Modelling and Computation of Electric and Magnetic Fields

2D FEM for Poisson’s equation

Arturo Popoli

arturo.popoli@unibo.it

Department of Electrical, Electronic and Information Engineering “Guglielmo Marconi”

University of Bologna

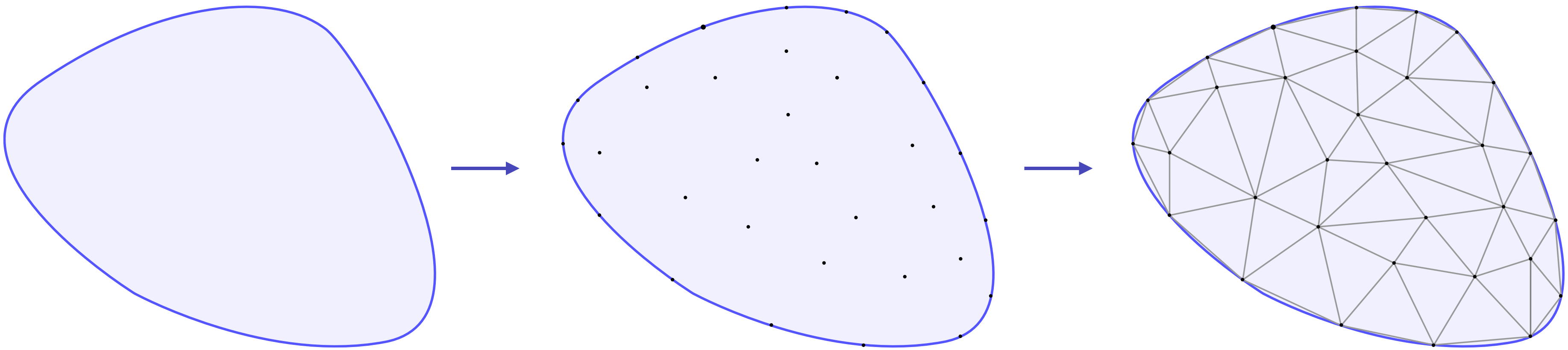
A.A. 2025-2026

From 1D to 2D

Goal: FEM discretization of Poisson's equation on 2D domains

- In 1D: $x \in [a, b]$

- In 2D



From 1D to 2D - strategy

(Strong) formulation + BC

Interpolation → Weighted residuals → $r(x, y) \neq 0$

Weak formulation

Galerkin's choice → $w \rightarrow L_k \; k = 1,2, \ldots, n$

Integration → Derive nodal equations

Element matrices → $[K_{el}]$, $[rhs_{el}]$

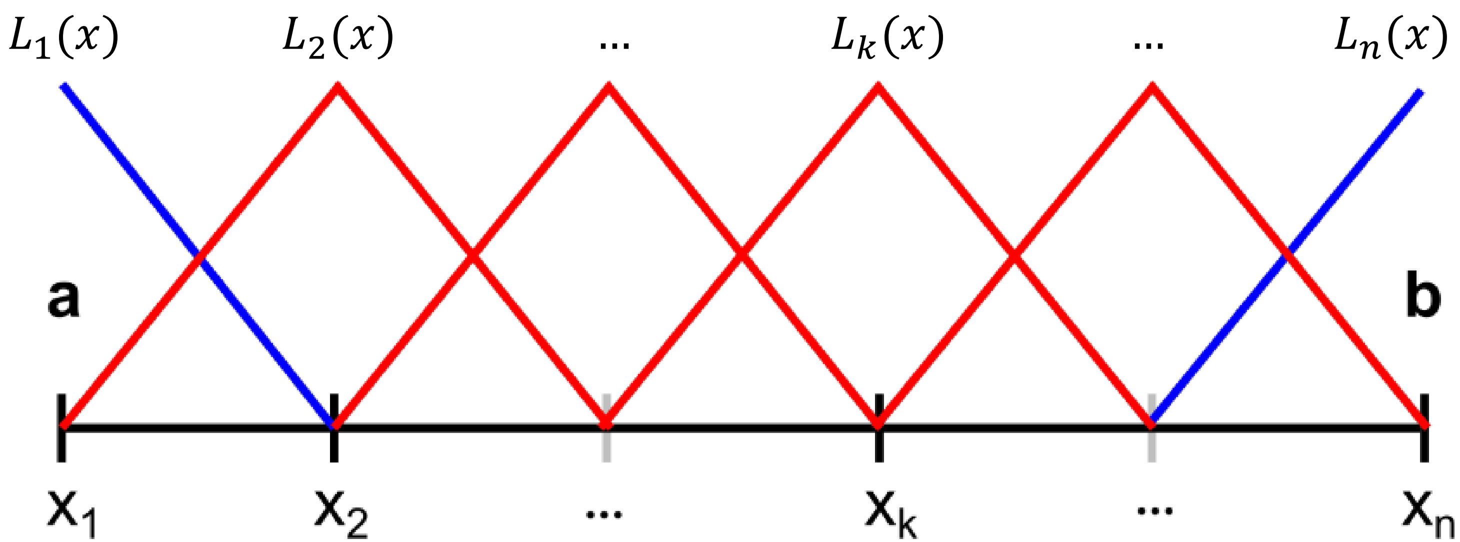
Linear system assembly and solution → $[K]\{\varphi\} = \{rhs\}$

Piecewise linear interpolation on triangles

Goal: find "equivalent" of 1D hat functions for a triangular mesh

- In 1D: approximation of a $\varphi(x)$ with piecewise function

$$\tilde{\varphi}(x) = \varphi_1 \, L_1(x) + \varphi_2 \, L_2(x) + \ldots + \varphi_n \, L_n(x) = \sum_{k=1}^{n} \varphi_k \, L_k(x)$$



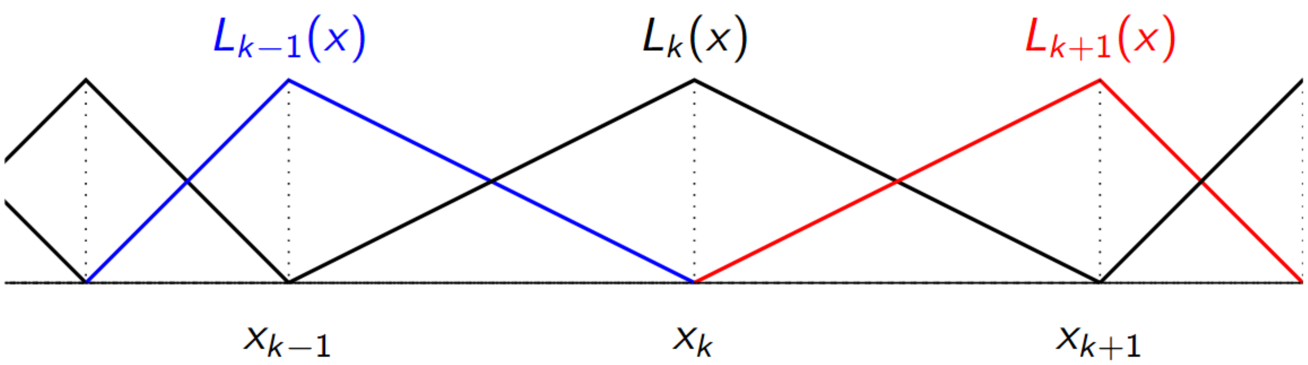
Piecewise linear interpolation on triangles

Goal: find "equivalent" of 1D hat functions for a triangular mesh

- In 1D: approximation of a $\varphi(x)$ with piecewise function

Internal nodes

$$L_k(x) = \begin{cases} 1 + \frac{x - x_k}{\Delta_-} & x \in [x_{k-1}, x_k] \\ 1 - \frac{x - x_k}{\Delta_+} & x \in [x_k, x_{k+1}] \\ 0 & x \notin [x_{k-1}, x_{k+1}] \end{cases}$$



Piecewise linear interpolation on triangles

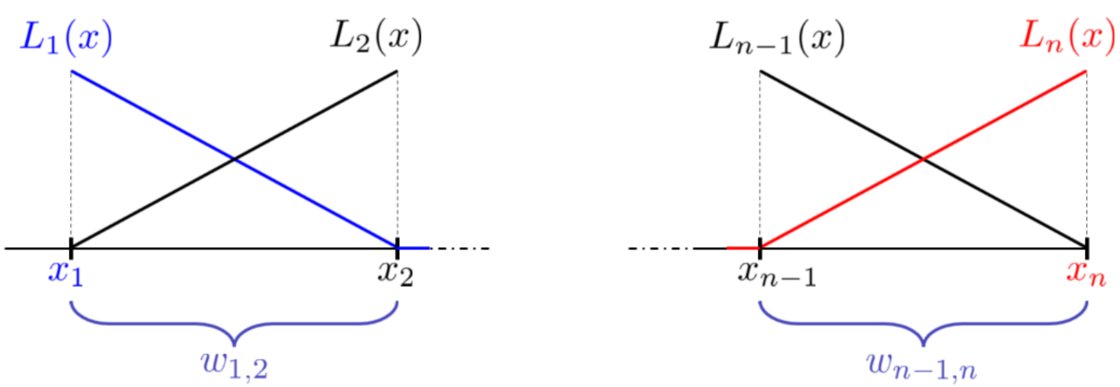
Goal: find "equivalent" of 1D hat functions for a triangular mesh

- In 1D: approximation of a $\varphi(x)$ with piecewise function

Boundary nodes

$$L_1(x) = \begin{cases} 1 - \frac{x - x_k}{\Delta_+} & x \in [x_1, x_2] \\ 0 & x \notin [x_1, x_2] \end{cases}$$

$$L_n(x) = \begin{cases} 1 + \frac{x - x_n}{\Delta_-} & x \in [x_{n-1}, x_n] \\ 0 & x \notin [x_{n-1}, x_n] \end{cases}$$

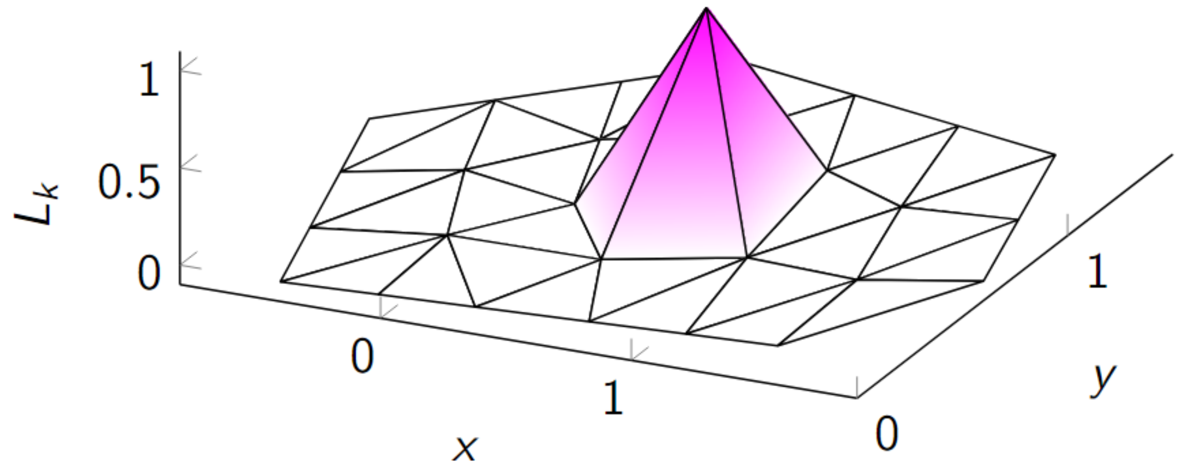


Piecewise linear interpolation on triangles

$$\tilde{\varphi}(x, y) = \varphi_1 L_1(x, y) + \varphi_2 L_2(x, y) + \cdots + \varphi_n L_n(x, y)$$

- Piecewise-linear functions in 2D

Example: $L_k(x, y)$

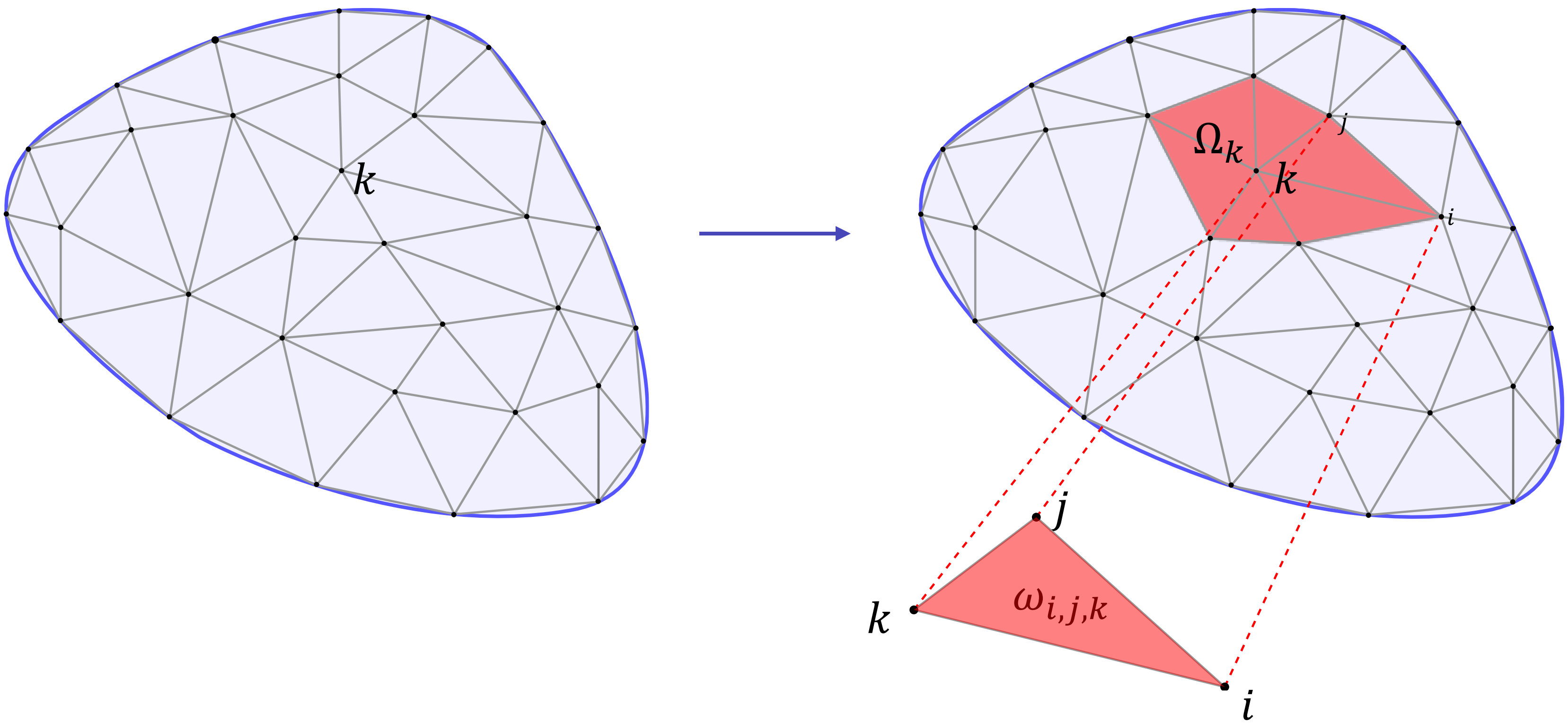


Requirements:

- Linear within Ω_k^*

*Support domain

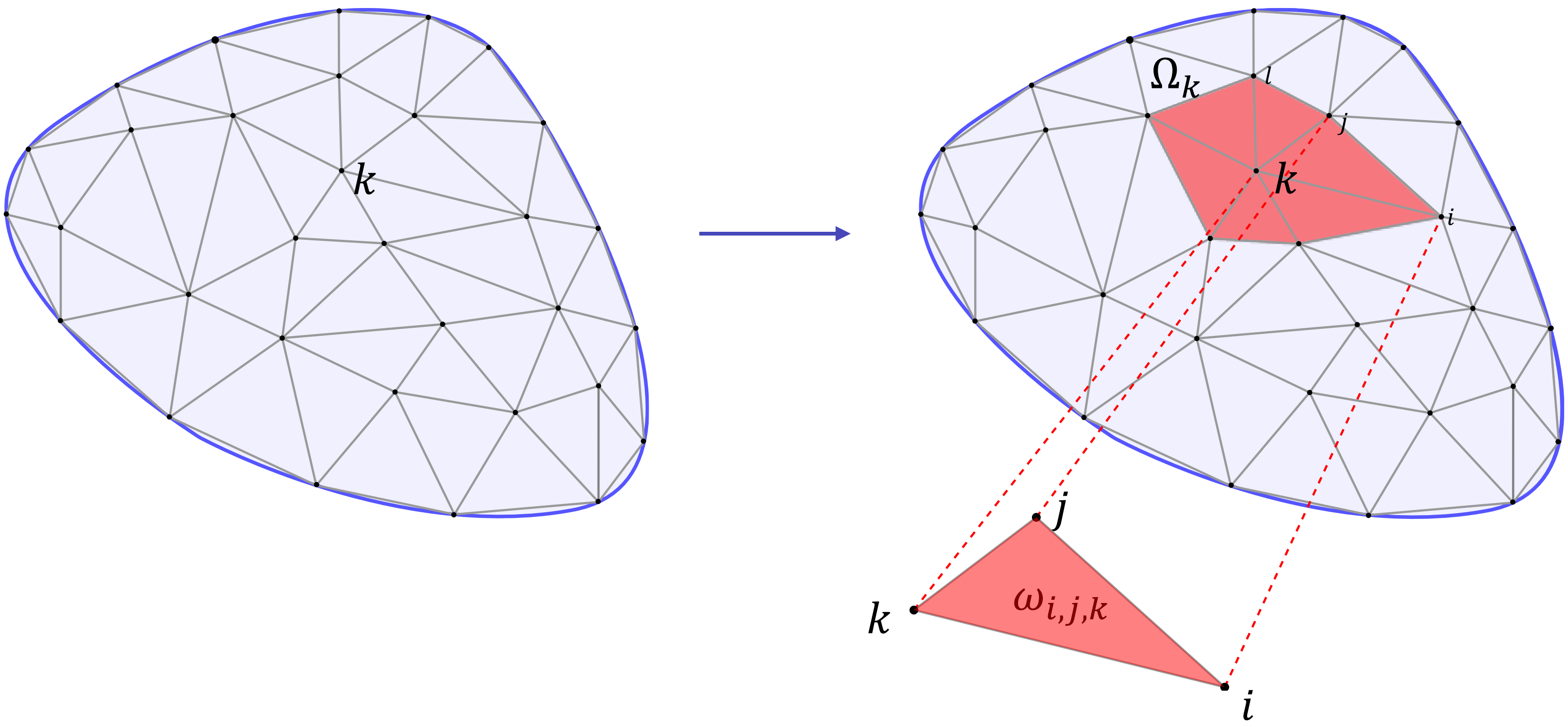
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



$$\Omega_k = \omega_{i,j,k} \cup \ \ldots$$

*Support domain

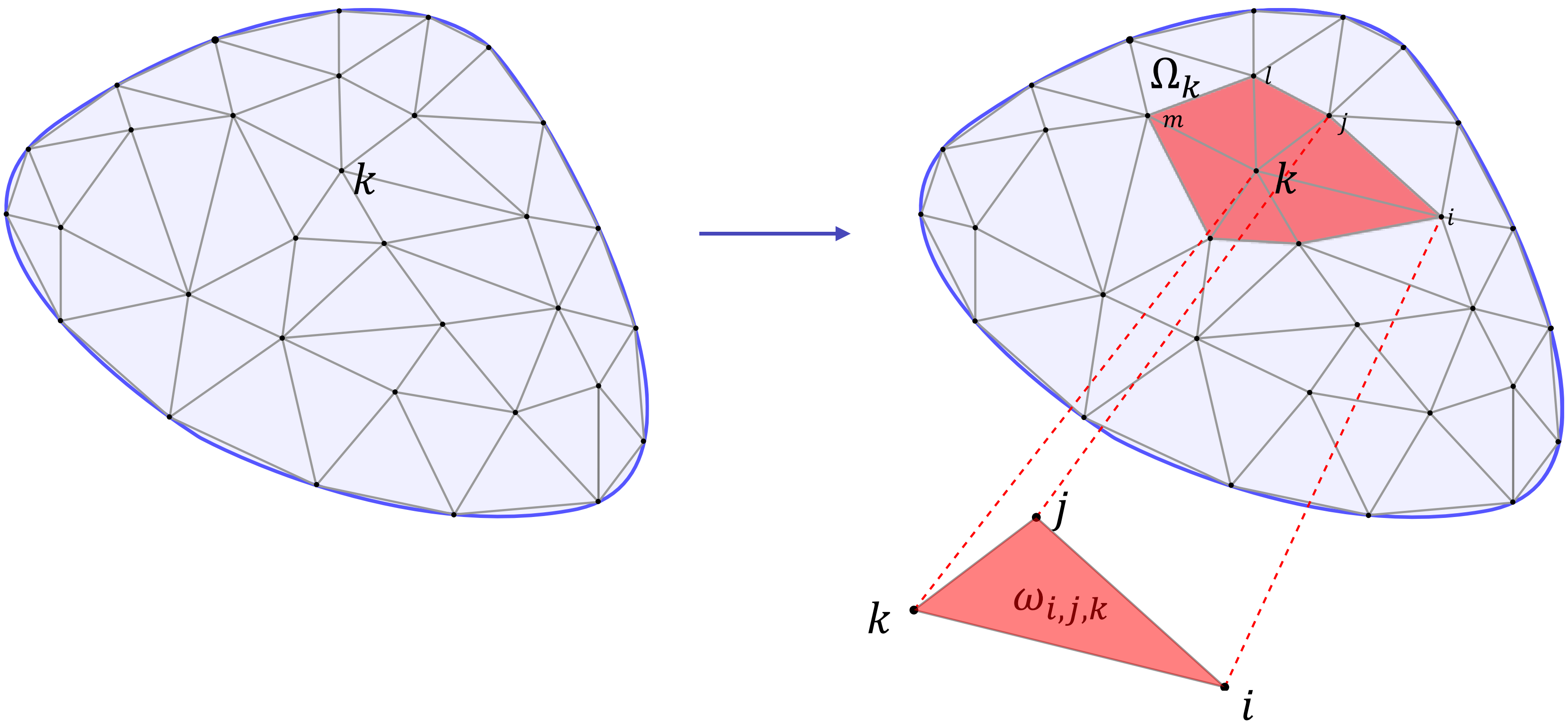
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



$$\Omega_k = \omega_{i,j,k} \cup \omega_{j,l,k} \cup \ \ldots$$

*Support domain

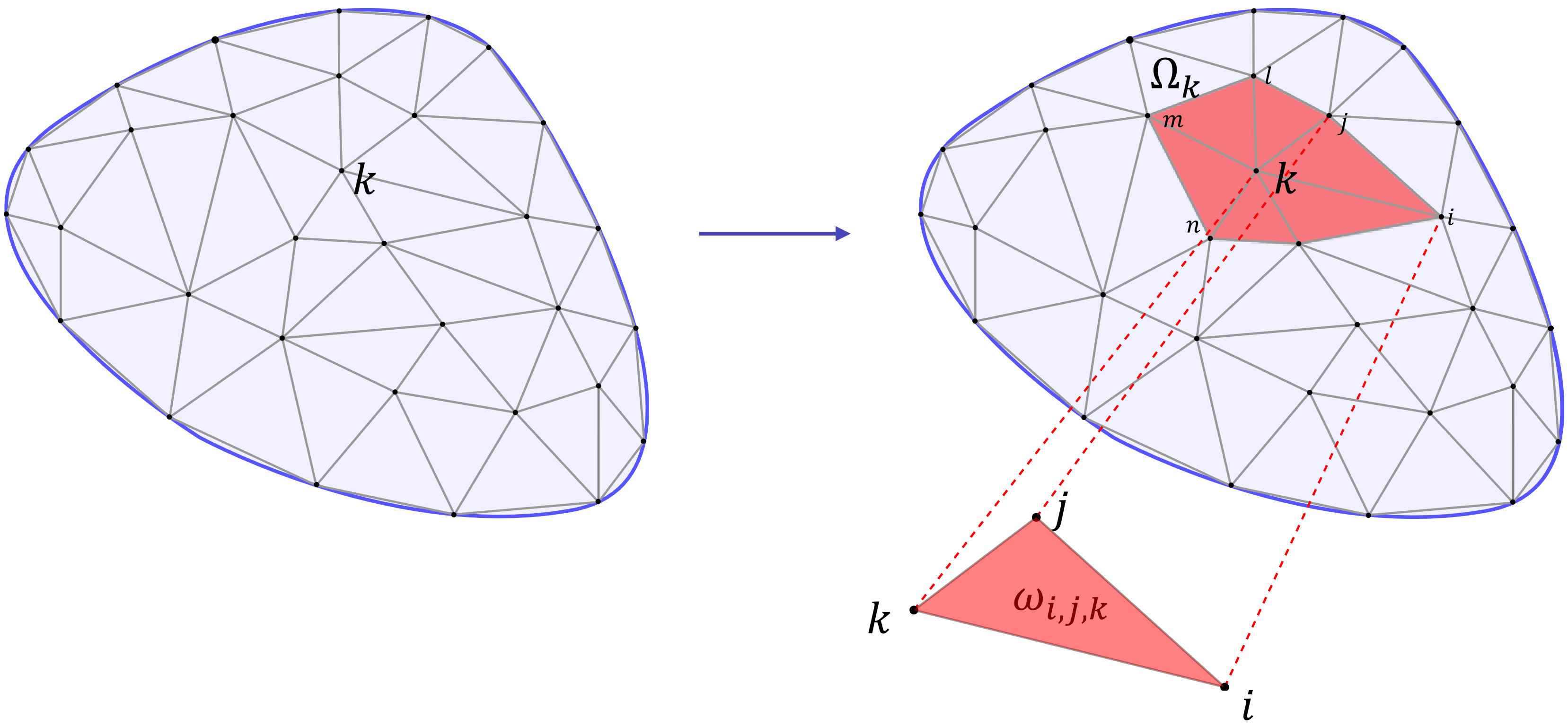
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



$$\Omega_k = \omega_{i,j,k} \cup \omega_{j,l,k} \cup \omega_{l,m,k} \cup \; \ldots$$

*Support domain

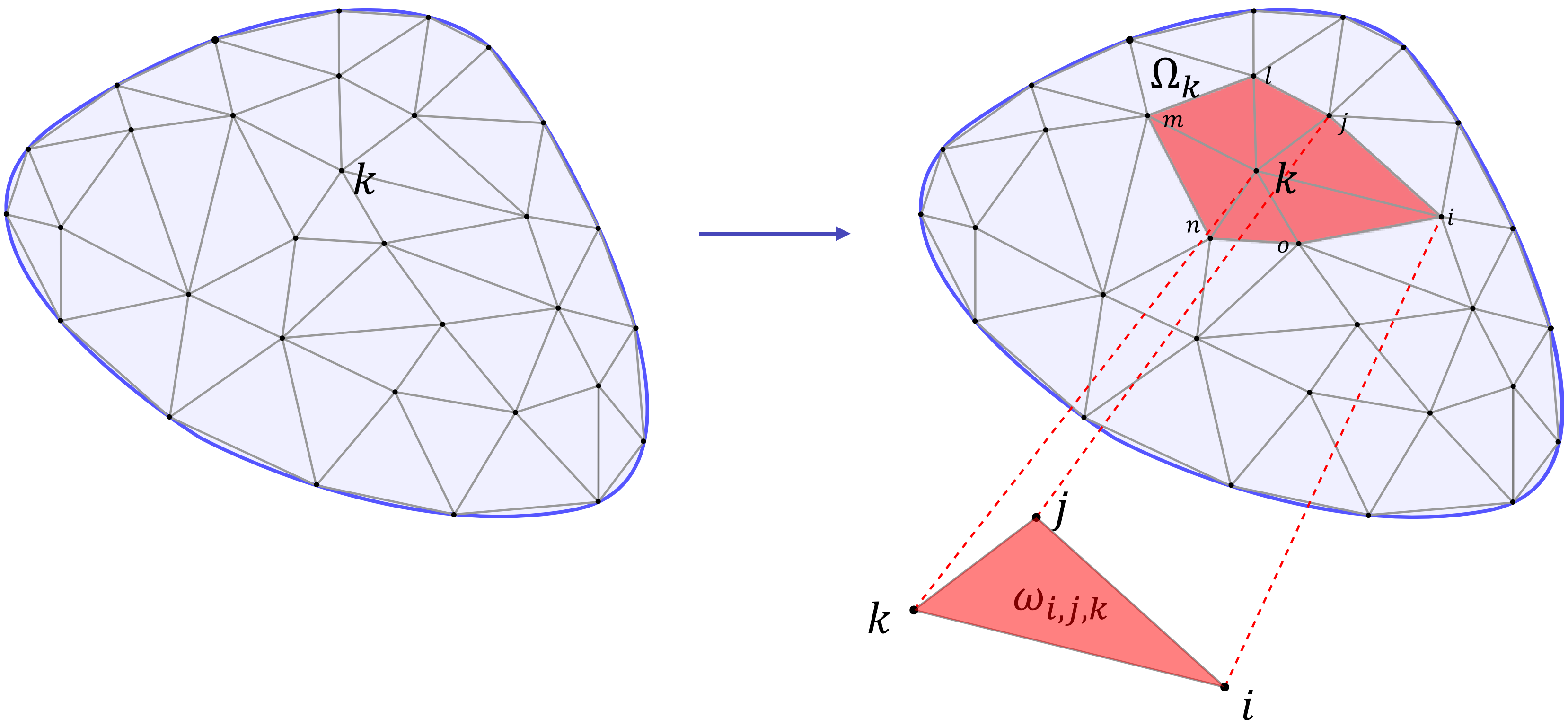
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



$$\Omega_k = \omega_{i,j,k} \cup \omega_{j,l,k} \cup \omega_{l,m,k} \cup \omega_{m,n,k} \cup \ \ldots$$

*Support domain

- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



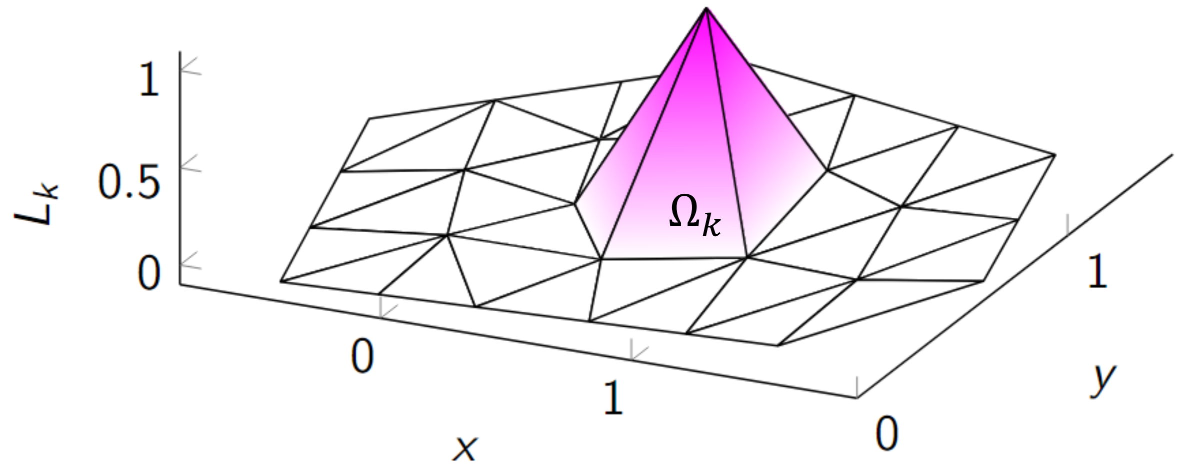
$$\Omega_k = \omega_{i,j,k} \cup \omega_{j,l,k} \cup \omega_{l,m,k} \cup \omega_{m,n,k} \cup \omega_{n,o,k}$$

Piecewise linear interpolation on triangles

$$\tilde{\varphi}(x, y) = \varphi_1 L_1(x, y) + \varphi_2 L_2(x, y) + \cdots + \varphi_n L_n(x, y)$$

- Piecewise-linear functions in 2D

Example: $L_k(x, y)$



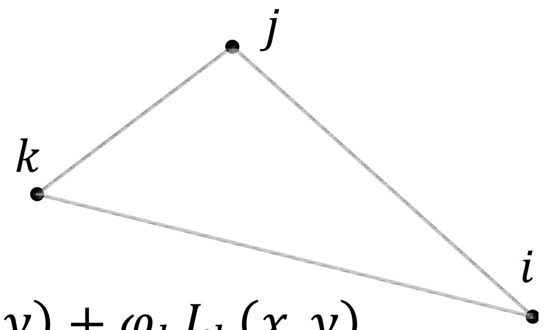
Requirements:

- Linear within Ω_k
- $L_k(x_k, y_k) = 1$ unit value on the node k
- $L_k(x_j, y_j) = 0, \forall j \neq k$ zero outside the support domain support domain Ω_k

2D interpolation – barycentric coordinates

Consider a single triangle $\omega_{i,j,k}$

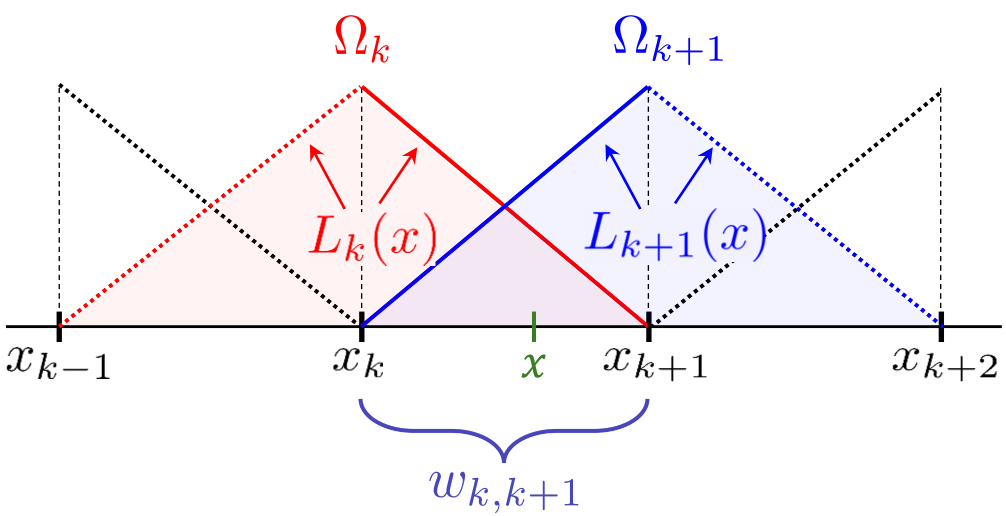
- Goal: express a function within triangle as a function nodal function values



$$\tilde{\varphi}(x,y) = \varphi_i L_i(x,y) + \varphi_j L_j(x,y) + \varphi_k L_k(x,y)$$

Note: same thing as in 1D case...

In 1D, for $x \in \omega_{k,k+1}$:



$$\varphi(x) = \varphi_k L_k(x) + \varphi_{k+1} L_{k+1}(x)$$

2D interpolation – barycentric coordinates

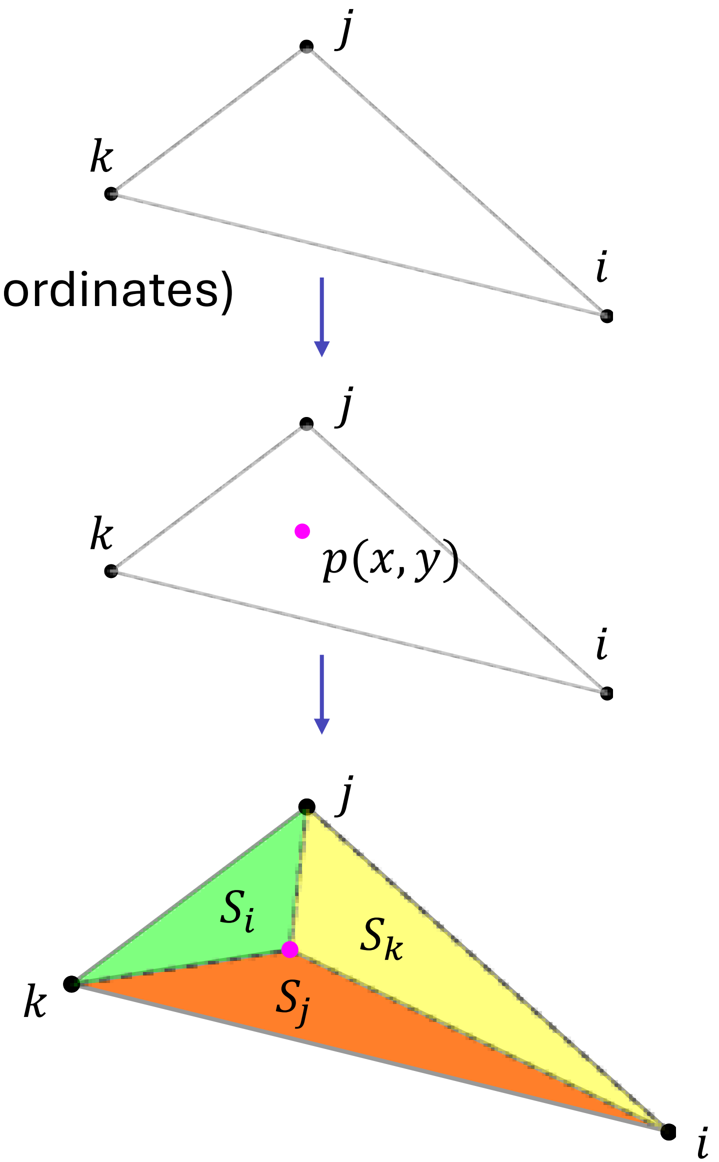
Consider a single triangle $\omega_{i,j,k}$

- Goal: derive expression for $L_k(x,y)$

New idea: **area coordinates** (barycentric coordinates)

- Introduce point $p(x,y)$ at arbitrary (x,y)
- p defines 3 sub-triangles $\in \omega_{i,j,k}$
- The three areas are functions of (x,y)
  - $S_i(x,y)$

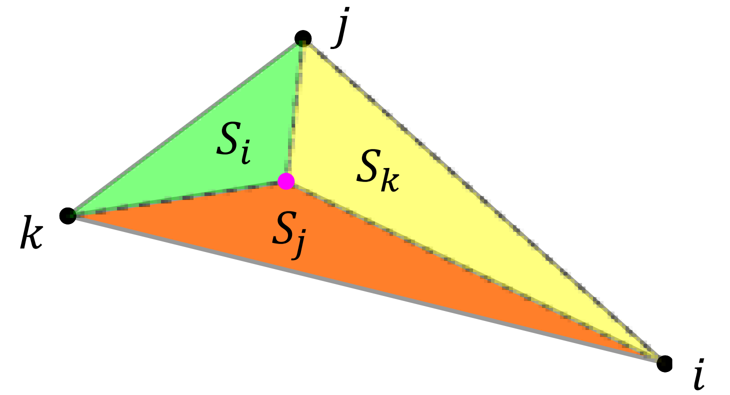
    Note: if $p(x,y)$ moves towards i, $S_i(x,y) \uparrow$
  - $S_j(x,y)$
  - $S_k(x,y)$



2D interpolation – barycentric coordinates

Consider a single triangle $\omega_{i,j,k}$

▶ Idea! Use **areas** to define shape functions



Area of sub-triangle i

$$L_i(x,y) = \frac{S_i(x,y)}{S} \qquad L_j(x,y) = \frac{S_j(x,y)}{S} \qquad L_k(x,y) = \frac{S_k(x,y)}{S}$$

Area of triangle $\omega_{i,k,k}$

▶ Computation of areas: for S we have:

$$S = \frac{1}{2}\det\begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix}$$

▶ Therefore, for the sub-triangles

$$S_i(x,y) = \frac{1}{2}\det\begin{bmatrix} 1 & x & y \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{bmatrix} \quad S_j(x,y) = \frac{1}{2}\det\begin{bmatrix} 1 & x_i & y_i \\ 1 & x & y \\ 1 & x_k & y_k \end{bmatrix} \quad S_k(x,y) = \frac{1}{2}\det\begin{bmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & y & y \end{bmatrix}$$

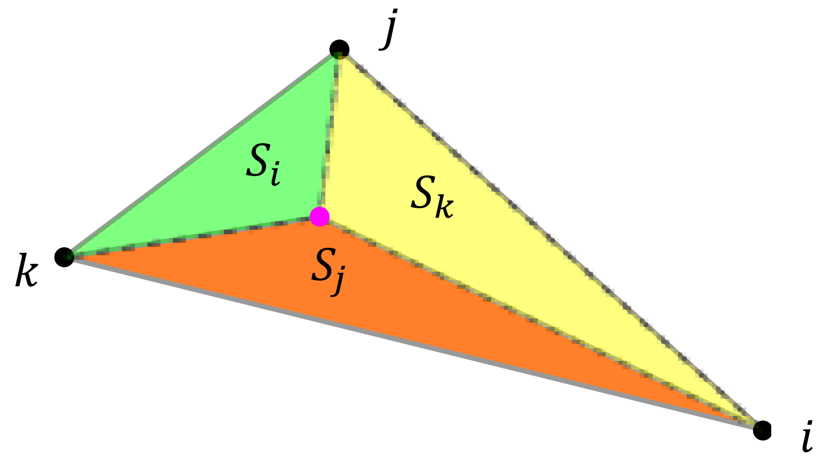
2D interpolation – barycentric coordinates

Recap – interpolation over triangles

$$\tilde{\varphi}(x,y) = \varphi_i L_i(x,y) + \varphi_j L_j(x,y) + \varphi_k L_k(x,y)$$

With:

$$\begin{cases} L_i(x,y) = \dfrac{S_i(x,y)}{S} \\ L_j(x,y) = \dfrac{S_j(x,y)}{S} \\ L_k(x,y) = \dfrac{S_k(x,y)}{S} \end{cases}$$



See previous page for calculation of $S_k(x,y)$ or S

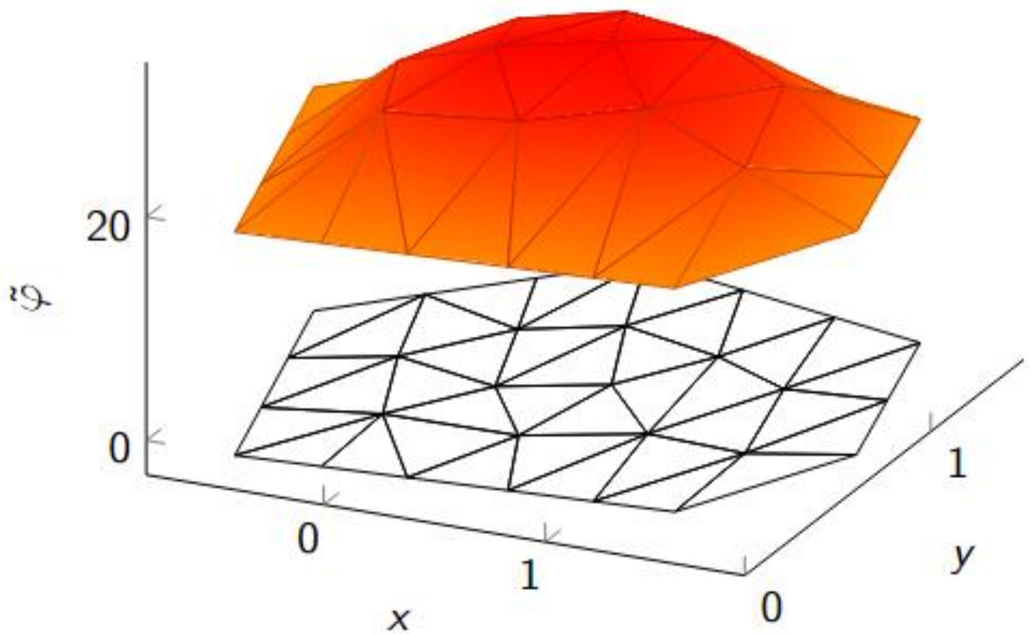
- Note (1): no “special” definitions at boundaries as in 1D
- Note (2): $L_i(x,y) + L_j(x,y) + L_k(x,y) = 1$

2D interpolation

▶ Result of interpolation

$$\tilde{\varphi}(x,y) = \varphi_i L_i(x,y) + \varphi_j L_j(x,y) + \varphi_k L_k(x,y)$$

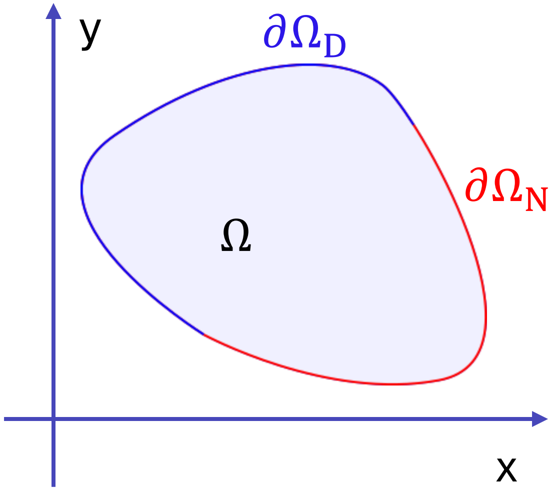
Extending to every triangle in a mesh, the interpolated 2D solution will look like this...



Formulation (2D)

Generalized Poisson problem (strong form)

$$\begin{cases} \nabla \cdot (p\nabla\varphi) = t & \Omega \\ \varphi = \varphi_0 & \partial\Omega_D \\ \dfrac{\partial\varphi}{\partial n} = \varphi_0' & \partial\Omega_N \end{cases}$$



Hp: $\frac{\partial}{\partial_z} = 0 \longrightarrow \begin{cases} \varphi = \varphi(x,y) \\ p = p(x,y) \\ t = t(x,y) \end{cases}$

The plan:

Piecewise linear interpolation

Weighted residuals

Weak form

Linear system (assembled via element matrices)

Weighted residuals

- To apply weighted residuals, we must *interpolate* the unknown function

$$\tilde{\varphi}(x,y) = \sum_k \varphi_k L_k(x,y)$$

- Residual: how much the interpolated function "fails" to satisfy formulation

$$r(x,y) = \nabla \cdot (p \nabla \tilde{\varphi}) - t$$

- We can require that the **weighted** residual is zero over the domain

$$\int_\Omega w(x,y) r(x,y)\, dS = 0$$

$$\downarrow$$

$$\int_\Omega w\, \nabla \cdot (p \nabla \tilde{\varphi})\, dS - \int_\Omega w\, t\, dS = 0$$

Towards weak formulation...

- **Problem**: $\tilde{\varphi}(x, y)$ is a piecewise-linear function $\in C_0$

...and we have a second derivative of $\tilde{\varphi}(x, y)$ in the formulation

$$\int_{\Omega} w \nabla \cdot (p \nabla \tilde{\varphi})\, dS - \int_{\Omega} w\, t\, dS = 0$$

- We need to "move" a derivative from $\tilde{\varphi}$ to w, to avoid $\pm\infty$

$$\int_{\Omega} \boxed{w \nabla \cdot (p \nabla \tilde{\varphi})}\, dS = \int_{\Omega} \nabla \cdot (w\, p \nabla \tilde{\varphi})\, dS - \int_{\Omega} p \nabla w \cdot \nabla \tilde{\varphi}\, dS$$

$$\nabla \cdot (g k \nabla f) = \boxed{g \nabla \cdot (k \nabla f)} + k \nabla f \cdot \nabla g$$ Product rule for divergence

$$\longrightarrow \boxed{g \nabla \cdot (k \nabla f)} = \nabla \cdot (g k \nabla f) - k \nabla f \cdot \nabla g$$

Towards weak formulation...

- We went from:

$$\boxed{\int_{\Omega} w \nabla \cdot (p \nabla \tilde{\varphi}) \, dS} - \int_{\Omega} w \, t \, dS = 0$$

- To...

$$\boxed{\int_{\Omega} \nabla \cdot (w \, p \nabla \tilde{\varphi}) \, dS - \int_{\Omega} p \nabla w \cdot \nabla \tilde{\varphi} \, dS} - \int_{\Omega} w \, t \, dS = 0$$

- Using divergence theorem...

$$\oint_{\partial\Omega} w \, p \nabla \tilde{\varphi} \cdot d\boldsymbol{l} - \int_{\Omega} p \nabla w \cdot \nabla \tilde{\varphi} \, dS - \int_{\Omega} w \, t \, dS = 0$$

- **Weak formulation**!

$$\int_{\Omega} p \nabla w \cdot \nabla \tilde{\varphi} \, dS = - \int_{\Omega} w \, t \, dS + \oint_{\partial\Omega} w \, p \nabla \tilde{\varphi} \cdot d\boldsymbol{l}$$

Galerkin approach

- Weighted residuals

$$\int_{\Omega} p\nabla w \cdot \nabla\tilde{\varphi}\, dS = -\int_{\Omega} w\, t\, dS + \oint_{\partial\Omega} w\, p\nabla\tilde{\varphi} \cdot d\boldsymbol{l}$$

We need to go from 1 equation to n-equations, otherwise cannot solve for φ_1, φ_2 ….

Galerkin's choice

$w(x,y) \rightarrow L_k(x,y),\ k = 1,2, \dots n$

$$\int_{\Omega} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = -\int_{\Omega} L_k\, t\, dS + \oint_{\partial\Omega} L_k p\nabla\tilde{\varphi} \cdot d\boldsymbol{l} \qquad k = 1,2, \dots, n$$

- Note (1): n-integrals, for $w = L_1(x), L_2(x), \dots$
- Note (2): every integral is on the whole 2D domain (Ω)
  - …we will be helped by the *locality* of the shape functions

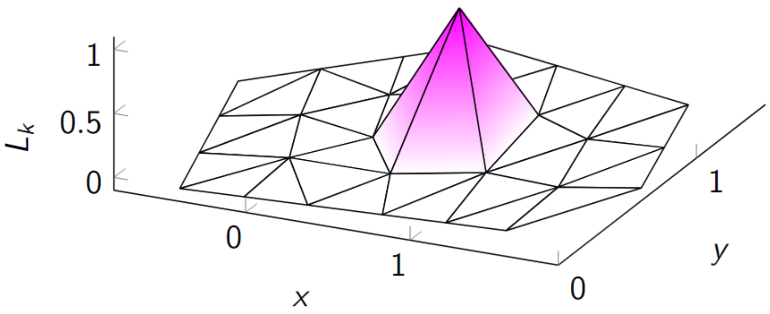
GOAL

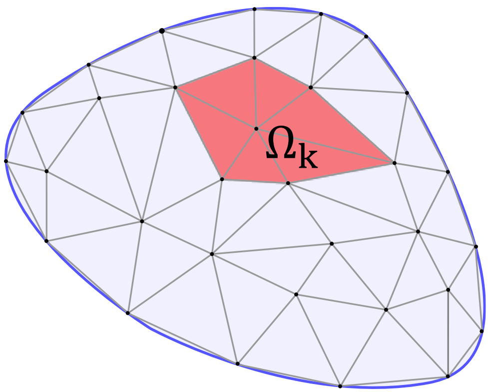
- …find approximated nodal values of φ

First term

$$\boxed{\int_{\Omega} p \nabla L_k \cdot \nabla \tilde{\varphi}\, dS} = -\int_{\Omega} L_k\, t\, dS + \oint_{\partial\Omega} L_k\, p \nabla \tilde{\varphi} \cdot d\boldsymbol{l}$$

L_k, its gradient, and, as a consequence, the kernel of the integral, assume non zero values only in its support domain Ω_k





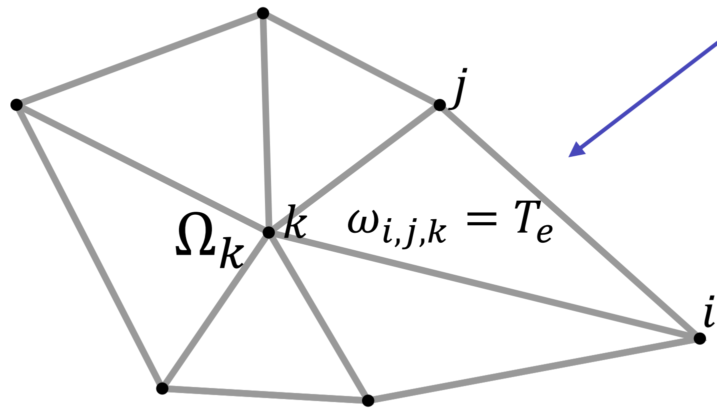
Therefore:

$$\boxed{\int_{\Omega} p \nabla L_k \cdot \nabla \tilde{\varphi}\, dS} = \int_{\Omega_k} p \nabla L_k \cdot \nabla \tilde{\varphi}\, dS$$

First term

- The integral can be expressed as a **sum of partial integrals** over each triangular element T_e within the support domain Ω_k

$T_e = e - th$ element of the triangular mesh

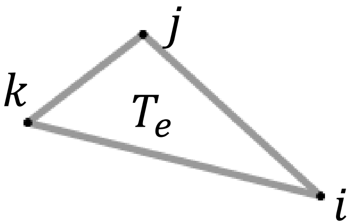


Sum of integrals over triangles within support domain

$$\int_{\Omega_k} p \nabla L_k \cdot \nabla \tilde{\varphi} \, dS = \sum_{T_k \in \Omega_k} \int_{T_k} p \nabla L_k \cdot \nabla \tilde{\varphi} \, dS$$

...we need to understand how to compute this integral!

Integral over an element



- On T_e (one of the elements $\in \Omega_k$):

$$\tilde{\varphi}(x,y) = \varphi_i L_i(x,y) + \varphi_j L_j(x,y) + \varphi_k L_k(x,y)$$

$$\nabla\tilde{\varphi}(x,y) = \varphi_i \nabla L_i(x,y) + \varphi_j \nabla L_j(x,y) + \varphi_k \nabla L_k(x,y)$$

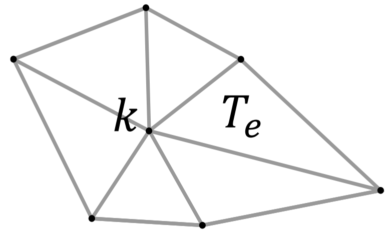
- Substituting in $\int_{T_e} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS$

$$\int_{T_e} p\nabla L_k \cdot \left(\varphi_i \nabla L_i + \varphi_j \nabla L_j + \varphi_k \nabla L_k\right) dS =$$

$$= \left[\int_{T_e} p\nabla L_k \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_k\, dS\right]\varphi_k$$

NOTE: three **constants** multiplying nodal values

…this is for **one** triangle within Ω_k

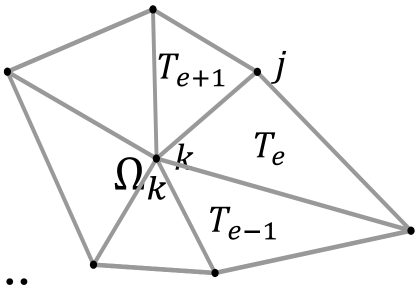


...wait, recap NOW!

We started from the weak form of the equation (whole domain!)

$$\boxed{\int_{\Omega} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS} = -\int_{\Omega} L_k\, t\, dS + \oint_{\partial\Omega} L_k\, p\nabla\tilde{\varphi} \cdot d\boldsymbol{l} \qquad k = 1,2,\dots,n$$

...then we focused only on the first term of k –th equation, (restricting integration domain to the support domain of a single node)



$$\int_{\Omega_k} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = \sum_{T_k \in \Omega_k} \int_{T_k} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS =$$

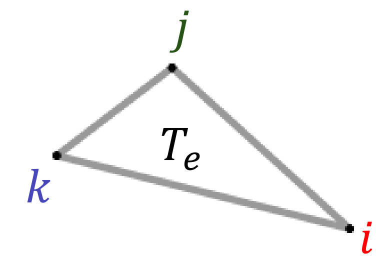
$$= \dots + \int_{T_{e-1}} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS + \int_{T_e} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS + \int_{T_{e+1}} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS + \dots$$

$$\int_{T_e} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS =$$

$$= \left[\int_{T_e} p\nabla L_k \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_k\, dS\right]\varphi_k$$

...so this is the ***contribution*** of triangle e to the equation for node k

Towards the element matrix



Goal: collect all contributions of T_e inside an element matrix $[K_{el}]$

...we already know the contribution to node k

$$\int_{T_e} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_k \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_k\, dS\right]\varphi_k$$

$$\tilde{\varphi} = \varphi_i L_i + \varphi_j L_j + \varphi_k L_k$$

...for nodes i and j:

$$\int_{T_e} p\nabla L_i \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_i \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_i \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_i \cdot \nabla L_k\, dS\right]\varphi_k$$

$$\int_{T_e} p\nabla L_j \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_j \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_j \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_j \cdot \nabla L_k\, dS\right]\varphi_k$$

...so each element is responsible for 9 coefficients!

Element matrix $[K]_{el}$

Goal: collect all contributions of a single element T_e inside an element matrix $[K_{el}]$

$$\int_{T_e} p\nabla L_i \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_i \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_i \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_i \cdot \nabla L_k\, dS\right]\varphi_k$$

$$\int_{T_e} p\nabla L_j \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_j \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_j \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_j \cdot \nabla L_k\, dS\right]\varphi_k$$

$$\int_{T_e} p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = \left[\int_{T_e} p\nabla L_k \cdot \nabla L_i\, dS\right]\varphi_i + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_j\, dS\right]\varphi_j + \left[\int_{T_e} p\nabla L_k \cdot \nabla L_k\, dS\right]\varphi_k$$

$$\downarrow$$

$$[K]_{el} = \begin{bmatrix} \int_{T_e} p\nabla L_i \cdot \nabla L_i\, dS & \int_{T_e} p\nabla L_i \cdot \nabla L_j\, dS & \int_{T_e} p\nabla L_i \cdot \nabla L_k\, dS \\ \int_{T_e} p\nabla L_j \cdot \nabla L_i\, dS & \int_{T_e} p\nabla L_j \cdot \nabla L_j\, dS & \int_{T_e} p\nabla L_j \cdot \nabla L_k\, dS \\ \int_{T_e} p\nabla L_k \cdot \nabla L_i\, dS & \int_{T_e} p\nabla L_k \cdot \nabla L_j\, dS & \int_{T_e} p\nabla L_k \cdot \nabla L_k\, dS \end{bmatrix}$$

Element matrix $[K]_{el}$

Compact form of the matrix

$$[K]_{el} = \int_e \begin{bmatrix} p\nabla L_i \cdot \nabla L_i & p\nabla L_i \cdot \nabla L_j & p\nabla L_i \cdot \nabla L_k \\ p\nabla L_j \cdot \nabla L_i & p\nabla L_j \cdot \nabla L_j & p\nabla L_j \cdot \nabla L_k \\ p\nabla L_k \cdot \nabla L_i & p\nabla L_k \cdot \nabla L_j & p\nabla L_k \cdot \nabla L_k \end{bmatrix} dS$$

Let's introduce the auxiliar matrix $[\nabla L]_{el}$ (shape function gradient matrix)

$$[\nabla L]_{el} = \begin{bmatrix} \frac{\partial L_i}{\partial x} & \frac{\partial L_j}{\partial x} & \frac{\partial L_k}{\partial x} \\ \frac{\partial L_i}{\partial y} & \frac{\partial L_j}{\partial y} & \frac{\partial L_k}{\partial y} \end{bmatrix}$$

To rewrite $[K_{el}]$ in an even more compact form

$$[K]_{el} = \int_{T_e} p(x,y)[\nabla L]^T[\nabla L]\, \mathrm{d}S$$

Second term (rhs)

Back to the weighted residuals

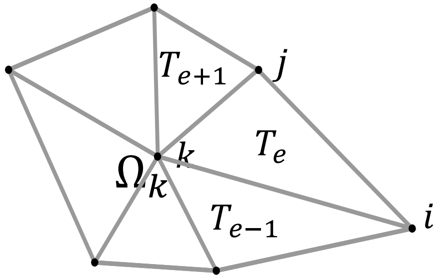
$$\int_\Omega p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = \boxed{-\int_\Omega L_k\, t\, dS} + \oint_{\partial\Omega} L_k\, p\nabla\tilde{\varphi}\cdot d\boldsymbol{l}$$

Non-zero values only on the support domain Ω_k

Again, can be decomposed in contributions of each triangle T_k in Ω_k

$$\boxed{\int_\Omega L_k\, t\, dS} = \sum_{T_k\in\Omega_k}\int_{T_k} L_k(x,y)t(x,y)dS =$$

$$= \cdots + \int_{T_{e-1}} L_k t\, dS + \underbrace{\int_{T_e} L_k t\, dS}_{} + \int_{T_{e+1}} L_k t\, dS + \cdots$$



...***contribution*** of triangle e to the (rhs of) equation for node k

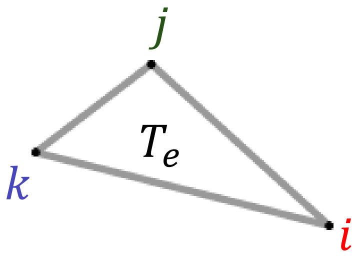
Second term (rhs)

T_e has other two vertices → contributions to nodal equations for i and j:

node k $\longrightarrow \int_{T_e} L_k(x,y)t(x,y)\,dS$

node i $\longrightarrow \int_{T_e} L_i(x,y)t(x,y)\,dS$

node j $\longrightarrow \int_{T_e} L_j(x,y)t(x,y)\,dS$



$$\{rhs\}_{el} = -\begin{Bmatrix} \int_{T_e} L_i(x,y)t(x,y)\,dS \\ \int_{T_e} L_j(x,y)t(x,y)\,dS \\ \int_{T_e} L_k(x,y)t(x,y)\,dS \end{Bmatrix}$$

Second term (rhs)

$$\{rhs\}_{el} = -\begin{Bmatrix} \int_{T_e} L_i(x,y)t(x,y)\,dS \\ \int_{T_e} L_j(x,y)t(x,y)\,dS \\ \int_{T_e} L_k(x,y)t(x,y)\,dS \end{Bmatrix}$$

In compact form

$$\{rhs\}_{el} = -\int_{T_e} \begin{Bmatrix} L_i(x,y)\,t(x,y) \\ L_j(x,y)\,t(x,y) \\ L_k(x,y)\,t(x,y) \end{Bmatrix} \mathrm{d}S$$

Introducing the element shape functions array $\{L\} = \begin{Bmatrix} L_i(x,y) \\ L_j(x,y) \\ L_k(x,y) \end{Bmatrix}$

Compact expression for the **element rhs array**

$$\{rhs\}_{el} = -\int_{T_e} \{L\}\, t(x,y)\,\mathrm{d}S$$

Third term (boundaries)

- ...what about third (and last) term?

$$\int_\Omega p\nabla L_k \cdot \nabla\tilde{\varphi}\, dS = -\int_\Omega L_k\, t\, dS + \boxed{\oint_{\partial\Omega} L_k\, p\nabla\tilde{\varphi}\cdot d\boldsymbol{l}}$$

$$[K]_{el} = \int_{T_e} p(x,y)[\nabla L]^T[\nabla L]\,\mathrm{d}S$$

$$\{rhs\}_{el} = -\int_{T_e} \{L\}\, t(x,y)\,\mathrm{d}S$$

Calculated on **boundary**

For any internal node k, L_k is zero at the boundary

Non-zero only if k is on boundary

Dirichlet BCs

$$\varphi_k = \varphi_0(x,y)$$

Neumann BCs

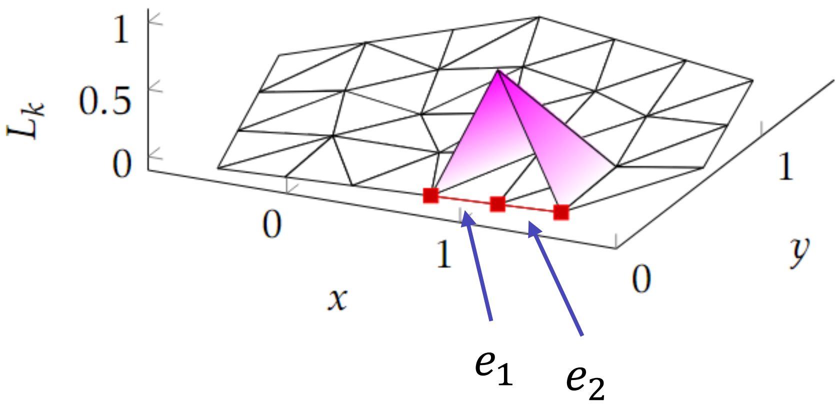
Boundary term:

$$\oint_{\partial\Omega} L_k \, p \nabla\tilde{\varphi} \cdot d\boldsymbol{l}$$

Since $L_k \neq 0$ only in support domain $\Omega_k \in \partial\Omega$ formed by e_1 and e_2

$$\oint_{\partial\Omega} L_k \, p \nabla\tilde{\varphi} \cdot d\boldsymbol{l} = \oint_{\partial\Omega} L_k \, p \frac{\partial\tilde{\varphi}}{\partial n} \, dl =$$

$$= \int_{e_1} L_k \, p \frac{\partial\tilde{\varphi}}{\partial n} \, dl + \int_{e_2} L_k \, p \frac{\partial\tilde{\varphi}}{\partial n} \, dl$$



Prescribed by Neumann BC! $\nabla\tilde{\varphi} \cdot \hat{\mathbf{n}} = \dfrac{\partial\tilde{\varphi}}{\partial n} = \varphi_0'(x, y)$

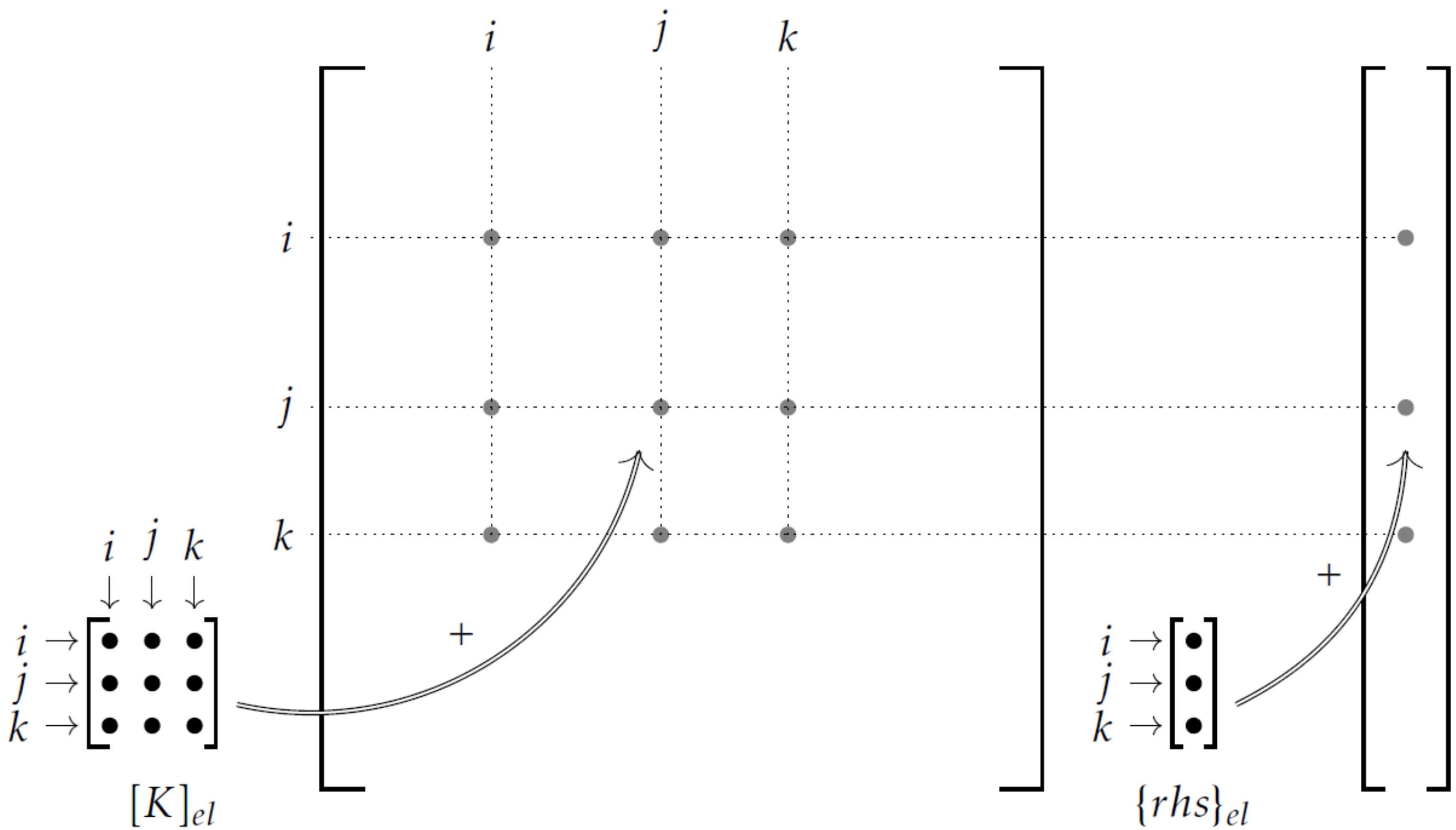
$$\rightarrow \oint_{\partial\Omega} L_k \, p \nabla\tilde{\varphi} \cdot d\boldsymbol{l} = \int_{e_1} L_k \, p \, \varphi_0' \, dl + \int_{e_2} L_k \, p \, \varphi_0' \, dl$$

Assembly of $[K]_{el}$ and $[rhs]_{el}$ into $[K]$ and $[rhs]$

A triangle contributes with 9 coefficients to 3 nodal equations

T_e is defined by nodes i, j and k:

→ 3 contributions for equation i, 3 contributions for eq. j,... k

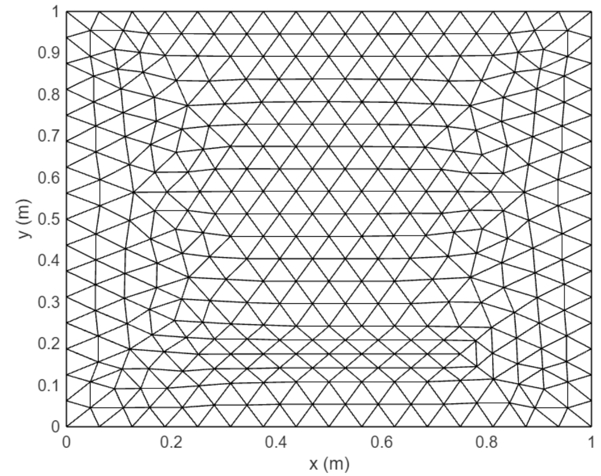


...and now let's try to make some practice!

- Install gmsh (https://gmsh.info/)
- Open the FEM_2D_main.mlx

set_BCs_on_nodes

set_BCs_on_nodes assignes BCs to boundary nodes based on BCs on edges



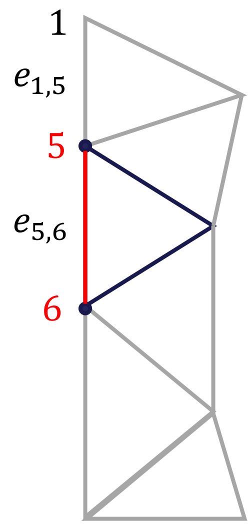

```
>> msh.LINES

ans =

     1     5     5
     5     6     5
     ...
```

edge

node 1 node 2 edge tag

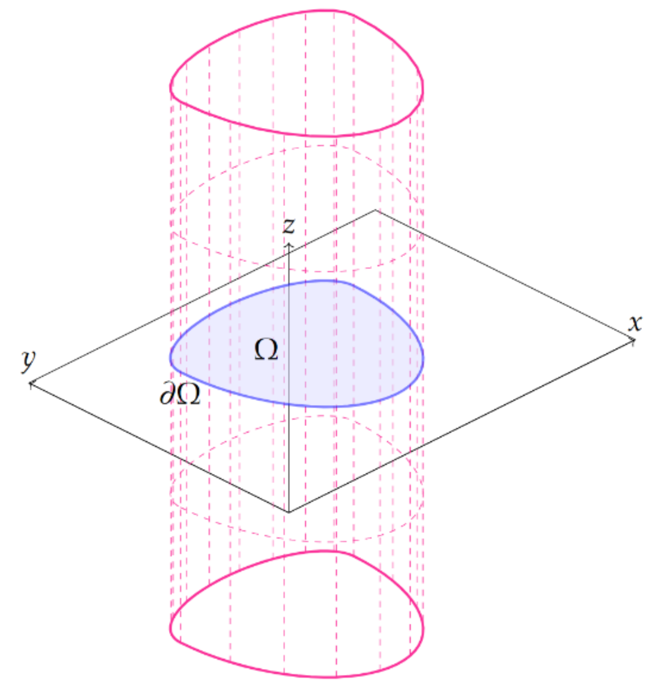


“Real meaning” of 2D…

Assume that $\frac{\partial}{\partial z} = 0$

Solution is the same for every *slice* along z

- $p(\boldsymbol{r}) \rightarrow p(x, y)$
- $t(\boldsymbol{r}) \rightarrow t(x, y)$
- $\varphi(\boldsymbol{r}) \rightarrow \varphi(x, y)$



We can study the problem in 2D!

$$\boxed{\int_V \nabla w \cdot [p(\boldsymbol{x})\, \nabla \tilde{\varphi}]\, \mathrm{d}V = -\int_V w(\boldsymbol{x})\, s(\boldsymbol{x})\, \mathrm{d}V + \oint_S w(\boldsymbol{x})\, p(\boldsymbol{x})\, \nabla \tilde{\varphi} \cdot \hat{\boldsymbol{n}} \mathrm{d}S}$$

$$\Big\downarrow \text{ 2D}$$

$$\boxed{\int_\Omega \nabla w \cdot [p(x,y)\, \nabla \tilde{\varphi}]\, \mathrm{d}S = -\int_\Omega w(x,y)\, s(x,y)\, \mathrm{d}S + \oint_{\partial\Omega} w(x,y)\, p(x,y)\, \nabla \tilde{\varphi} \cdot \hat{n} \mathrm{d}l}$$

Appendix: weak form derivation in 3D

$$\int_V w(\boldsymbol{r})\, r(\boldsymbol{r})\, \mathrm{d}V = \int_V w(\boldsymbol{r})\, \nabla \cdot [p(\boldsymbol{x}) \nabla \tilde{\varphi}] \mathrm{d}V - \int_V w(\boldsymbol{r})\, s(\boldsymbol{r})\, \mathrm{d}V = 0$$

$$\nabla \cdot (g \bar{F}) = \nabla g \cdot \bar{F} + g\, \nabla \cdot \bar{F}$$

$$\int_V \nabla \cdot (g \bar{F})\, dV = \int_V \nabla g \cdot \bar{F}\, dV + \int_V g\, \nabla \cdot \bar{F}\, dV$$

Gauss Theorem

$$\oint_S g \bar{F} \cdot \hat{n}\, dS = \int_V \nabla g \cdot \bar{F}\, dV + \int_V g\, \nabla \cdot \bar{F}\, dV$$

Reordering:

$$\int_V g\, \nabla \cdot \bar{F}\, dV = -\int_V \nabla g \cdot \bar{F}\, dV + \oint_S g\, F_n\, dS$$