2.29 Numerical Fluid Mechanics
Spring 2015 – **Lecture 23**

REVIEW Lecture 22:

- Grid Generation
  - Basic concepts and structured grids, Cont'd
    - General coordinate transformation
    - Differential equation methods
    - Conformal mapping methods
  - Unstructured grid generation
    - Delaunay Triangulation
    - Advancing Front method
- Finite Element Methods

$$\tilde{u}(x) = \sum_{i=1}^{n} a_i \, \phi_i(x) \quad \Rightarrow \quad L\left(\tilde{u}(x)\right) - f(x) = R(x) \neq 0$$

  - Introduction
  - Method of Weighted Residuals:

$$\int_t \int_V R(\mathbf{x}) \, w_i(\mathbf{x}) \, d\mathbf{x} \, dt = 0, \quad i = 1, 2, \ldots, n$$

    - Galerkin, Subdomain and Collocation
  - General Approach to Finite Elements:
    - Steps in setting-up and solving the discrete FE system
    - Galerkin Examples in 1D and 2D

TODAY (Lecture 23): Intro. to Finite Elements, Cont'd

- Finite Element Methods
  - Introduction
  - Method of Weighted Residuals: Galerkin, Subdomain and Collocation
  - General Approach to Finite Elements:
    - Steps in setting-up and solving the discrete FE system
    - Galerkin Examples in 1D and 2D
  - Computational Galerkin Methods for PDE: general case
    - Variations of MWR: summary
    - Finite Elements and their basis functions on local coordinates (1D and 2D)
    - Isoparametric finite elements and basis functions on local coordinates (1D, 2D, triangular)
  - High-Order: Motivation
  - Continuous and Discontinuous Galerkin FE methods:
    - CG vs. DG
    - Hybridizable Discontinuous Galerkin (HDG): Main idea and example
  - DG: Worked simple example
- Finite Volume on Complex geometries

References and Reading Assignments Finite Element Methods

- Chapters 31 on "Finite Elements" of "Chapra and Canale, Numerical Methods for Engineers, 2006."
- Lapidus and Pinder, 1982: Numerical solutions of PDEs in Science and Engineering.
- Chapter 5 on "Weighted Residuals Methods" of Fletcher, Computational Techniques for Fluid Dynamics. Springer, 2003.

- Some Refs on Finite Elements only:
  - Hesthaven J.S. and T. Warburton. Nodal discontinuous Galerkin methods, vol. 54 of Texts in Applied Mathematics. Springer, New York, 2008. Algorithms, analysis, and applications
  - Mathematical aspects of discontinuous Galerkin methods (Di Pietro and Ern, 2012)
  - Theory and Practice of Finite Elements (Ern and Guermond, 2004)

General Approach to Finite Elements

1. Discretization: divide domain into "finite elements"
   - Define nodes (vertex of elements) and nodal lines/planes
2. Set-up Element equations
   i. Choose appropriate basis functions $\phi_i(x)$: $\tilde{u}(x) = \sum_{i=1}^{n} a_i \, \phi_i(x)$
      - 1D Example with Lagrange's polynomials: Interpolating functions $N_i(x)$

$$\tilde{u} = a_0 + a_1 \, x = u_1 N_1(x) + u_2 N_2(x) \quad \text{where } \mathrm{N}_1(x) = \frac{x_2 - x}{x_2 - x_1} \text{ and } \mathrm{N}_2(x) = \frac{x - x_1}{x_2 - x_1}$$

      - With this choice, we obtain for example the 2nd order CDS and Trapezoidal rule: $\frac{d\,\tilde{u}}{dx} = a_1 = \frac{u_2 - u_1}{x_2 - x_1}$ and $\int_{x_1}^{x_2} \tilde{u}\, dx = \frac{u_1 + u_2}{2}(x_2 - x_1)$

   ii. Evaluate coefficients of these basis functions by approximating the equations to be solved in an optimal way
      - This develops the equations governing the element's dynamics
      - Two main approaches: Method of Weighted Residuals (MWR) or Variational Approach

      $\Rightarrow$ Result: relationships between the unknown coefficients a_i so as to satisfy the PDE in an optimal approximate way

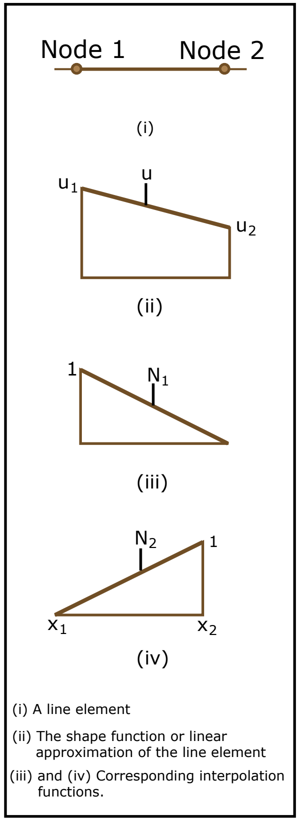


Image by MIT OpenCourseWare.

General Approach to Finite Elements, Cont'd

2. Set-up Element equations, Cont'd
   – Mathematically, combining i. and ii. gives the element equations: a set of (often linear) algebraic equations for a given element *e:*

$$\mathbf{K}_e \, \mathbf{u}_e = \mathbf{f}_e$$

   where $\mathbf{K}_\mathrm{e}$ is the element property matrix (stiffness matrix in solids), $\mathbf{u}_\mathrm{e}$ the vector of unknowns at the nodes and $\mathbf{f}_\mathrm{e}$ the vector of external forcing

3. Assembly:
   – After the individual element equations are derived, they must be assembled: i.e. impose continuity constraints for contiguous elements
   – This leas to: $\mathbf{K}\,\mathbf{u} = \mathbf{f}$

   where $\mathbf{K}$ is the assemblage property or coefficient matrix, $\mathbf{u}$ the vector of unknowns at the nodes and $\mathbf{f}$ the vector of external forcing

4. Boundary Conditions: Modify " $\mathbf{K}\,\mathbf{u} = \mathbf{f}$ " to account for BCs
5. Solution: use LU, banded, iterative, gradient or other methods
6. Post-processing: compute secondary variables, errors, plot, etc

Galerkin's Method: Simple Example

Differential Equation

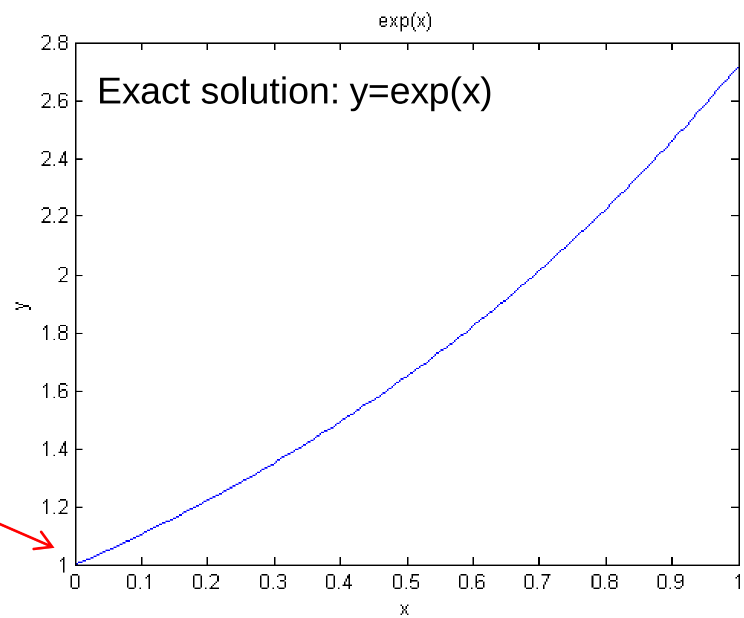
$$\frac{dy}{dx} - y = 0$$

Boundary Conditions

$$y = 1\ , \quad x = 0$$

<u>1. Discretization</u>:

Generic N (here 3) equidistant nodes along x, at x = [0, 0.5,1]



In this simple example, a single element is chosen to cover the whole domain ⇒ the element/mass matrix is the full one (**K**=**K**$_e$)

<u>2. Element equations:</u>

<u>i. Basis (Shape) Functions</u>:
Power Series (Modal basis)

$$\tilde{y} = 1 + \sum_{j=1}^{N} a_j x^j$$

Boundary Condition

Note: this is equivalent to imposing the BC on the full sum

$$\tilde{y} = \sum_{j=0}^{N} a_j x^j$$

$$a_0 = 1$$

Galerkin's Method: Simple Example, Cont'd

ii. Optimal coefficients with MWR: set weighted residuals (remainder) to zero

Remainder: $$R = \frac{d\,\tilde{y}}{dx} - \tilde{y}$$

$$R = -1 + \sum_{j=1}^{N} a_j (j x^{j-1} - x^j)$$

Galerkin ⇒ set remainder orthogonal to each shape function:

Denoting inner products as: $(f, g) = \langle f, g \rangle = \int_0^1 f\, g\, dx$

leads to: $(R, x^{k-1}) = 0, \quad k = 1, \ldots, N$

which then leads to the Algebraic Equations:

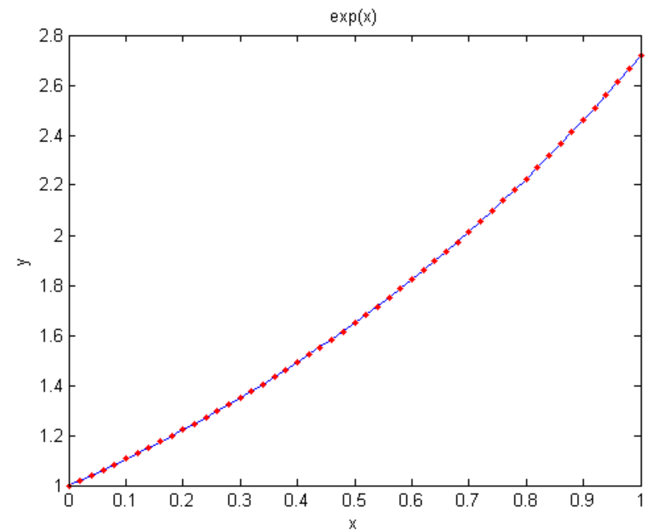
$$\mathbf{Ma} = \mathbf{d} \qquad k = 1, \ldots, N \qquad j = 1, \ldots, N$$

$$d_k = (1, x^{k-1})$$

$$m_{kj} = (j x^{j-1} - x^j, x^{k-1}) = \frac{j}{j+k-1} - \frac{1}{j+k}$$

exp_eq.m

```
N=3;
d=zeros(N,1);
m=zeros(N,N);
for k=1:N
    d(k)=1/k;
    for j=1:N
        m(k,j) = j/(j+k-1)-1/(j+k);
    end
end
a=inv(m)*d;
y=ones(1,n);
for k=1:N
    y=y+a(k)*x.^k
end
```



Galerkin's Method Simple Example, Cont'd

3 - 4. Assembly and boundary conditions:

Already done (element fills whole domain)

5. Solution: For $N = 3$

BCs already set

$$\mathbf{a}^T = [1.0141, 0.4225, 0.2817];$$

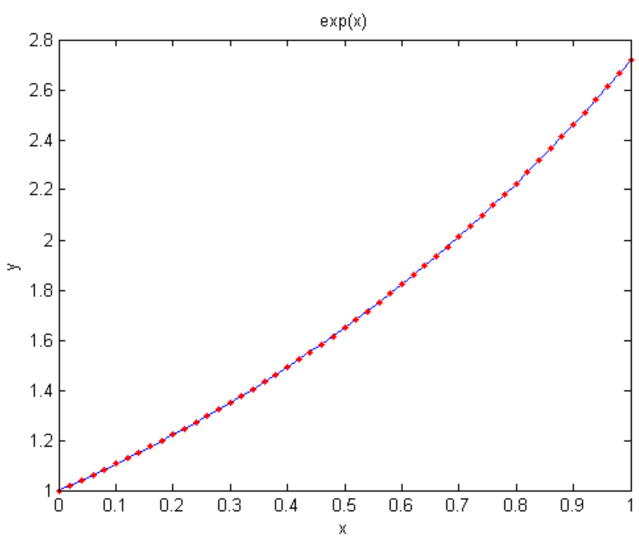
$$\tilde{y} = 1 + 1.0141x + 0.4225x^2 + 0.2817x^3$$

L$_2$ Error:

$$L_2 = ||y - \tilde{y}||_2 = \sqrt{\sum_{\ell=1}^{L} (y(x_\ell) - \tilde{y}(x_\ell))^2}$$

exp_eq.m

```
N=3;
d=zeros(N,1);
m=zeros(N,N);
for k=1:N
    d(k)=1/k;
    for j=1:N
        m(k,j) = j/(j+k-1)-1/(j+k);
    end
end
a=inv(m)*d;
y=ones(1,n);
for k=1:N
    y=y+a(k)*x.^k
end
```



Comparisons with other Weighted Residual Methods

$$\frac{dy}{dx} - y = 0$$

$$\tilde{y} = 1 + \sum_{j=1}^{N} a_j x^j$$

Least Squares

$$\begin{bmatrix} 1/3 & 1/4 & 1/5 \\ 1/4 & 8/15 & 2/3 \\ 1/5 & 2/3 & 33/35 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 2/3 \\ 3/4 \end{bmatrix}$$

Subdomain Method

$$\begin{bmatrix} 5/18 & 8/81 & 11/324 \\ 3/18 & 20/81 & 69/324 \\ 1/18 & 26/81 & 163/324 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Galerkin

$$\begin{bmatrix} 1/2 & 2/3 & 3/4 \\ 1/6 & 5/12 & 11/20 \\ 1/12 & 3/10 & 13/30 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1/2 \\ 1/3 \end{bmatrix}$$

Collocation

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.75 & 0.625 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Comparisons with other Weighted Residual Methods

Comparison of coefficients for approximate solution of $dy/dx - y = 0$

Scheme \ Coefficient	a_1	a_2	a_3
Least squares	1.0131	0.4255	0.2797
Galerkin	1.0141	0.4225	0.2817
Subdomain	1.0156	0.4219	0.2813
Collocation	1.0000	0.4286	0.2857
Taylor series	1.0000	0.5000	0.1667
Optimal $L_{2,d}$	1.0138	0.4264	0.2781

Image by MIT OpenCourseWare.

Comparison of approximate solutions of $dy/dx - y = 0$

x	Least squares	Galerkin	Subdomain	Collocation	Taylor series	Optimal $L_{2,d}$	Exact
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.2	1.2219	1.2220	1.2223	1.2194	1.2213	1.2220	1.2214
0.4	1.4912	1.4913	1.4917	1.4869	1.4907	1.4915	1.4918
0.6	1.8214	1.8214	1.8220	1.8160	1.8160	1.8219	1.8221
0.8	2.2260	2.2259	2.2265	2.2206	2.2053	2.2263	2.2255
1.0	2.7183	2.7183	2.7187	2.7143	2.6667	2.7183	2.7183
$\Vert y_a - y \Vert_{2,d}$	0.00105	0.00103	0.00127	0.0094	0.0512	0.00101	

Image by MIT OpenCourseWare.

Galerkin's Method in 2 Dimensions

Differential Equation

$$L(u) = 0$$

Boundary Conditions

$$S(u) = 0$$

Shape/Test Function Solution (u_o satisifies BC)

$$\tilde{u} = u_0(x,y) + \sum_{j=1}^{N} a_j \phi_j(x,y)$$

Remainder (if L is linear Diff. Eqn.)

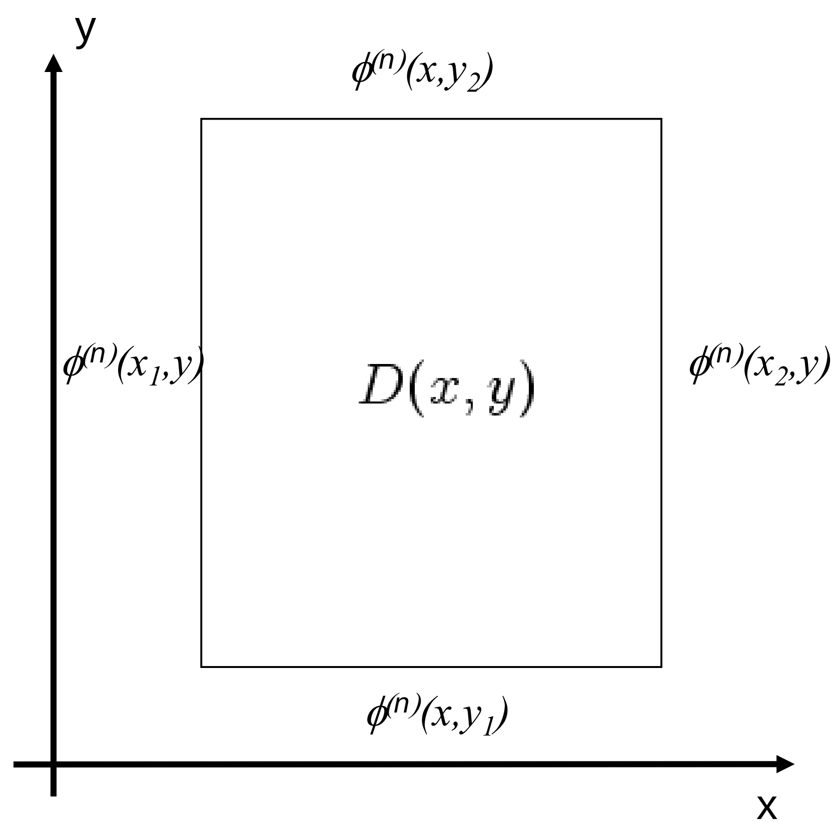
$$R(u_0, a_1, \ldots a_N, x, y) = L(\tilde{u}) = L(u_0) + \sum_{j=1}^{N} a_j L(\phi_j(x,y))$$

Inner Product: $(f,g) = \int\int_D f g \, dx dy$

Galerkin's Method

$$(R, \phi_k) = 0$$

$$\sum_{j=1}^{N} a_j (L(\phi_j), \phi_k) = -(L(u_0), \phi_k)$$



Galerkin's method: 2D Example
Fully-developed Laminar Viscous Flow in Duct

Steady, Very Viscous, fully-developed
Fluid Flow in Duct

$$\cancel{u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial x}} + \frac{1}{\rho}\frac{\partial p}{\partial z} = \nu\left\{\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right\}$$

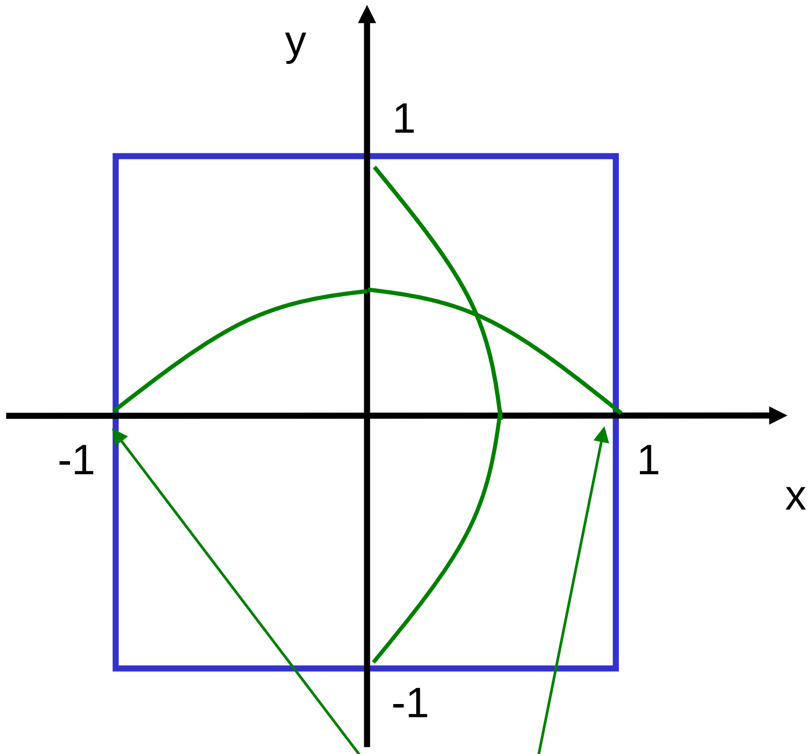
$$\frac{\partial p}{\partial z} = \text{const}$$

Non-dimensionalization, yields a Poisson Equation:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = -1$$

Shape/Test Functions

$$\tilde{w} = \sum_{i=1,3,5\ldots}^{N} \sum_{j=1,3,5\ldots}^{N} a_{ij} \cos i\frac{\pi}{2}x \cos j\frac{\pi}{2}y$$



Shape/Test functions satisfy boundary conditions

4 BCs: No-slip (zero flow) at the walls

Again: in this example, single element fills the whole domain

Galerkin's Method: Viscous Flow in Duct, Cont'd

```
x=[-1:h:1]';
y=[-1:h:1];
n=length(x); m=length(y); w=zeros(n,m);
Nt=5;
for j=1:n
 xx(:,j)=x; yy(j,:)=y;
end
for i=1:2:Nt
    for j=1:2:Nt
      w=w+(8/pi^2)^2*
        (-1)^((i+j)/2-1)/(i*j*(i^2+j^2))
        *cos(i*pi/2*xx).*cos(j*pi/2*yy);
    end
end
```

duct_galerkin.m

Remainder:

$$R = -\left[\sum_{i=1,3,5\ldots}^{N}\sum_{j=1,3,5\ldots}^{N} a_{ij}\cos i\frac{\pi}{2}x\cos j\frac{\pi}{2}y\left\{\left(i\frac{\pi}{2}\right)^2+\left(j\frac{\pi}{2}\right)^2\right\}-1\right]$$

Inner product (set to zero):

$$\left(R, \cos k\frac{\pi}{2}x\cos \ell\frac{\pi}{2}y\right) \ ,\ k,\ \ell = 1,3,5,\ldots$$

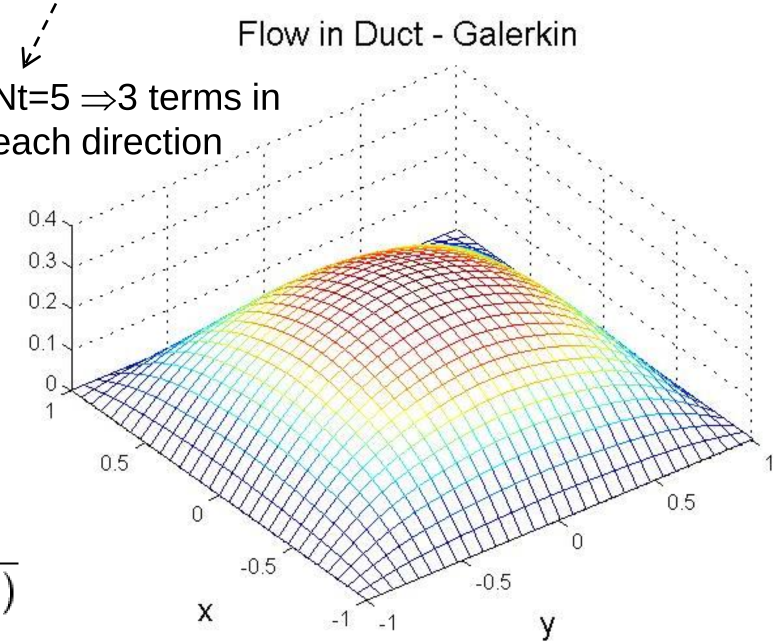
Analytical Integration:

$$a_{ij} = \left(\frac{8}{\pi^2}\right)^2 \frac{(-1)^{(i+j)/2-1}}{ij(i^2+j^2)}$$

Galerkin Solution:

$$\tilde{w} = \left(\frac{8}{\pi^2}\right)^2 \sum_{i=1,3,5\ldots}^{N}\sum_{j=1,3,5\ldots}^{N} \frac{(-1)^{(i+j)/2-1}}{ij(i^2+j^2)}\cos i\frac{\pi}{2}x\cos j\frac{\pi}{2}y$$

Nt=5 ⇒3 terms in each direction



Flow Rate:

$$\begin{aligned}\dot{q} &= \int_{-1}^{1}\int_{-1}^{1}\tilde{w}(x,y)dxdy \\ &= 2\left(\frac{8}{\pi^2}\right)^3 \sum_{i=1,3,5\ldots}^{N}\sum_{j=1,3,5\ldots}^{N}\frac{1}{i^2j^2(i^2+j^2)}\end{aligned}$$

Computational Galerkin Methods: Some General Notes

Differential Equation: $L(u) = 0$

Residuals

- PDE: $L(\tilde{u}) = R$
- ICs: $I(\tilde{u}) = R_I$
- BCs: $S(\tilde{u}) = R_B$

Boundary problem

- PDE satisfied exactly
- Boundary Element Method
  - Panel Method
  - Spectral Methods

$$R = 0$$

Inner problem

- Boundary conditions satisfied exactly
- Finite Element Method
- Spectral Methods

$$R_B = 0$$

Mixed Problem

- Finite Element Method

$$R, R_B \neq 0$$

Global Shape/Test Function:

$$\tilde{u}(\mathbf{x}, t) = u_0(\mathbf{x}, t) + \sum_{j=1}^{N} a_j \phi(\mathbf{x}, t)$$

Time Marching + spatial discretization (separable):

$$\tilde{u}(\mathbf{x}, t) = u_0(\mathbf{x}, t) + \sum_{j=1}^{N} a_j(t) \phi(\mathbf{x})$$

Weighted Residuals

$$(R, w_k(\mathbf{x})) = 0, k = 1, \dots N$$

$$\lim_{N \to \infty} ||\tilde{u} - u||_2 = 0$$

Different forms of the Methods of Weighted Residuals: Summary

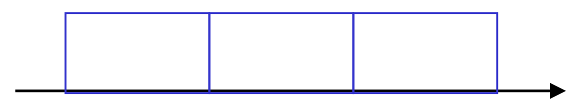
Inner Product $(L(u), w) = 0$

Discrete Form $(f, g) = \sum_{i=1}^{N} f_i g_i$

$k = 1, 2, \ldots, n$

Subdomain Method:

$$w_k = \begin{cases} 1 & \text{in } D_k \\ 0 & \text{outside } D_k \end{cases}$$



$$\int_t \int_{D_k} R(\mathbf{x})\, d\mathbf{x}\, dt = 0$$

Collocation Method:

$$w_k(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_k)$$

$$R(\mathbf{x}_k) = 0$$

$$R(\mathbf{x}_k) = 0$$

Least Squares Method:

$$\left.\begin{aligned} &(\mathrm{R,R}) = \text{minimum} \\ &\frac{\partial}{\partial a_i}\left(\int_t \int_{V_k} R(\mathbf{x})\, R(\mathbf{x}) d\mathbf{x}\, dt \right) = 0 \end{aligned}\right\} \Rightarrow \quad w_k = \frac{\partial R}{\partial a_k}$$

In the least-square method, the coefficients are adjusted so as to minimize the integral of the residuals. It amounts to a continuous form of regression.

Method of Moments: $w_k(\mathbf{x}) = x^k \;, \quad k = 0, 1, \ldots, N$

Galerkin: $w_k(\mathbf{x}) = \phi_k(\mathbf{x})$

In Galerkin, weight functions are basis functions: they sum to one at any position in the element. In many cases, Galerkin's method yields the same result as variational methods

How to obtain solution for Nodal Unknowns? Modal ϕ_k vs. Nodal (Interpolating) N_j Basis Fcts.

- $\tilde{u}(x,y) = \sum_{j=1}^{N} \bar{u}_j \; N_j(x,y)$

- $\tilde{u}(x,y) = \sum_{k=1}^{N} a_k \; \phi_k(x,y)$

$$\Rightarrow \quad \bar{u}_j = \sum_{k=1}^{N} a_k \; \phi_k(x_j, y_j)$$

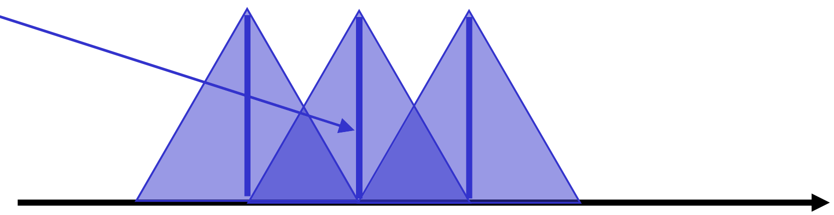
$$\Rightarrow \quad \bar{\mathbf{u}} = \mathbf{\Phi}\,\mathbf{a} \quad \Rightarrow \quad \mathbf{a} = \mathbf{\Phi}^{-1}\,\bar{\mathbf{u}}$$

- $\tilde{u}(x,y) = \sum_{k=1}^{N} \left(\sum_{j=1}^{N} \left(\mathbf{\Phi}^{-1}\right)_{kj} \bar{u}_j \right) \phi_k(x,y)$

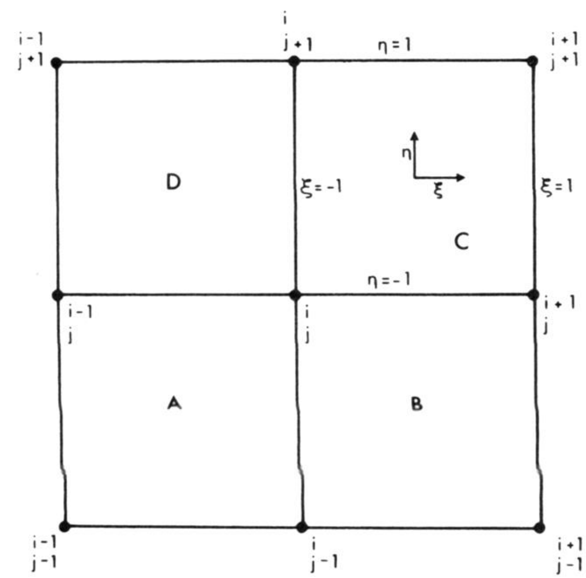
$$= \sum_{j=1}^{N} \bar{u}_j \left(\sum_{k=1}^{N} \left(\mathbf{\Phi}^{-1}\right)_{kj} \phi_k(x,y) \right)$$

$$\Rightarrow \quad N_j(x,y) = \sum_{k=1}^{N} \left(\mathbf{\Phi}^{-1}\right)_{kj} \phi_k(x,y)$$

1 Dimension



2 Dimensions



Finite Elements
1-dimensional Elements

Trial Function Solution

$$\tilde{u} = \sum_{j=1}^{N} N_j(x)\overline{u}_j$$

Interpolation (Nodal) Functions

$$N_2 = \frac{x - x_1}{x_2 - x_1}$$

$$N_2 = \frac{x - x_3}{x_2 - x_3}$$

$$N_3 = \frac{x - x_2}{x_3 - x_2}$$

$$N_3 = \frac{x - x_4}{x_3 - x_4}$$

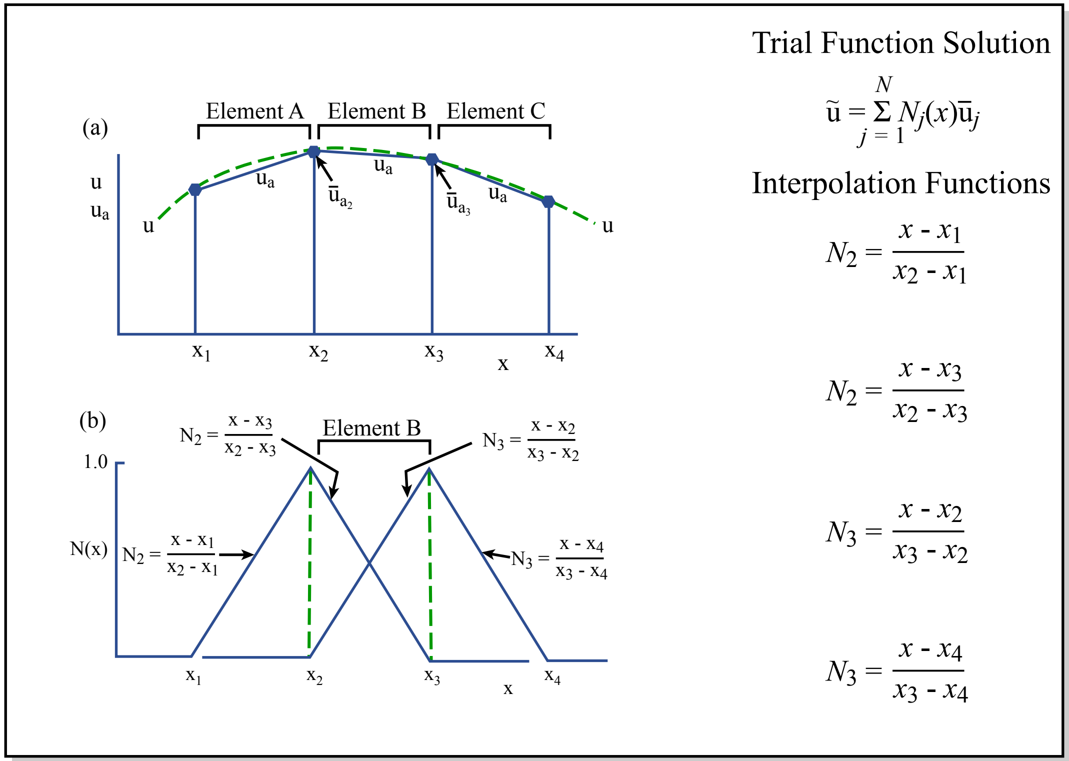


Image by MIT OpenCourseWare.

Two functions per element

Finite Elements
1-dimensional Elements

Quadratic Interpolation (Nodal) Functions

$$N_3 = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

$$N_3 = \frac{(x - x_4)(x - x_5)}{(x_3 - x_4)(x_3 - x_5)}$$

$$N_4 = \frac{(x - x_3)(x - x_5)}{(x_4 - x_3)(x_4 - x_5)}$$

$$N_2 = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}$$

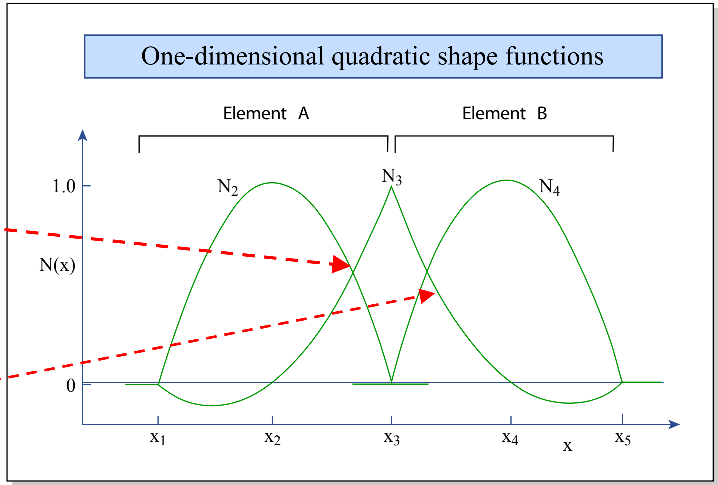


Image by MIT OpenCourseWare.

Three functions per element

Complex Boundaries
Isoparametric Elements

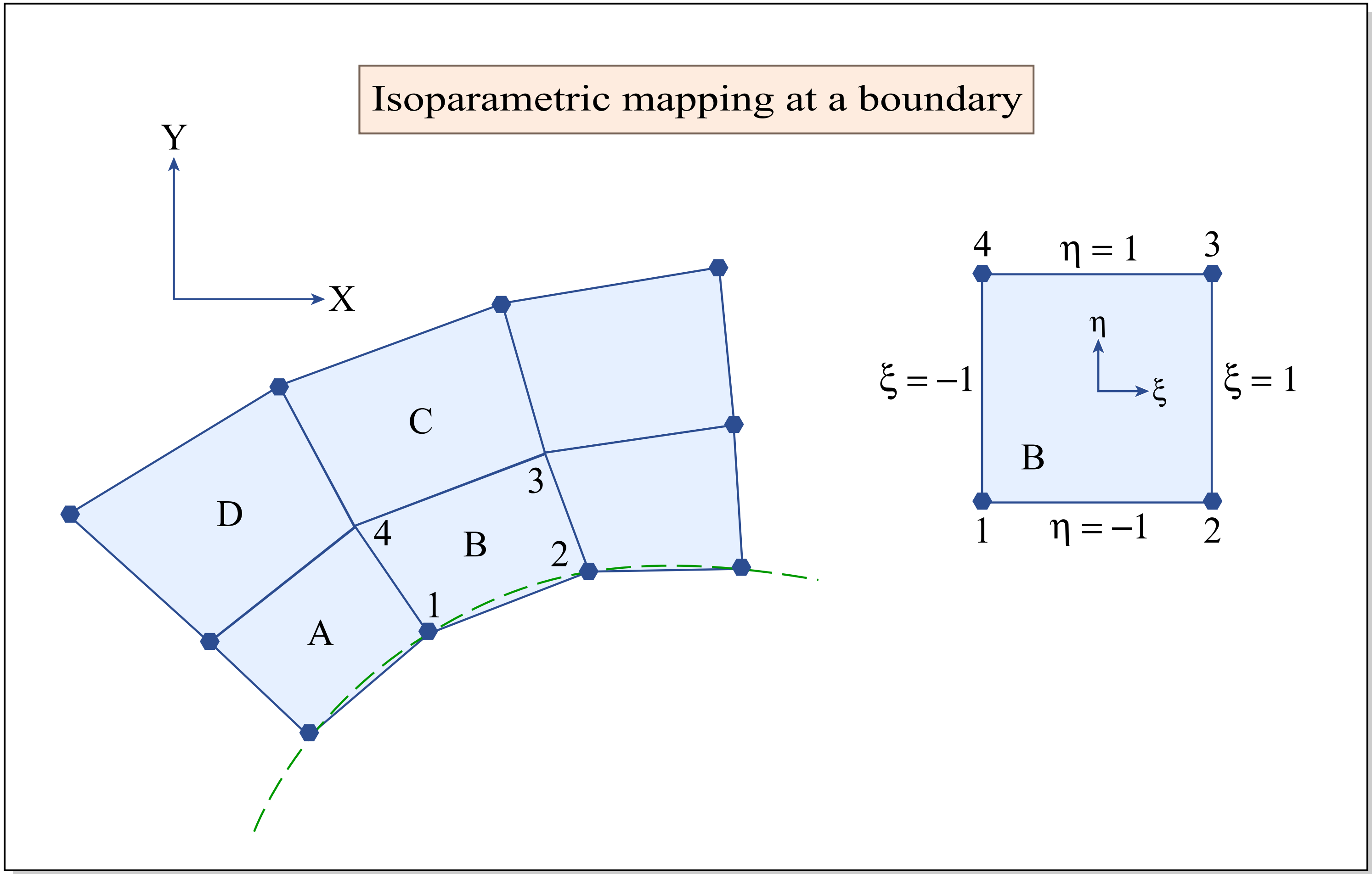


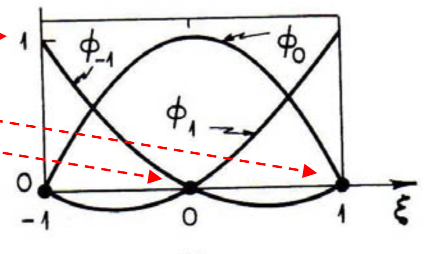
Image by MIT OpenCourseWare.

Nodal Basis Functions in the Local Coordinate System

(2.2.29) $$\phi(\xi) = a + b\xi + c\xi^2.$$

By introducing the requirement that $\phi_{-1}|_{-1} = 1$ and $\phi_{-1}|_0 = \phi_{-1}|_1 = 0$, we obtain the matrix equation

(2.2.30) $$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix},$$



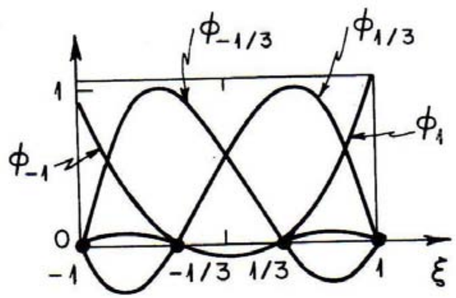


Figure 2.7. Quadratic (*a*) and cubic (*b*) basis functions defined in local ξ coordinate system.

TABLE 2.4. Linear, Quadratic, and Cubic Basis Functions Defined in the Dimensionless ξ Coordinate System

Degree	Function	Form $(-1 \leq \xi \leq 1)$
Linear	$\phi_{-1}(\xi)$	$\frac{1}{2}(1-\xi)$
	$\phi_1(\xi)$	$\frac{1}{2}(1+\xi)$
Quadratic	$\phi_{-1}(\xi)$	$-\frac{1}{2}\xi(1-\xi)$
	$\phi_0(\xi)$	$1-\xi^2$
	$\phi_1(\xi)$	$\frac{1}{2}\xi(1+\xi)$
Cubic	$\phi_{-1}(\xi)$	$\frac{1}{16}(-9\xi^3+9\xi^2+\xi-1)$ or $\frac{1}{16}(1-\xi)(9\xi^2-1)$
	$\phi_{-1/3}(\xi)$	$\frac{9}{16}(3\xi^3-\xi^2-3\xi+1)$ or $\frac{9}{16}(3\xi-1)(\xi^2-1)$
	$\phi_{1/3}(\xi)$	$\frac{9}{16}(-3\xi^3-\xi^2+3\xi+1)$ or $-\frac{9}{16}(3\xi+1)(\xi^2-1)$
	$\phi_1(\xi)$	$\frac{1}{16}(9\xi^3+9\xi^2-\xi-1)$ or $\frac{1}{16}(1+\xi)(9\xi^2-1)$

© Wiley-Interscience. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
Source: pp. 63-65 in Lapidus, L., and G. Pinder. *Numerical Solution of Partial Differential Equations in Science and Engineering.* 1st ed. Wiley-Interscience, 1982. [Preview with Google Books].

Finite Elements
2-dimensional Elements

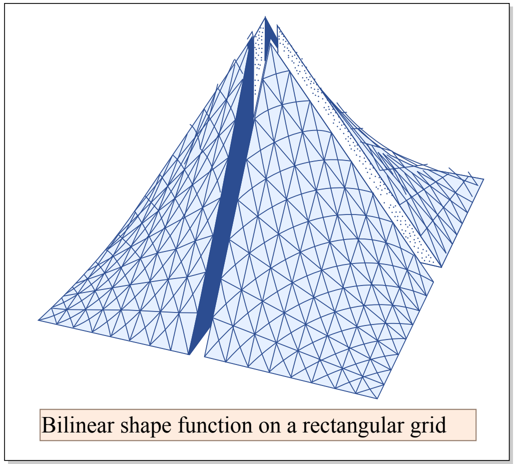


Image by MIT OpenCourseWare.

$$\tilde{u} = \sum_{i=1}^{N}\sum_{j=1}^{N} N_{ij}(x)\overline{u}_{ij}$$

$$\tilde{u} = \sum_{\ell=1}^{4} N_\ell(\xi,\eta)\overline{u}_\ell$$

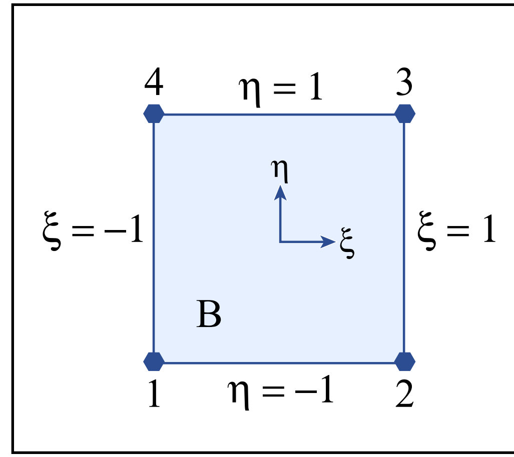


Image by MIT OpenCourseWare.

Linear Interpolation (Nodal) Functions

$$
\begin{aligned}
N_1 &= 0.25(1-\xi)(1-\eta)\\
N_2 &= 0.25(1+\xi)(1-\eta)\\
N_3 &= 0.25(1-\xi)(1+\eta)\\
N_4 &= 0.25(1+\xi)(1+\eta)
\end{aligned}
$$

$$N_\ell = 0.25(1+\xi_\ell\xi)(1+\eta_\ell\eta)$$

Quadratic Interpolation (Nodal) Functions

$$\prod_{r\neq i}\frac{(\xi-\xi_r)(\eta-\eta_r)}{(\xi_i-\xi_r)(\eta_i-\eta_r)}$$

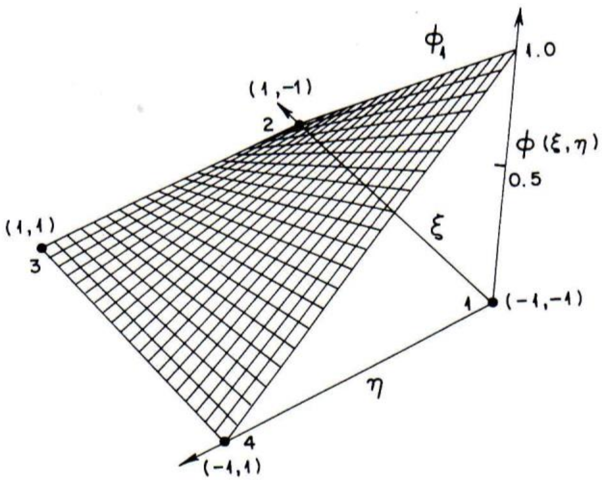
corner nodes $\quad N_i = 0.25\xi_i\xi(1+\xi_i\xi)\eta_i\eta(1+\eta_i\eta)$

side nodes $\quad N_i = 0.5(1-\xi^2)\eta_i\eta(1+\eta_i\eta)\ ,\ \xi_i = 0$

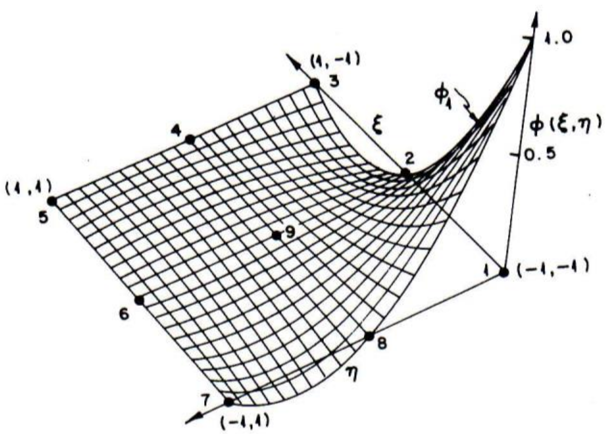
$\quad N_i = 0.5(1-\eta^2)\xi_i\xi(1+\xi_i\xi)\ ,\ \eta_i = 0$

interior node $\quad N_i = (1-\xi^2)(1-\eta^2)$

Finite Elements in 2D: Nodal Basis Functions in the Local Coordinate System

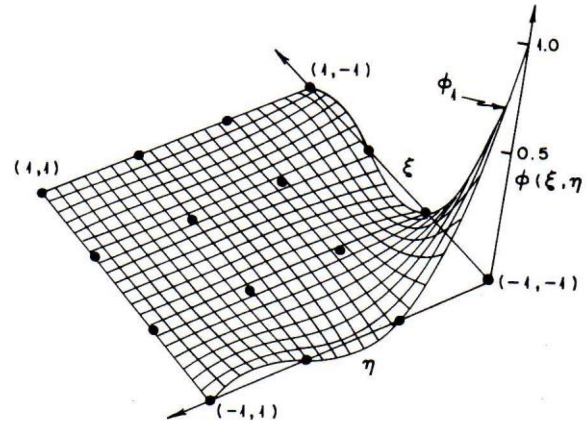


(a)



(b)

Figure 2.11. (*a*) Two-dimensional basis function that is linear along each side. (*b*) Two-dimensional Lagrangian basis function that is quadratic along each side. Note the occurrence of a central node where the basis function must be zero.



(c)

Figure 2.11. (*Continued*)

Figure 2.11. (*c*) Two-dimensional Lagrangian basis function that is cubic along each side. Note the occurrence of four interior nodes where the basis function is defined to be zero.

© Wiley-Interscience. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/fairuse.

Finite Elements in 2D: Nodal Basis Functions in the Local Coordinate System

© Wiley-Interscience. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
Source: Table 2.7A in Lapidus, L., and G. Pinder. *Numerical Solution of Partial Differential Equations in Science and Engineering*. 1st ed. Wiley-Interscience, 1982. [Preview with Google Books].

Two-Dimensional Finite Elements
Example: Flow in Duct, Bilinear Basis functions

Finite Element Solution

$$\tilde{w} = \sum_{j=1}^{N} \overline{w}_j N_j(x,y)$$

$$N_j = 0.25(1+\xi_j\xi)(1+\eta_j\eta)$$

$$\left(\frac{\partial^2 \tilde{w}}{\partial x^2}, N_k\right) + \left(\frac{\partial^2 \tilde{w}}{\partial y^2}, N_k\right) = (-1, N_k)$$

Integration by Parts

$$\left(\frac{\partial^2 w}{\partial x^2}, N_k\right) \equiv \int_{-1}^{1} \frac{\partial^2 w}{\partial x^2} N_k \, dx = \left[\frac{\partial w}{\partial x} N_k\right]_{-1}^{1} - \int_{-1}^{1} \frac{\partial w}{\partial x}\frac{dN_k}{dx} \, dx$$

$$\left(\frac{\partial^2 \tilde{w}}{\partial x^2}, N_k\right) = -\left(\frac{\partial \tilde{w}}{\partial x}, \frac{\partial N_k}{\partial x}\right) \qquad \text{(for center nodes)}$$

Algebraic Equations for center nodes

$$-\sum_{j=1}^{N}\left(\int_{-1}^{1}\int_{-1}^{1} \frac{\partial N_j}{\partial x}\frac{\partial N_k}{\partial x} + \frac{\partial N_j}{\partial y}\frac{\partial N_k}{\partial y} \, dx dy\right)\overline{w}_j = -\int_{-1}^{1}\int_{-1}^{1} 1 N_k dx dy, k = 1, \ldots N$$

Finite Elements
2-dimensional Triangular Elements

Triangular Coordinates

Linear Polynomial Modal Basis Functions:

$$u(x,y) = a_0 + a_{1,1}\, x + a_{1,2}\, y$$

$$\left.\begin{aligned} u_1(x,y) &= a_0 + a_{1,1}\, x_1 + a_{1,2}\, y_1 \\ u_2(x,y) &= a_0 + a_{1,1}\, x_2 + a_{1,2}\, y_2 \\ u_3(x,y) &= a_0 + a_{1,1}\, x_3 + a_{1,2}\, y_3 \end{aligned}\right\} \Rightarrow \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_{1,1} \\ a_{1,2} \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

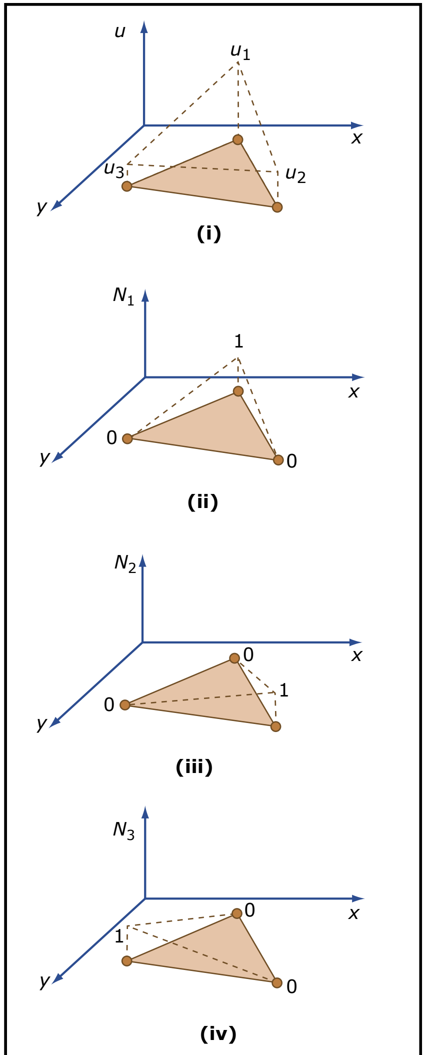
Nodal Basis (Interpolating) Functions:

$$u(x,y) = u_1\, N_1(x,y) + u_2\, N_2(x,y) + u_3\, N_3(x,y)$$

$$N_1(x,y) = \frac{1}{2A_T}\left[(x_2 y_3 - x_3 y_2) + (y_2 - y_3)\, x + (x_3 - x_2)\, y\right]$$

$$N_2(x,y) = \frac{1}{2A_T}\left[(x_3 y_1 - x_1 y_3) + (y_3 - y_1)\, x + (x_1 - x_3)\, y\right]$$

$$N_3(x,y) = \frac{1}{2A_T}\left[(x_1 y_2 - x_2 y_1) + (y_1 - y_2)\, x + (x_2 - x_1)\, y\right]$$



A linear approximation function (i) and its interpolation functions (ii)-(iv).

Image by MIT OpenCourseWare.

Higher-order: Increased accuracy for same Efficiency

Equation: $\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0$

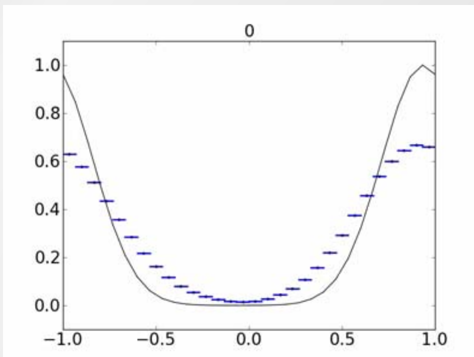
Low order

High order

Equal degrees of freedom (Approx. equal computational. efficiency)

Polynomial degree = 0
30 elements

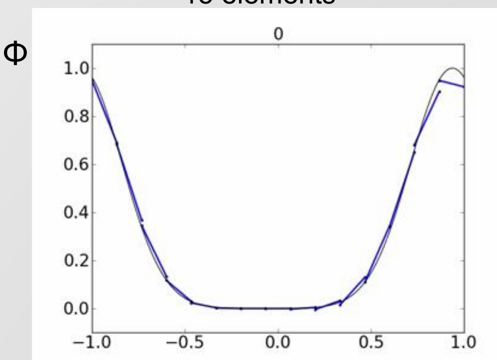
Φ



x

Polynomial degree = 1
15 elements

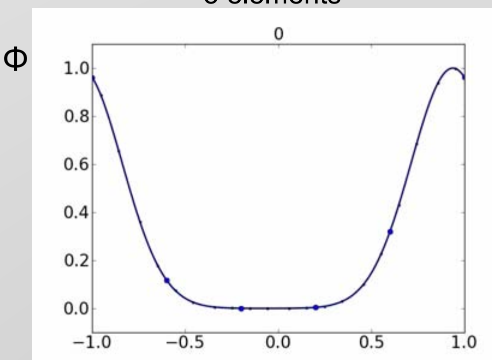
Φ



x

Polynomial degree = 5
5 elements

Φ



x

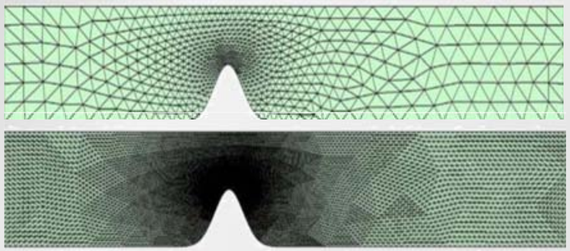
- Higher-order and low-order should be compared:
  - At the same accuracy (most comp. efficient scheme wins)
  - At the same comp. efficiency (most accurate scheme wins)

Rarely done jointly in literature, difficult

- Higher-order can be more accurate for the same comp. efficiency

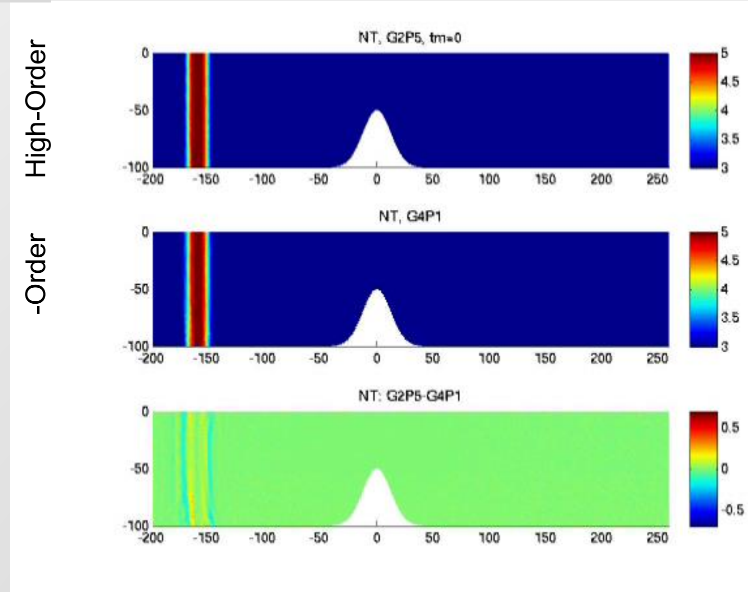
Accurate numerical modeling of Phytoplankton

- Biological patch (NPZ model)
  - ~19 tidal cycles (8.5 days)
  - Mean flow + daily tidal cycle
- Two discretizations of similar cost
  - 6th order scheme on coarse mesh
  - 2nd order scheme on fine mesh

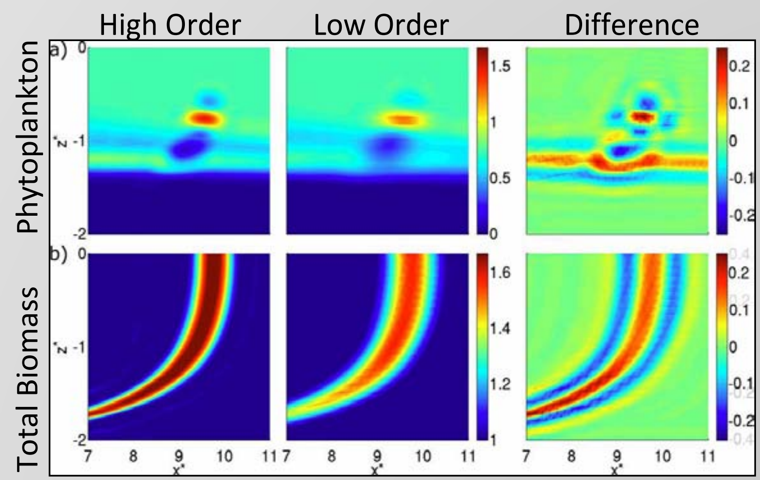


P=5, $O(\Delta x^6)$
21,000 DOFs

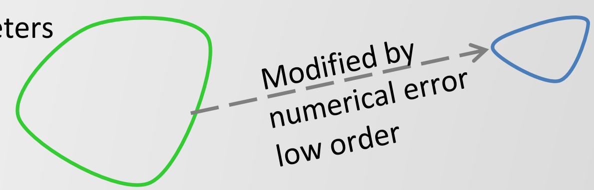
P=1, $O(\Delta x^2)$
67,200 DOFs



- Numerical diffusion of lower-order scheme modifies the concentration of biomass in patch



True Limit cycle
from parameters



© Springer-Verlag. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.
Source: Ueckermann, Mattheus P., and Pierre FJ Lermusiaux. "High-order Schemes for 2D Unsteady Biogeochemical Ocean Models." *Ocean Dynamics* 60, no. 6 (2010): 1415-45.

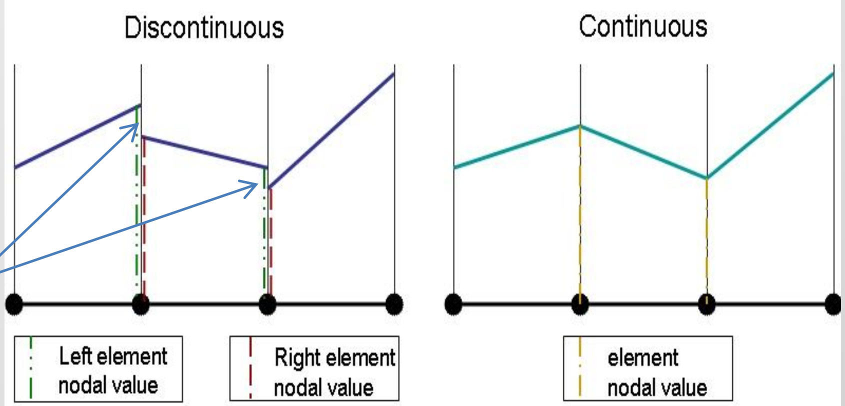
(Ueckermann and Lermusiaux, OD, 2010)

Discontinuous Galerkin (DG) Finite Elements

- The **basis** can be continuous or discontinuous across elements

$$\phi(\mathbf{x},t) \approx \phi_h(\mathbf{x},t) = \sum_{i=1}^{n_b} \phi_i(t)\underbrace{\theta_i(\mathbf{x})}_{\text{basis}}$$

- CG: Continuous Galerkin
- DG: Discontinuous Galerkin



Main challenge with DG: Defining numerical flux → $\hat{\phi} = f(\phi^+, \phi^-)$

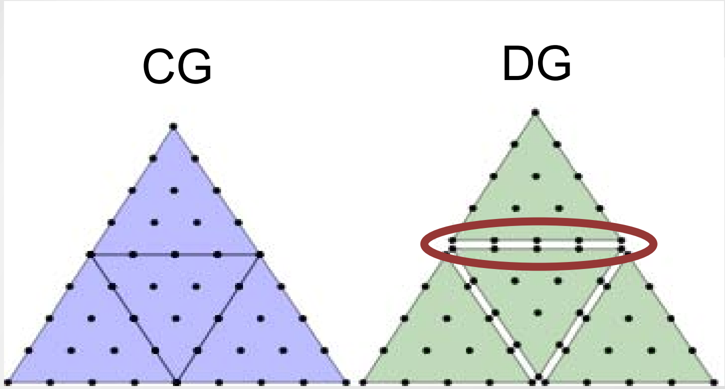
© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

- Advantages of DG:
  - Efficient data-structures for parallelization and computer architectures
  - Flexibility to add stabilization for advective terms (upwinding, Riemann solvers)
  - Local conservation of mass/momentum
- Disadvantages:
  - Difficult to implement
  - Relatively new (Reed and Hill 1978, Cockburn and Shu 1989-1998)
    - Standard practices still being developed
  - **Expensive compared to Continuous Galerkin for elliptic problems**
  - **Numerical stability issues due to Gibbs oscillations**

Hybrid Discontinuous Galerkin (HDG) computationally competitive with CG

Biggest concern with DG: Efficiency for elliptic problems

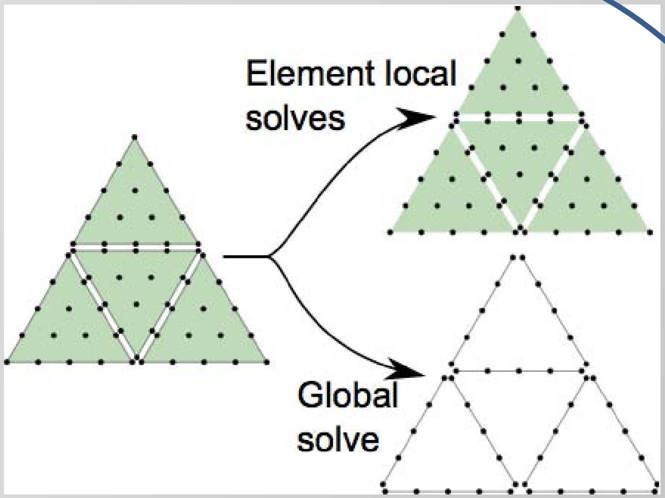
- For DG, unknowns are **duplicated** at edges of element



HDG

- HDG is competitive to CG while retaining properties of DG
- HDG parameterizes element-local solutions using new edge-space **λ**

Nguyen et al. (JCP2009)
Cockburn et al. (SJNA2009)



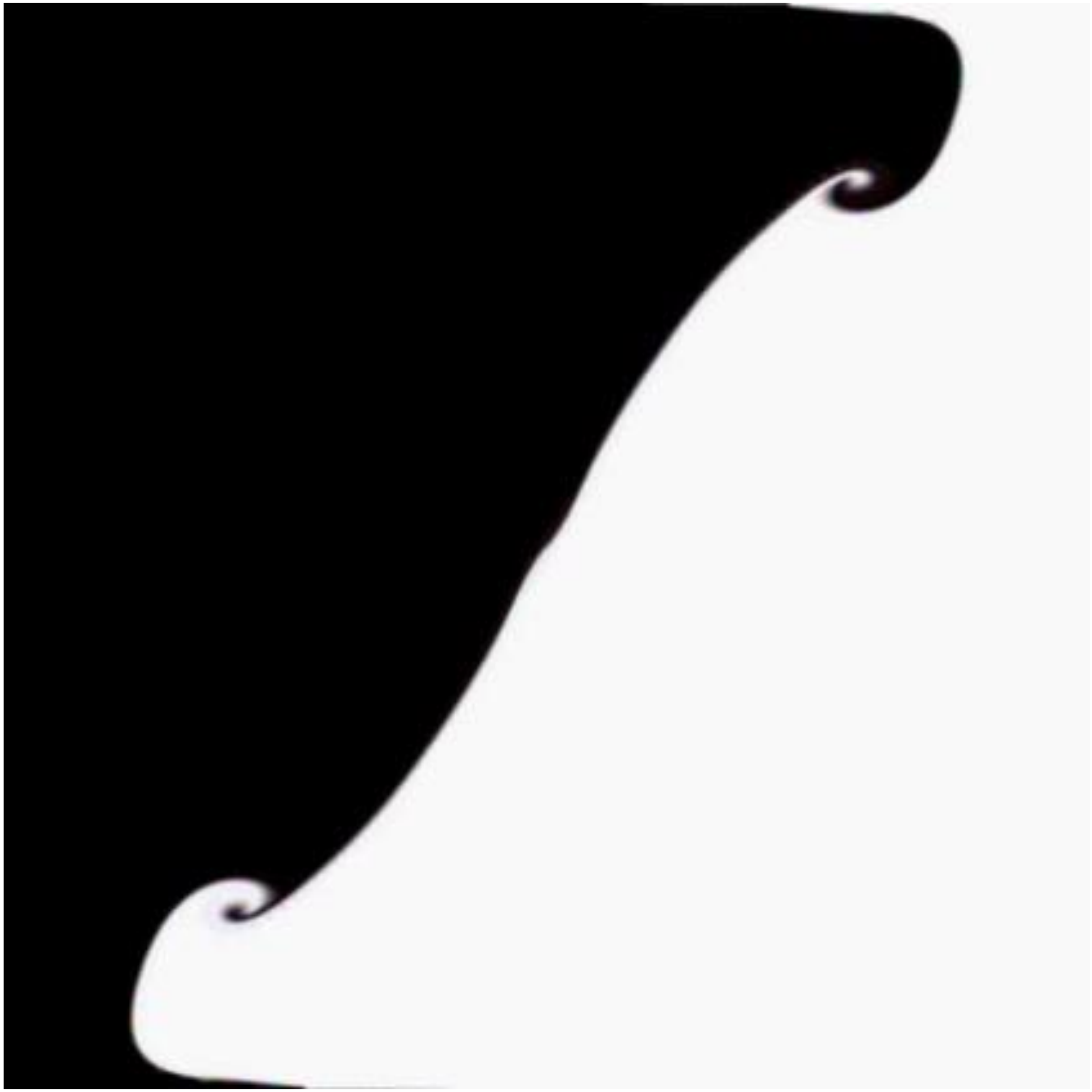
Key idea: Given initial and boundary conditions for a domain, the interior solution can be calculated (with HDG, also in each local element)

Continuity on the edge space of:
- Fields
- Normal component of total fluxes, e.g. numerical trace of total stress

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Next-generation CFD for Regional Ocean Modeling: Hybrid Discontinuous Galerkin (HDG) FEMs

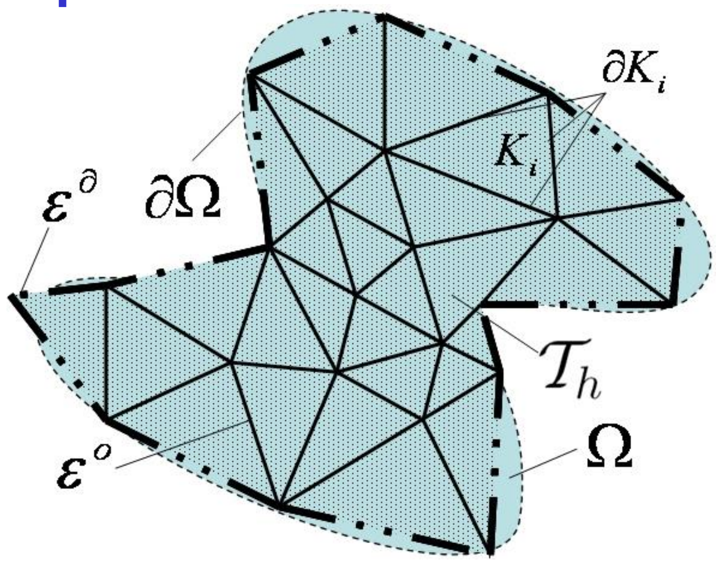
The Lock Exchange Problem: Gr=1.25e7



DG – Worked Example

- Choose function space

$$W_h^p = \{w \in L^2(\Omega) : w\,|_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_h\}$$



Original eq. : $$\frac{\partial u}{\partial t} + \nabla \cdot (\vec{c}u) = 0$$

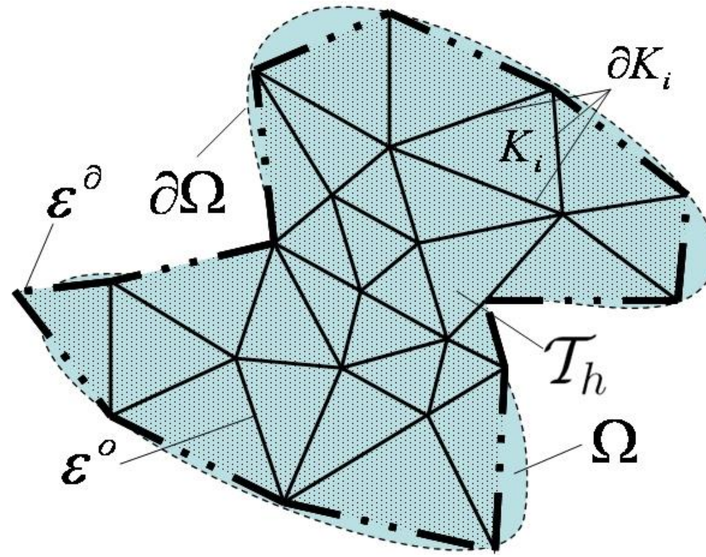
MWR : $$\int_K w\frac{\partial u}{\partial t}dK + \int_K w\nabla \cdot (\vec{c}u)dK = 0$$

Integrate by parts : $$\int_K w\frac{\partial u}{\partial t}dK - \int_K \nabla w \cdot (\vec{c}u)dK + \int_K \nabla \cdot (\vec{c}uw)dK = 0$$

Divergence theorem, leads to “weak form” : $$\int_K w\frac{\partial u}{\partial t}dK - \int_K \nabla w \cdot (\vec{c}u)dK + \int_{\partial K} w\hat{n} \cdot \vec{c}\hat{u}d\partial K = 0$$

DG – Worked Example, Cont'd

Substitute basis and test functions, which are the same for Galerkin FE methods



Weak form : $$\int_K w\frac{\partial u}{\partial t}dK - \int_K \nabla w\cdot(\vec{c}u)dK + \int_{\partial K} w\hat{n}\cdot\vec{c}\hat{u}d\partial K = 0$$

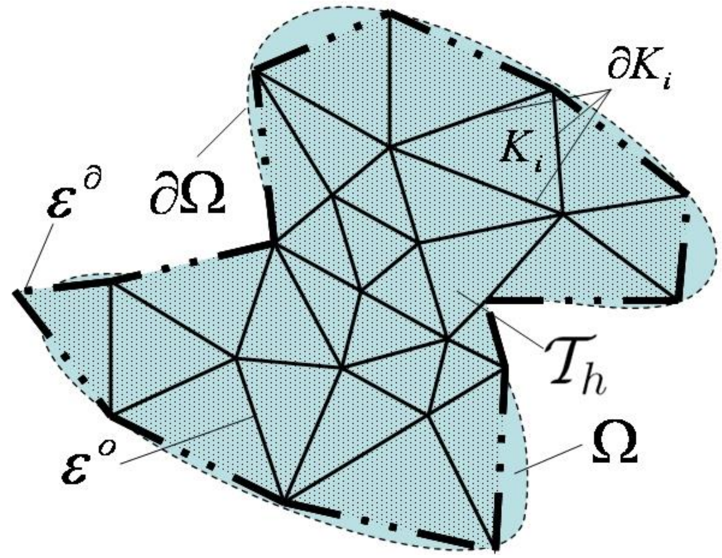
Shape fcts. : $$\int_K w\frac{\partial u_j}{\partial t}\theta_j dK - \int_K \nabla w\cdot(\vec{c}u_j\theta_j)dK + \int_{\partial K} w\hat{n}\cdot(\vec{c}\hat{u}_j\theta_j)d\partial K = 0$$

Basis fcts. : $$\int_K \theta_i\frac{\partial u_j}{\partial t}\theta_j dK - \int_K \nabla\theta_i\cdot(\vec{c}u_j\theta_j)dK + \int_{\partial K} \theta_i n\cdot(\vec{c}\hat{u}_j\theta_j)d\partial K = 0$$

Final FE eq. : $$\frac{\partial u_j}{\partial t}\int_K \theta_i\theta_j dK - \vec{c}u_j\int_K \nabla\theta_i\cdot(\theta_j)dK + \vec{c}\hat{u}_j\cdot\int_{\partial K} \hat{n}(\theta_j\theta_i)d\partial K = 0$$

DG – Worked Example, Cont'd

- Substitute for matrices
  - **M**- Mass matrix
  - **K**- Stiffness or Convection matrix
- Solve specific case of 1D eq.



$$
\begin{aligned}
\frac{\partial u_j}{\partial t}\int_K \theta_i\theta_j dK - \vec{c}u_j\int_K \nabla\theta_i\cdot(\theta_j)dK + \vec{c}\hat{u}_j\cdot\int_{\partial K}\hat{n}(\theta_j\theta_i)d\partial K &= 0\\
\mathbf{M}_{ij}\frac{\partial u_j}{\partial t} - \mathbf{K}_{ij}\vec{c}u_j + \vec{c}\hat{u}_j\cdot\int_{\partial K}\hat{n}(\theta_j\theta_i)d\partial K &= 0
\end{aligned}
$$

Euler time-integration :

$$1D:\quad \mathbf{M}_{ij}\frac{\partial u_j}{\partial t} - \mathbf{K}_{ij}\vec{c}u_j + \vec{c}\hat{u}_j\cdot\hat{n}(\theta_j\theta_i) = 0$$

$$1D:\quad u_j^{n+1} = u_j^n + \Delta t\mathrm{M}^{-1}(\mathbf{K}_{ij}\vec{c}u_j - \vec{c}\hat{u}_j\cdot\hat{n}\delta_{ij})$$

Fluxes: central with upwind :

$$\hat{u}_j = \frac{u_j^+ + u_h^-}{2} - \frac{c}{|c|}\frac{u_j^+ - u_h^-}{2}$$

DG – Worked Example - Code

```
clear all, clc, clf, close all

syms x
%create nodal basis
%Set order of basis function
%N >=2
N = 3;

%Create basis
if N==3
 theta = [1/2*x^2-1/2*x;
             1- x^2;
          1/2*x^2+1/2*x];
else
xi = linspace(-1,1,N);
 for i=1:N
  theta(i)=sym('1');
  for j=1:N
   if j~=i
    theta(i) = ...
      theta(i)*(x-xi(j))/(xi(i)-xi(j));
   end
  end
 end
end
```

```
%Create mass matrix
for i = 1:N
 for j = 1:N
  %Create integrand
  intgr  = int(theta(i)*theta(j));
  %Integrate
  M(i,j) =...
          subs(intgr,1)-subs(intgr,-1);
 end
end
%create convection matrix
for i = 1:N
 for j = 1:N
  %Create integrand
  intgr = ...
          int(diff(theta(i))*theta(j));
  %Integrate
  K(i,j) = ...
          subs(intgr,1)-subs(intgr,-1);
 end
end
```

DG – Worked Example – Code Cont'd

```
%% Initialize u
Nx = 20;
dx = 1./Nx;
%Multiply Jacobian through mass matrix.
%Note computationl domain has length=2,
actual domain length = dx
M=M*dx/2;

%Create "mesh"
x = zeros(N,Nx);
for i = 1:N
    x(i,:) =...
 dx/(N-1)*(i-1):dx:1-dx/(N-1)*(N-i);
end
%Initialize u vector
u = exp(-(x-.5).^2/.1^2);

%Set timestep and velocity
dt=0.002;   c=1;
%Periodic domain
ids = [Nx,1:Nx-1];
```

```
%Integrate over time
for i = 1:10/dt
    u0=u;
    %Integrate with 4th order RK
  for irk=4:-1:1
    %Always use upwind flux
    r = c*K*u;
    %upwinding
    r(end,:) = r(end,:) - c*u(end,:);
    %upwinding
    r(1,:) = r(1,:) + c*u(end,ids);
    %RK scheme
    u = u0 + dt/irk*(M\r);
    end
    %Plot solution
    if ~mod(i,10)
        plot(x,u,'b')
        drawnow
    end
  end
end
```

MIT OpenCourseWare
http://ocw.mit.edu

2.29 Numerical Fluid Mechanics
Spring 2015

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.