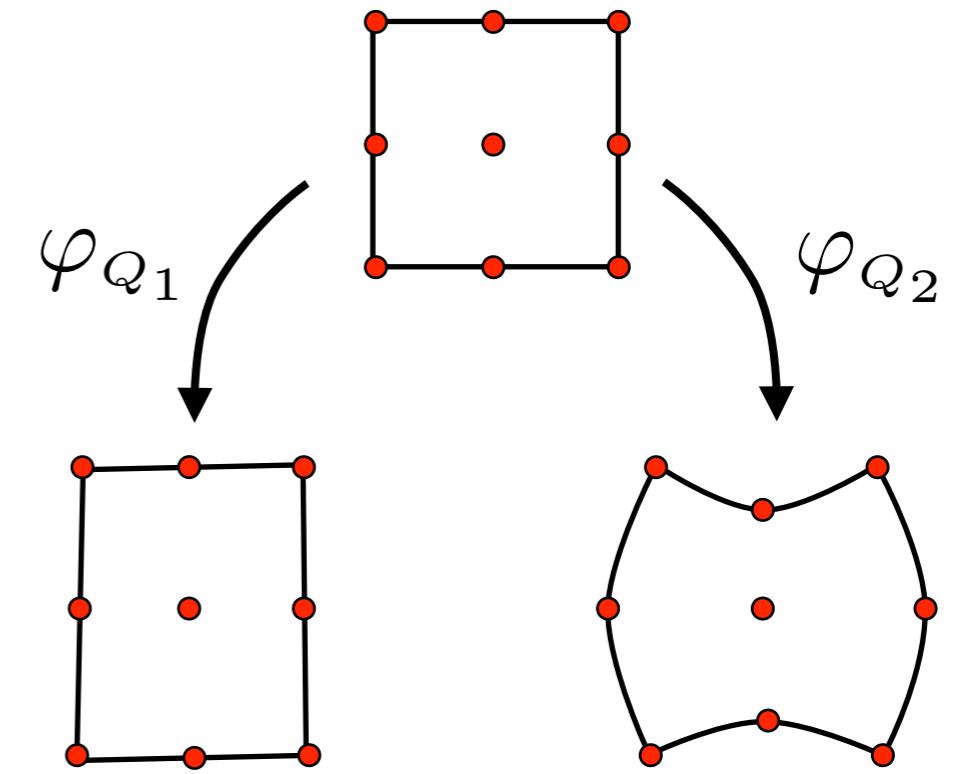
Integration on a cell: the Quadrature classes

- QGauss<dim> n-Order Gauss quadrature
- Other rules
 - QGaussLobattom<dim> Gauss Lobatto
 - QSimpson<dim> Simpson
 - QTrapez<dim> Trapezoidal
 - QMidpoint Midpoint
 - ...

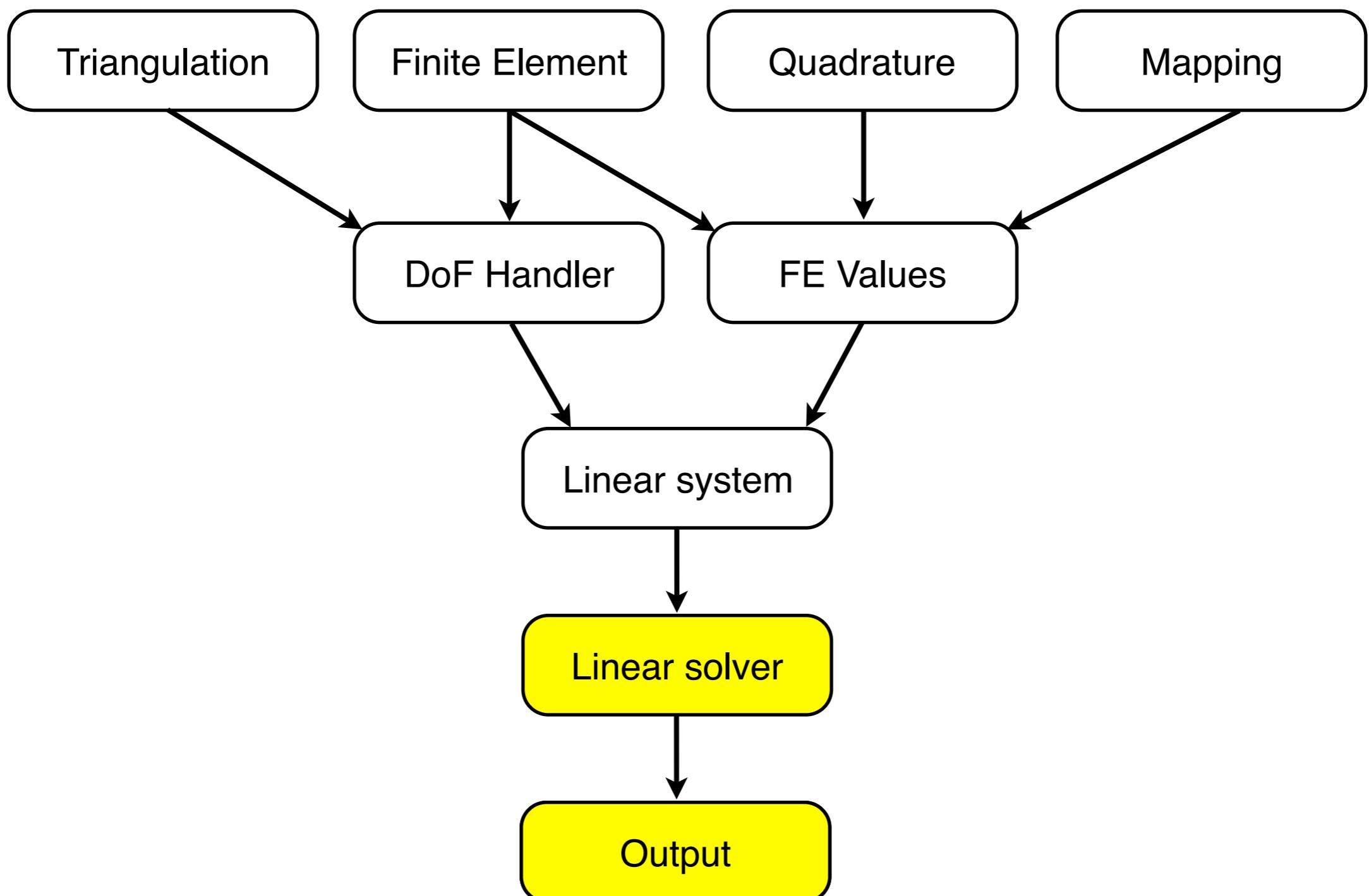


Integration on a cell: the Mapping classes

- n-order mappings
 - Increase accuracy of:
 - Integration schemes
 - Surface basis vectors
- Lagrangian / Eulerian
 - Latter useful for fluid and contact problems, data visualization
- Boundary and interior manifolds



Structure of a prototypical FE problem



Solving Poisson's equation

- Demonstration: Step-3
https://www.dealii.org/9.0.0/doxygen/deal.II/step_3.html
<http://www.math.colostate.edu/~bangerth/videos.676.10.html>
- Key points
 - Local assembly + quadrature rules
 - Distribution of local contributions to the global linear system
 - Application of boundary conditions
 - Solving a linear system
 - Output for visualisation

