DD2370 Computational Methods for Electromagnetics
FEM Method in 2D – Interpolation Functions and Assembly Matrix

Stefano Markidis, KTH Royal Institute of Technology

Problem Statement

- Consider the problem of calculating the static electric potential φ due to electric charge density ρ_e distributed in domain Ω.

$$-\nabla \cdot (\varepsilon \nabla \varphi) = \rho_e \qquad \text{on } \Omega$$

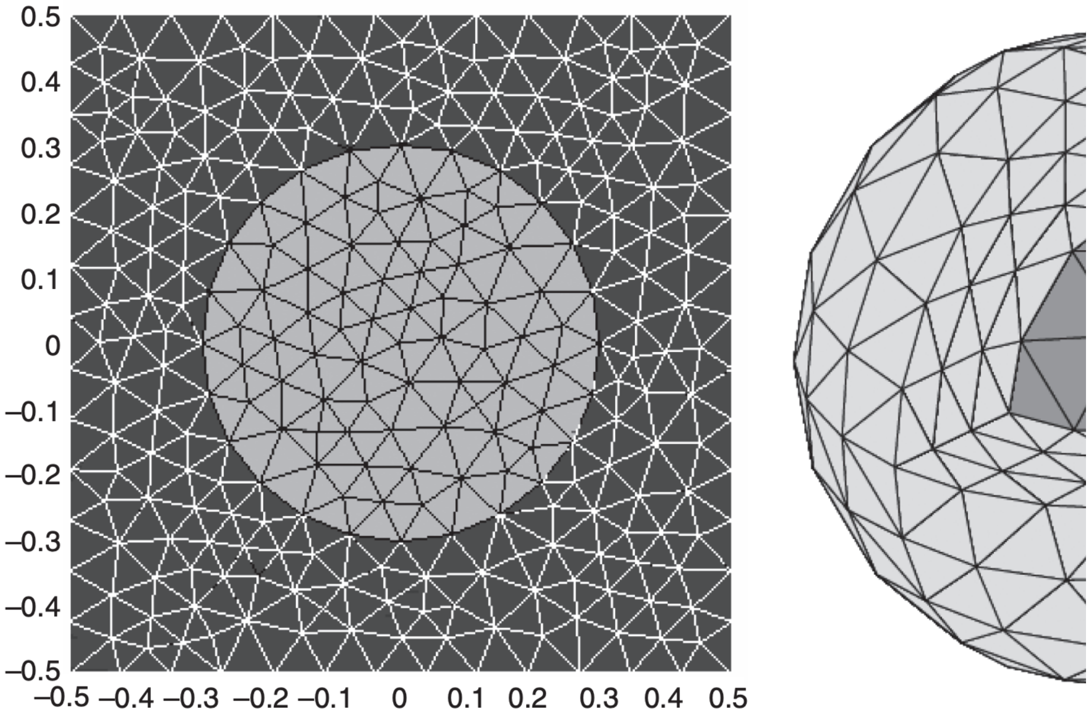
- In addition, φ must satisfy certain boundary conditions to have a unique solution. Typical boundary conditions include **the Dirichlet condition and the Neumann condition**, which prescribes the normal derivative of the potential.

$$\varphi = \varphi_D \qquad \text{on } \Gamma_D$$

$$\hat{n} \cdot (\varepsilon \nabla \varphi) = \kappa_N \qquad \text{on } \Gamma_N$$

FEM Formulation

- Divide the entire domain into many small subdomains = finite elements
- Seek an approximate solution in each of the subdomains.
- The commonly used subdomains are **triangular elements** in two dimensions and **tetrahedral elements** in three dimensions



(a)

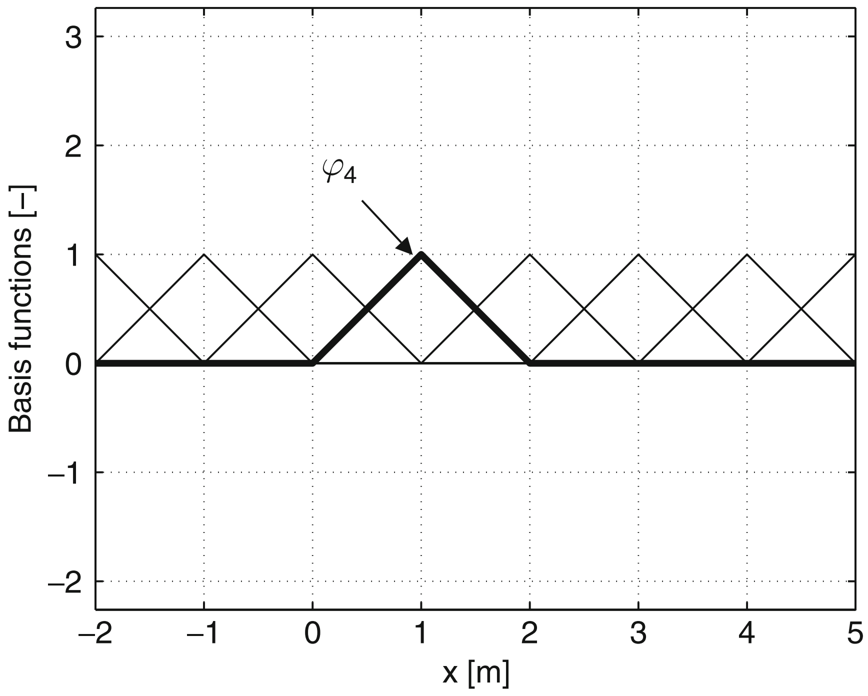
Organization of this Lecture

1. Formulate our FEM basis functions = interpolation functions
2. Use Weighted Residual Method to find the coefficients (weights/amplitude) of of our interpolation

Interpolation Functions in 1D

- ***Interpolation functions*** which are linear on each interval, one at node *i* and zero at all other nodes.
- We seek approximate solutions that are expanded in the basis functions (in the following, f will denote this approximate solution):

$$f(x) = \sum_{j=1}^{8} f_j \varphi_j(x)$$

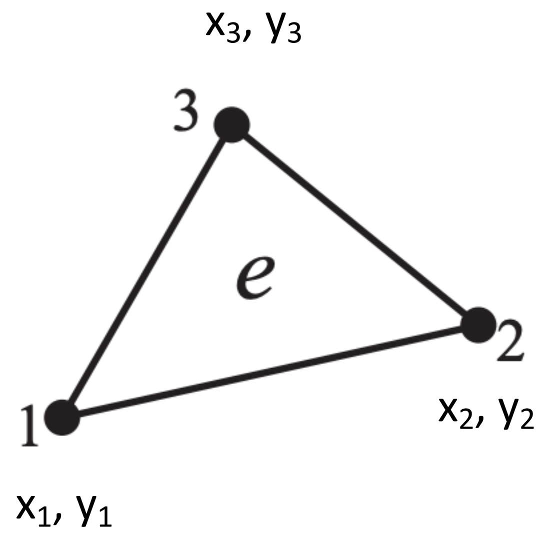


Moving to 2D - Potential in Triangle Element

- For example, the **potential in a triangular element** can be approximated as

Here Phi is potential,
not the basis function!

$$\boxed{\varphi^{(e)}(x, y) = a + bx + cy}$$



- **Applying this equation to each of the three vertices (nodes)** of the triangular element yields

$$\varphi_1^{(e)} = a + bx_1^{(e)} + cy_1^{(e)}$$

$$\varphi_2^{(e)} = a + bx_2^{(e)} + cy_2^{(e)}$$

$$\varphi_3^{(e)} = a + bx_3^{(e)} + cy_3^{(e)}$$

Interpolating Functions I

- Solving **equations for a, b and c**

$$\varphi_1^{(e)} = a + bx_1^{(e)} + cy_1^{(e)}$$

$$\varphi_2^{(e)} = a + bx_2^{(e)} + cy_2^{(e)}$$

$$\varphi_3^{(e)} = a + bx_3^{(e)} + cy_3^{(e)}$$

$$\varphi^{(e)}(x, y) = \underbrace{N_1^{(e)}}_{\text{Interpolating functions}}(x, y)\varphi_1^{(e)} + \underbrace{N_2^{(e)}}_{\text{Interpolating functions}}(x, y)\varphi_2^{(e)} + \underbrace{N_3^{(e)}}_{\text{Interpolating functions}}(x, y)\varphi_3^{(e)}$$

$$N_l^{(e)}(x, y) = \frac{1}{2\Delta^{(e)}}(a_l^{(e)} + b_l^{(e)}x + c_l^{(e)}y)$$

$$\Delta^{(e)} = \frac{1}{2}(b_1^{(e)}c_2^{(e)} - b_2^{(e)}c_1^{(e)})$$

- Substituting in

$$\varphi^{(e)}(x, y) = a + bx + cy$$

Interpolation functions are going to be our basis functions!

Interpolating Functions II

- The interpolation functions, are completely determined by the **coordinates of the three nodes**
- It can be shown that the interpolation functions so derived have the following property

$$\boxed{N_l^{(e)}(x_k^{(e)}, y_k^{(e)}) = \begin{cases} 1 & l = k \\ 0 & l \neq k \end{cases}}$$

continuity on the interpolation on the node

$$N_l^{(e)}(x, y) = \frac{1}{2\Delta^{(e)}}(a_l^{(e)} + b_l^{(e)}x + c_l^{(e)}y)$$

$$\begin{aligned}
a_1^{(e)} &= x_2^{(e)}y_3^{(e)} - x_3^{(e)}y_2^{(e)}, & b_1^{(e)} &= y_2^{(e)} - y_3^{(e)}, & c_1^{(e)} &= x_3^{(e)} - x_2^{(e)} \\
a_2^{(e)} &= x_3^{(e)}y_1^{(e)} - x_1^{(e)}y_3^{(e)}, & b_2^{(e)} &= y_3^{(e)} - y_1^{(e)}, & c_2^{(e)} &= x_1^{(e)} - x_3^{(e)} \\
a_3^{(e)} &= x_1^{(e)}y_2^{(e)} - x_2^{(e)}y_1^{(e)}, & b_3^{(e)} &= y_1^{(e)} - y_2^{(e)}, & c_3^{(e)} &= x_2^{(e)} - x_1^{(e)}
\end{aligned}$$

$$\Delta^{(e)} = \frac{1}{2}(b_1^{(e)}c_2^{(e)} - b_2^{(e)}c_1^{(e)})$$

Also this is equivalent to area of the element, see textbook

Find Potential over the Domain

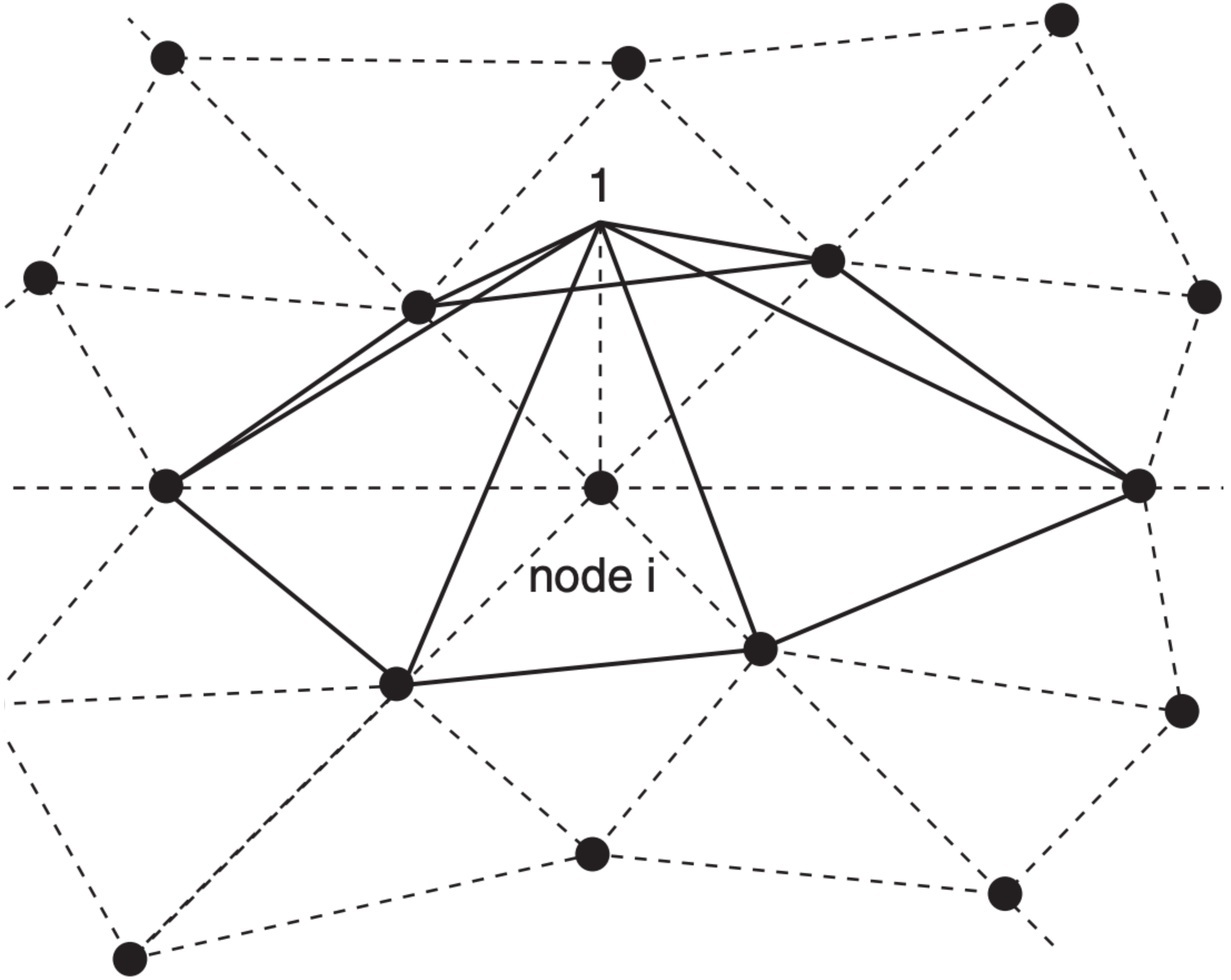
- When the potential in each element is interpolated using its values at the nodes, the potential in the entire domain can be expressed as

$$\varphi = \sum_{j=1}^{N} N_j \varphi_j + \sum_{j=1}^{N_D} N_j^D \varphi_j^D$$

- *N* denotes the total number of nodes at which the potential is unknown, and N_D denotes the number of nodes on Γ_D where the potential is given.
- **φ_j denotes the value of the potential (our unknown)** at node j and N_j is the corresponding interpolation or basis function; φ_j^D and N_j^D denote the corresponding quantities associated with Γ_D.

Interpolation Functions III

- φ_j denotes the value of the potential at node j and N_j is the corresponding interpolation or basis function
- This interpolation function comprises **the interpolation functions in all the elements that are directly connected to the associated node**



Until now it was only about picking it up the FEM functions!

Let’s apply the Weighted Residual Method to the Poisson Equation

Inner Product with Test Functions

$$-\int_{\Omega} w_i \left[\nabla\cdot(\varepsilon\nabla\varphi)\right] d\Omega = \int_{\Omega} w_i \rho_e \, d\Omega$$

Vector identity

$$w_i \left[\nabla\cdot(\varepsilon\nabla\varphi)\right] = \nabla\cdot(w_i \varepsilon\nabla\varphi) - \varepsilon\nabla\varphi\cdot\nabla w_i$$

Gauss Theorem

$$\int_{\Omega} \nabla\cdot(w_i \varepsilon\nabla\varphi)\, d\Omega = \oint_{\Gamma} \hat{n}\cdot(w_i \varepsilon\nabla\varphi)\, d\Gamma$$

$$\int_{\Omega} \varepsilon\nabla w_i \cdot\nabla\varphi \, d\Omega = \int_{\Omega} w_i \rho_e \, d\Omega + \oint_{\Gamma} \hat{n}\cdot(\varepsilon\nabla\varphi) w_i \, d\Gamma$$

$$\hat{n}\cdot(\varepsilon\nabla\varphi) = \kappa_N \qquad \text{on } \Gamma_N$$

$$\boxed{\int_{\Omega} \varepsilon\nabla w_i \cdot\nabla\varphi \, d\Omega = \int_{\Omega} w_i \rho_e \, d\Omega + \int_{\Gamma_D} \hat{n}\cdot(\varepsilon\nabla\varphi) w_i \, d\Gamma + \int_{\Gamma_N} \kappa_N w_i \, d\Gamma.}$$

Weak-Form Representation

The corresponding solution is called the weak- form solution, which satisfies our Eqs. in the <u>weighted average sense</u>

Galerkin Method

- Using Galerkin ' s method, we choose

$$w_i = N_i \qquad i = 1, 2, \ldots, N$$

- where N_i is the interpolation function associated with unknown φ_i

$$w_i = N_i \qquad i = 1, 2, \ldots, N$$

$$\int_\Omega \varepsilon \nabla w_i \cdot \nabla \varphi \, d\Omega = \int_\Omega w_i \rho_e \, d\Omega + \int_{\Gamma_D} \hat{n} \cdot (\varepsilon \nabla \varphi) w_i \, d\Gamma + \int_{\Gamma_N} \kappa_N w_i \, d\Gamma. \Rightarrow \boxed{\sum_{j=1}^{N} \varphi_j \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega = \int_\Omega \rho_e N_i \, d\Omega + \int_{\Gamma_N} \kappa_N N_i \, d\Gamma - \sum_{j=1}^{N_D} \varphi_j^D \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j^D \, d\Omega}$$

$$\varphi = \sum_{j=1}^{N} N_j \varphi_j + \sum_{j=1}^{N_D} N_j^D \varphi_j^D$$

Our Linear System to Solve

$$\sum_{j=1}^{N} \varphi_j \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega = \int_\Omega \rho_e N_i \, d\Omega + \int_{\Gamma_N} \kappa_N N_i \, d\Gamma - \sum_{j=1}^{N_D} \varphi_j^D \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j^D \, d\Omega$$

$$\sum_{j=1}^{N} K_{ij} \varphi_j = b_i \qquad i = 1, 2, \ldots, N$$

$$K_{ij} = \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega \qquad b_i = \int_\Omega \rho_e N_i \, d\Omega + \int_{\Gamma_N} \kappa_N N_i \, d\Gamma - \sum_{j=1}^{N_D} \varphi_j^D \int_\Omega \varepsilon \nabla N_i \cdot \nabla N_j^D \, d\Omega$$

$$\boxed{[K]\{\varphi\} = \{b\}}$$

Matrix Entry = Sum Over Elements

- In the actual implementation of the finite element method, we rewrite K_{ij} as

$$K_{ij} = \int_{\Omega} \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega \quad \Rightarrow \quad K_{ij} = \underline{\sum_{e=1}^{M}} \int_{\Omega^{(e)}} \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega$$

- where Ω(e) denotes the domain of element e and M denotes the total number of elements in Ω.

Assembly of Matrix K

$$K_{ij} = \sum_{e=1}^{M} \int_{\Omega^{(e)}} \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega$$

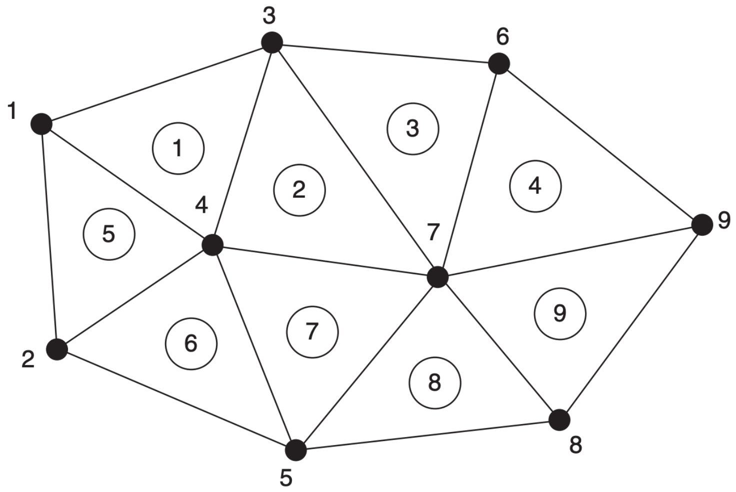
- We can process elements one by one and calculate the contribution of each element to K.
  - This process is called **assembly**.

Connectivity Array/Matrix

- To facilitate the assembly we define a connectivity array for each finite element mesh
  - the relation between the element numbers and the node numbers.
  - This connectivity array can be defined as n(3,e) where e = 1, 2, . . . , M
  - The value of n(1,e) is assigned as the node number of the first node in element *e*
  - n (2, e) the node number of the second node in element e
  - …

Connectivity Array/Matrix

We number both element and nodes



e	$n(1, e)$	$n(2, e)$	$n(3, e)$
1	1	4	3
2	4	7	3
3	3	7	6
4	7	9	6
5	2	4	1
6	2	5	4
7	5	7	4
8	5	8	7
9	7	8	9

- The numbering in this connectivity array is not **unique**. For instance, the nodes in the first element can be numbered as 4, 3, 1 or 3, 1, 4.
- Numbering does not affect the result as long as the nodes are numbered **counterclockwise**

$$\Delta^{(e)} = \frac{1}{2}(b_1^{(e)}c_2^{(e)} - b_2^{(e)}c_1^{(e)}) > 0$$

What "Meshing" means in FEM Methods?

1. Build connectivity matrix
   - element index → **node global indices**
2. Build a node to coordinate matrix
   - node global index → **coordinate**

- Remember **we need coordinate points for creating interpolation functions**!
- We provide a 2D mesh for your assignment: check these structures

Assuming that
i = n(l,e) and j = n(k,e)

$$K_{ij} = \sum_{e=1}^{M} \int_{\Omega^{(e)}} \varepsilon \nabla N_i \cdot \nabla N_j \, d\Omega$$

Only the elements that are connected to nodes i and j have non-trivial contributions to the value of K_{ij}

$$K_{ij} = \sum_{e=1}^{M} \int_{\Omega^{(e)}} \varepsilon \nabla N_l^{(e)} \cdot \nabla N_k^{(e)} d\Omega$$

where l , k = 1, 2, 3

$$K_{lk}^{(e)} = \int_{\Omega^{(e)}} \varepsilon \nabla N_l^{(e)} \cdot \nabla N_k^{(e)} d\Omega$$

Once each K_{lk} is calculated, it is added to K_{ij} where the values of i and j are given by i=n(l,e) and j = n (k,e)

Integral to be Evaluated

The integral can be evaluated either numerically or analytically

$$K_{lk}^{(e)} = \int_{\Omega^{(e)}} \varepsilon \nabla N_l^{(e)} \cdot \nabla N_k^{(e)} d\Omega$$