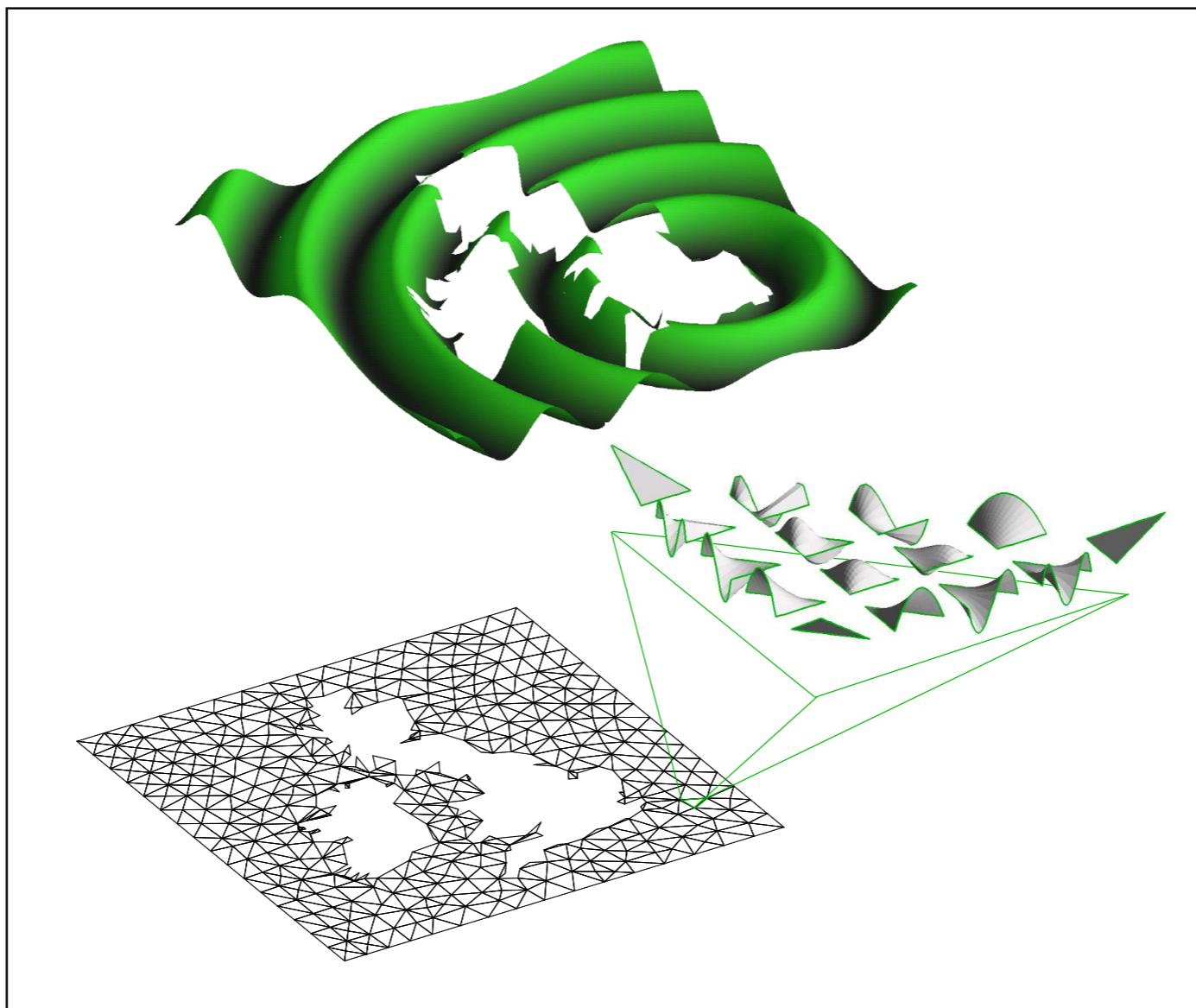
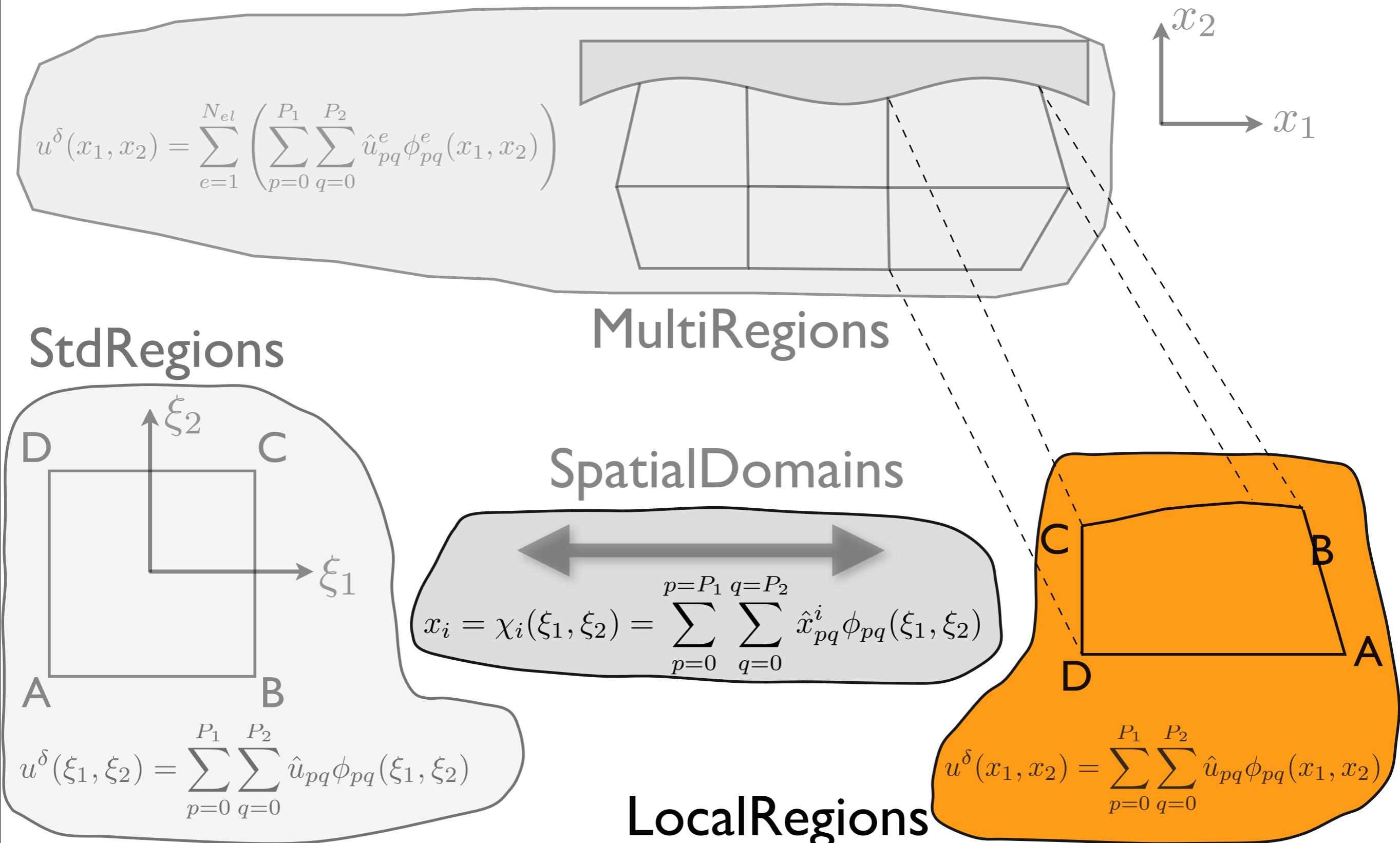


Expansions in Local Regions

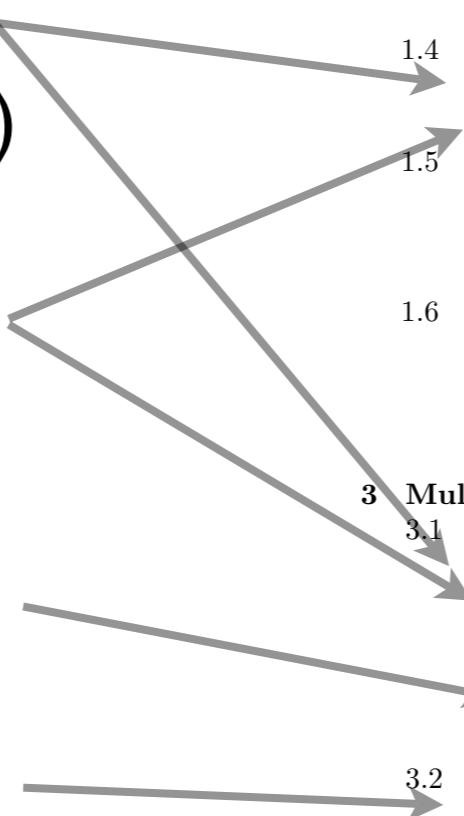


The big picture



Outline

- 2D Integration
 - Metrics (Jacobian)
- 2D Differentiation
 - Metrics
- Elemental projection
- Global Assembly



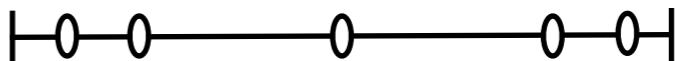
1	Fundamental Concepts in One Dimension	1
1.1	Method of Weighted Residuals	2
1.2	Galerkin Formulation	5
1.2.1	Descriptive Formulation	5
1.2.2	Two-Domain Linear Finite Element Example	8
1.2.3	Mathematical Formulation	11
1.2.4	Mathematical Properties of the Galerkin Approximation	13
1.2.5	Residual Equation for C^0 Test and Trial Functions	15
1.3	One-Dimensional Expansion Bases	16
1.3.1	Elemental Decomposition: The h -Type Extension	16
1.3.2	Polynomial Expansions: The p -Type Extension	22
1.3.3	Modal Polynomial Expansions	28
1.3.4	Nodal Polynomial Expansions	31
1.4	Elemental Operations	34
1.4.1	Numerical Integration	34
1.4.2	Differentiation	39
1.5	Error Estimates	42
1.5.1	h -Convergence of Linear Finite Elements	42
1.5.2	L^2 Error of the p -Type Interpolation in a Single Element	44
1.5.3	General Error Estimates for hp Elements	45
1.6	Implementation of a 1D Spectral/ hp Element Solver:	46
1.6.1	Exercises	46
1.6.2	Convergence Examples	50
3	Multi-dimensional Formulation	98
3.1	Local Elemental Operations	99
3.1.1	Integration within the Standard Region Ω_{st}	99
3.1.2	Differentiation in the Standard Region Ω_{st}	104
3.1.3	Operations within General Shaped Elements	109
3.1.4	Discrete Evaluation of the Surface Jacobian	115
3.1.5	Elemental Projections and Transformations	118
3.1.6	Sum Factorisation/Tensor Product Operations	129
3.2	Global Operations	134
3.2.1	Global Assembly and Connectivity	135
3.2.2	Global Matrix System	146
3.2.3	Static Condensation/Substructuring	149
3.2.4	Global Boundary System Numbering and Ordering to Enforce Dirichlet Boundary Conditions	153
3.3	Pre- and Post-Processing Issues	157
3.3.1	Boundary Condition Discretisation	158
3.3.2	Elemental Boundary Transformation	158
3.3.3	Mesh Generation for Spectral/ hp Element Discretisation	161
3.3.4	Global Coarse Meshing	162
3.3.5	High-Order Mesh Generation	163
3.3.6	Particle Tracking in Spectral/ hp Element Discretisations	172
3.4	Exercises: Implementation of a 2D Spectral/ hp Element solver for a Global Projection Problem Using a C^0 Galerkin Formulation	181

Integration: Gaussian Quadrature

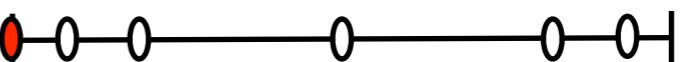
$$\int_{-1}^1 (1 - \xi)^\alpha (1 + \xi)^\beta f(\xi) d\xi = \sum_{p=0}^{Q-1} w_p^{\alpha, \beta} f(\xi_p) + R(f)$$

$$w_i^{\alpha, \beta} = \int_{-1}^1 (1 - \xi)^\alpha (1 + \xi)^\beta h_p(\xi) d\xi,$$
$$R(f) = \int_{-1}^1 (1 - \xi)^\alpha (1 + \xi)^\beta s(\xi) r(\xi) d\xi,$$

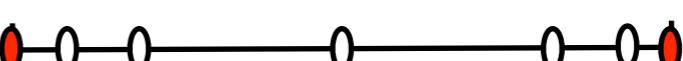
- *Gauss quadrature* is obtained if $k = 1$. Gauss quadrature use zeros which lie inside the domain, $-1 < \xi_p < 1$ for $p = 0, \dots, Q - 1$. Gauss quadrature is exact for $f(\xi)$ being a polynomial order $P \leq 2Q - 1$;



- *Gauss-Radau quadrature* is given by $k = 2$. This quadrature uses one point at the end of the domain, usually $\xi_0 = -1$. The rest lie inside the domain, $-1 < \xi_p < 1$ for $p = 1, \dots, Q - 1$. Gauss-Radau quadrature is exact for $f(\xi)$ being a polynomial order $P \leq 2Q - 2$;



- *Gauss-Lobatto quadrature* is defined by $k = 3$. Here both end-points of the domain are used, $\xi_0 = -1$ and $\xi_{Q-1} = 1$. The remaining zeros lie inside the domain, $-1 < \xi_i < 1$ for $i = 1, \dots, Q - 2$. Gauss-Lobatto quadrature is exact for $f(\xi)$ being a polynomial order $P \leq 2Q - 3$.

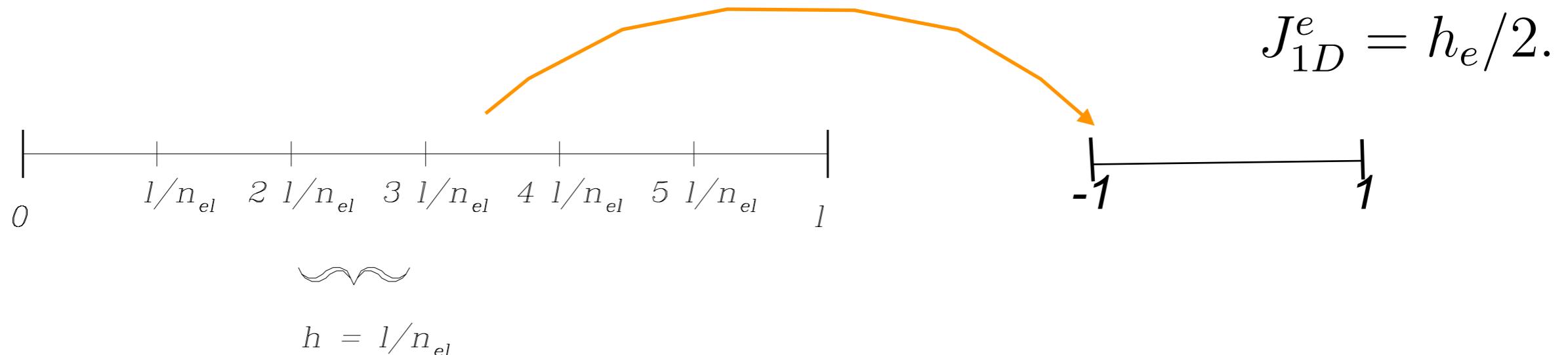


Polylib at <http://www.nektar.info>

Numerical Integration: general segment

$$\int_{\Omega^e} f(x) dx ,$$

Elemental mapping $\int_{\Omega^e} f(x) dx , = \int_{-1}^1 J_{1D}^e f(\xi) d\xi , \quad J_{1D}^e = \partial x / \partial \xi$

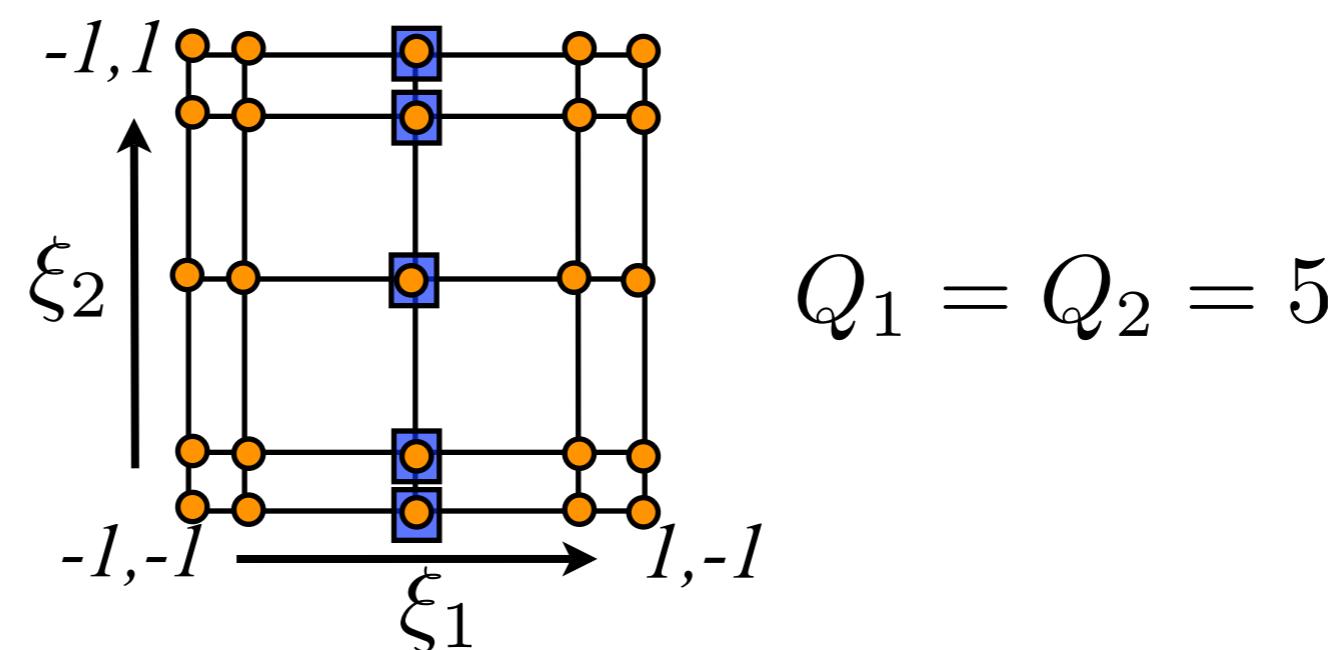


Integration: Standard Quadrilateral

StdRegions::StdQuadExp::Integral()

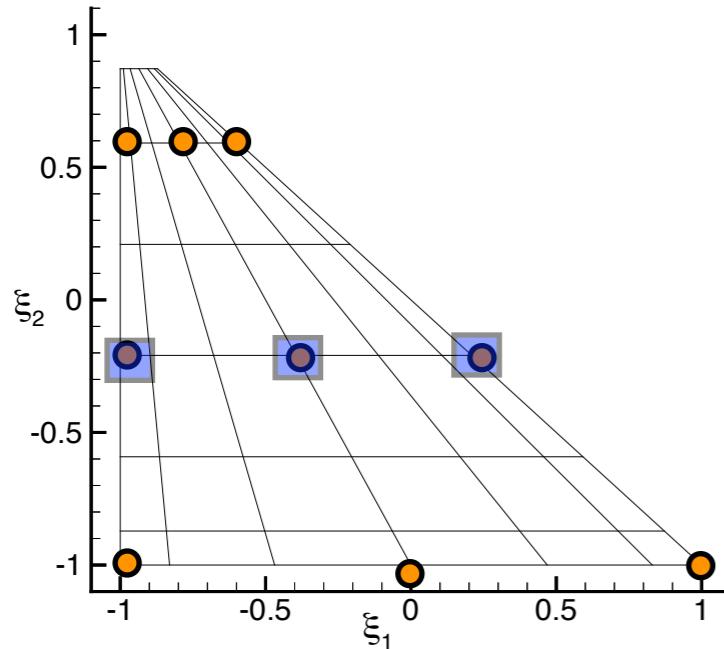
$$\int_{Q^2} u(\xi_1, \xi_2) \, d\xi_1 \, d\xi_2 = \int_{-1}^1 \left\{ \int_{-1}^1 u(\xi_1, \xi_2) \Big|_{\xi_2} \, d\xi_1 \right\} d\xi_2.$$

$$\int_{Q^2} u(\xi_1, \xi_2) \, d\xi_1 \, d\xi_2 \simeq \sum_{j=0}^{Q_2-1} w_j \left\{ \sum_{i=0}^{Q_1-1} w_i u(\xi_{1i}, \xi_{2j}) \right\}$$



Integration: Standard Triangle

StdRegions::StdTriExp::Integral()



$$\begin{aligned} \int_{T^2} u(\xi_1, \xi_2) \, d\xi_1 d\xi_2 &= \int_{-1}^1 \int_{-1}^{-\xi_2} u(\xi_1, \xi_2) \, d\xi_1 d\xi_2 \\ &= \int_{-1}^1 \int_{-1}^1 u(\eta_1, \eta_2) \left| \frac{\partial(\xi_1, \xi_2)}{\partial(\eta_1, \eta_2)} \right| \, d\eta_1 d\eta_2, \end{aligned}$$

$$\frac{\partial(\xi_1, \xi_2)}{\partial(\eta_1, \eta_2)} = \left(\frac{1 - \eta_2}{2} \right).$$

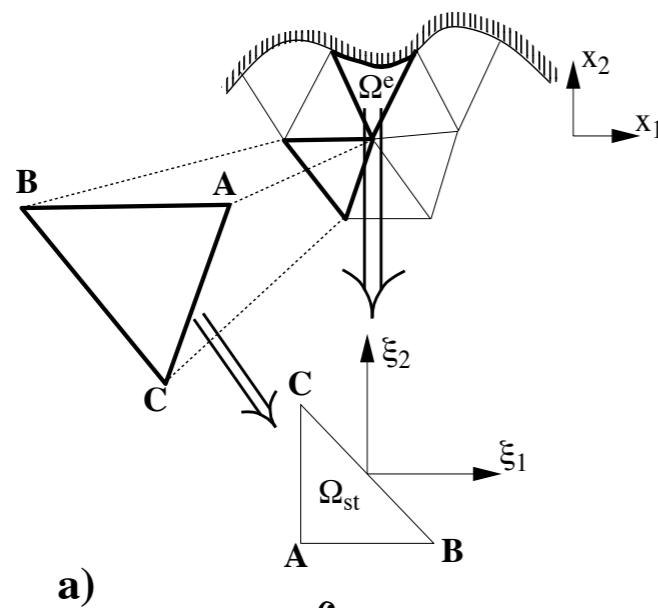
$$\eta_1 = 2 \frac{(1 + \xi_1)}{(1 - \xi_2)} - 1, \quad \eta_2 = \xi_2.$$

$$\int_{-1}^1 \int_{-1}^1 u(\eta_1, \eta_2) \left(\frac{1 - \eta_2}{2} \right) \, d\eta_1 d\eta_2 = \sum_{i=0}^{Q_1-1} w_i \left\{ \sum_{j=0}^{Q_2-1} w_j u(\eta_{1i}, \eta_{2j}) \left(\frac{1 - \eta_{2j}}{2} \right) \right\}$$

Weighted quadrature: $\int_{-1}^1 (1 - \xi)^\alpha (1 + \xi)^\beta u(\xi) \, d\xi = \sum_{i=0}^{Q-1} w^{\alpha, \beta} u(\xi_i^{\alpha, \beta}),$

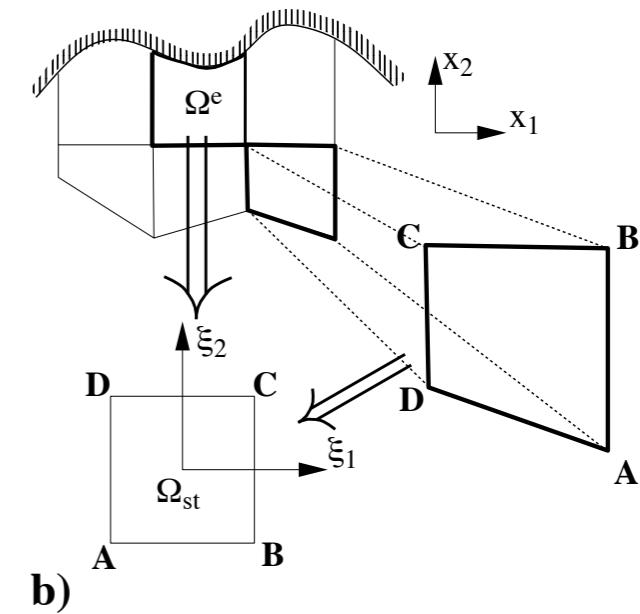
$$\int_{-1}^1 \int_{-1}^1 u(\eta_1, \eta_2) \left(\frac{1 - \eta_2}{2} \right) \, d\eta_1 d\eta_2 = \sum_{i=0}^{Q_1-1} w_i^{0,0} \left\{ \sum_{j=0}^{Q_2-1} \hat{w}_j^{1,0} u(\eta_{1i}, \eta_{2j}) \right\} \quad \hat{w}_j^{1,0} = \frac{w_j^{1,0}}{2}.$$

Mapping metrics: Jacobian



a)

$$\chi_1(\xi_1, \xi_2)$$
$$\chi_2(\xi_1, \xi_2)$$



b)

$$\int_{\Omega^e} u(x_1, x_2) \, dx_1 \, dx_2 = \int_{\Omega_{st}} u(\xi_1, \xi_2) |J_{2D}| \, d\xi_1 \, d\xi_2,$$

Jacobian:
$$J_{2D} = \begin{vmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{vmatrix} = \frac{\partial x_1}{\partial \xi_1} \frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2} \frac{\partial x_2}{\partial \xi_1}.$$

Note Jacobian requires differentiation with respect to standard coordinates

Course Notes: Section 3.1.3.1 & 3.1.3.3

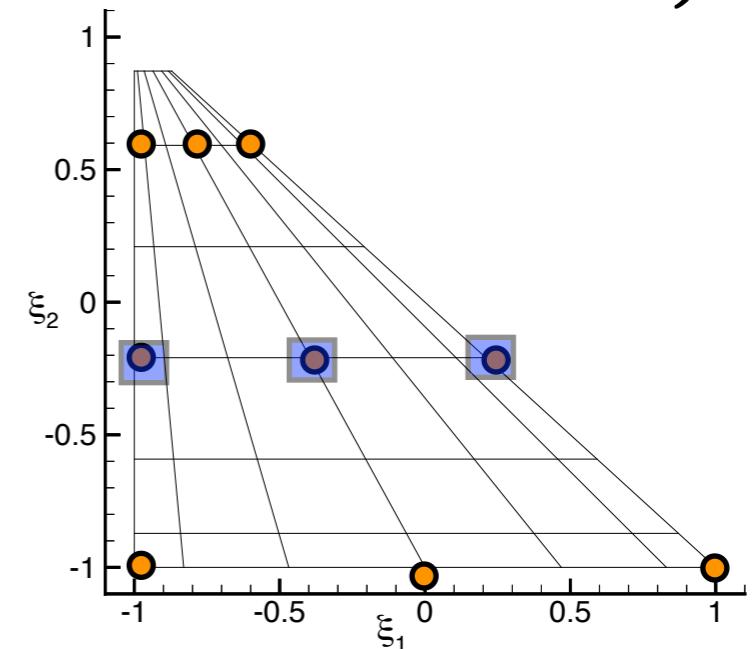
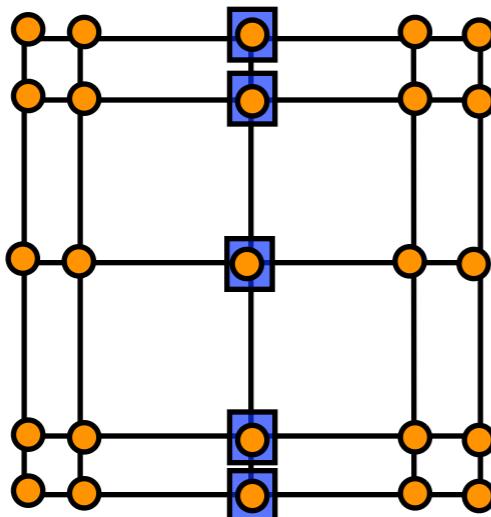
Integration: General 2D Elements

LocalRegions::QuadExp::Integral()

LocalRegions::TriExp::Integral()

$$\int_{\Omega^e} u(x_1, x_2) \, dx_1 \, dx_2 = \int_{\Omega_{st}} u(\xi_1, \xi_2) |J_{2D}| \, d\xi_1 \, d\xi_2,$$

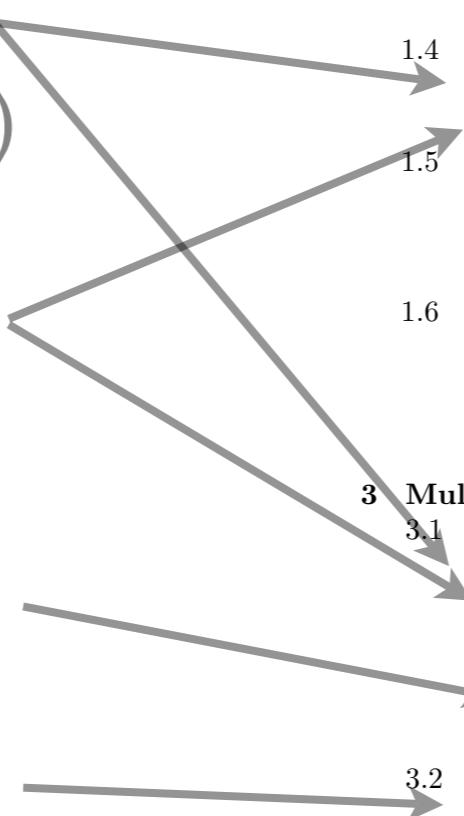
$$\int_{\Omega^e} u(x_1, x_2) \, dx_1 \, dx_2 \simeq \sum_{j=0}^{Q_2-1} w_j \left\{ \sum_{i=0}^{Q_1-1} w_i u(\xi_{1i}, \xi_{2j}) |J_{2D}|(\xi_{1i}, \xi_{2j}) \right\}$$



Course Notes: Section 3.1.3.3

Outline

- 2D Integration
 - Metrics (Jacobian)
- 2D Differentiation
 - Metrics
- Elemental projection
- Global Assembly



1	Fundamental Concepts in One Dimension	1
1.1	Method of Weighted Residuals	2
1.2	Galerkin Formulation	5
1.2.1	Descriptive Formulation	5
1.2.2	Two-Domain Linear Finite Element Example	8
1.2.3	Mathematical Formulation	11
1.2.4	Mathematical Properties of the Galerkin Approximation	13
1.2.5	Residual Equation for C^0 Test and Trial Functions	15
1.3	One-Dimensional Expansion Bases	16
1.3.1	Elemental Decomposition: The h -Type Extension	16
1.3.2	Polynomial Expansions: The p -Type Extension	22
1.3.3	Modal Polynomial Expansions	28
1.3.4	Nodal Polynomial Expansions	31
1.4	Elemental Operations	34
1.4.1	Numerical Integration	34
1.4.2	Differentiation	39
1.5	Error Estimates	42
1.5.1	h -Convergence of Linear Finite Elements	42
1.5.2	L^2 Error of the p -Type Interpolation in a Single Element	44
1.5.3	General Error Estimates for hp Elements	45
1.6	Implementation of a 1D Spectral/ hp Element Solver:	46
1.6.1	Exercises	46
1.6.2	Convergence Examples	50
3	Multi-dimensional Formulation	98
3.1	Local Elemental Operations	99
3.1.1	Integration within the Standard Region Ω_{st}	99
3.1.2	Differentiation in the Standard Region Ω_{st}	104
3.1.3	Operations within General Shaped Elements	109
3.1.4	Discrete Evaluation of the Surface Jacobian	115
3.1.5	Elemental Projections and Transformations	118
3.1.6	Sum Factorisation/Tensor Product Operations	129
3.2	Global Operations	134
3.2.1	Global Assembly and Connectivity	135
3.2.2	Global Matrix System	146
3.2.3	Static Condensation/Substructuring	149
3.2.4	Global Boundary System Numbering and Ordering to Enforce Dirichlet Boundary Conditions	153
3.3	Pre- and Post-Processing Issues	157
3.3.1	Boundary Condition Discretisation	158
3.3.2	Elemental Boundary Transformation	158
3.3.3	Mesh Generation for Spectral/ hp Element Discretisation	161
3.3.4	Global Coarse Meshing	162
3.3.5	High-Order Mesh Generation	163
3.3.6	Particle Tracking in Spectral/ hp Element Discretisations	172
3.4	Exercises: Implementation of a 2D Spectral/ hp Element solver for a Global Projection Problem Using a C^0 Galerkin Formulation	181

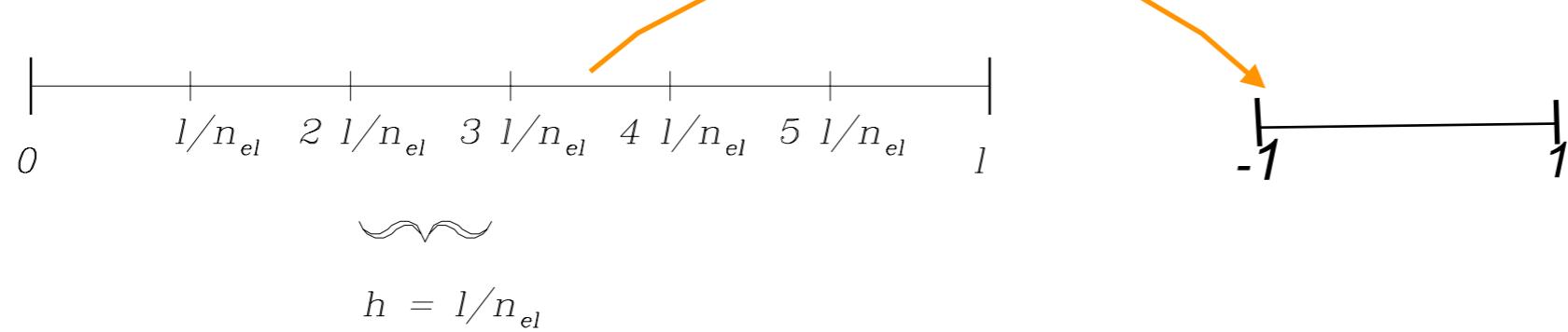
Numerical Differentiation

Elemental representation:

$$u^\delta(x) = \sum_{p=0}^P \hat{u}_p \phi_p(\chi^{-1}) = \sum_{p=0}^P \hat{u}_p \phi_p(\xi), \quad = \sum_{p=0}^P u(\xi_p) h_p(\xi)$$

Chain rule:

$$\frac{du^\delta}{dx} = \boxed{\frac{du^\delta}{d\xi}} \frac{d\xi}{dx}$$



$$\frac{d\xi}{dx} = \frac{h}{2}$$

Collocation differentiation:

$$\boxed{\frac{du(\xi)}{d\xi}} \Big|_{\xi=\xi_i} = \sum_{j=0}^{Q-1} d_{ij} u(\xi_j), \quad d_{ij} = \boxed{\frac{dh_j(\xi)}{d\xi}} \Big|_{\xi=\xi_i}$$

Collocation property:

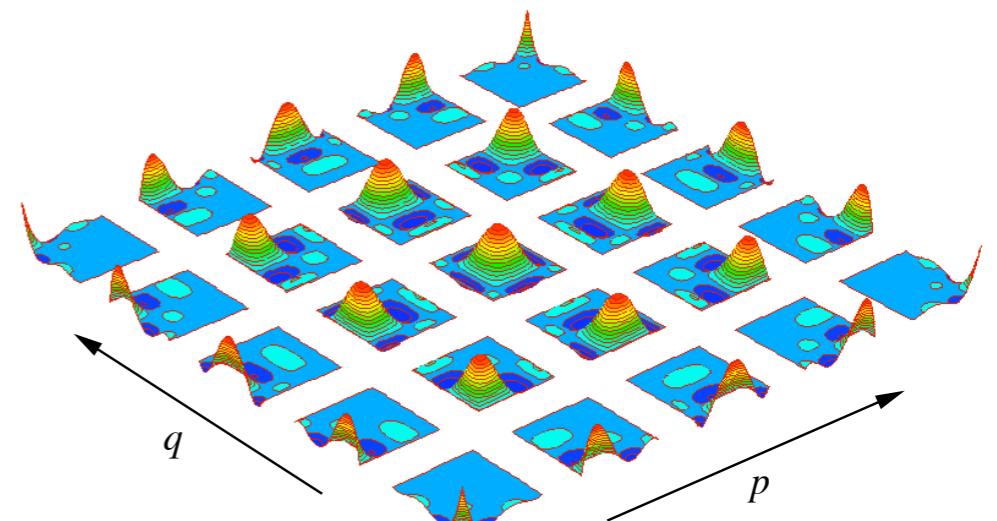
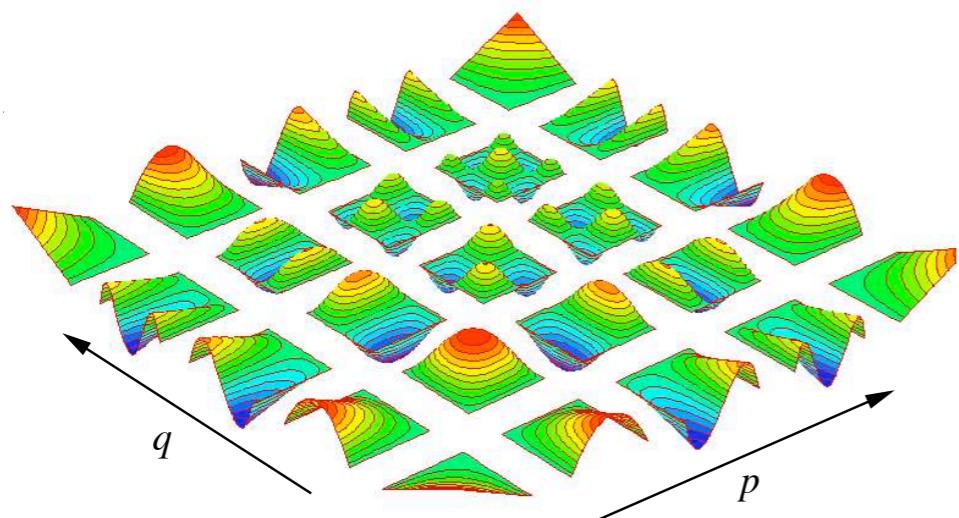
$$u(\xi) \frac{du}{dx} = \sum_{p=0}^P u(x_p) h_p(\xi) \cdot \sum_{p=0}^P \frac{du}{d\xi}(\xi_p) h_p(\xi) \simeq \sum_{p=0}^P u(\xi_p) \frac{du}{d\xi}(x_p) h_p(\xi)$$

Numerical Differentiation: Standard 2D regions

$$u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \hat{u}_{pq} \phi_{pq}(\xi_1, \xi_2),$$



$$u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{Q_1-1} \sum_{q=0}^{Q_2-1} u_{pq} h_p(\xi_1) h_q(\xi_2),$$
$$u_{pq} = u^\delta(\xi_{1p}, \xi_{2q}),$$



Course Notes: Section 3.1.2.1

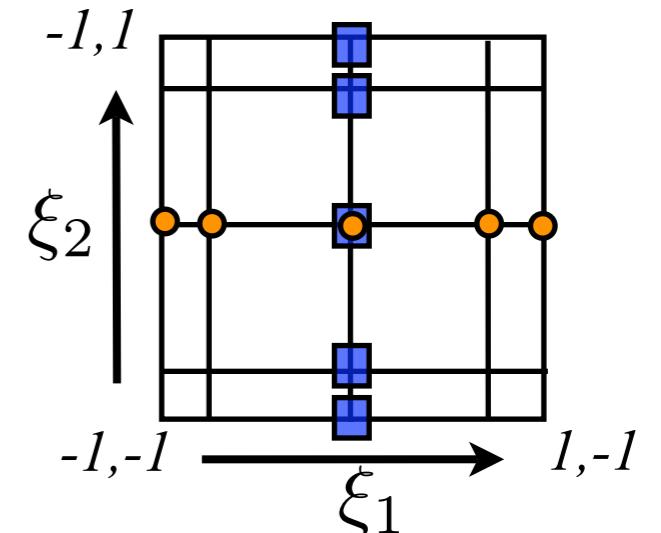
Numerical Differentiation: Standard Quad

StdRegions::StdQuadExp::PhysDeriv()

$$u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \hat{u}_{pq} \phi_{pq}(\xi_1, \xi_2), \quad \longleftrightarrow \quad u^\delta(\xi_1, \xi_2) = \sum_{p=0}^{Q_1-1} \sum_{q=0}^{Q_2-1} u_{pq} h_p(\xi_1) h_q(\xi_2),$$

$$u_{pq} = u^\delta(\xi_{1p}, \xi_{2q}),$$

$$\frac{\partial u^\delta}{\partial \xi_1}(\xi_1, \xi_2) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} u_{pq} \frac{dh_p(\xi_1)}{d\xi_1} h_q(\xi_2).$$



$$\frac{\partial u^\delta}{\partial \xi_1}(\xi_{1i}, \xi_{2j}) = \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} \left\{ u_{pq} \left. \frac{dh_p(\xi_1)}{d\xi_1} \right|_{\xi_{1i}} \delta_{qj} \right\} = \sum_{p=0}^{P_1} u_{pj} \left. \frac{dh_p(\xi_1)}{d\xi_1} \right|_{\xi_{1i}}$$

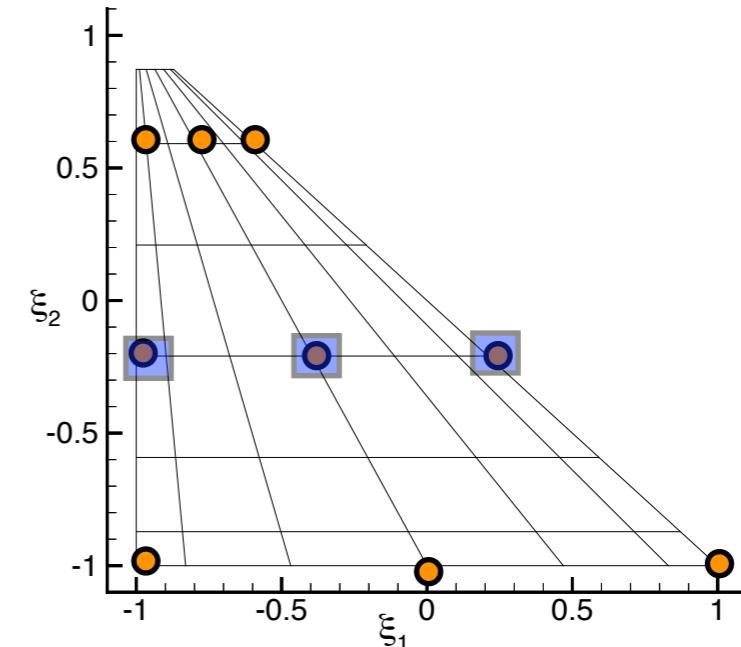
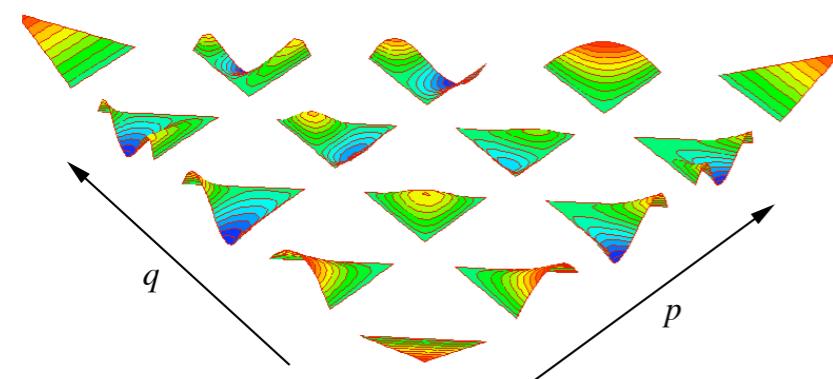
$$\frac{\partial u^\delta}{\partial \xi_2}(\xi_{1i}, \xi_{2j}) = \sum_{q=0}^{P_2} u_{iq} \left. \frac{dh_q(\xi_2)}{d\xi_2} \right|_{\xi_{2j}}.$$

Course Notes: Section 3.1.2.1

Numerical Differentiation: Standard Tri

StdRegions::StdTriExp::PhysDeriv()

$$u^\delta(\xi_1, \xi_2) = \sum_{p,q}^{P_1, P_2} \hat{u}_{pq} \phi_{pq}(\eta_1, \eta_2) \quad \leftrightarrow \quad \sum_{p=0}^{P_1} \sum_{q=0}^{P_2} u_{pq} h_p(\eta_1) h_q(\eta_2), \quad u_{pq} = u^\delta(\eta_{1p}, \eta_{2q}),$$



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

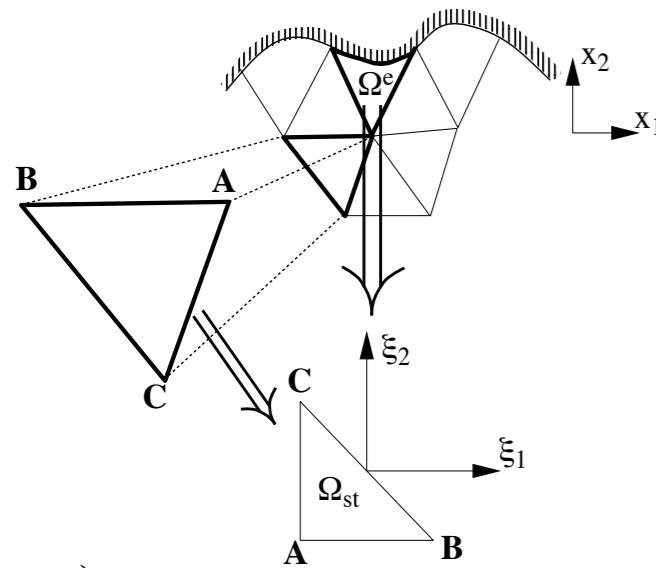
Apply chain rule:

$$\begin{pmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \end{pmatrix} = \begin{pmatrix} \frac{2}{(1 - \eta_2)} \frac{\partial}{\partial \eta_1} \\ 2 \frac{(1 + \eta_1)}{(1 - \eta_2)} \frac{\partial}{\partial \eta_1} + \frac{\partial}{\partial \eta_2} \end{pmatrix}$$

Course Notes: Section 3.1.2.1

Numerical Differentiation: Local Regions

LocalRegions::QuadExp/TriExp::PhysDeriv()



$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} \frac{\partial}{\partial \xi_1} & + & \frac{\partial \xi_2}{\partial x_1} \frac{\partial}{\partial \xi_2} \\ \frac{\partial \xi_1}{\partial x_2} \frac{\partial}{\partial \xi_1} & + & \frac{\partial \xi_2}{\partial x_2} \frac{\partial}{\partial \xi_2} \end{bmatrix} \quad \begin{aligned} &\chi_1(\xi_1, \xi_2) \\ &\chi_2(\xi_1, \xi_2) \end{aligned}$$

a)

$$\begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} \begin{bmatrix} d\xi_1 \\ d\xi_2 \end{bmatrix}$$

$$dx_1(\xi_1, \xi_2) = \frac{\partial x_1}{\partial \xi_1} d\xi_1 + \frac{\partial x_1}{\partial \xi_2} d\xi_2$$

$$\begin{bmatrix} d\xi_1 \\ d\xi_2 \end{bmatrix} = \frac{1}{J} \begin{bmatrix} \frac{\partial x_2}{\partial \xi_2} & -\frac{\partial x_1}{\partial \xi_2} \\ -\frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_1} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix},$$

$$d\xi_1(x_1, x_2) = \frac{\partial \xi_1}{\partial x_1} dx_1 + \frac{\partial \xi_1}{\partial x_2} dx_2$$

$$\begin{bmatrix} d\xi_1 \\ d\xi_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix}.$$

$$\underline{\frac{\partial \xi_1}{\partial x_1} = \frac{1}{J} \frac{\partial x_2}{\partial \xi_2}}, \quad \underline{\frac{\partial \xi_1}{\partial x_2} = -\frac{1}{J} \frac{\partial x_1}{\partial \xi_2}},$$

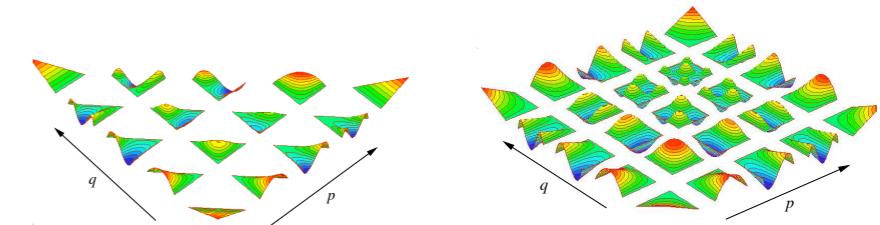
$$\underline{\frac{\partial \xi_2}{\partial x_1} = -\frac{1}{J} \frac{\partial x_2}{\partial \xi_1}}, \quad \underline{\frac{\partial \xi_2}{\partial x_2} = \frac{1}{J} \frac{\partial x_1}{\partial \xi_1}}.$$

Elemental forward transform

LocalRegions::QuadExp/TriExp::FwdTrans()

Expansion:

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



Approximation:

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

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

Method of weighted residual:

$$(a, b) = \int_{\Omega^e} ab d\mathbf{x}$$

$$(v, \sum_{p,q} \hat{u}_{pq} \phi_{pq}) - (v, u) = (v, \cancel{R(u)}),$$

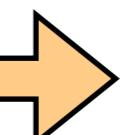
$$(v, \sum_{p,q} \hat{u}_{pq} \phi_{pq}) = (v, u).$$

Course Notes: Sections I.I & 3.I.5.3

Elemental forward transform

LocalRegions::QuadExp/TriExp::FwdTrans()

$$(v, \sum_{p,q} \hat{u}_{pq} \phi_{pq}) = (v, u).$$

Galerkin weight: $v(\xi_1, \xi_2) = \phi_{rs}(\xi_1, \xi_2).$  $(\phi_{rs}, \sum_{p,q} \hat{u}_{pq} \phi_{pq}) = (\phi_{rs}, u)$

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

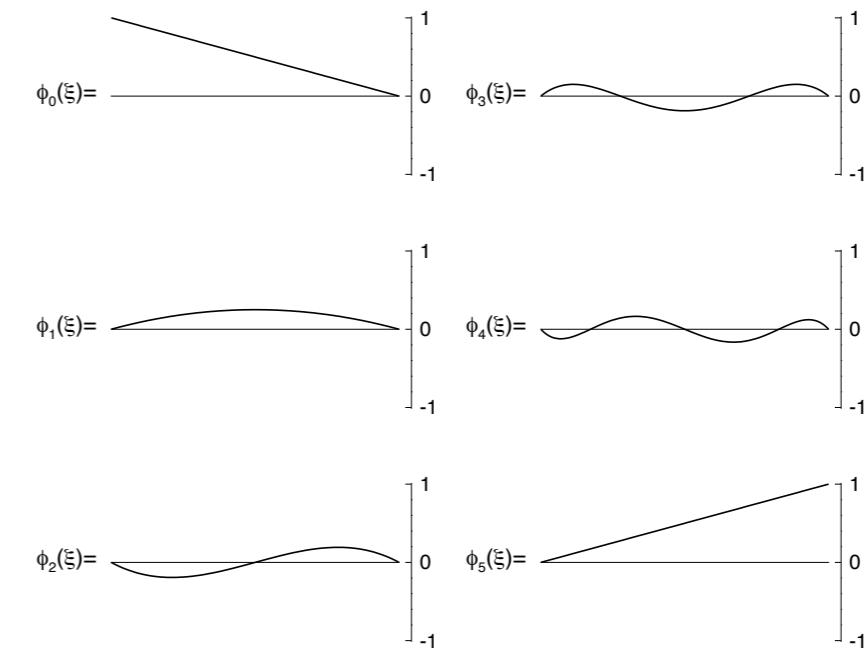
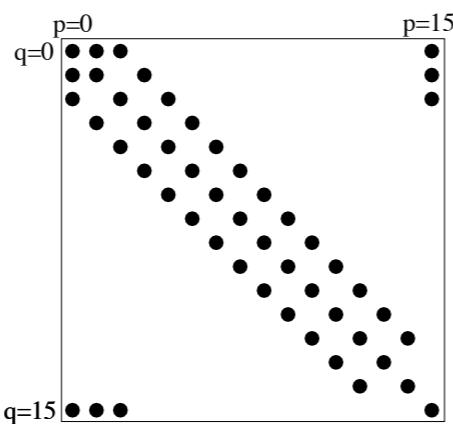
$$\mathbf{M}[i][j] = \int_{\Omega^e} \phi_{i(r,s)}(x_1, x_2) \phi_{j(p,q)}(x_1, x_2) dx_1 dx_2 \quad \hat{\mathbf{u}}[i] = \hat{u}_{i(p,q)}$$
$$\mathbf{f}[i] = \int_{\Omega^e} \phi_{i(p,q)} u(x_1, x_2) dx_1 dx_2$$

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

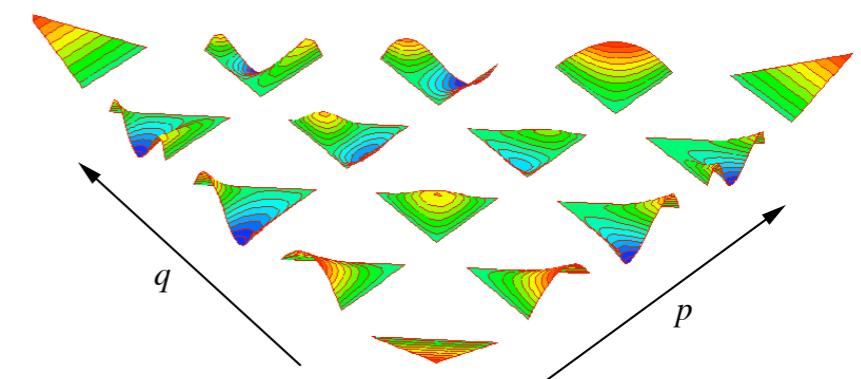
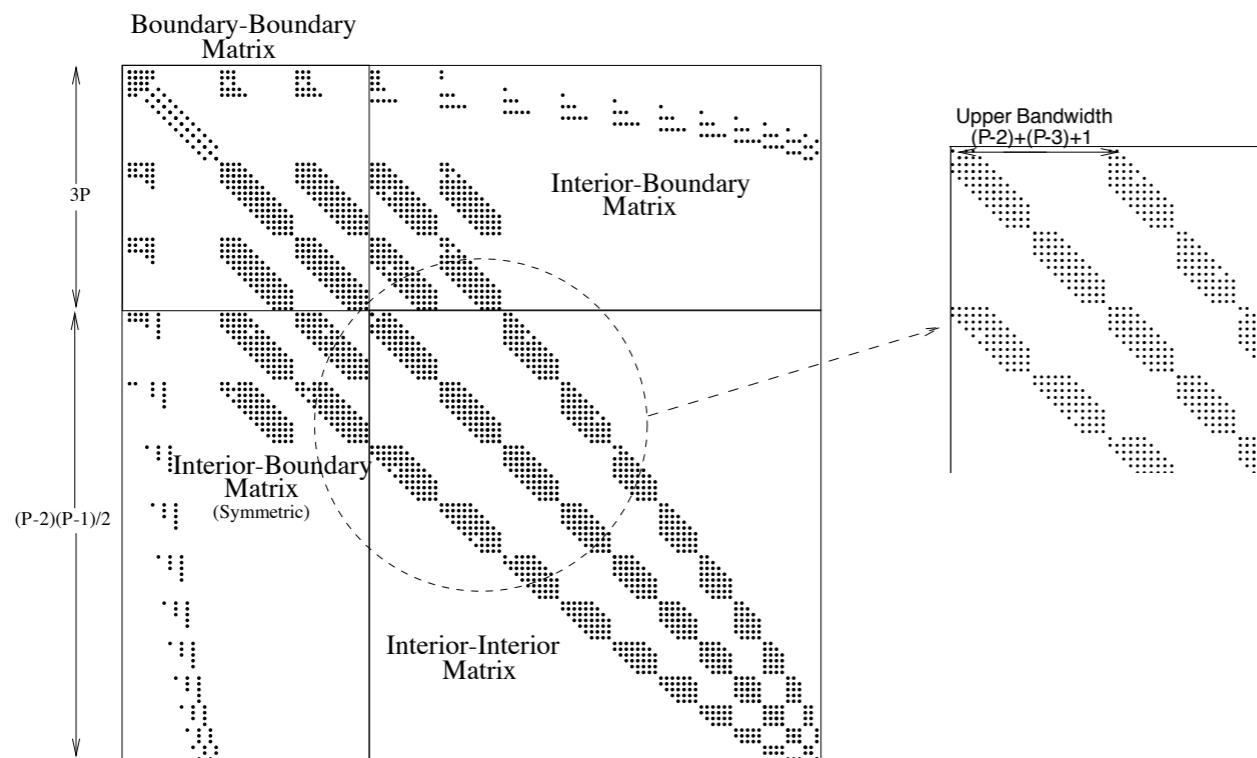
Course Notes: Section 3.1.5.3

Elemental Mass matrix structure

Segment expansion $P = 15$

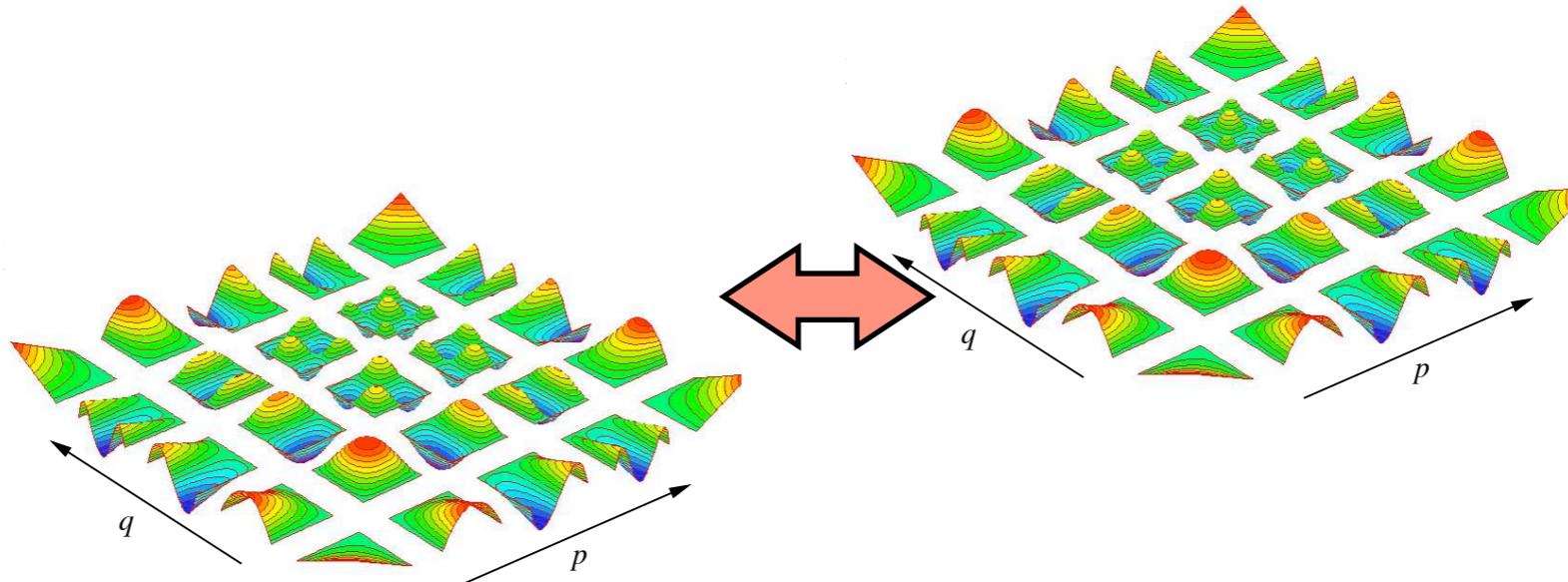


Triangular expansion $P=14$

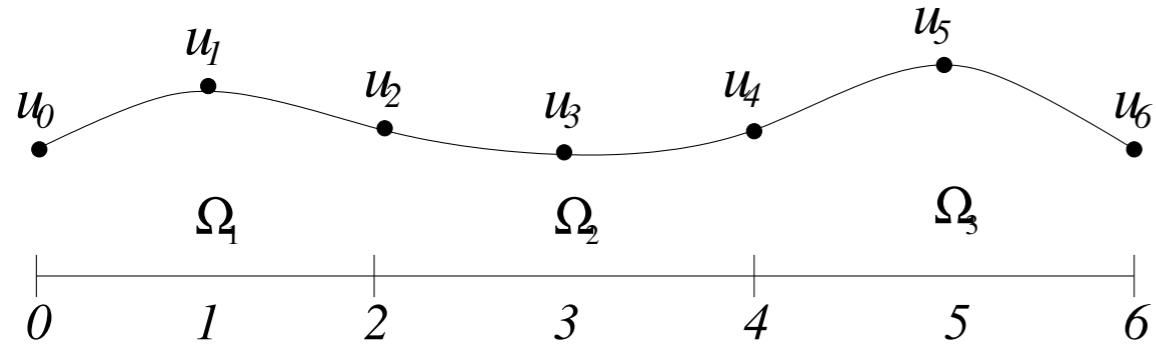


Global Assembly

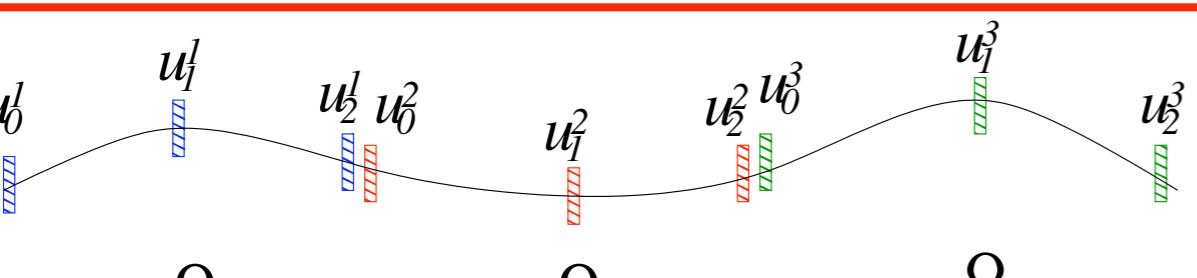
- Classical continuous Galerkin FEM
- 2nd order PDE: $\nabla^2 u - \lambda u = -f$
 - C_0 continuity sufficient.
 - Use boundary interior decomposition to make continuous expansion.



Global Assembly



$$u(\mathbf{x}) = \sum_{i=0}^6 u_i \Phi_i(\mathbf{x}) \quad \text{Global Numbering}$$



$$u(\mathbf{x}) = \sum_{e=0}^2 \sum_{i=2}^6 u_i^e \phi_e(\xi)$$

Local Numbering

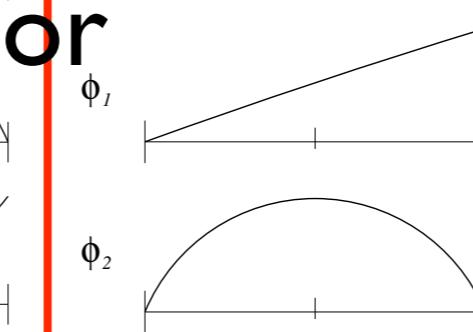
$$u(\mathbf{x}) = \sum_{e=0}^2 \sum_{i=2}^6 u_i^e \phi_e(\xi)$$

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

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

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

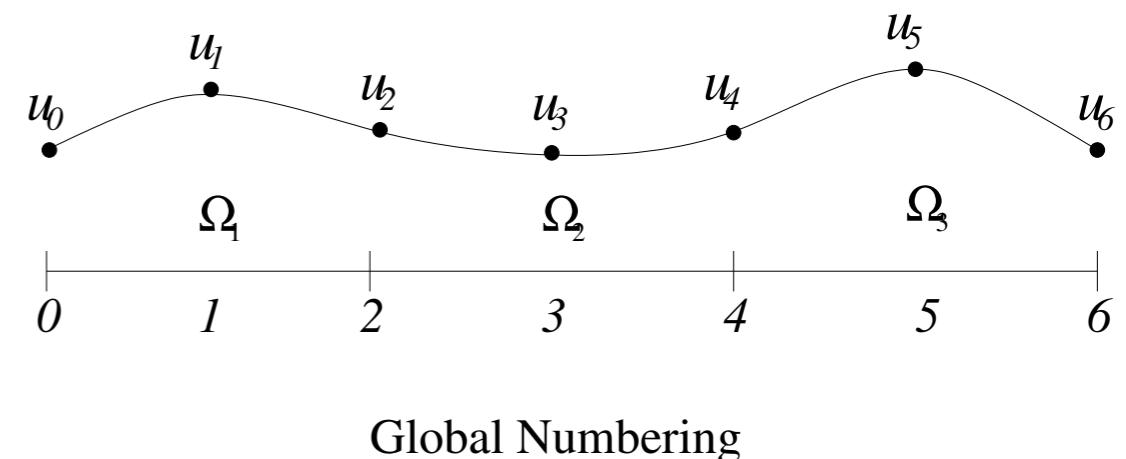
$$\mathcal{A}^T \mathcal{A} \neq \mathbf{I}$$



Course Notes: Sections I.3.I.4 & 3.2.I

Global Assembly

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



$$\text{map}[1][p] = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}, \quad \text{map}[2][p] = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \quad \text{map}[3][p] = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

```

for e = 1 : N_el
    for p = 0 : P^e
        for q = 0 : P^e
            for e = 1 : N_el
                for p = 0 : P^e
                    for q = 0 : P^e
                         $\tilde{\mathbf{u}}^e[p] = \tilde{\mathbf{u}}_g[\text{map}[e][p]]$ 
                         $\mathbf{M}_g[\text{map}[e][p]][\text{map}[e][q]] =$ 
                        end
                         $\mathbf{M}_g[\text{map}[e][p]][\text{map}[e][q]] + \mathbf{M}^e[p][q]$ 
                    end
                end
            end
        end
    end
end

```

Nektar++ code

