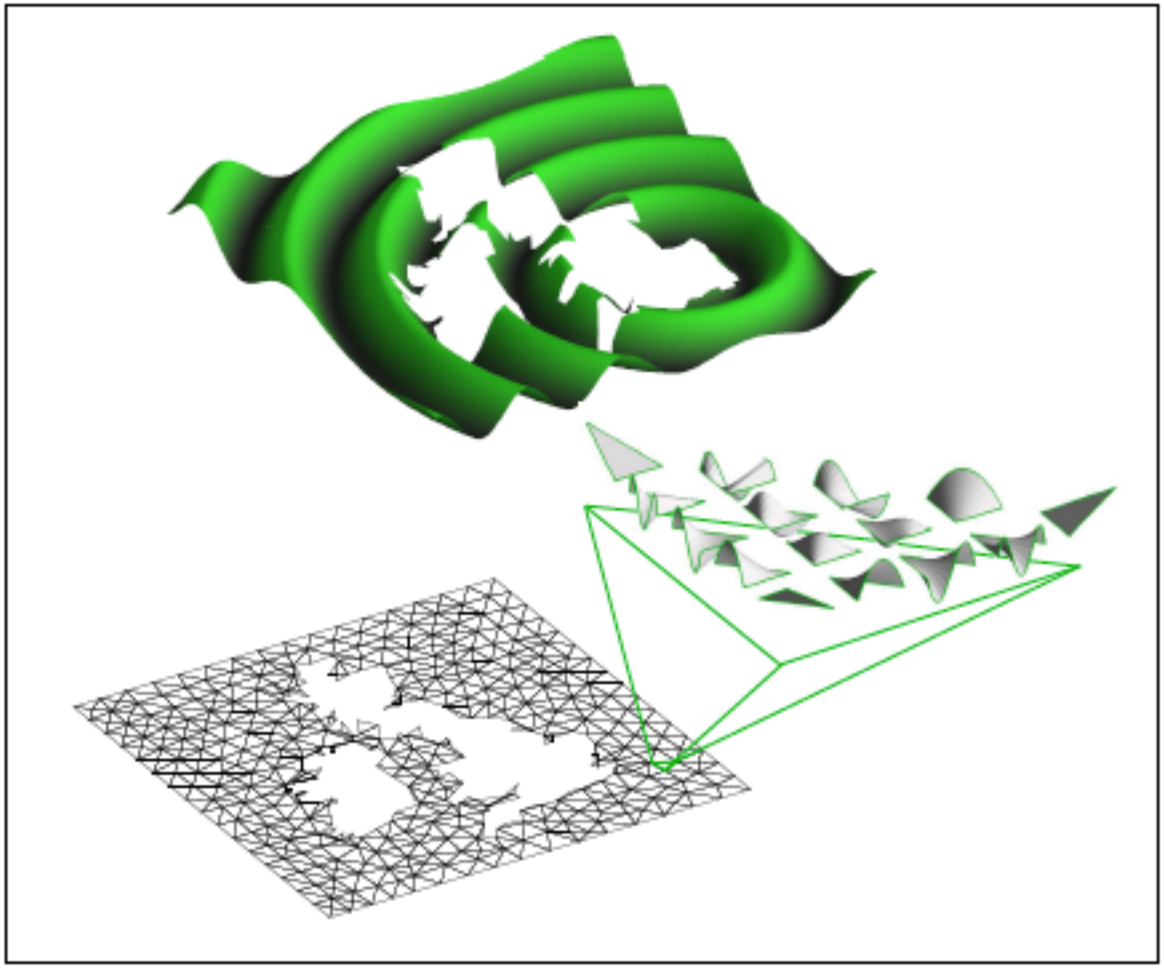
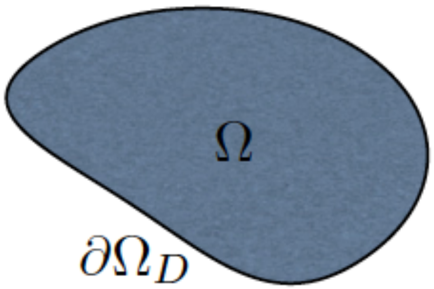
Generalities on Spectral/*hp* element method



Driving example: Poisson Problem

$$\mathbb{L}(u) = \nabla^2 u + f = 0 \quad x \in \Omega$$

$$u = g(x) \quad x \in \partial\Omega_D$$



Starting Point: Weak Formulation of the problem

Find $u \in \mathcal{V}$ such that u satisfies the boundary conditions and such that for all $v \in \mathcal{V}_0$,

$$(\nabla u, \nabla v) = (f, v)$$

Finite Element Method

Approximation of the solution and force by piecewise functions over *N* subdomains Ω_i :

$$u^{\delta}(x) = \sum_{i=0}^{N} \hat{u}_i \Phi_i(x)$$

$$f(x) = \sum_{i=0}^{N} \hat{f}_i \Phi_i(x)$$

Substitution into the weak form of the problem allows to evaluate <u>unknown</u> coefficients $\hat{u}_i$.

Two different approaches for the problem:

- ***h*-type methods**
- ***p*- type methods**

hp-type methods

h-type methods

- Fixed order polynomial in every element
- Convergence achieved by reducing the size of elements

+

p-type methods

- Fixed mesh
- Convergence achieved by increasing the order of the polynomial in every element

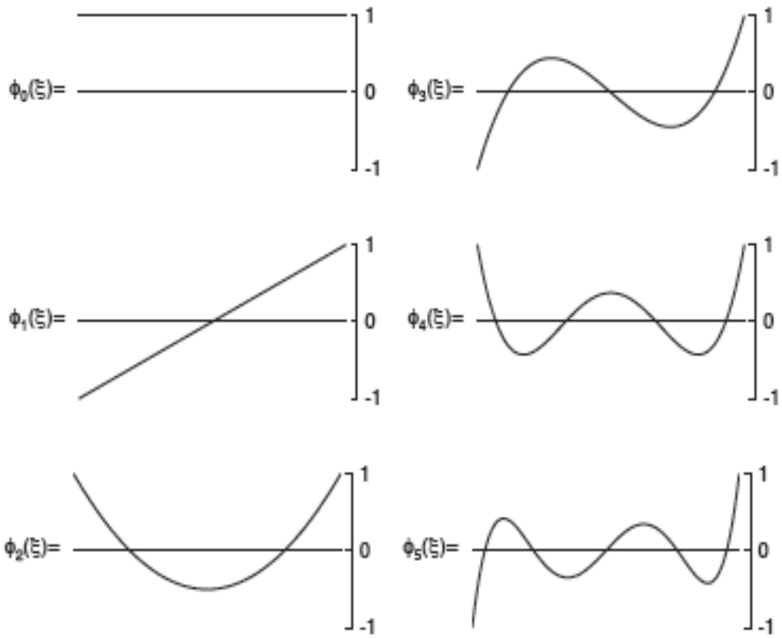
↓

hp-type methods: *h*-type decomposition to generate the initial mesh upon which *p*-type extension is applied

Legendre expansion

Legendre expansion within the standard region

$$\Omega_{st} = \{\xi | -1 \leq \xi \leq 1\}$$



$$\phi_p(\xi) \mapsto L_p(\xi) \equiv P_p^{(0,0)}(\xi), \quad 0 \leq \ p \leq P$$

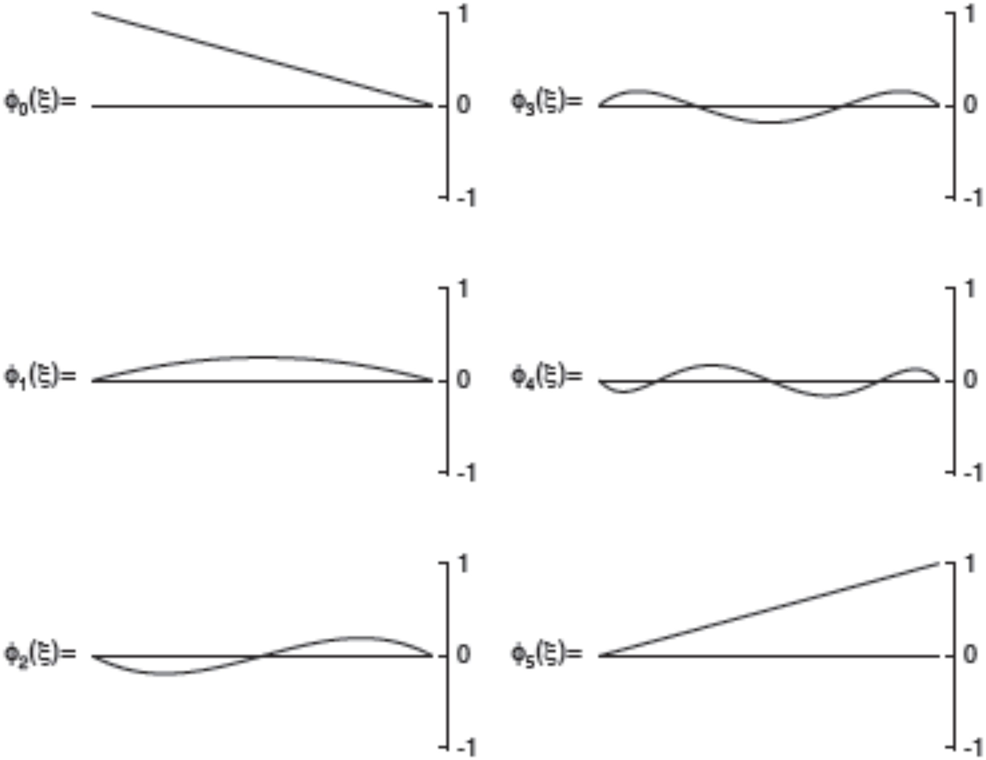
where,

$$P_p^{(0,0)} = \frac{(-1)^p}{2^p p!} \frac{d^n}{dx^p} [(1-x^2)^p]$$

Property of Jacobi polynomials

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta P_n^{\alpha,\beta}(x) P_i^{\alpha,\beta} d\xi = C\delta_{ni}$$

P-type finite elements

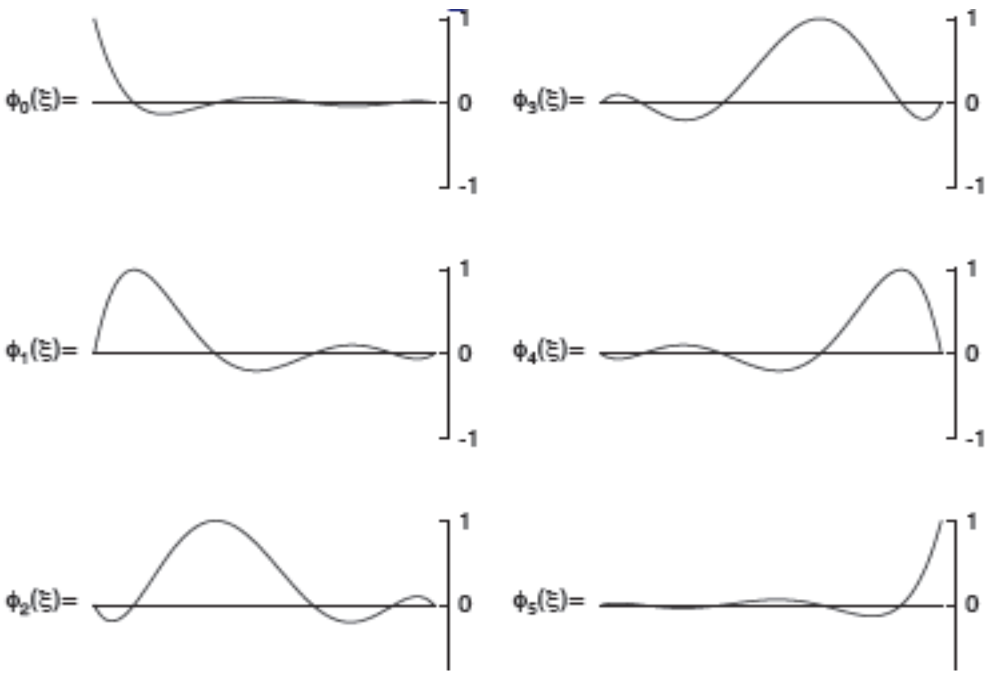


$$\Omega_{st} = \{\xi | -1 \leq \xi \leq 1\}$$

$$\phi_p(\xi) \mapsto \psi_p^{(}\xi) = \begin{cases} \frac{1-\xi}{2}, & p = 0 \\ \left(\frac{1-\xi}{2}\right)\left(\frac{1+\xi}{2}\right) P_{p-1}^{(1,1)}(\xi) & 0 < p < P \\ \frac{1+\xi}{2} & p = P \end{cases}$$

Spectral/*hp* element method

Spectral/hp element method uses the Lagrange polynomial through the zeros of the Gauss-Labotto polynomials.

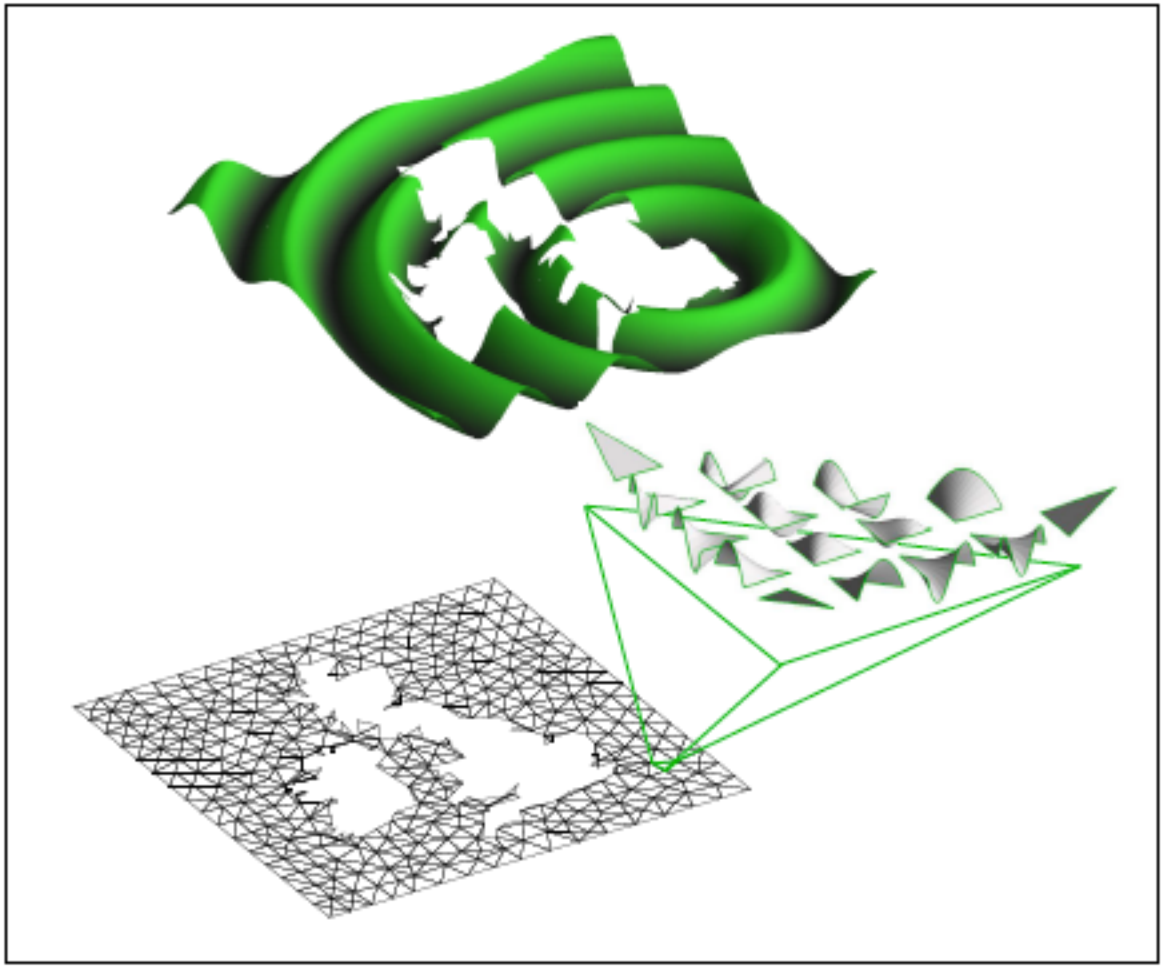


$$\Omega_{st} = \{\xi | -1 \leq \xi \leq 1\}$$

$$h_p(x) = \frac{\prod_{q=0,q\neq p}^{Q-1}(x - x_p)}{\prod_{q=0,q\neq p}^{Q-1}(x_p - x_p)}$$

$$\phi_p(\xi) \mapsto h_p^{gl}(\xi) = \begin{cases} 1, & \xi = \xi_p \\ \frac{(\xi-1)(\xi+1)\frac{\partial L_p}{\partial \xi}}{P(P+1)L_p(\xi_p)(\xi_p-\xi)} & otherwise \end{cases} \qquad 0 < p < P$$

Nektar++



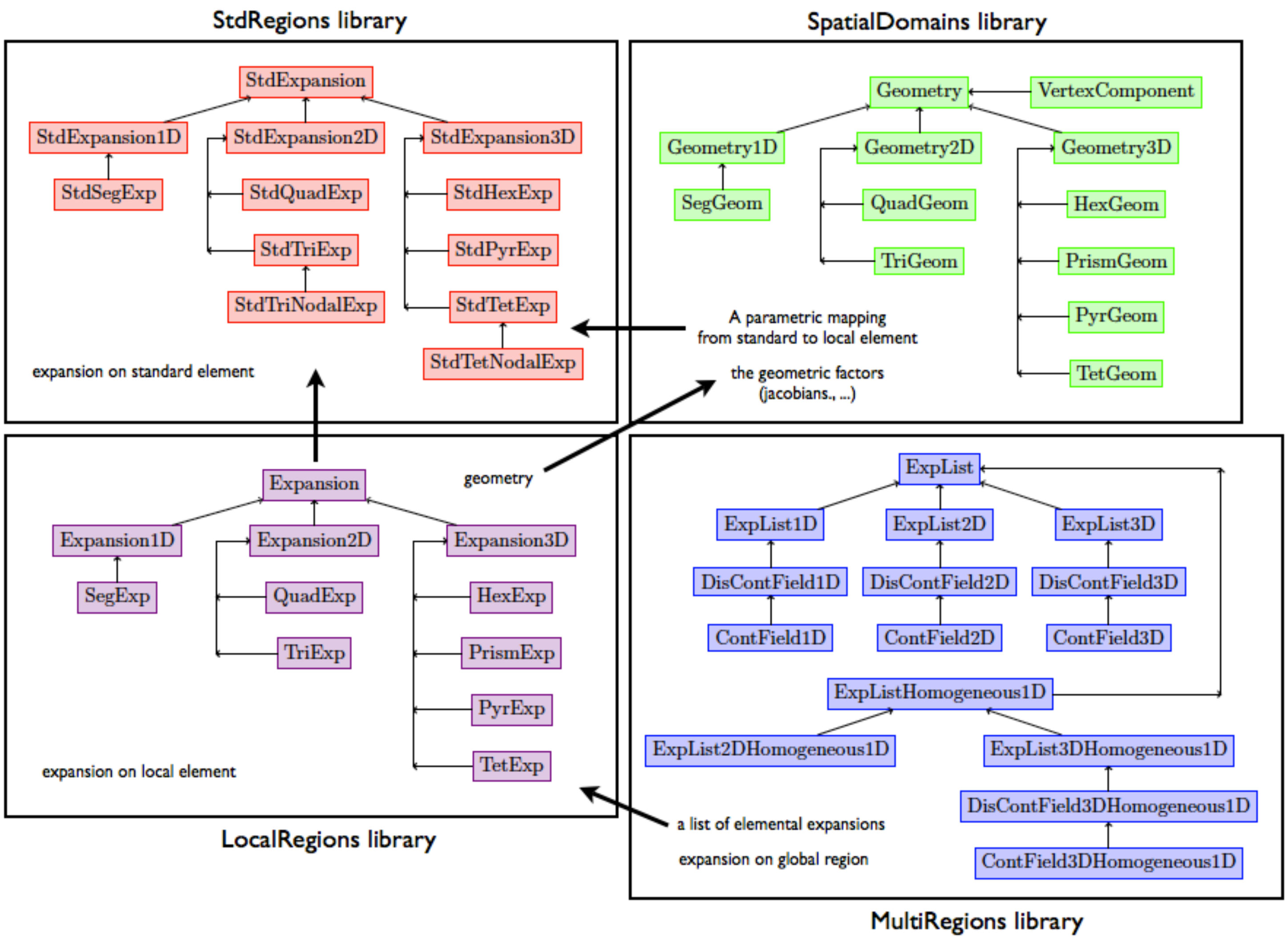
Nektar++
What is it?

Nektar++ is an open source software library currently being developed and designed to provide a bridge to the community to provide a toolbox of data structures and algorithms which implement the spectral/hp element method yielding fast error convergence. It is implemented as a C++ object-oriented toolkit which allows developers to implement spectral element solvers for a variety of different engineering problems.

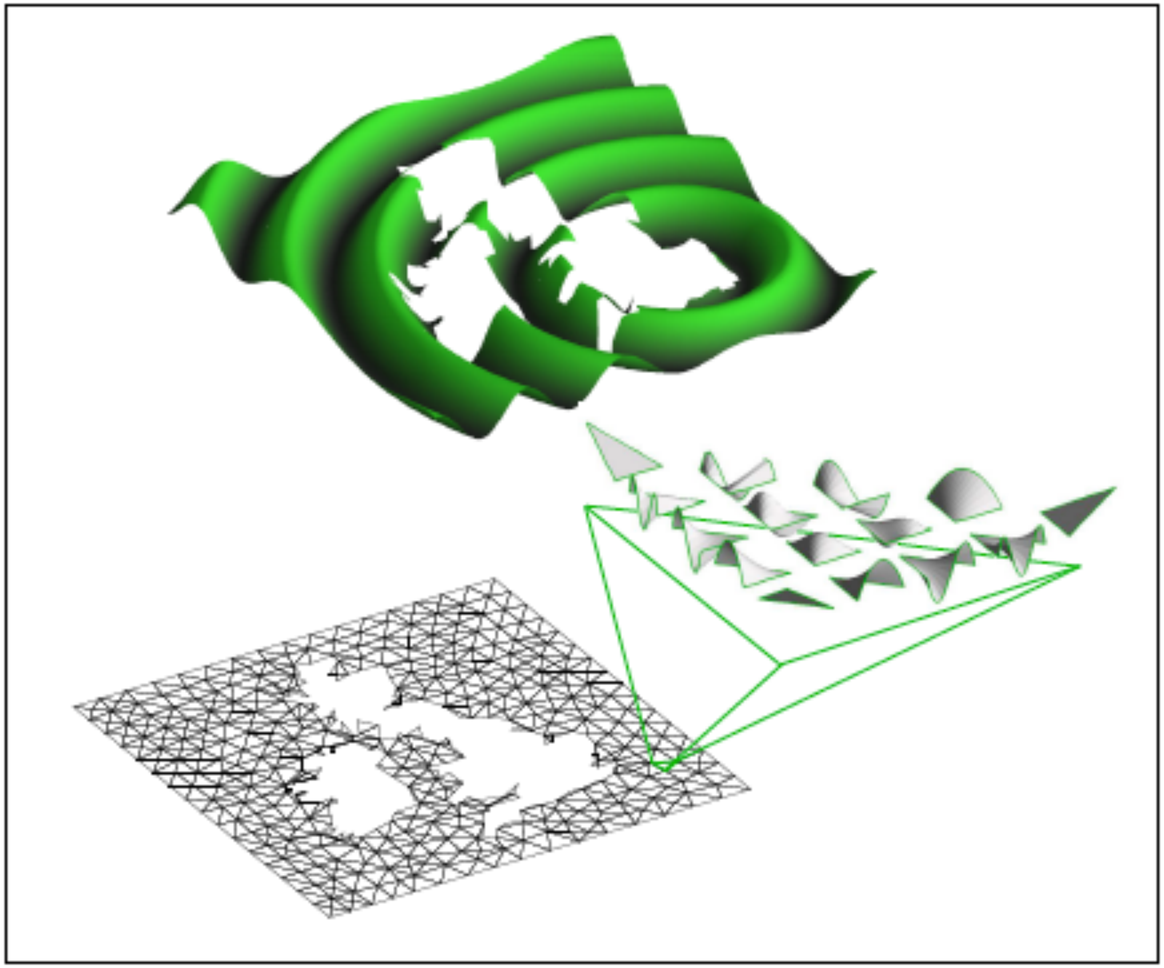
For more information, go to:

www.nektar.info

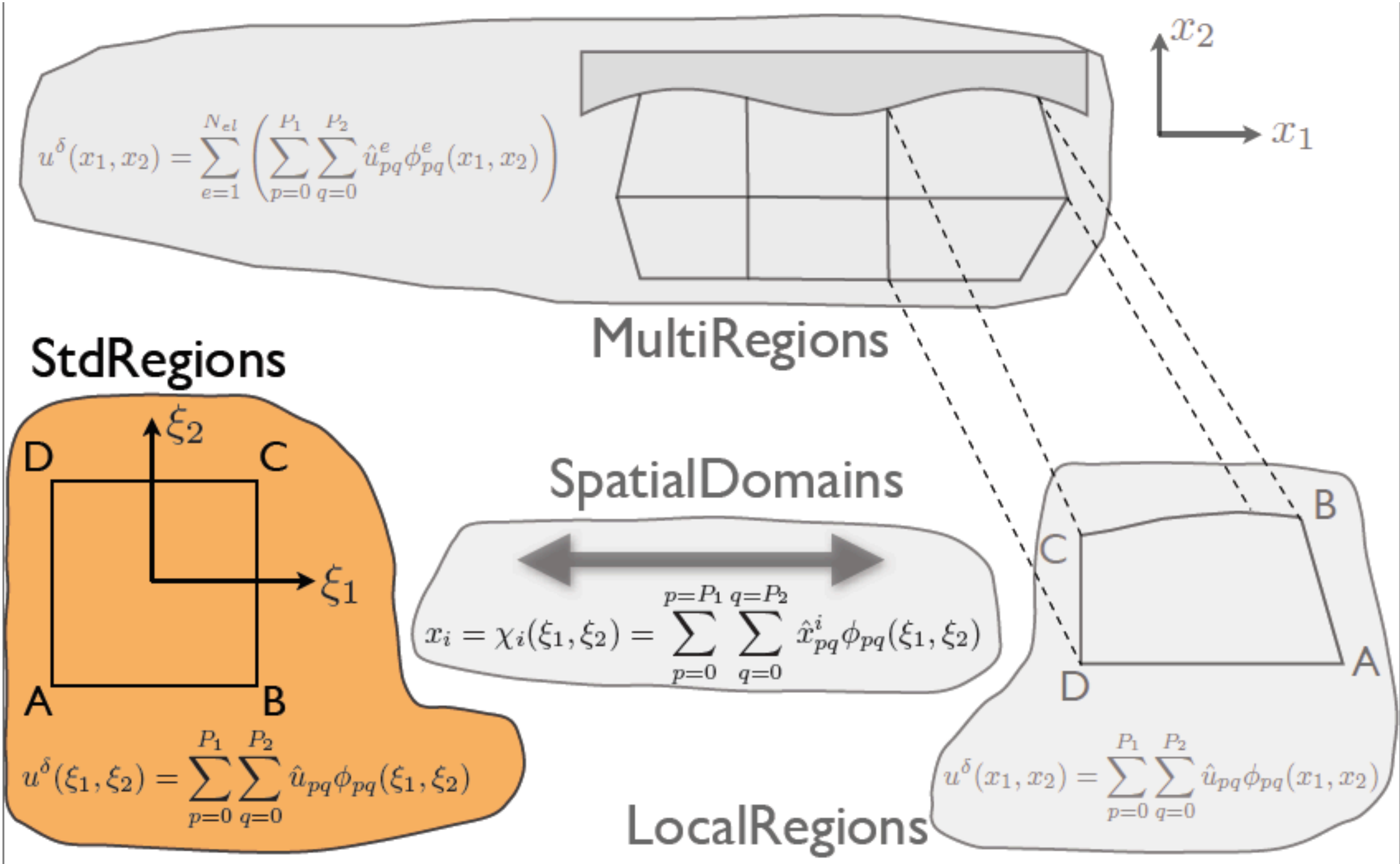
Overview of Nektar++ Structure



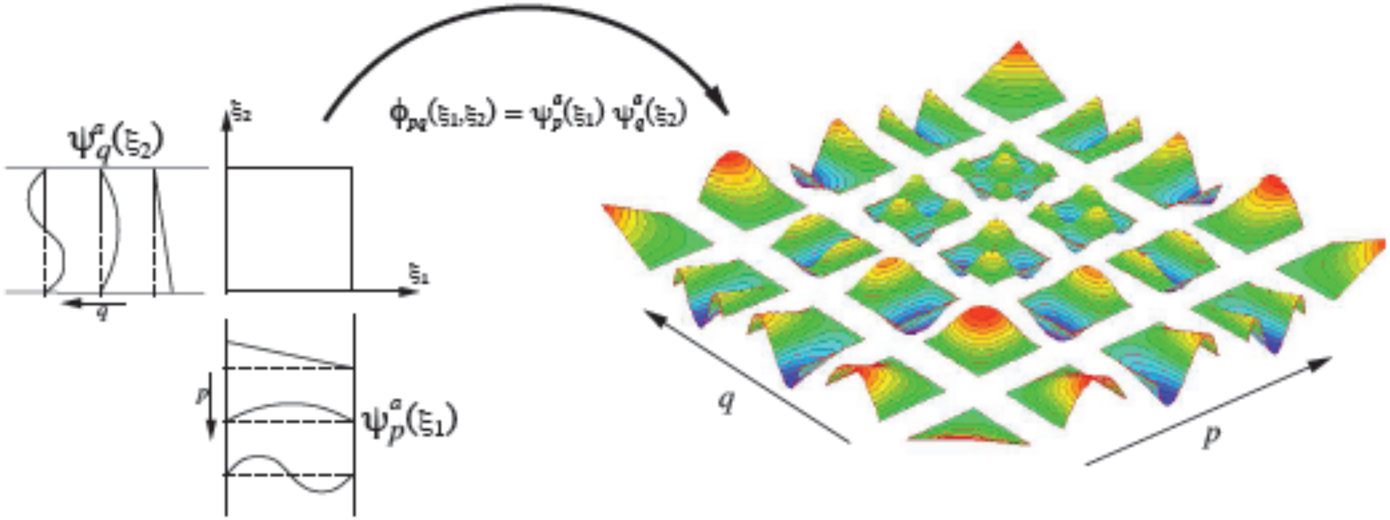
Expansions in Standard Regions (*StdRegions*)



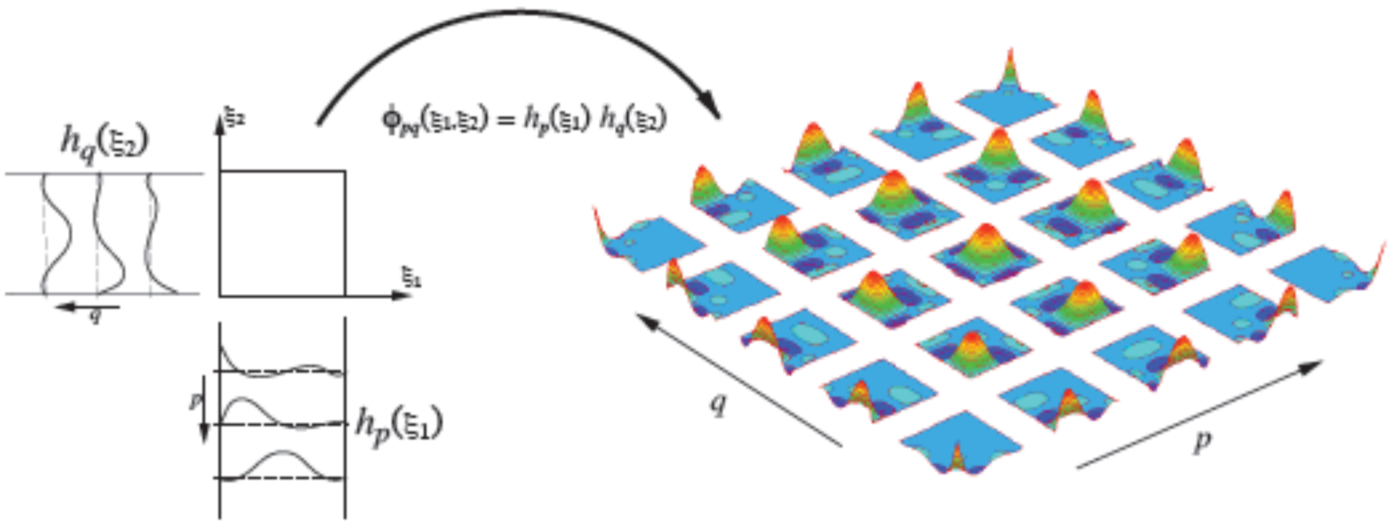
StdRegions Library



Spectral element/*P*-type finite element

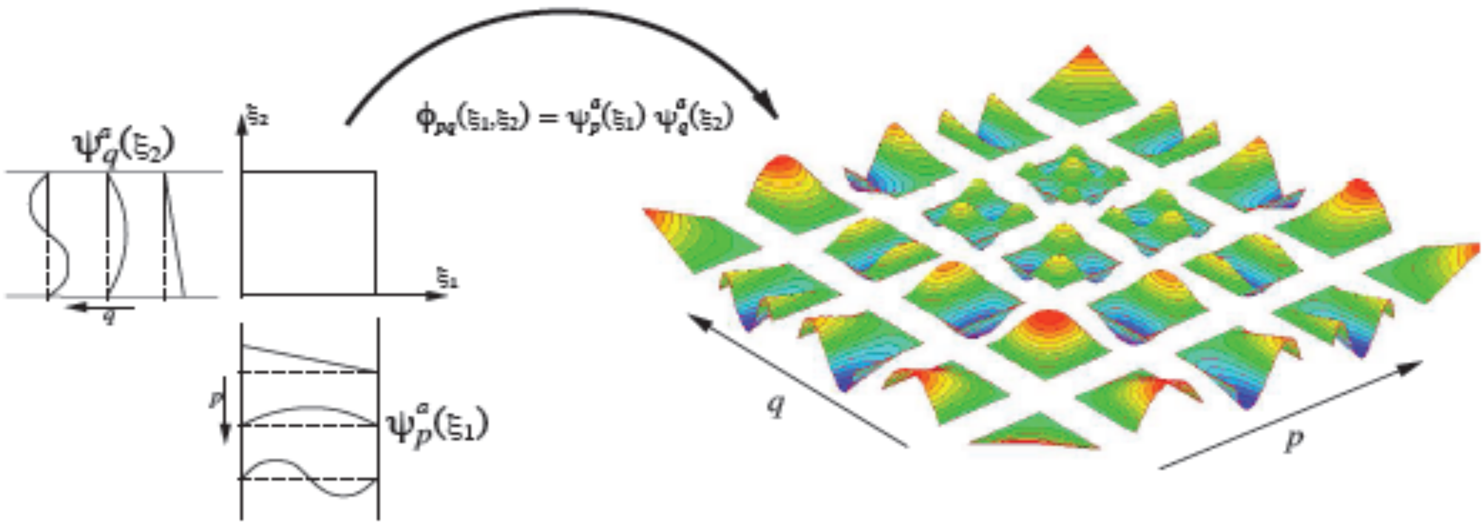


P-type finite element- hierarchical basis



Spectral element- collocation basis

Tensor product bases & sum factorisation



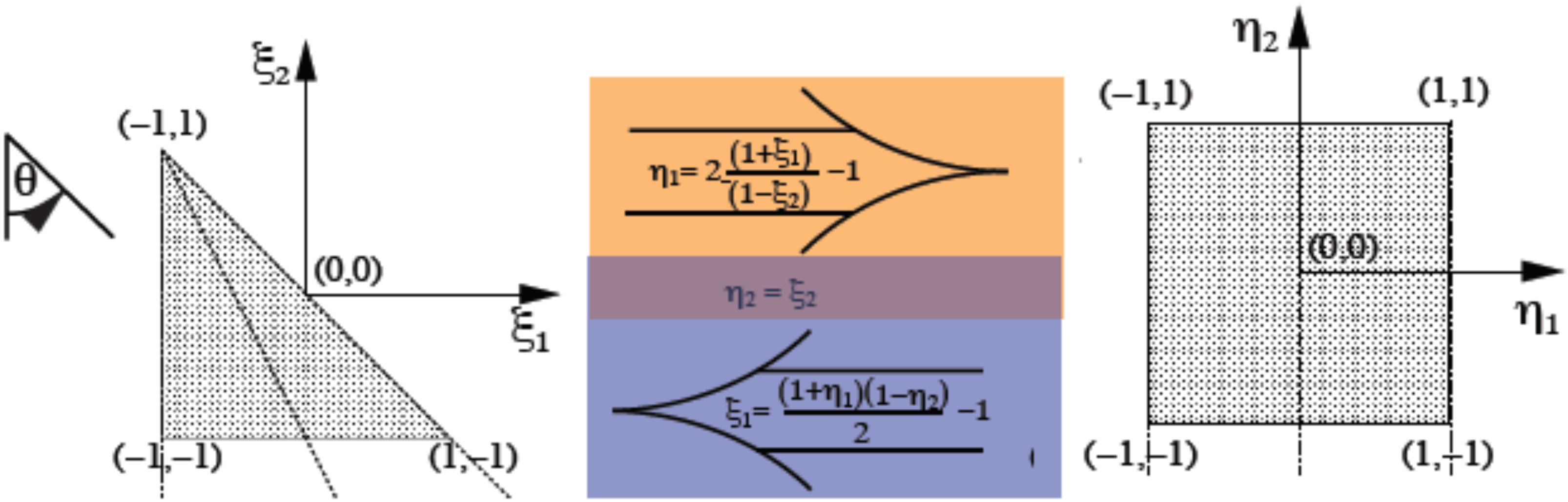
Inner product:

$$I_{pq} = \int_{\Omega^e} \phi_{pq}(\xi_1,\xi_2)u(\xi_1,\xi_2) = \sum_i \sum_j \phi_{pq}(\xi_{1,i},\xi_{2,j})u(\xi_{1,i},\xi_{2,j})$$

$$I_{pq} = \sum_i \sum_j \psi_p^a(\xi_{1,i})\psi_q^a(\xi_{2,j})u(\xi_{1,i},\xi_{2,j}) \sim O(P^4)$$

$$I_{pq} = \sum_i \psi_p^a(\xi_{1,i}) f(\xi_{2,i}) \sim O(P^3)$$

Collapsed coordinate system



$$\mathcal{T}_{st} = \{(\xi_1, \xi_2) | -1 \leq \xi_1, \xi_2; \xi_1 + \xi_2 \leq 0\} .$$

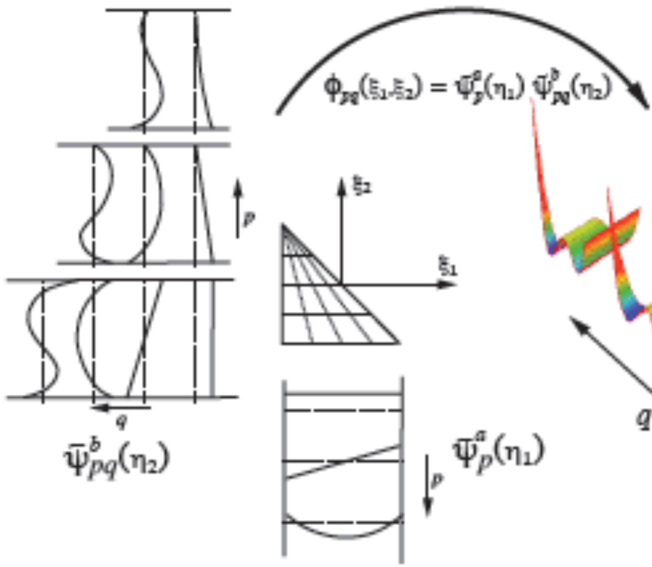
$$\mathcal{T}_{st} = \{(\eta_1, \eta_2) | -1 \leq \eta_1, \eta_2 \leq 1\} ,$$

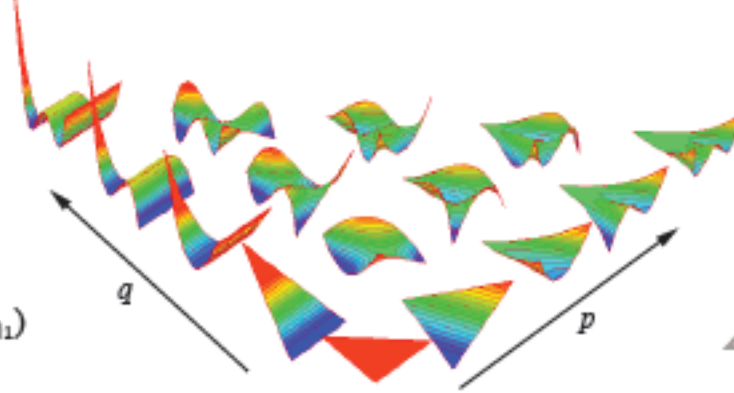
$$\eta_1 = 2\frac{(1+\xi_1)}{(1-\xi_2)} - 1 , \qquad \eta_2 = \xi_2 ,$$

$$\xi_1 = \frac{(1+\eta_1)(1-\eta_2)}{2} - 1 , \qquad \xi_2 = \eta_2 .$$

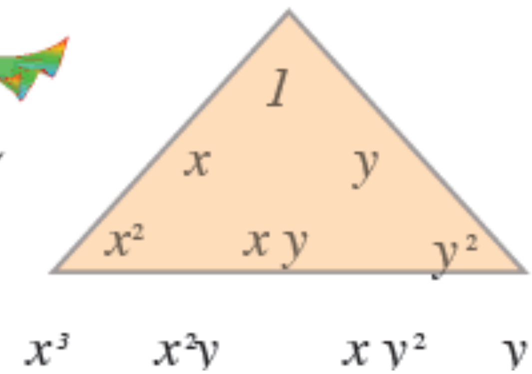
- **2 coordinates in 2 dimensions.**
- **Not rotationally symmetric.**
- **Analogous system used in cylindrical coordinate system.**

Orthogonal Expansion





$$\xi_1 = \frac{(1+\eta_1)(1-\eta_2)}{2} - 1\,, \qquad \xi_2 = \eta_2\,.$$



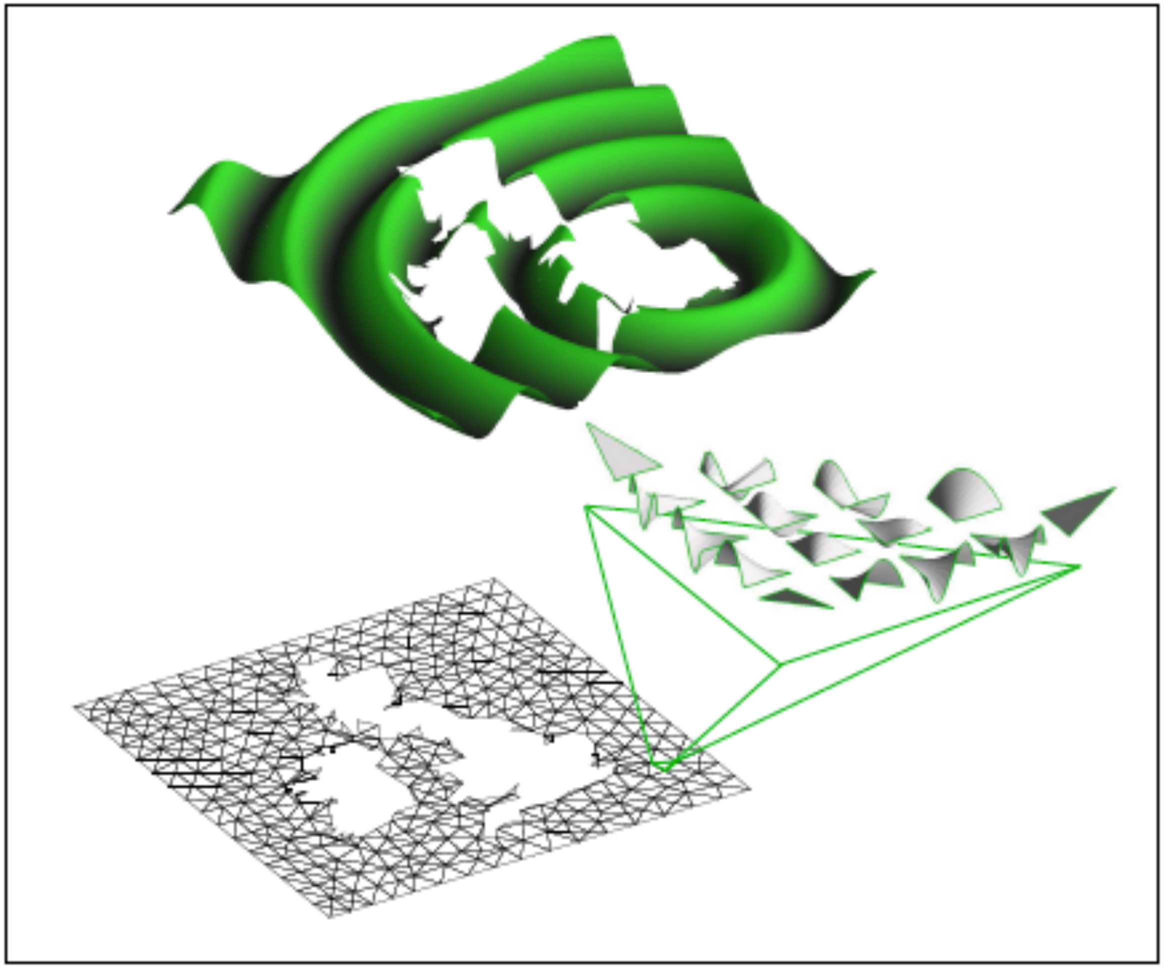
Principal functions:

$$\tilde{\psi}^a_p = P_p^{(0,0)}(z) \qquad\qquad \tilde{\psi}^a_{pq} = \left(\frac{1-z}{2}\right) P_q^{(2p+1,0)}(z)$$

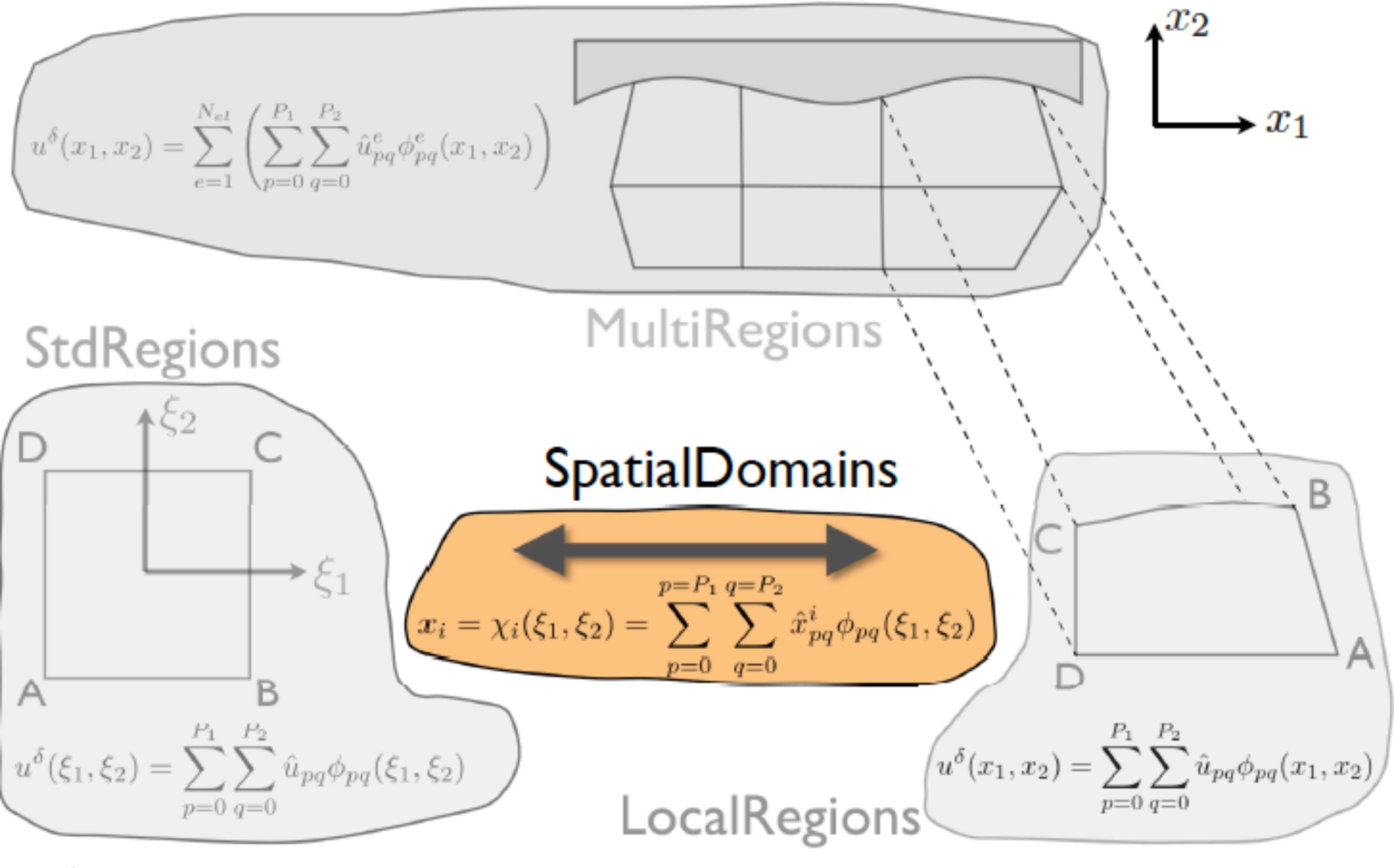
Generalised tensor products:

- Quadrilateral expansion: $\phi_{pq}(\xi_1,\xi_2) = \tilde{\psi}^a_p(\xi_1)\tilde{\psi}^a_q(\xi_2), \quad 0 \le p,q \le P$
- Triangular expansion: $\phi_{pq}(\xi_1,\xi_2) = \tilde{\psi}^a_p(\eta_1)\tilde{\psi}^a_{pq}(\eta_2), \quad 0 \le p, p+q \le P$

Spatial Construction of Elements (*SpatialDomains*)



StdRegions Library



Approximating the Geometry

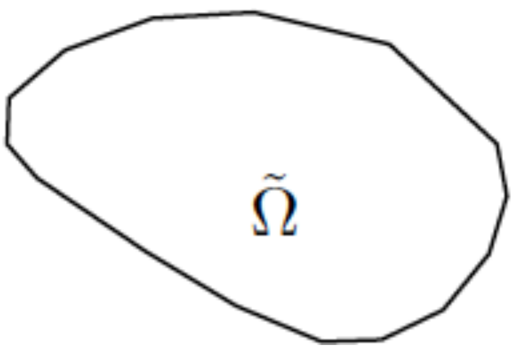
Find u such $\mathcal{L}(u) = f$ on Ω subject to boundary and initial conditions

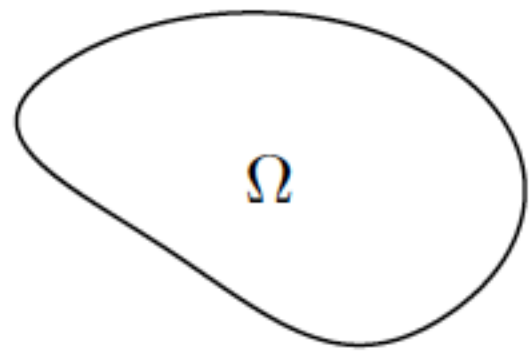
$$\tilde{\Omega} = \mathcal{T}(\Omega)$$

Approximation of Domain

Tessellation of Domain

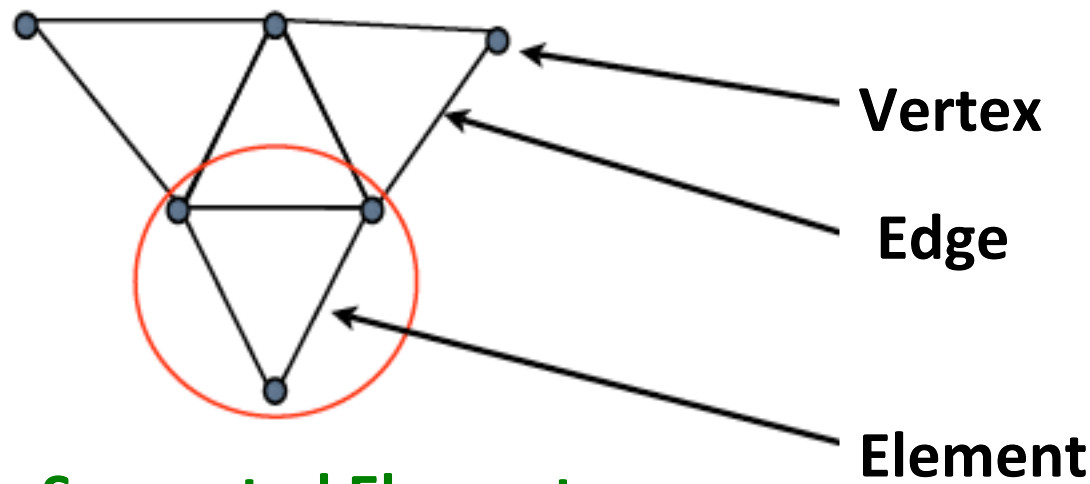
True Domain





Mesh

Mesh- a collection of vertices, edges and faces which form elements



Supported Element:

Dimension	Elements
1-D	• Segments
2-D	• Triangles • Quadrilaterals
3-D	• Tetrahedra • Hexahedra • Prisms • Pyramids

World Space vs Reference Space

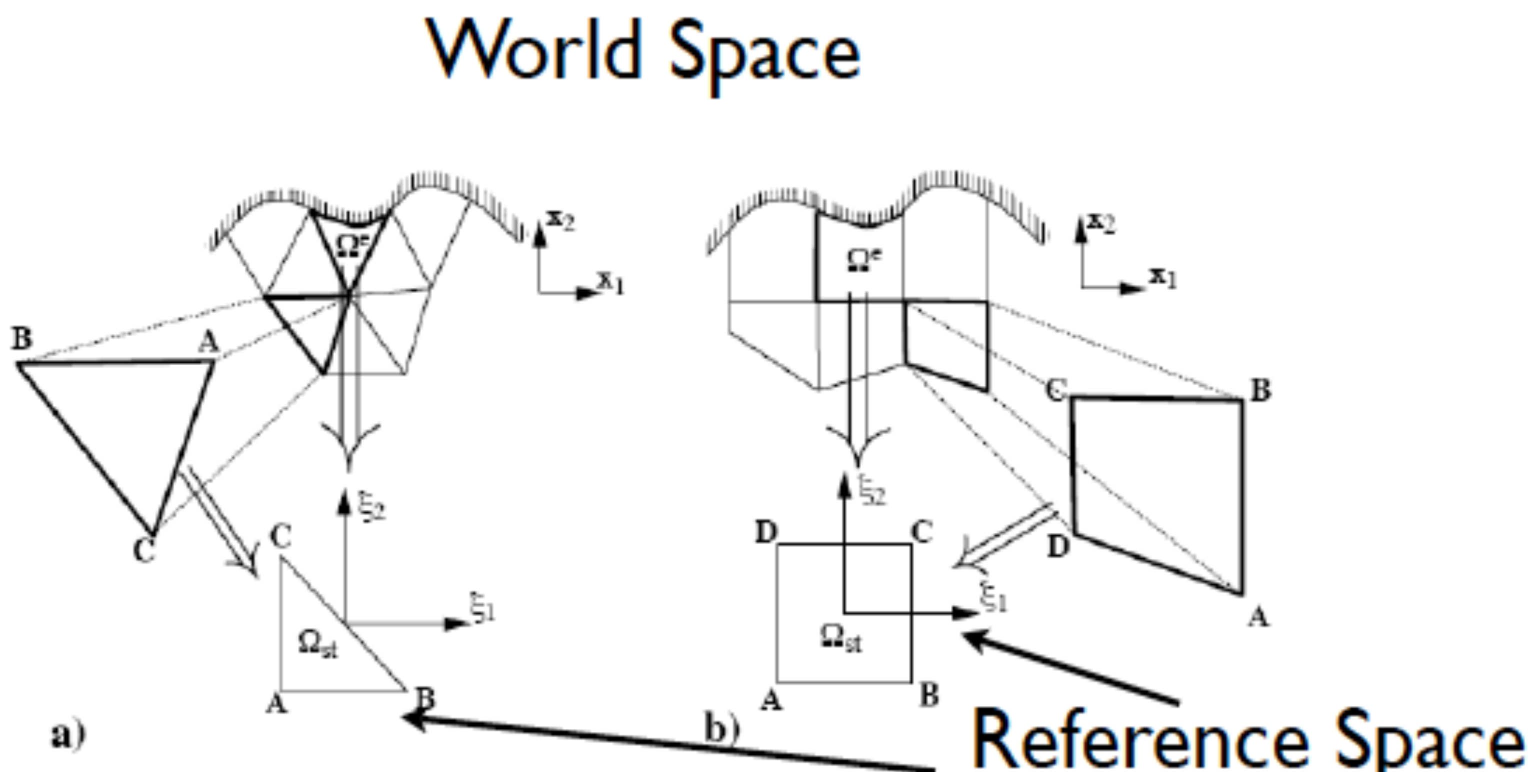
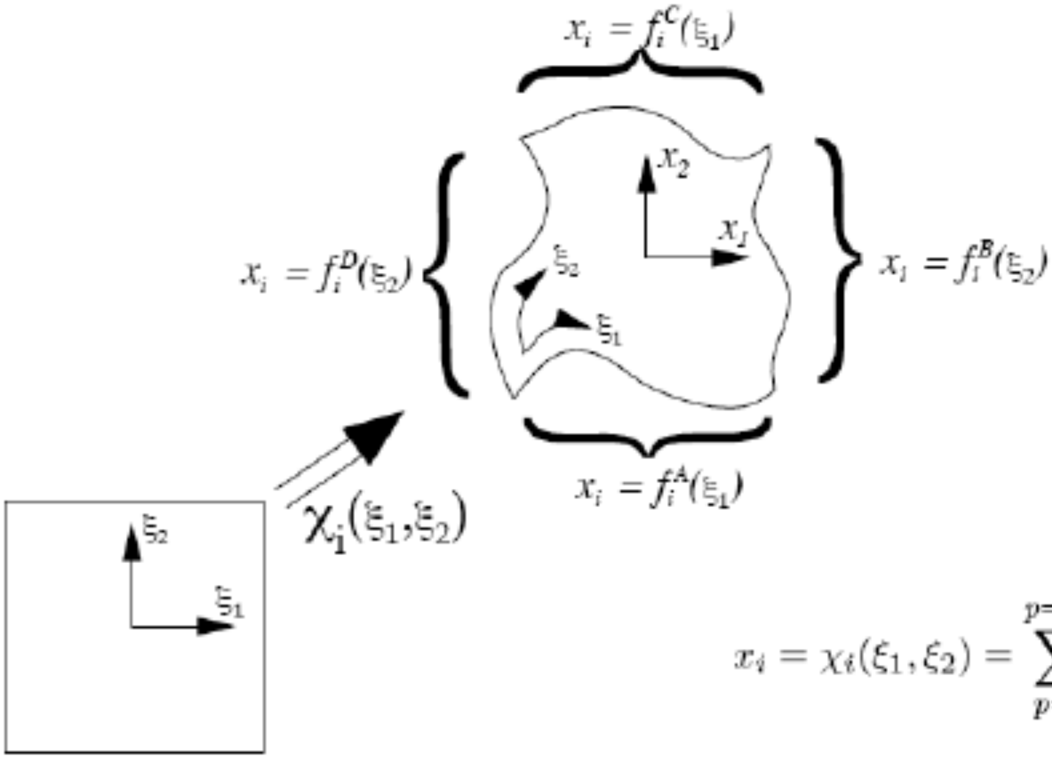


Figure 3.2 To construct a C^0 expansion from multiple elements of specified shapes (for example, triangles or rectangles), each elemental region Ω^e is mapped to a standard region Ω_{st} in which all local operations are evaluated.

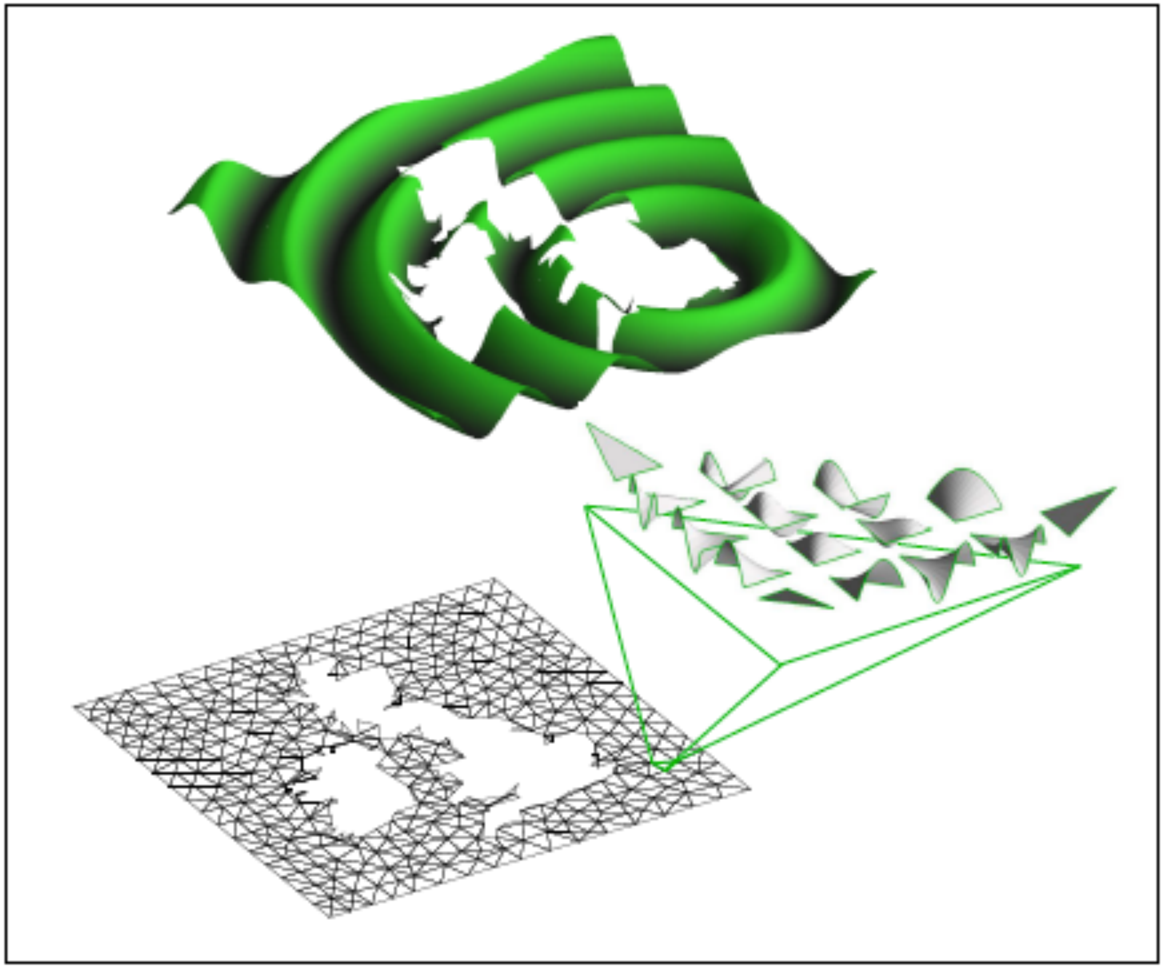
Mapping from reference space to world space



$$x_i = \chi_i(\xi_1, \xi_2) = \sum_{p=0}^{p=P_1} \sum_{q=0}^{q=P_2} \hat{x}_{pq}^i \phi_{pq}(\xi_1, \xi_2)$$

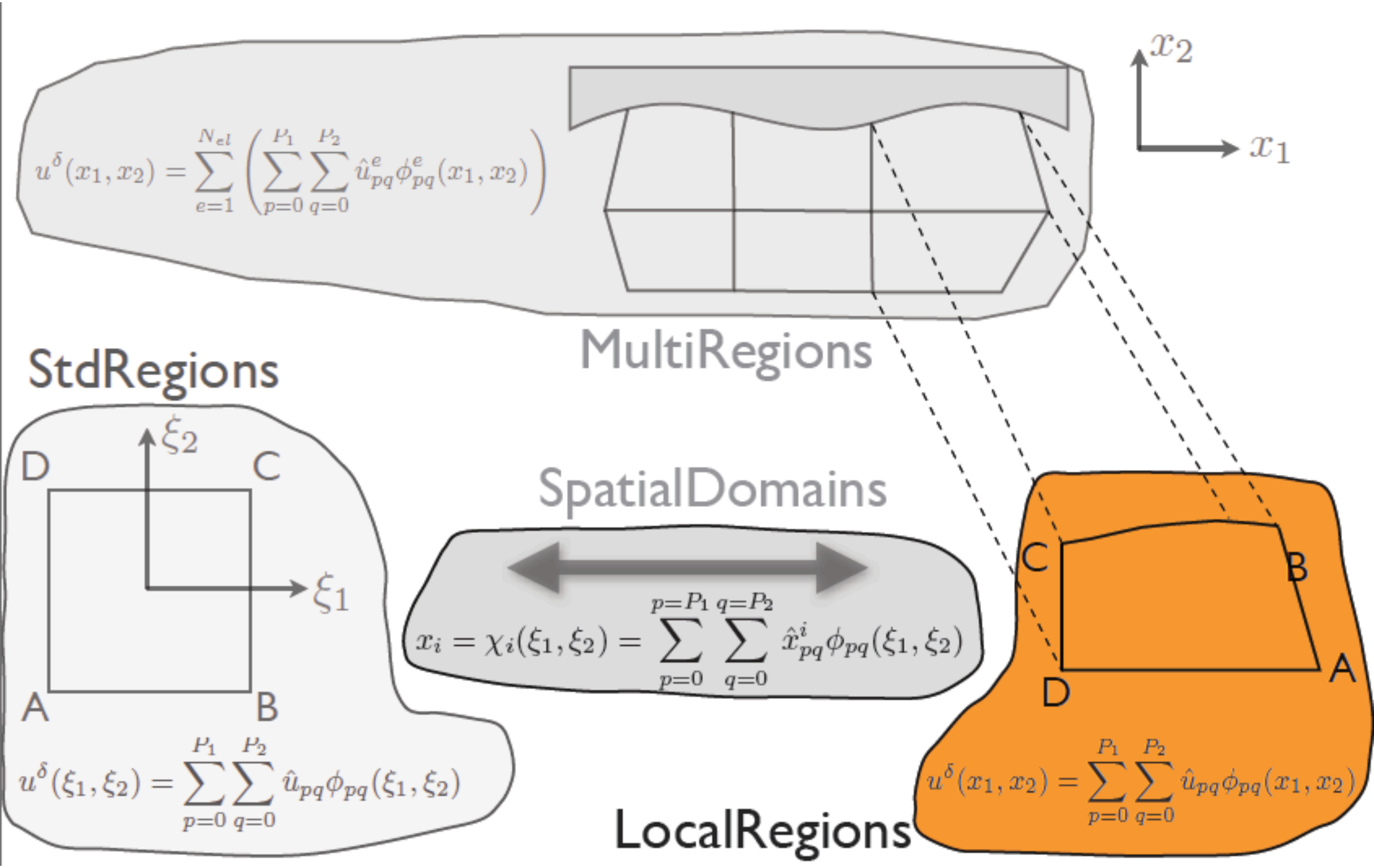
Figure 3.4 A general curved element can be described in terms of a series of parametric functions $f^A(\xi_1), f^B(\xi_2), f^C(\xi_1)$, and $f^D(\xi_2)$. Representing these functions as a discrete expansion we can construct an iso-parametric mapping $\chi_i(\xi_1, \xi_2)$ relating the standard region (ξ_1, ξ_2) to the deformed region (x_1, x_2).

Expansions in Local Regions (*LocalRegions*)



Nektar++

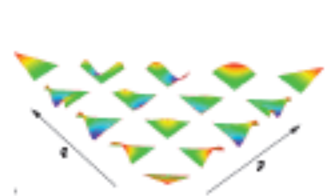
LocalRegions Library

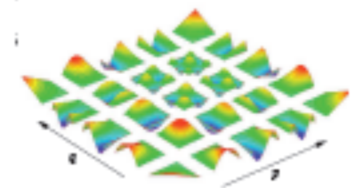


Elemental forward transform

LocalRegions::QuadExp/TriExp::FwdTrans()

Expansion: $u^\delta = \sum_{p,q} \hat{u}_{pq}\phi_{pq}$





Approximation: $u^\delta(\xi_1,\xi_2) - u(\xi_1,\xi_2) = R(u)$

$$\left(\sum_{p,q} \hat{u}_{pq}\phi_{pq}(\xi_1,\xi_2)\right) - u(\xi_1,\xi_2) = R(u)$$

Method of weighted residual: $$\left(v, \sum_{p,q} \hat{u}_{pq}\phi_{pq}\right) - (v,u) = (v, R(u))$$

Setting $(v, R(u)) = 0$ $\Rightarrow$ $$\left(v, \sum_{p,q} \hat{u}_{pq}\phi_{pq}\right) = (v,u)$$

Elemental forward transform

LocalRegions::QuadExp/TriExp::FwdTrans()

Considering the Galerkin weight: $v(\xi_1, \xi_2) = \phi_{rs}(\xi_1, \xi_2)$

$$\left(\phi_{rs}, \sum_{p,q} \hat{u}_{pq}\phi_{pq}\right) = (\phi_{rs}, u) \Rightarrow \sum_{pq} (\phi_{rs}, \phi_{pq})\hat{u}_{pq} = (\phi_{rs}, u)$$

$$\sum_{pq} (\phi_{rs}, \phi_{pq})\hat{u}_{pq} = (\phi_{rs}, u)$$

$$\mathbf{M}[i][j] = \int_{\Omega^e} \phi_{i(r,s)}(x_1, x_2)\phi_{j(p,q)}(x_1, x_2)dx_1 dx_2 \qquad \hat{\mathbf{u}}[i] = \hat{u}_{i(p,q)}$$

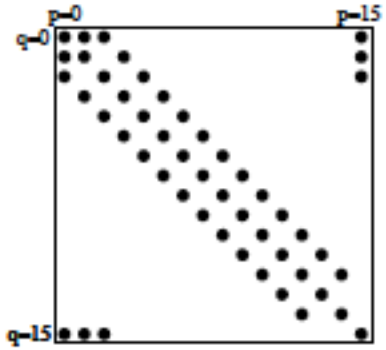
$$\mathbf{f}[i] = \int_{\Omega^e} \phi_{i,(p,q)} u(x_1, x_2)dx_1 dx_2$$

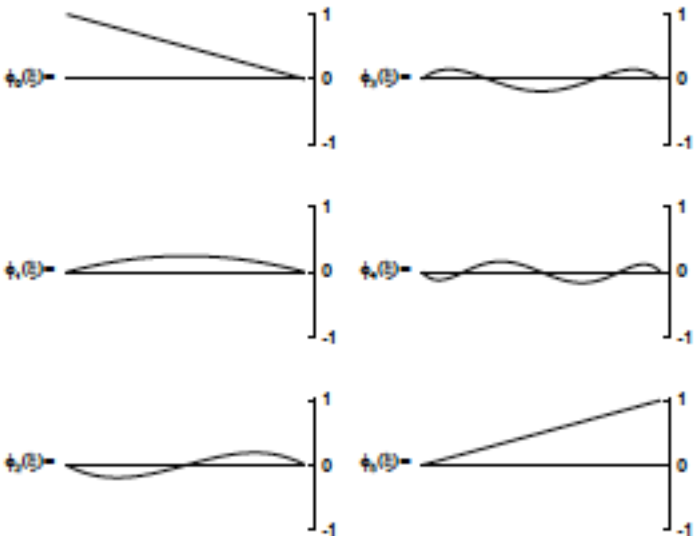
$$\downarrow$$

$$\boxed{\hat{\mathbf{u}} = \mathbf{M}^{-1}\mathbf{f}}$$

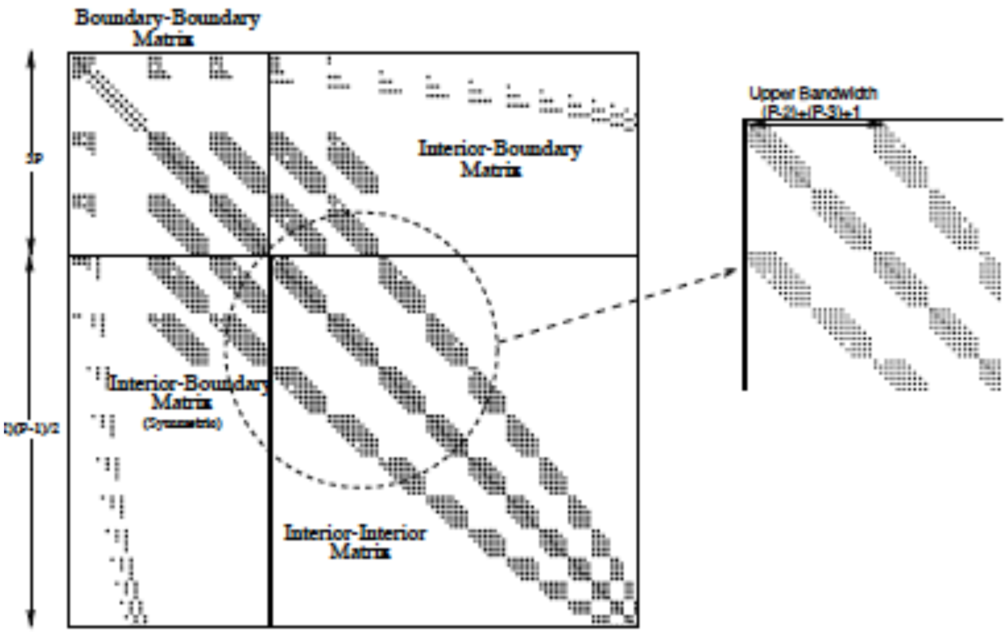
Elemental Mass Matrix Structure

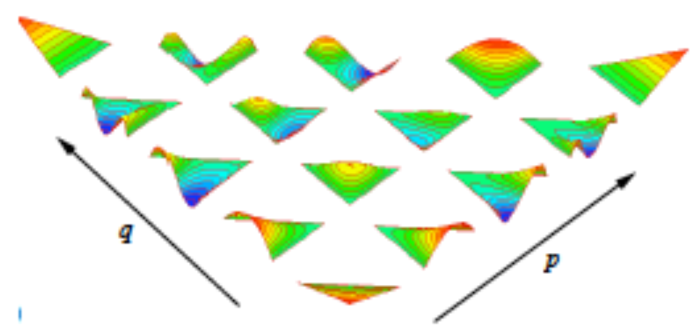
Segment expansion *P*=15





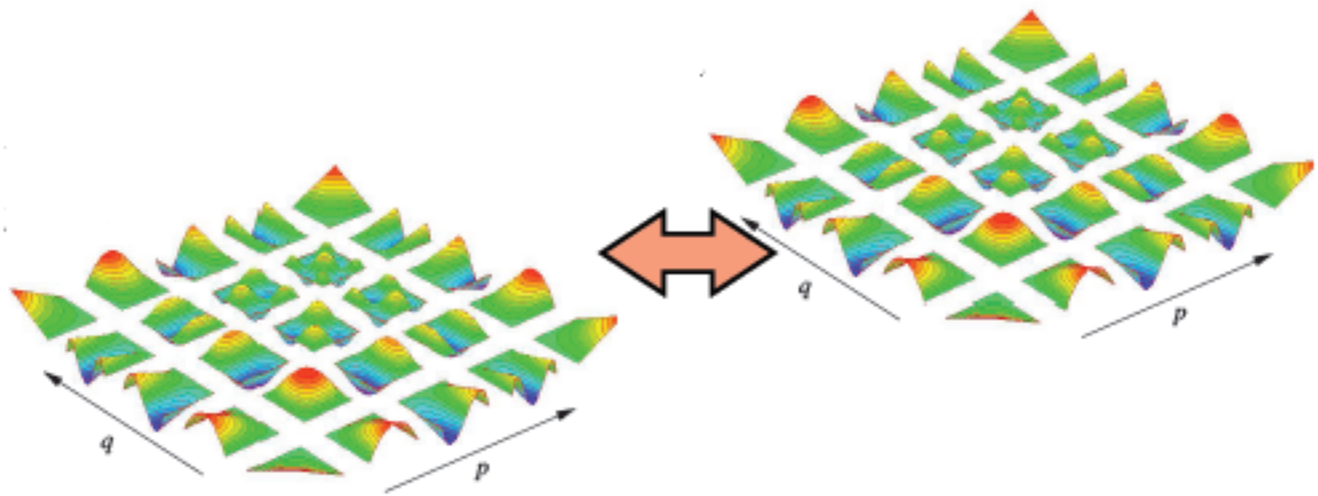
Triangular expansion *P*=14



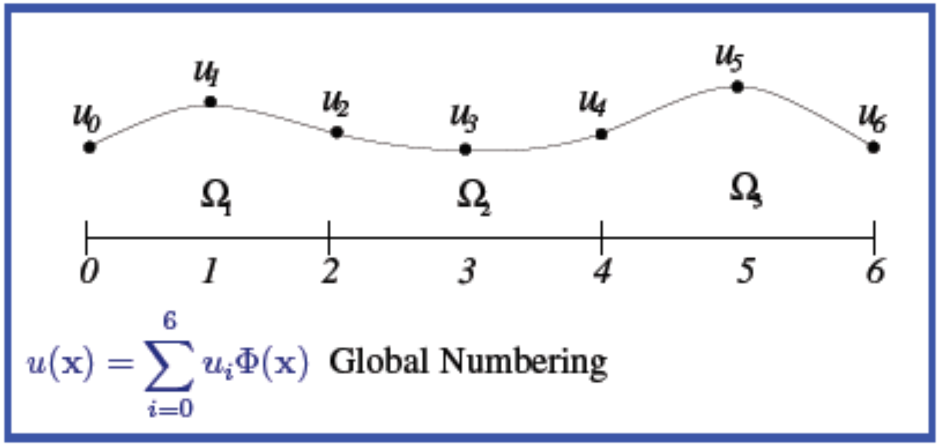


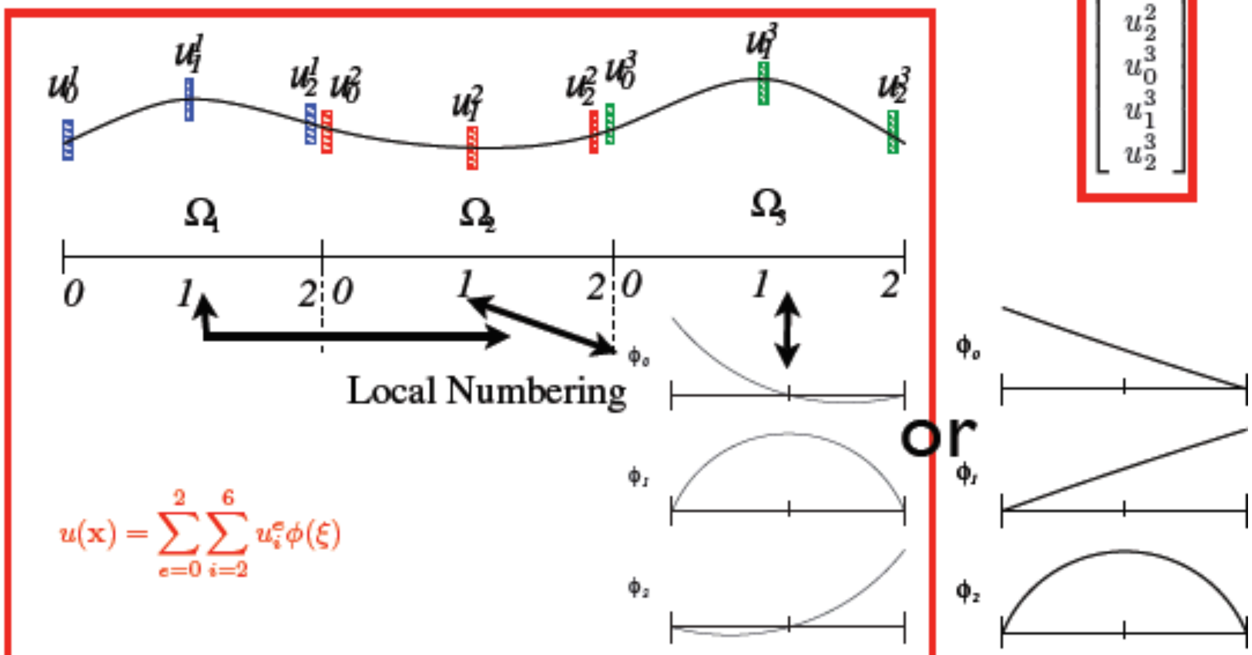
Global Assembly

- Classical continuous Galerkin FEM
- II order Partial differential equation
- C^0 continuity sufficient
- Use boundary interior decomposition to make continuous expansion.



Global Assembly





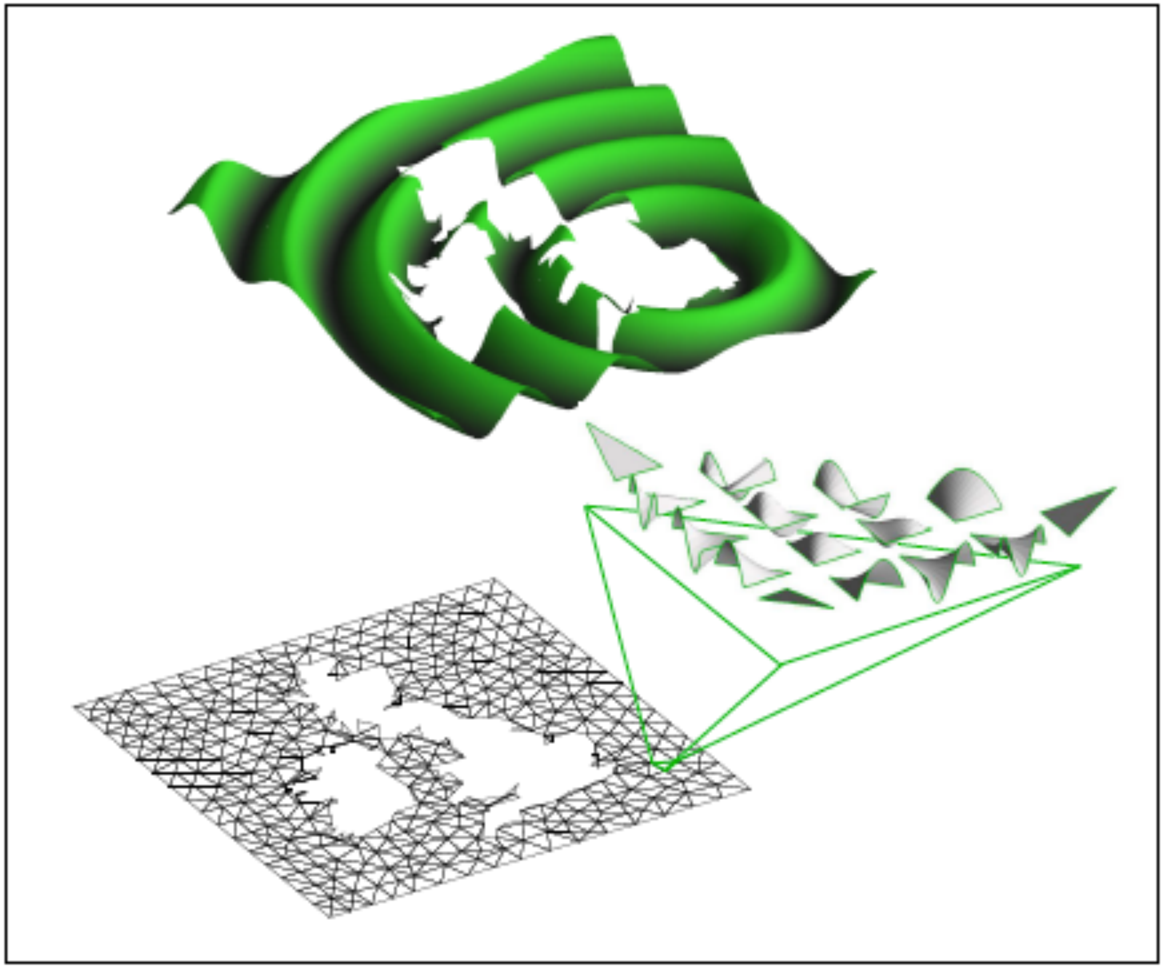
$$\tilde{\mathbf{u}}_l = \mathcal{A}\tilde{\mathbf{u}}_g$$

$$\tilde{\mathbf{u}}_l = \boxed{\begin{bmatrix} u_0^1 \\ u_1^1 \\ u_2^1 \\ u_0^2 \\ u_1^2 \\ u_2^2 \\ u_0^3 \\ u_1^3 \\ u_2^3 \end{bmatrix}} = \mathcal{A}\tilde{\mathbf{u}}_g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boxed{\begin{bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{u}_3 \\ \tilde{u}_4 \\ \tilde{u}_5 \\ \tilde{u}_6 \end{bmatrix}}$$

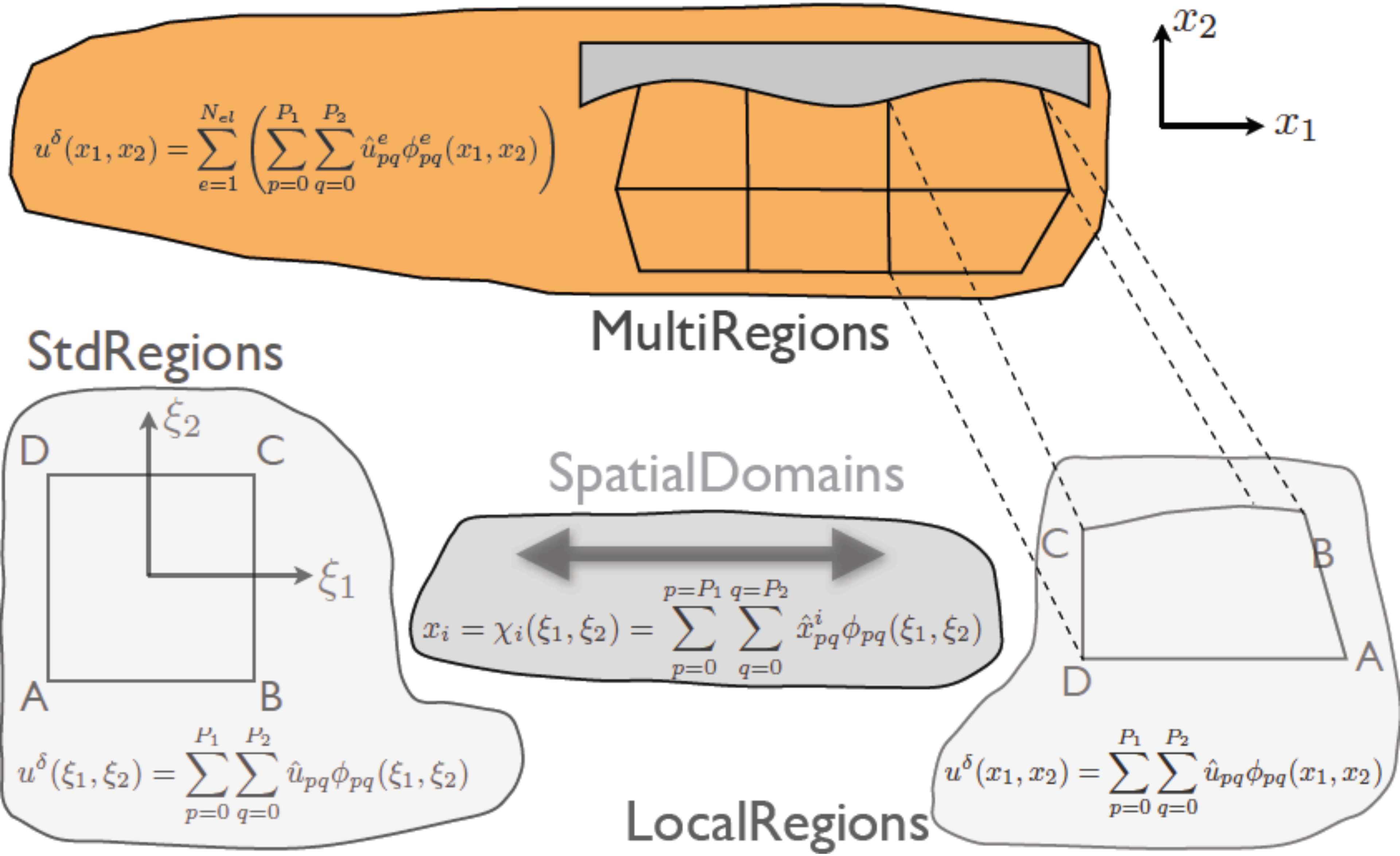
$$\tilde{\mathbf{i}}_g = \mathcal{A}^T \tilde{\mathbf{i}}_l$$

$$\mathcal{A}^T \mathcal{A} \neq \mathbf{I}$$

Expansions in Multiple Regions (*MultiRegions*)



MultiRegion Library



Poisson problem

1-D Poisson Problem ⇒
$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} u = f \\ u(0) = g_{\mathcal{D}} \\ \dfrac{\partial u}{\partial x}(l) = g_{\mathcal{N}} \end{cases}$$

Weak Formulation:
$$\int_0^l \frac{\partial v}{\partial x}\frac{\partial u}{\partial x} = \int_0^l v f \, dx + \left[v \frac{\partial u}{\partial x} \right]_0^l$$

Enforcing Neumann Boundary conditions:
$$\left[v \frac{\partial u}{\partial x} \right]_0^l = v(l) \left. \frac{\partial u}{\partial x} \right|_l - v(0) \left. \frac{\partial u}{\partial x} \right|_0$$

Enforcing Dirichlet Boundary conditions (Lift BC)
$$u^\delta = u^{\mathcal{D}} + u^{\mathcal{H}} \qquad u^{\mathcal{D}} = g_{\mathcal{D}} \qquad u^{\mathcal{H}} = 0$$

Discrete Approximation

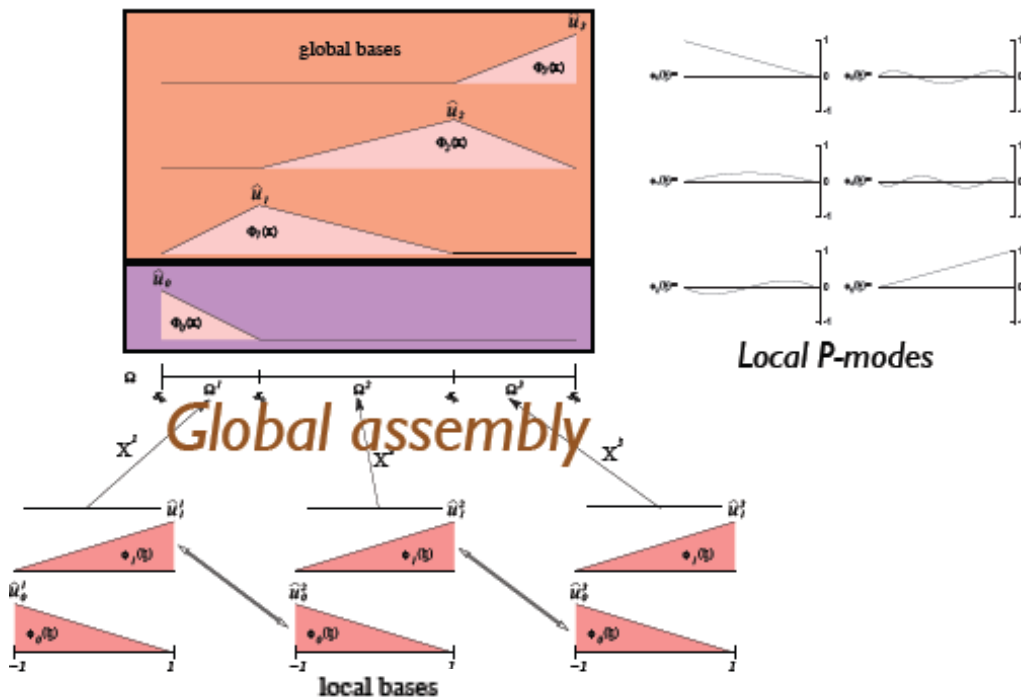
Substituting the decomposition in the weak formulation:

$$\int_0^l \frac{\partial v^{\mathcal{H}}}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x}dx = \int_0^l v f\,dx + v(l)g_{\mathcal{N}} - \int_0^l \frac{\partial v^{\mathcal{D}}}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x}dx$$

Global approximation – C^0:

$$u \implies u^\delta = \sum_i \hat{u}_i \Phi(x)$$

$$v \implies v^\delta = \sum_i \hat{v}_i \Phi(x)$$



Local approximation

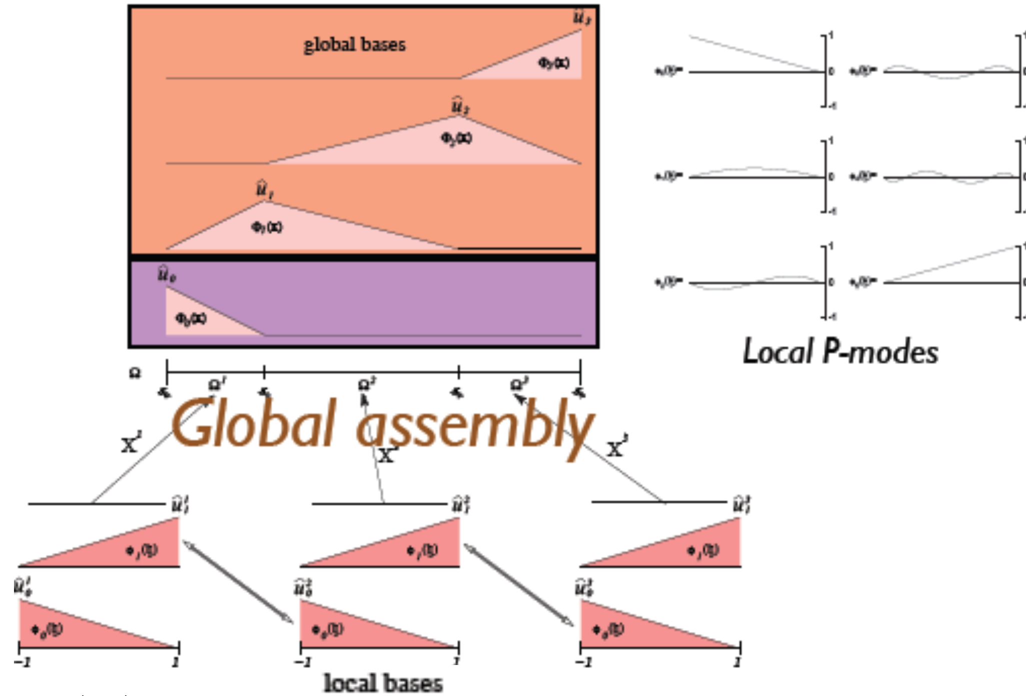
$$u^\delta = \sum_i \hat{u}_i \Phi_i(x) = \sum_e^{N_{el}} \sum_p \hat{u}_p^e \phi_p(x)$$

Discrete spaces

Global approximation – C^0:

$$u \implies u^\delta = \sum_i \hat{u}_i \Phi(x)$$

$$v \implies v^\delta = \sum_i \hat{v}_i \Phi(x)$$



Local approximation:

$$u^\delta = \sum_i \hat{u}_i \Phi_i(x) = \sum_e^{N_{el}} \sum_p \hat{u}_p^e \phi_p(x)$$

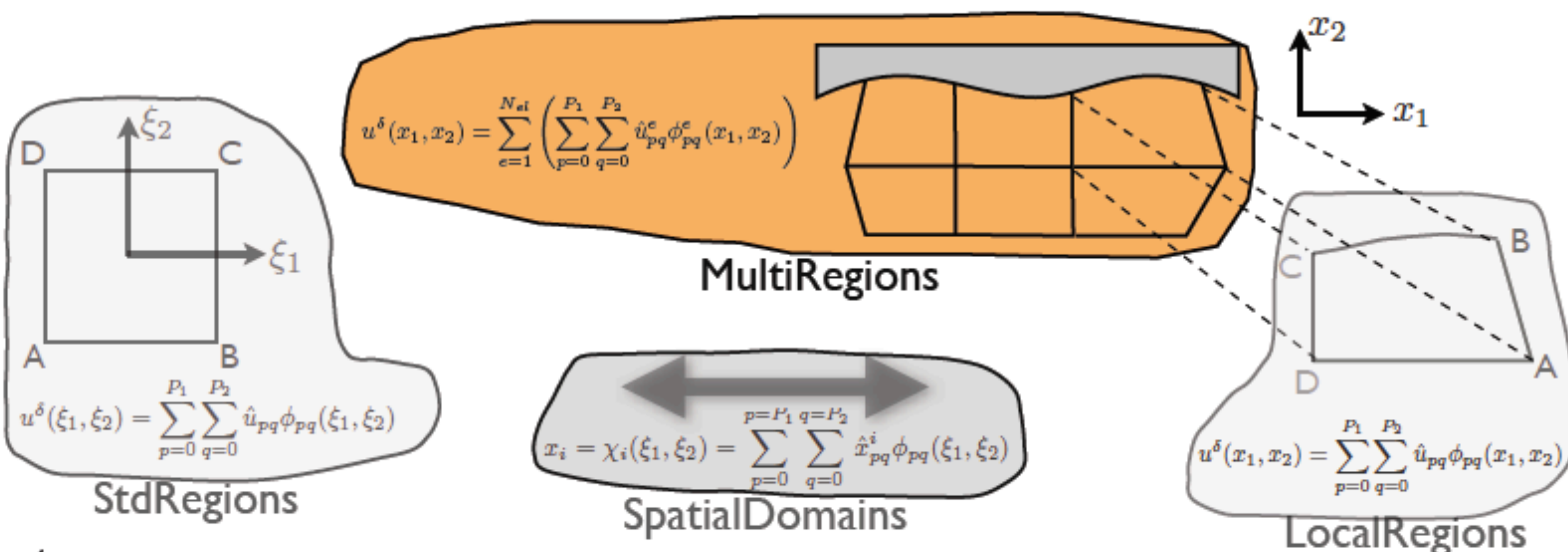
$$\boxed{\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x} \frac{\partial \Phi_j^{\mathcal{H}}}{\partial x} \hat{u}_j dx = \int_0^l \Phi_i^{\mathcal{H}} f^* dx \right\}}$$

$$\mathbf{L}[i][j] \qquad \mathbf{f}[i]$$

Evaluation of f[*i*]

$$\boxed{\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x}\frac{\partial \Phi_j^{\mathcal{H}}}{\partial x}\hat{u}_j\, dx = \int_0^l \Phi_i^{\mathcal{H}} f^*\, dx \right\}}$$

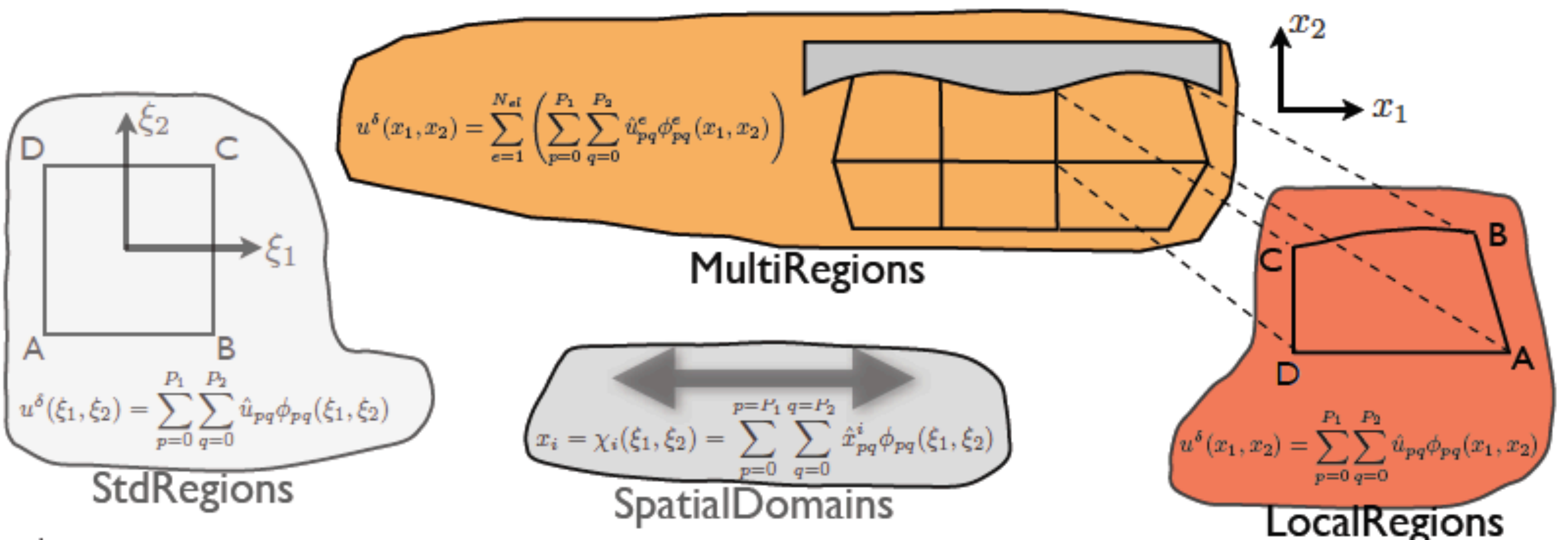
$$\mathbf{f}[i] = \int_0^l \Phi_i f^*\, dx$$



Evaluation of f[*i*]

$$\boxed{\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x} \frac{\partial \Phi_j^{\mathcal{H}}}{\partial x} \hat{u}_j \, dx = \int_0^l \Phi_i^{\mathcal{H}} f^* \, dx \right\}}$$

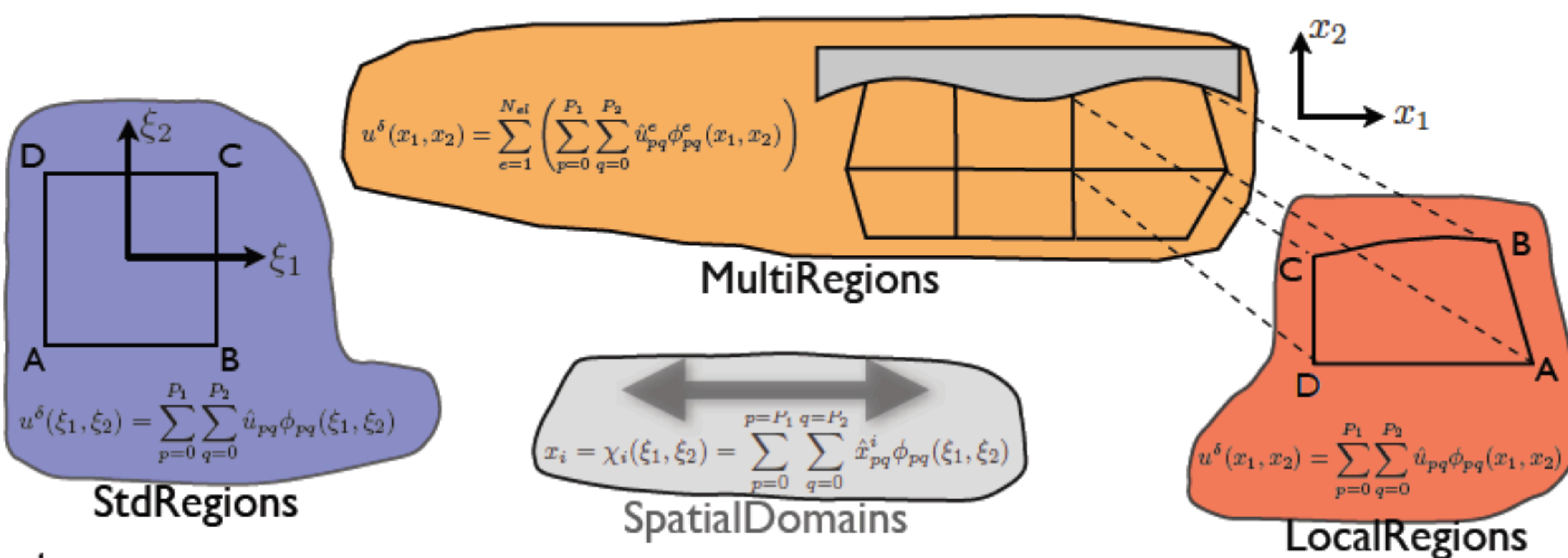
$$\mathbf{f}[i] = \int_0^l \Phi_i f^* dx = \sum_e^{N_{el}} \sum_i \int_{\Omega_e} \phi_p(x) f^* dx$$



Evaluation of f[*i*]

$$\boxed{\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x} \frac{\partial \Phi_j^{\mathcal{H}}}{\partial x} \hat{u}_j \, dx = \int_0^l \Phi_i^{\mathcal{H}} f^* \, dx \right\}}$$

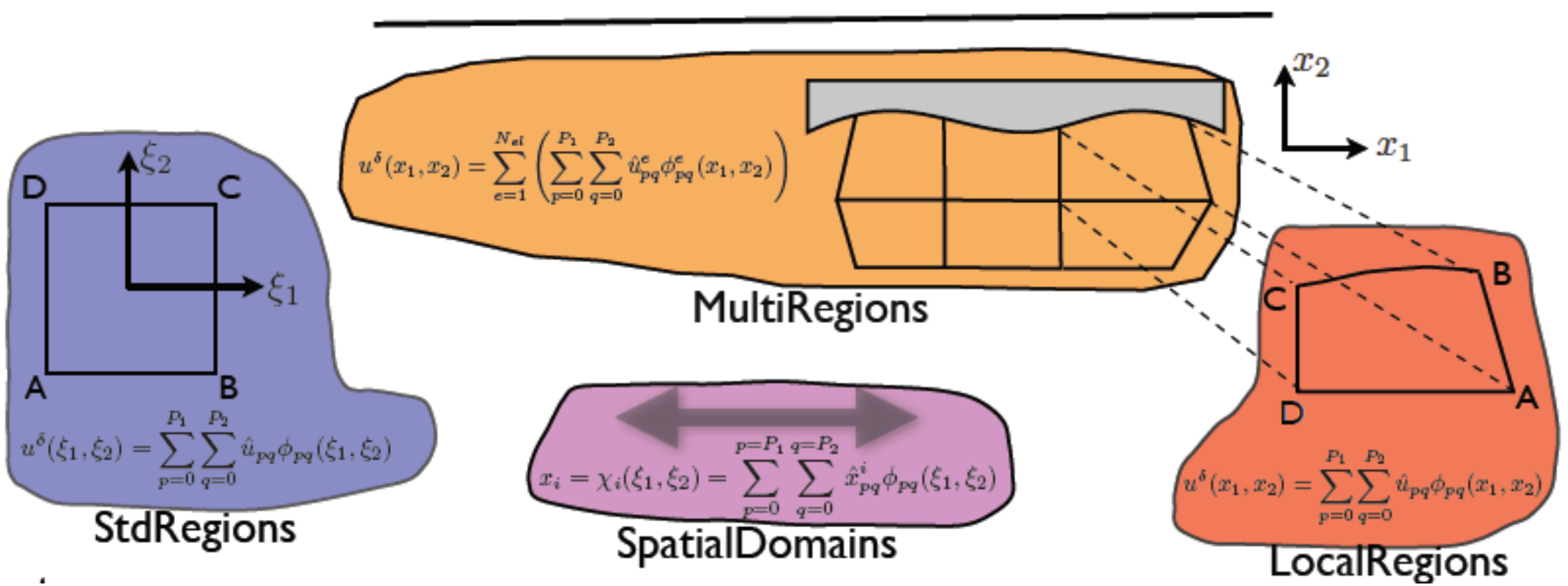
$$\mathbf{f}[i] = \int_0^l \Phi_i f^* dx = \sum_e^{N_{el}} \sum_i \int_{\Omega_e} \phi_p(x) f^* dx = \sum_e^{N_{el}} \sum_i \int_{\Omega_e} \phi_p(\chi^e(\xi)) f^* J^e d\xi$$



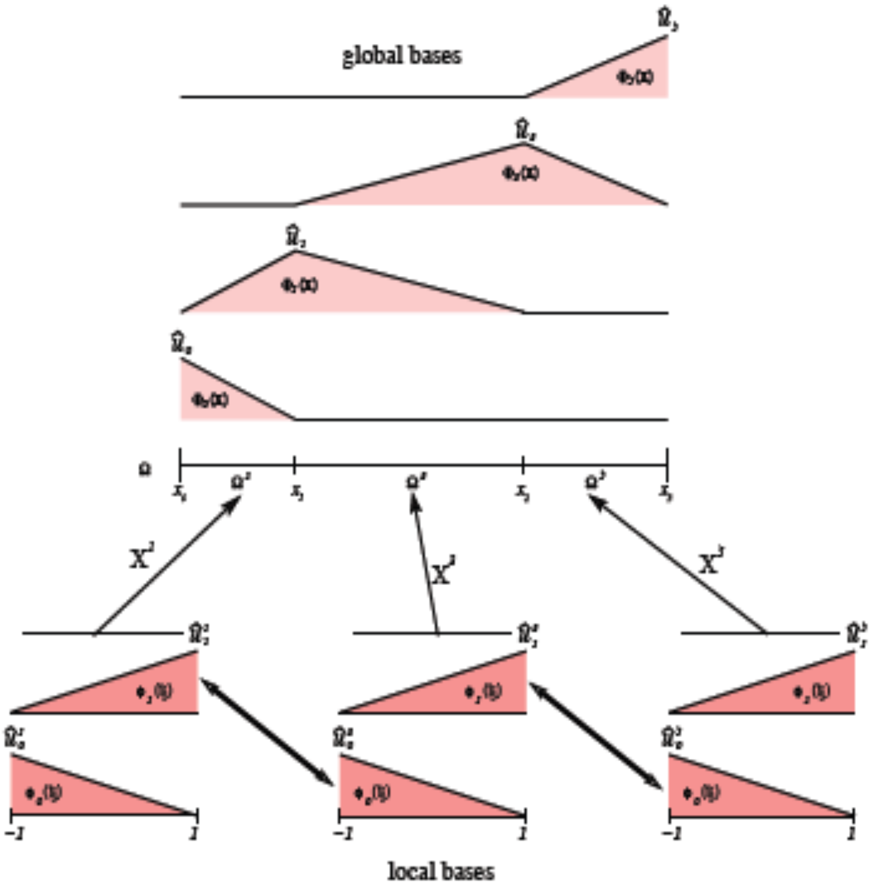
Evaluation of f[i]

$$\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x} \frac{\partial \Phi_j^{\mathcal{H}}}{\partial x} \hat{u}_j dx = \int_0^l \Phi_i^{\mathcal{H}} f^* dx \right\}$$

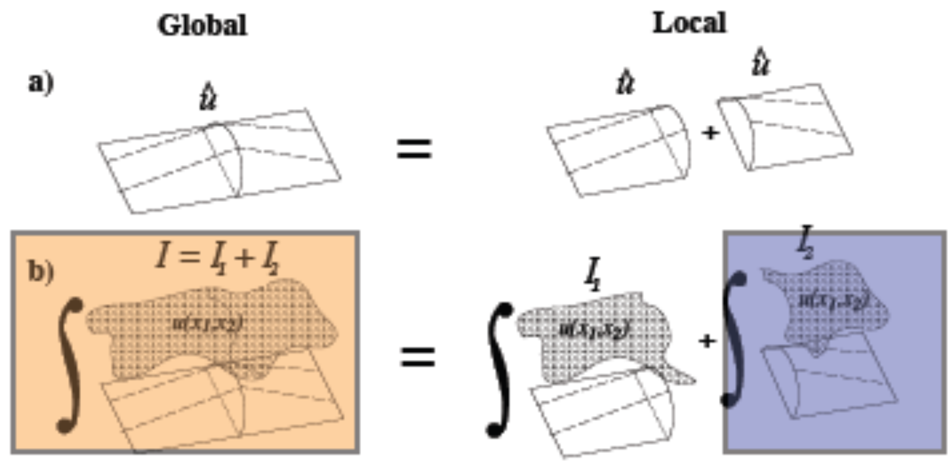
$$\mathbf{f}[i] = \int_0^l \Phi_i f^* dx = \sum_e^{N_{el}} \sum_i \int_{\Omega_e} \phi_p(x) f^* dx = \sum_e^{N_{el}} \sum_i \int_{\Omega_e} \phi_p(\chi^e(\xi)) f^* J^e d\xi$$



Global Assembly



$$\mathrm{map}[1][i] = \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\} \qquad \mathrm{map}[2][i] = \left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\} \qquad \mathrm{map}[3][i] = \left\{ \begin{matrix} 2 \\ 3 \end{matrix} \right\}$$



$$I^e[i] = \sum_e^{N_{el}} \sum_p \int_{\Omega_e} \phi_p(\chi^e(\xi)) f^* J^e d\xi$$

$$f[i] = \int_0^1 \Phi_i f^* dx$$

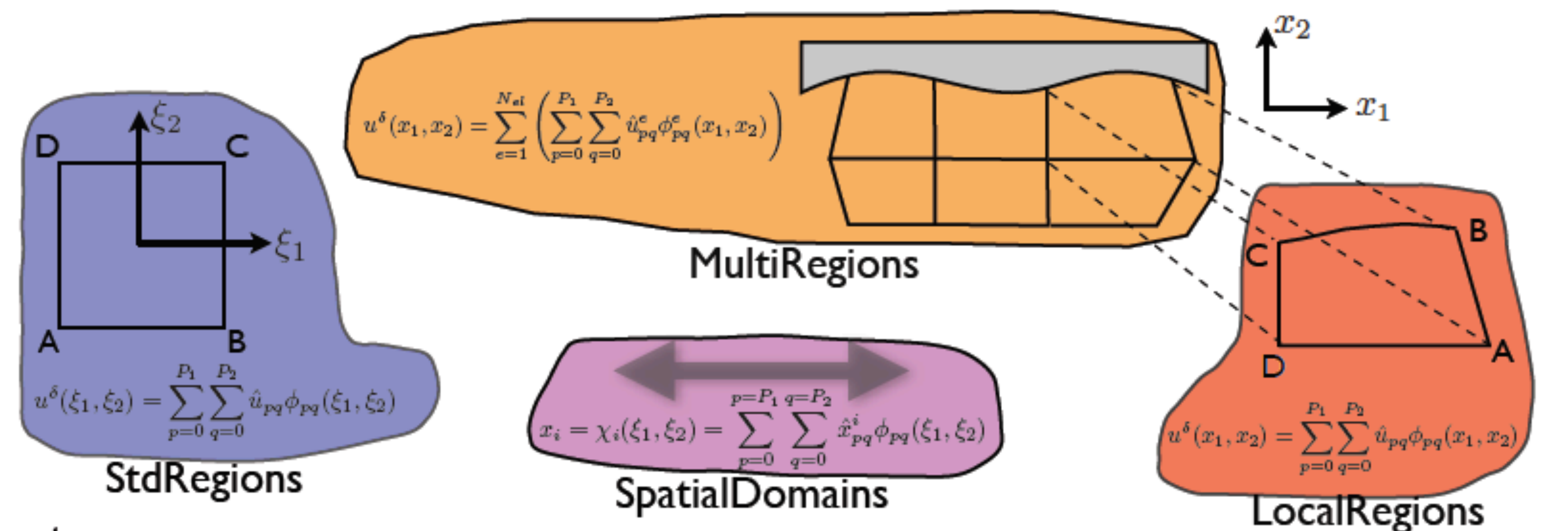
$$
\begin{array}{l}
Do \quad e = 1, N_{el} \\
\quad Do \quad i = 1, N_{el} - 1 \\
\quad f[map[e][i]] = f[map[e][i]] + I^e[i] \\
\quad continue \\
continue
\end{array}
$$

Matrix Construction

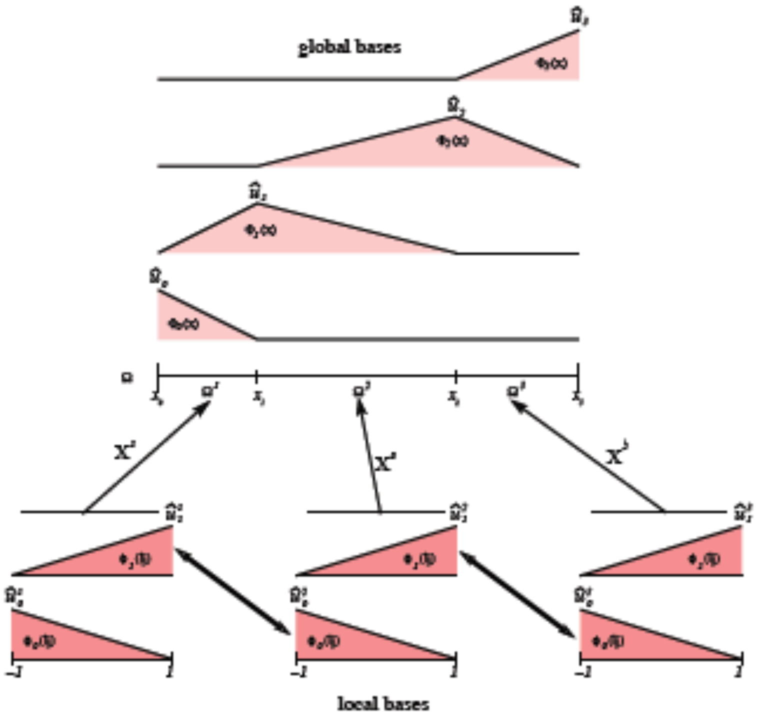
$$\boxed{\sum_i \hat{v}_j \left\{ \sum_j \int_0^l \frac{\partial \Phi_i^{\mathcal{H}}}{\partial x}\frac{\partial \Phi_j^{\mathcal{H}}}{\partial x}\hat{u}_j dx = \int_0^l \Phi_i^{\mathcal{H}} f^* dx \right\}}$$

$$\mathbf{L}[i][j] = \int_0^l \frac{\partial \Phi_i}{\partial x}\frac{\partial \Phi_j}{\partial x} dx = \sum_e^{N_{el}} \sum_p \sum_q \int_{\Omega^e} \frac{\partial \phi_p}{\partial x}\frac{\partial \phi_q}{\partial x} dx =$$

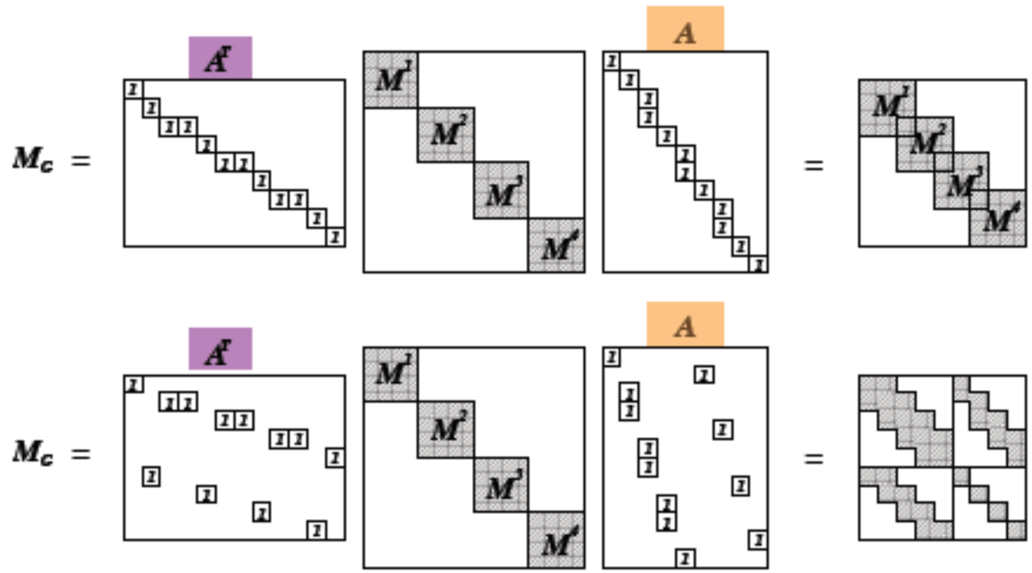
$$= \underline{\sum_e^{N_{el}} \sum_p \sum_q \int_{-1}^{1} \frac{\partial \phi_p(\chi^e(\xi))}{\partial x}\frac{\partial \phi_q(\chi^e(\xi))}{\partial x} J^e d\xi}$$



Matrix Construction



$\text{map}[1][i] = \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\}$ $\quad \text{map}[2][i] = \left\{ \begin{array}{c} 1 \\ 2 \end{array} \right\}$ $\quad \text{map}[3][i] = \left\{ \begin{array}{c} 2 \\ 3 \end{array} \right\}$



$$\left.\begin{array}{l} \text{Do } e = 1, N_{el} \\ \quad \text{Do } i = 0, N_m^e - 1 \\ \quad\quad \hat{u}^e[i] = \hat{u}_g[\text{map}[e][i]] \\ \quad \text{continue} \\ \text{continue} \end{array}\right\} \Leftrightarrow \hat{u}_l = \mathcal{A}\hat{u}_g,$$

$$\left.\begin{array}{l} \text{Do } e = 1, N_{el} \\ \quad \text{Do } i = 0, N_m^e - 1 \\ \quad\quad \hat{u}_g[\text{map}[e][i]] = \hat{u}_g[\text{map}[e][i]] + \hat{u}^e[i] \\ \quad \text{continue} \\ \text{continue} \end{array}\right\} \Leftrightarrow \hat{u}_g = \mathcal{A}^T \hat{u}_l$$