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

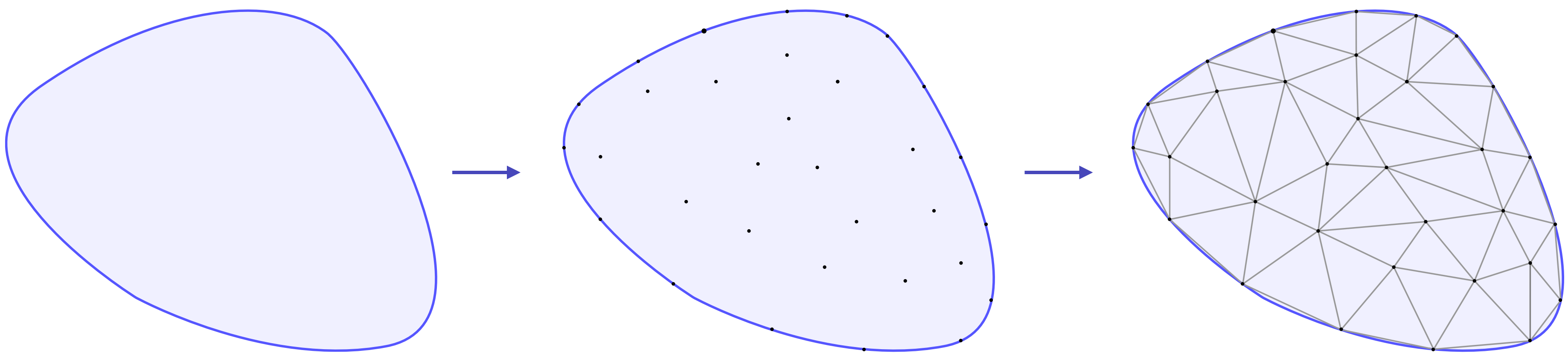
A.A. 2025-2026

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 → $w \rightarrow L_k \; k = 1,2, \dots, n$

↓

Integration → Derive nodal equations

↓

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

↓

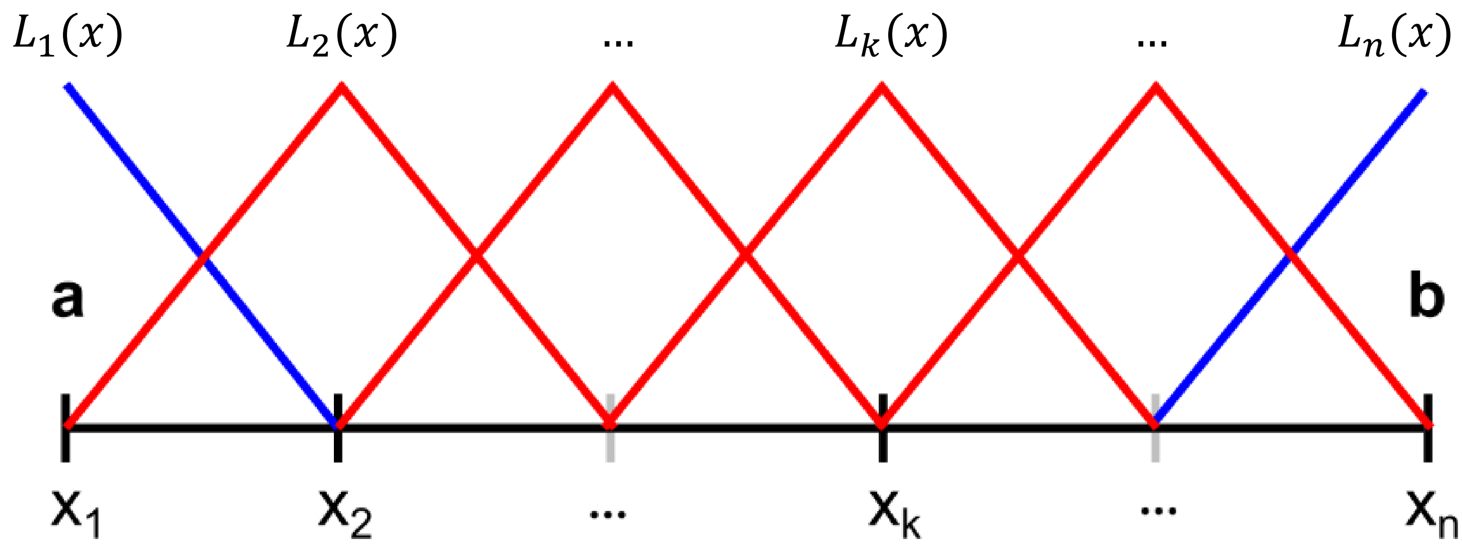
Linear system assembly and solution → $[K]\{\varphi\} = \{rhs\}$

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

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



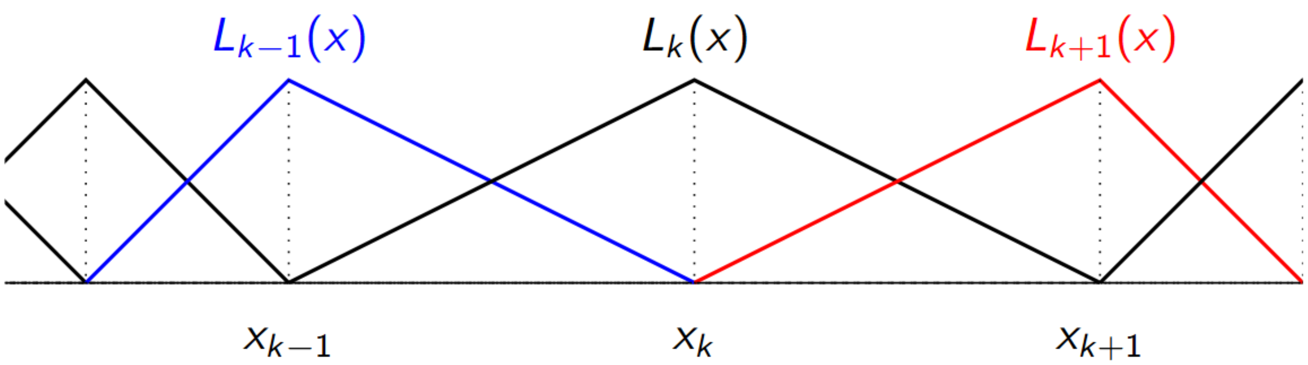
Piecewise linear interpolation on triangles

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

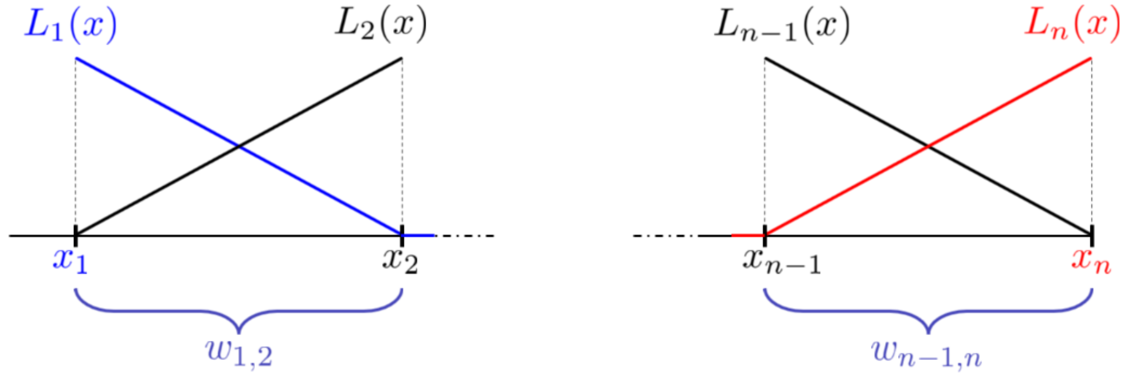
Goal: find "equivalent" of 1D hat functions for a triangular mesh

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

Boundary nodes

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

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

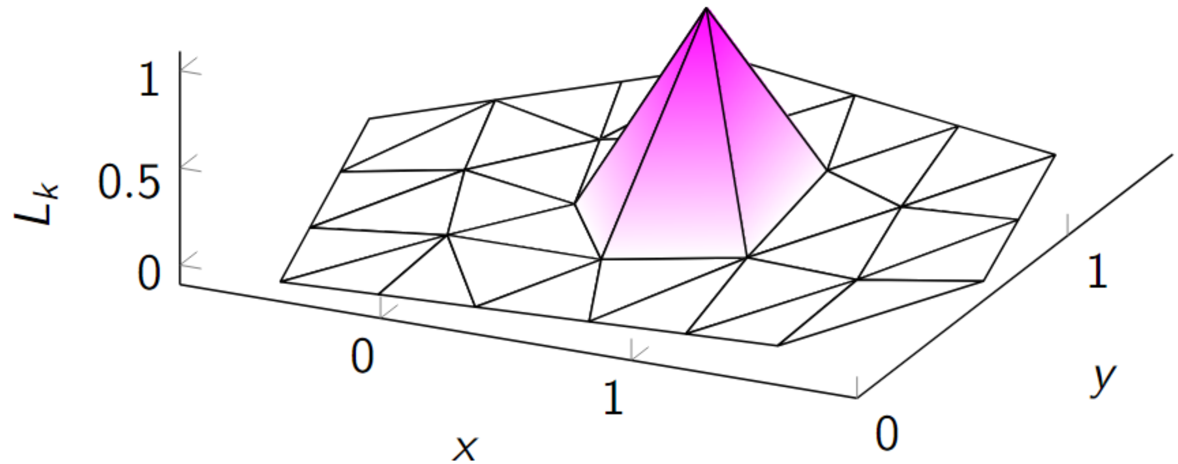


Piecewise linear interpolation on triangles

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

- Piecewise-linear functions in 2D

Example: $L_k(x, y)$

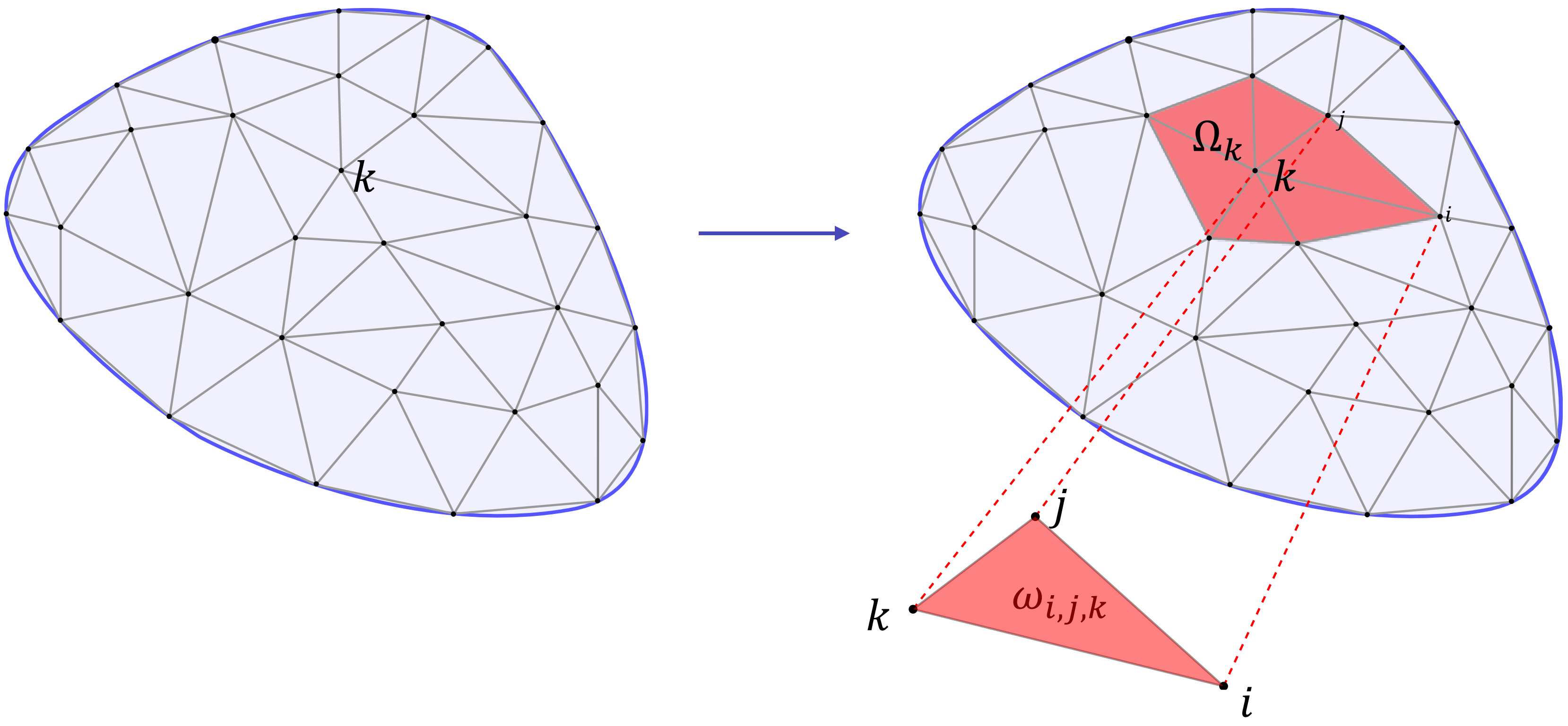


Requirements:

- Linear within Ω_k^*

*Support domain

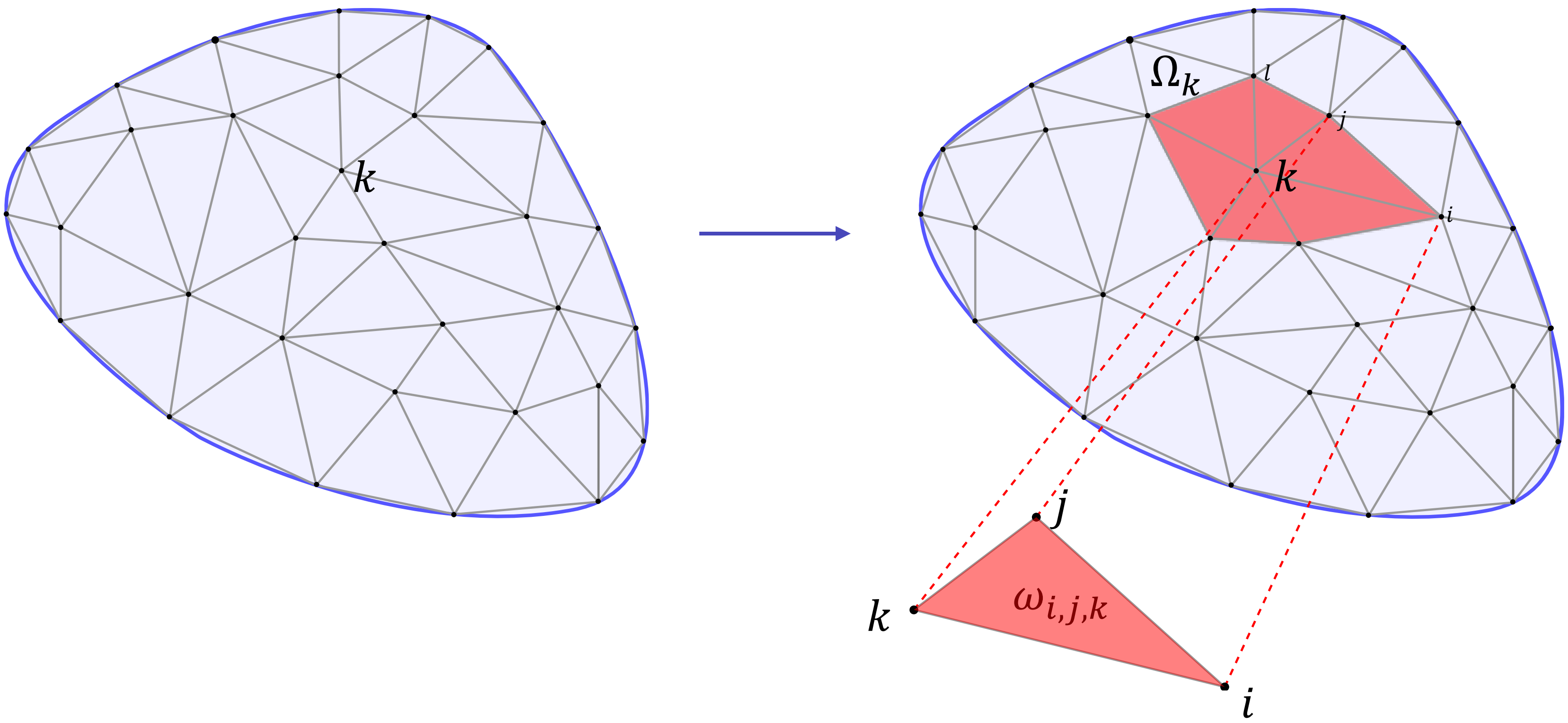
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



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

*Support domain

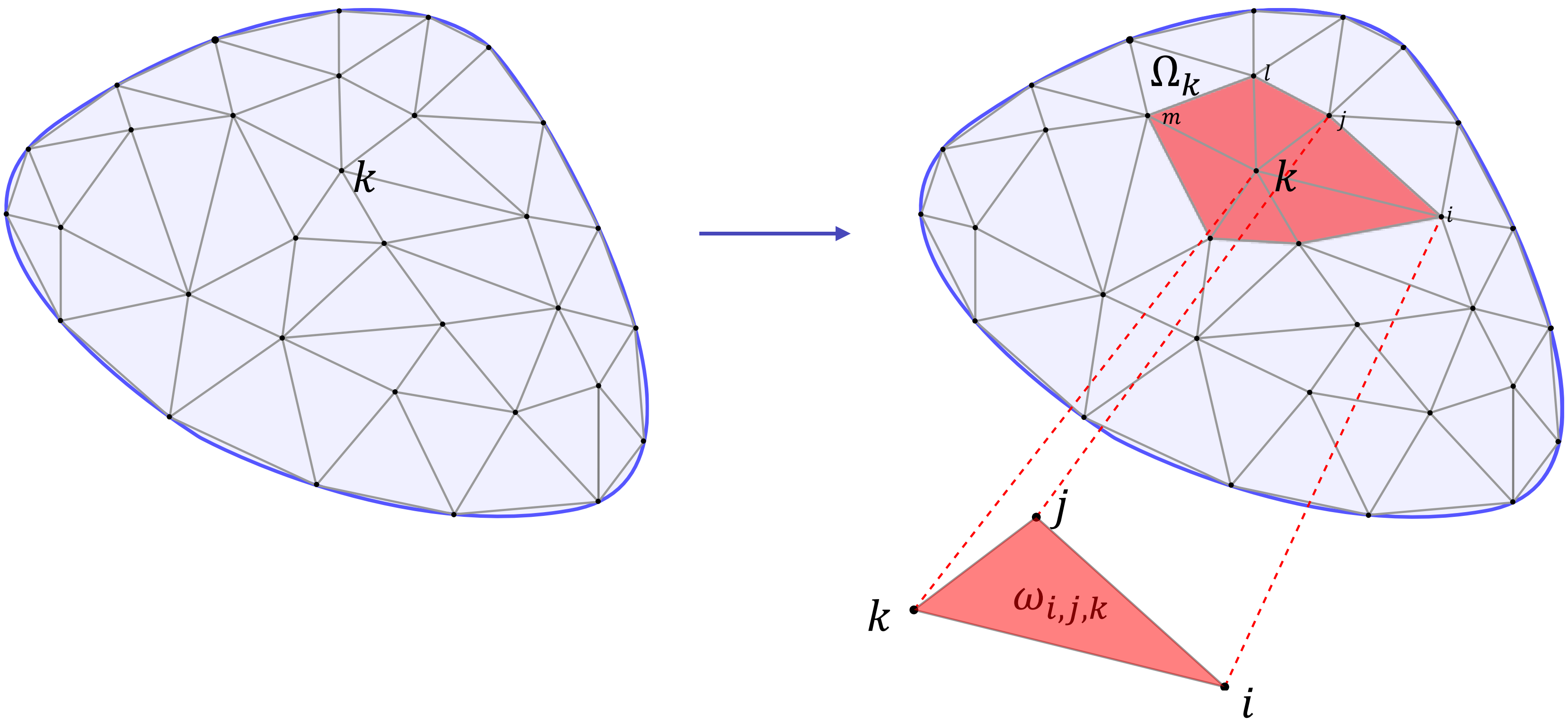
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



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

*Support domain

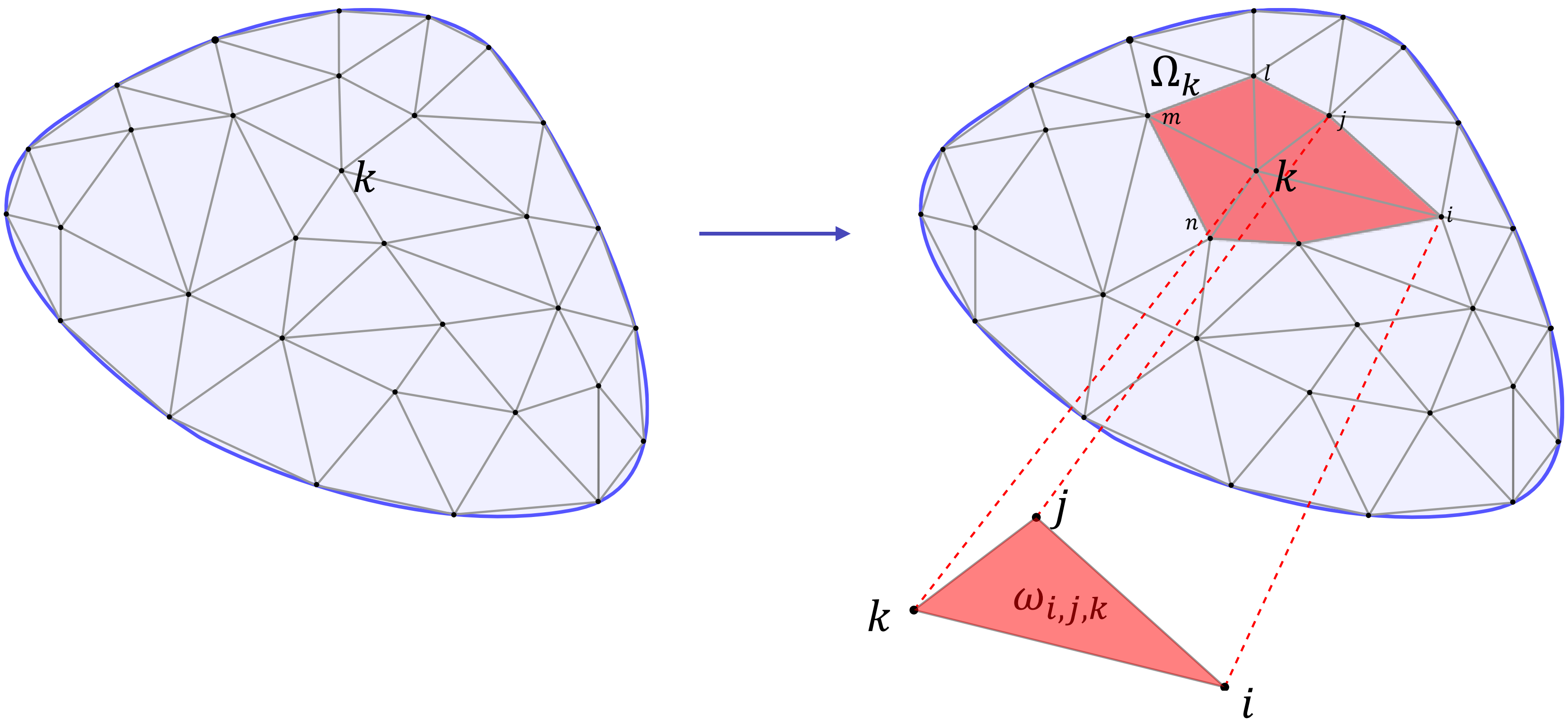
- Support domain Ω_k: *union of all elements* $\omega_{i,j,k}$ *that have* k *as one of their vertices*



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

*Support domain

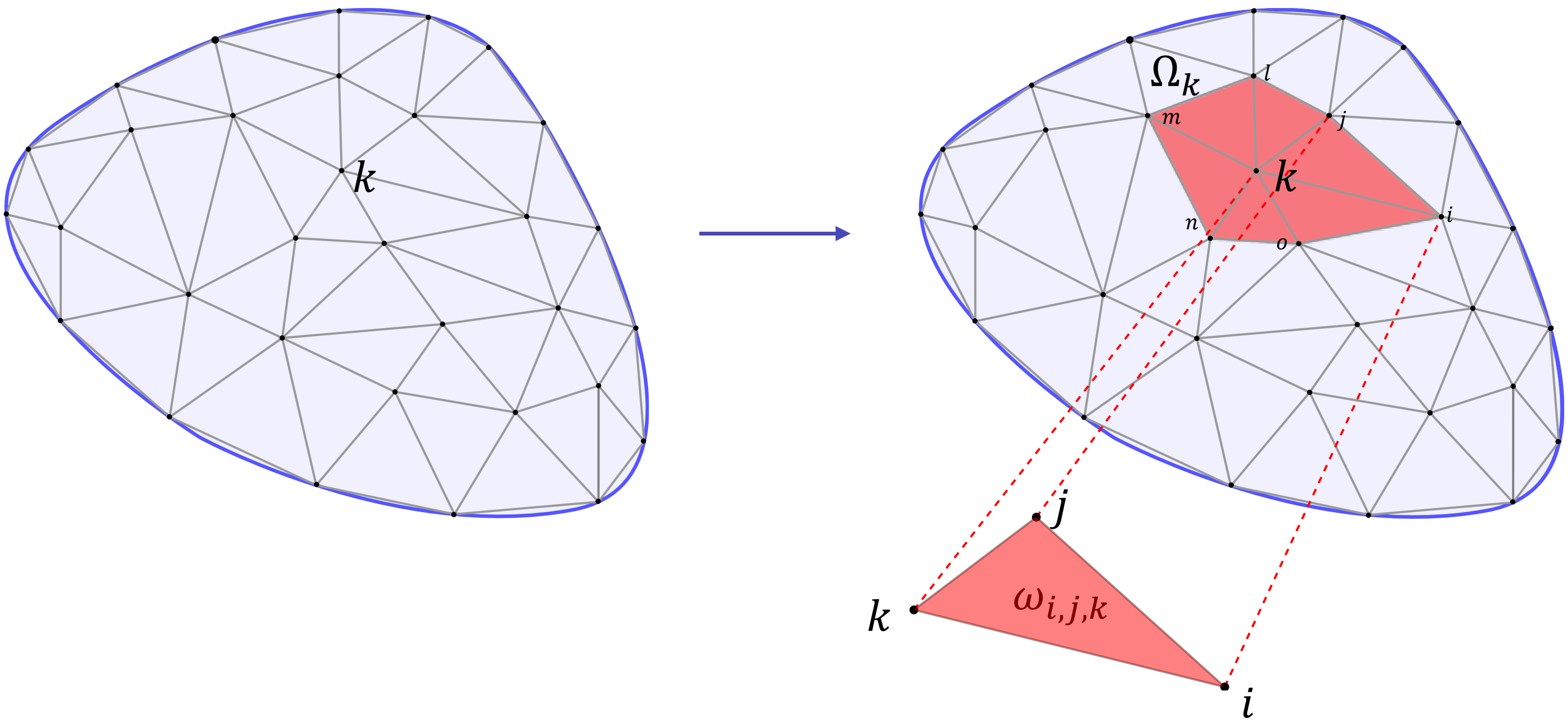
- Support domain Ω_k: *union of all elements $\omega_{i,j,k}$ that have k as one of their vertices*



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

*Support domain

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



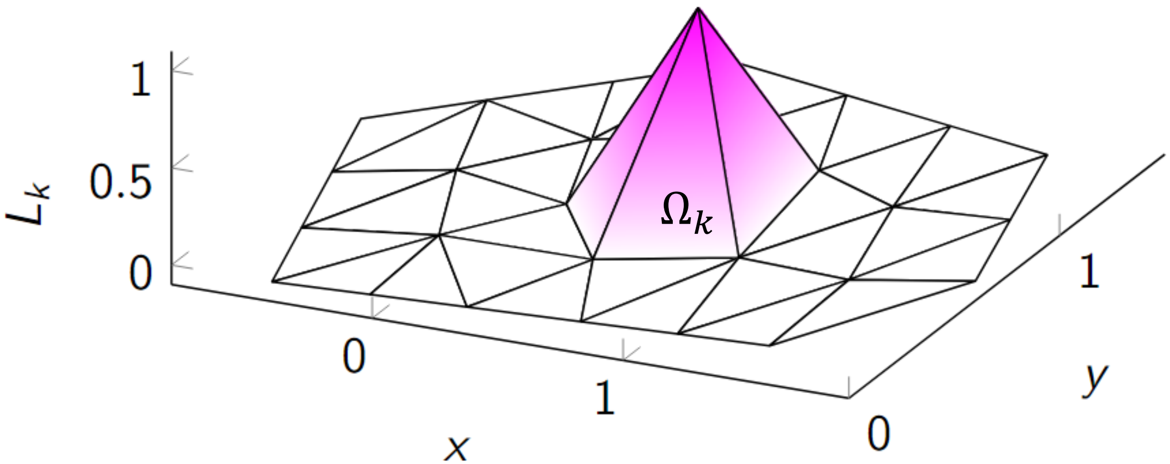
$$\Omega_k = \omega_{i,j,k} \cup \omega_{j,l,k} \cup \omega_{l,m,k} \cup \omega_{m,n,k} \cup \omega_{n,o,k}$$

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: $L_k(x,y)$



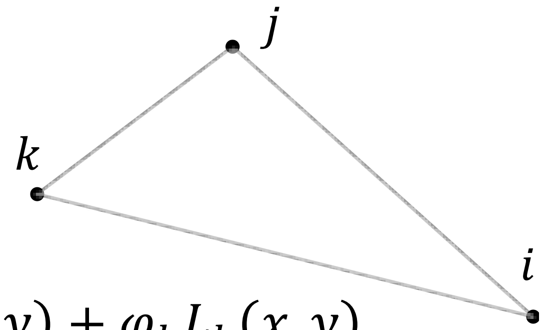
Requirements:

- 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 $\omega_{i,j,k}$

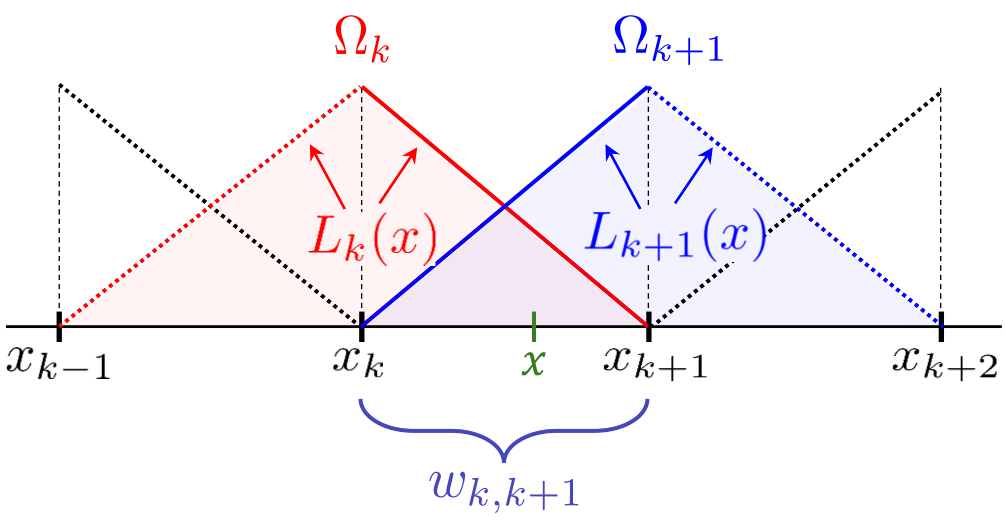
- Goal: express a function within triangle as a function nodal function values



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

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

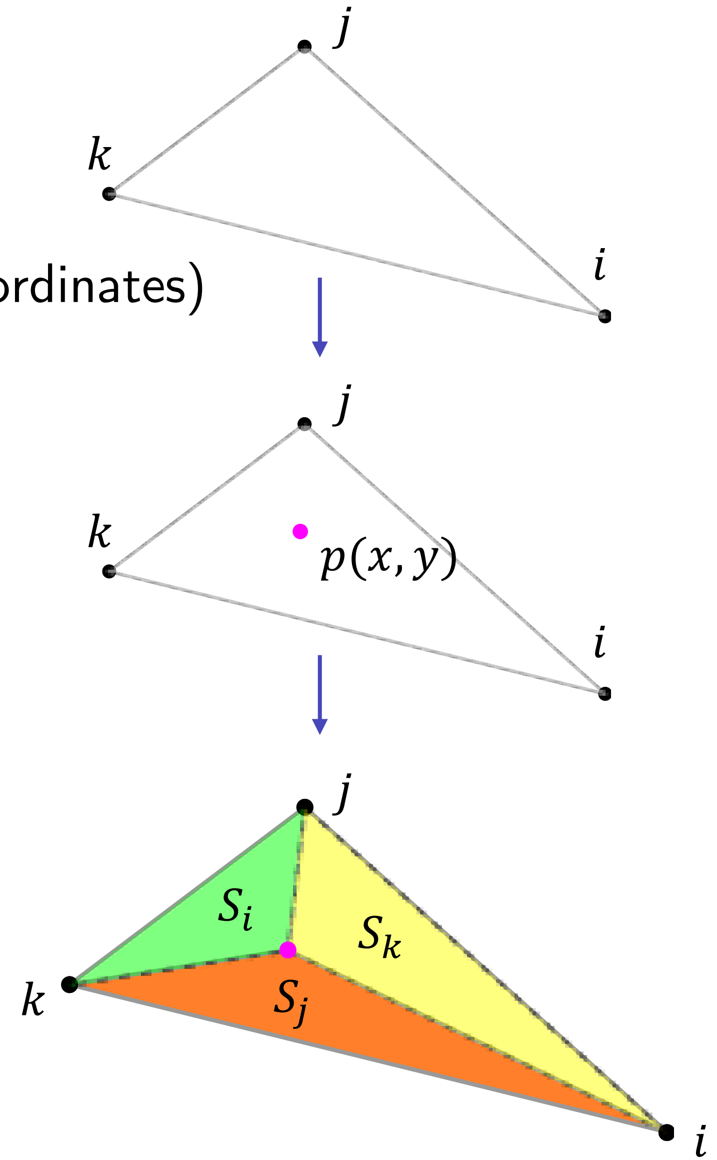
Consider a single triangle $\omega_{i,j,k}$

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

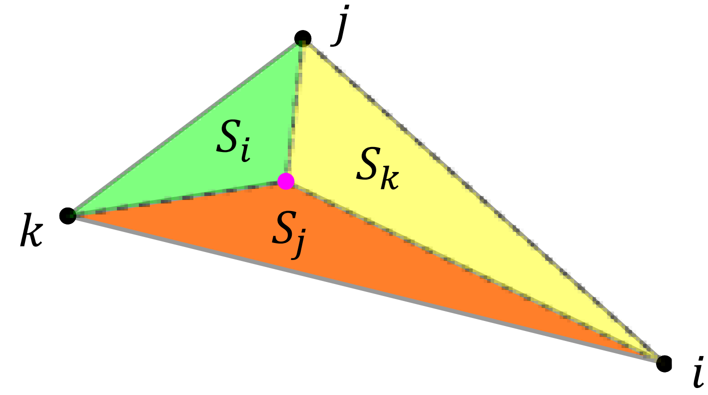
New idea: **area coordinates** (barycentric coordinates)

- 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

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

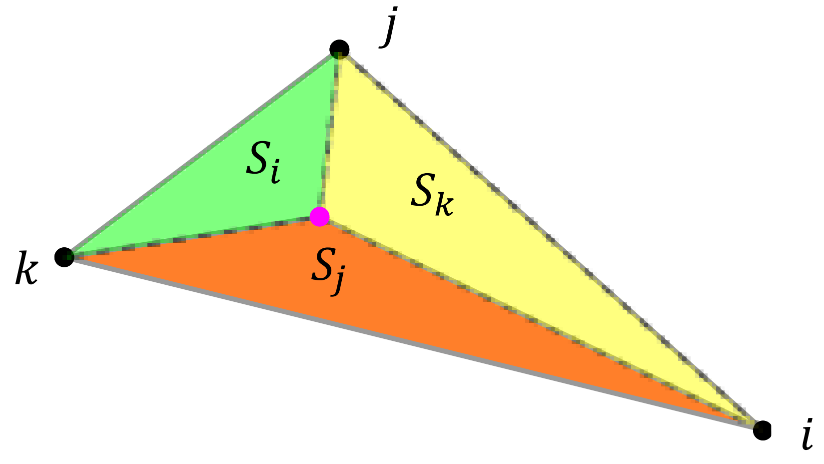
2D interpolation – barycentric coordinates

Recap – interpolation over triangles

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

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

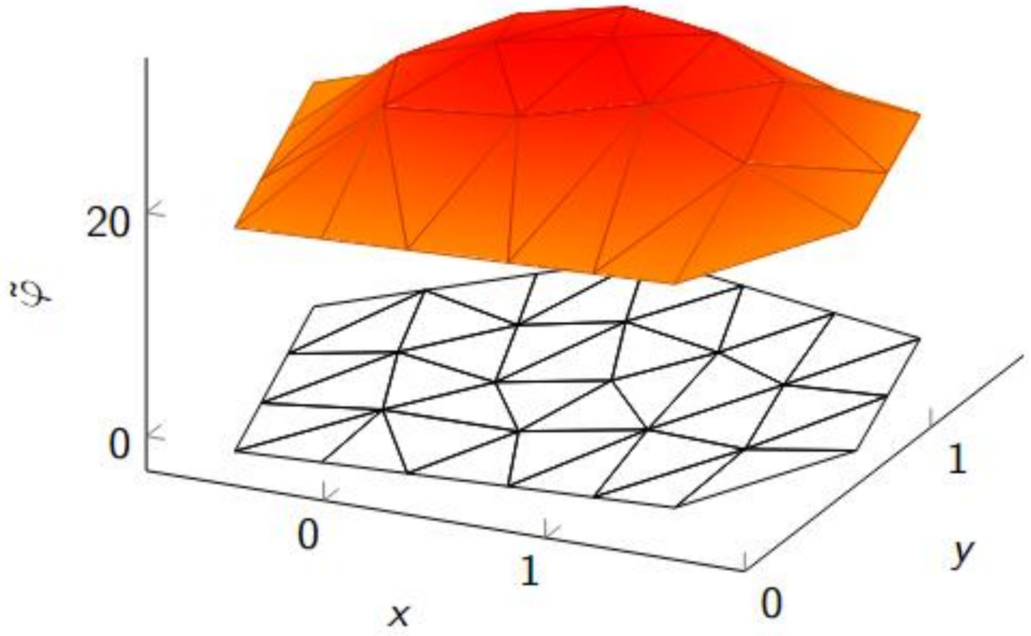
- Note (1): no "special" definitions at boundaries as in 1D
- Note (2): $L_i(x,y) + L_j(x,y) + L_k(x,y) = 1$

2D interpolation

- Result of interpolation

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

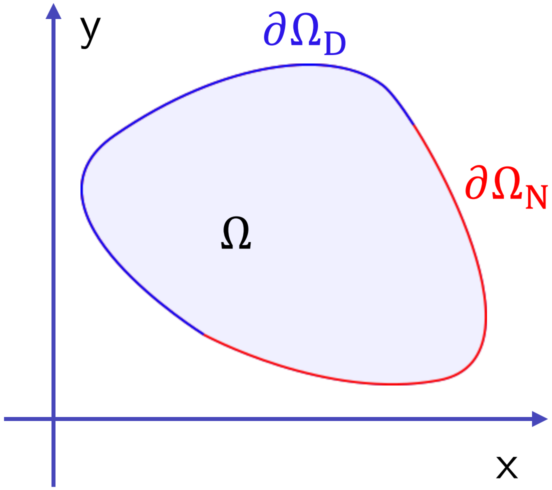
Extending to every triangle in a mesh, the interpolated 2D solution will look like this...



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

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

- 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 $\pm\infty$

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

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

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

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...

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

- **Weak formulation**!

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