Finite Element Modelling for Geosciences: 1d to 2d and Isoparametric Elements

Patrick Sanan

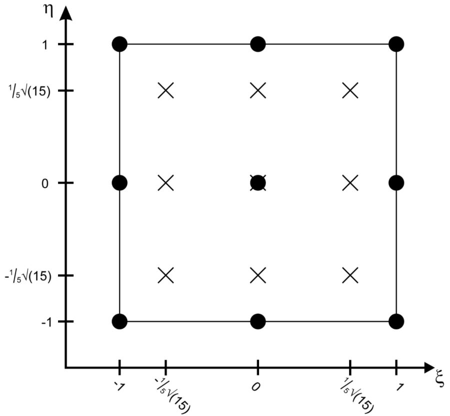
ETH Zurich

From 1d to 2d

- Most problems of scientific or engineering interest are posed in 2 or more dimensions
- The good news is that almost everything remains the same, regardless or dimension
  - Choose physics (weak/variational form)
  - Mesh domain
  - Choose basis functions
  - Define element matrices and vectors
  - Assemble element matrices/vectors into global matrices/vectors
  - Solve the system
  - Plot, postprocess, analyze, ...
- Conceptual complication: quadrature in 2D (this lecture)
- Practical complication: indexing (See e.g. Figure 5.1 in the notes)
- (Conceptual complication: Neumann Boundary conditions in 2d) (Not emphasized in this course)

Numerical Quadrature in 2d

- 2d quadrature rules over reference elements can be defined as products of 1d rules.



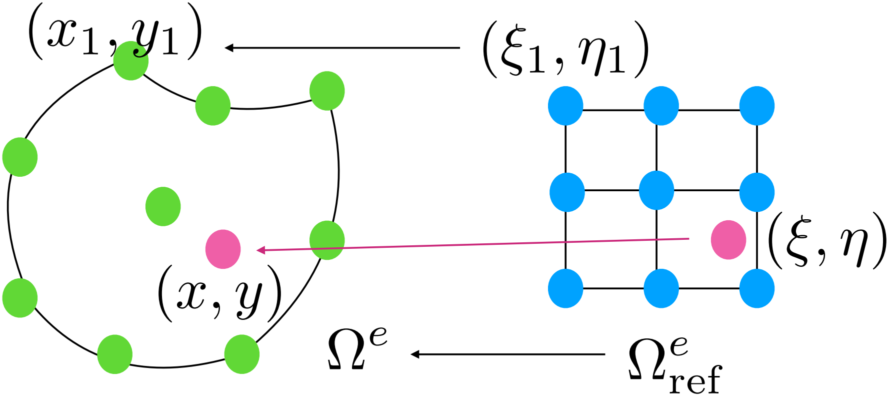
$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)d\vec{\eta} \approx \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} w_i w_j f(\xi_i,\eta_j)$$

Numerical Quadrature in 2d

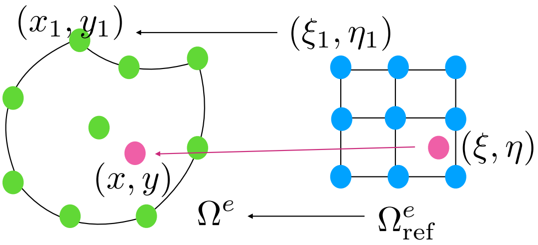
- It's convenient to introduce a linear numbering, so that you can loop over all points in your code

$$\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)d\vec{\eta} \approx \sum_{i=1}^{n_x}\sum_{j-1}^{n_y} w_i w_j f(\xi_i,\eta_j)$$
$$= \sum_{k=1}^{N_{\text{ip}}} f(\xi_k,\eta_k)W_k, \quad N_{\text{ip}} \doteq n_x n_y$$

Isoparametric Elements



Isoparametric Elements



- In constructing our method, we can make two independent choices
  1. What our basis functions N_i are, on each physical element
  2. How to map from a reference element, to help with quadrature.
- Key: Use the **same** mapping (ϕ, earlier) to define the basis functions, by siimply mapping basis functions on the reference element.
- This ends up being very convenient, computationally, as you'll see when writing your codes.
- Terminology: the word "element" is sometimes used to mean the basis functions, as well as the physical subdomains!

Isoparametric Elements: Jacobian Computation

$$
\begin{aligned}
x &= N_1(\xi,\eta)x_1 + N_2(\xi,\eta)x_2 + \dots \\
y &= N_1(\xi,\eta)x_2 + N_2(\xi,\eta)y_2 + \dots \\
&\qquad\qquad\qquad\qquad\qquad\qquad \dots
\end{aligned}
$$

$$
\begin{bmatrix} x \\ y \\ \vdots \end{bmatrix} = \begin{bmatrix} N_1(\vec{\xi}) & N_2(\vec{\xi}) & N_3(\vec{\xi}) & \dots \end{bmatrix} \begin{bmatrix} x_1 & y_1 & \dots \\ x_2 & y_2 & \dots \\ x_3 & y_3 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}
$$

$$
\mathbf{J} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \dots & \frac{\partial N_9}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \dots & \frac{\partial N_9}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_9 & y_9 \end{bmatrix}
$$

Isoparametric Elements: Jacobian Computation

$$\mathbf{J} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \cdots & \frac{\partial N_9}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \cdots & \frac{\partial N_9}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \vdots & \vdots \\ x_9 & y_9 \end{bmatrix}$$

- Note that the matrix on the left is constant for each quadrature point in the reference domain.
- Note that the matrix on the right is constant for each element.

Derivatives of shape functions

- In assembling our stiffness matrix, we need to evaluate integrals of the form

$$\int_{\Omega^e} (\nabla \mathbf{N}(\vec{x}))^T \mathbf{D} (\nabla \mathbf{N}_j(\vec{x})) d\vec{x}$$

  (Compare with (5.13) in the notes)
  This is a matrix, with entries

$$\int_{\Omega^e} k_{ij}(\vec{x}) \frac{\partial N_a}{\partial x_i}(\vec{x}) \frac{\partial N_b}{\partial x_j}(\vec{x}) d\vec{x}$$

- How to we compute things like $\frac{\partial N_a}{\partial x_i}$? (i.e. the entries in $\nabla \mathbf{N}$ (5.16))

Derivatives of shape functions

- Recall how the Jacobian appears in the chain rule (5.23)

$$\begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \cdots \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \vdots \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \vdots \end{bmatrix}$$

Derivatives of shape functions

- We know the derivatives of the basis functions in the reference domain, e.g. $\frac{\partial N_i}{\partial \xi}$
- So, we invert the previous relationship (5.24)

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \vdots \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \vdots \end{bmatrix}$$

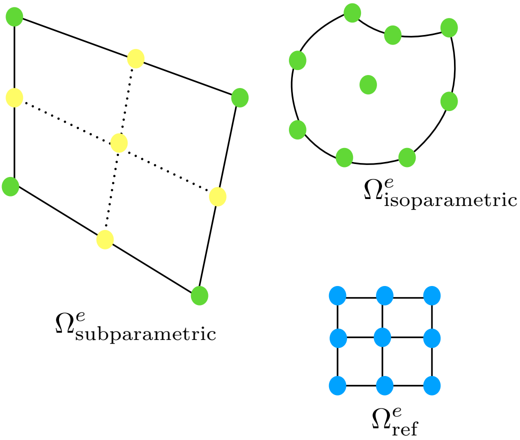
- You can use this to compute e.g. $\frac{\partial N_i}{\partial x}$.

$$\begin{bmatrix} \frac{\partial N_i(x,y)}{\partial x} \\ \frac{\partial N_i(x,y)}{\partial y} \\ \vdots \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_i(\xi,\eta)}{\partial \xi} \\ \frac{\partial N_i(\xi,\eta)}{\partial \eta} \\ \vdots \end{bmatrix}$$

Derivatives of shape functions

- Programming tip in MATLAB: If you are clever in how your arrange your matrix representing the derivatives of N_i in the reference domain, you can use the backlash operator to apply $\mathbf{J}^{-1}$ to each column of a matrix, without calling `inv()`.

What would a **non**-isoparametric element look like?



- Use quadratic basis functions
- Define the yellow points as the midpoints of the edges
- Thus, the physical element is defined by 4 points, even though there are 9 basis functions, so this is called a *subparametric* element