Efficient Numerical Integration in Polygonal Finite Element Method

S. E. Mousavi[a] **and N. Sukumar**[b]

[a] University of Texas, Austin

[b] University of California, Davis

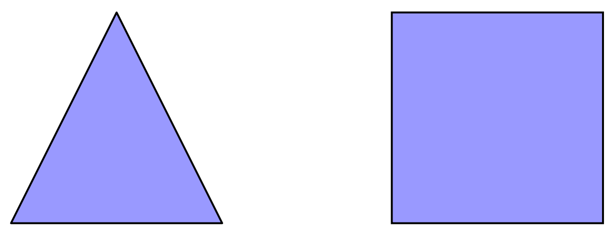
NSF Workshop on Barycentric Coordinates | Columbia University, NY | July 26, 2012

Need for Integration

Mass matrix: $$\mathbf{M}^e = \int_{\Omega_0} \mathbf{N}(\xi;\eta)^T \mathbf{N}(\xi;\eta)\det(\mathbf{J})\,d\xi\,d\eta$$

Stiffness matrix: $$\mathbf{K}^e = \int_{\Omega_0} (\mathbf{J}^{-1}\mathbf{B}(\xi;\eta))^T (\mathbf{J}^{-1}\mathbf{B}(\xi;\eta))\det(\mathbf{J})\,d\xi\,d\eta$$

Force vector: $$\mathbf{F}^e = \int_{\Omega_0} \mathbf{N}(\xi;\eta)^T f(\xi;\eta)\det(\mathbf{J})\,d\xi\,d\eta$$

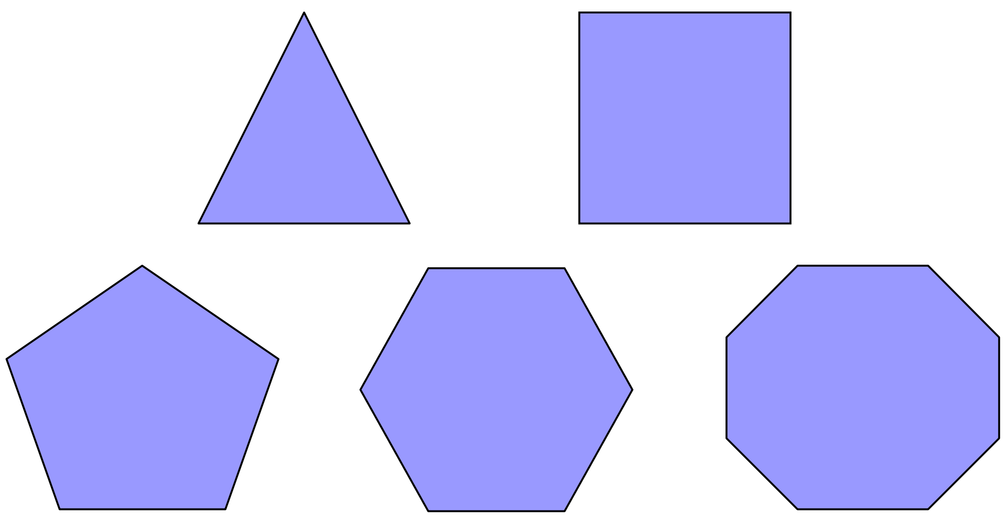


Need for Integration

Mass matrix: $$\mathbf{M}^e = \int_{\Omega_0} \mathbf{N}(\xi,\eta)^T \mathbf{N}(\xi,\eta) \det(\mathbf{J})\, d\xi\, d\eta$$

Stiffness matrix: $$\mathbf{K}^e = \int_{\Omega_0} (\mathbf{J}^{-1}\mathbf{B}(\xi,\eta))^T (\mathbf{J}^{-1}\mathbf{B}(\xi,\eta)) \det(\mathbf{J})\, d\xi\, d\eta$$

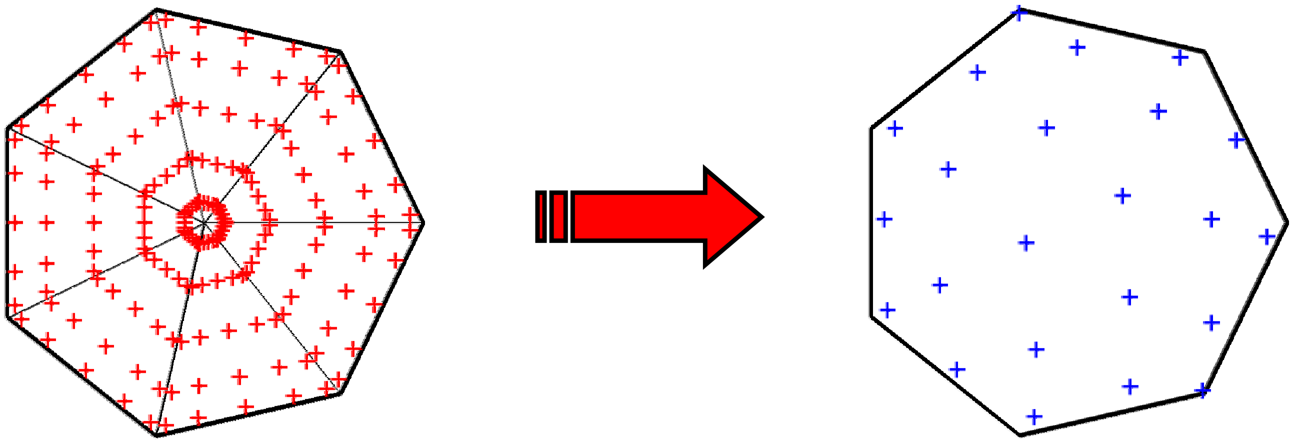
Force vector: $$\mathbf{F}^e = \int_{\Omega_0} \mathbf{N}(\xi,\eta)^T f(\xi,\eta) \det(\mathbf{J})\, d\xi\, d\eta$$



Goal

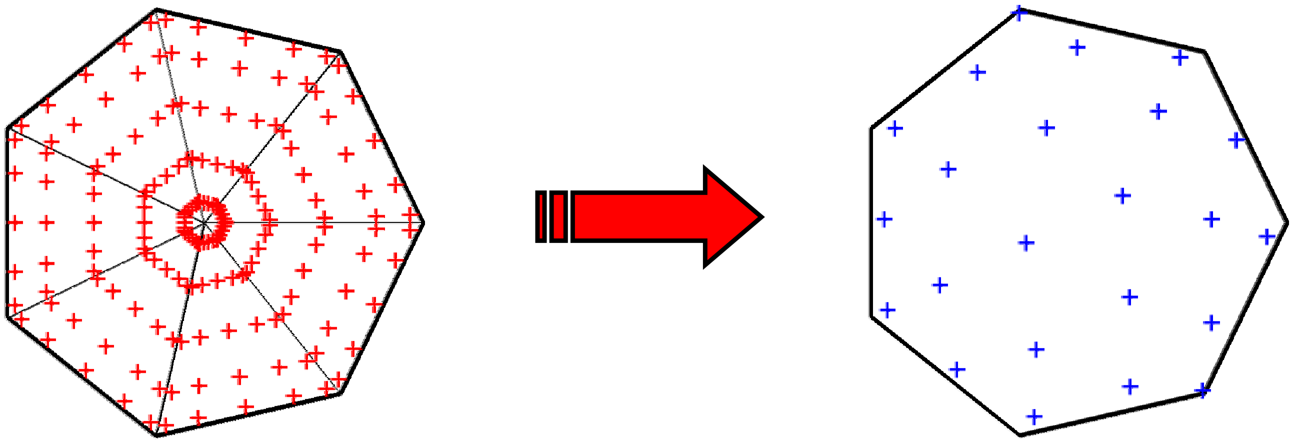
- **Constructing Gaussian-like quadratures for n-gons, n>3**

  (polynomial precision)



Goal

- Constructing Gaussian-like quadratures for n-gons, n>3 (polynomial precision)



- Weighted quadratures: quadratures for polygonal basis functions (rational polynomials)

Outline

- Moment equations
- Node elimination algorithm
- Quadratures on the fly
- Weighted quadratures

Moment Equations

$$Q(f) = \sum_{i=1}^{n} f(\mathbf{x}_i) w_i \approx \int_\Omega \omega(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

For a set of basis functions $\Phi = \{\phi_j\}_{j=1}^{m}$ **over the domain** Ω **, find the quadrature** $Q = \{\mathbf{x}_i, w_i\}_{i=1}^{n}$ **such that:**

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

Moment Equations

$$Q(f) = \sum_{i=1}^{n} f(\mathbf{x}_i) w_i \approx \int_\Omega \omega(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

For a set of basis functions $\Phi = \{\phi_j\}_{j=1}^m$ **over the domain** Ω **, find the quadrature** $Q = \{\mathbf{x}_i, w_i\}_{i=1}^n$ **such that:**

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\, d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\, d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\, d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

- **Newton iterations**
- **Least squares solution**

Moment Equations

$$Q(f) = \sum_{i=1}^{n} f(\mathbf{x}_i) w_i \approx \int_\Omega \omega(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

For a set of basis functions $\Phi = \{\phi_j\}_{j=1}^{m}$ over the domain Ω , find the quadrature $Q = \{\mathbf{x}_i, w_i\}_{i=1}^{n}$ such that:

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

- **Newton iterations**
  - **time-consuming, fewer points: n ≈ m/(d+1)**
- **Least squares solution**
  - **faster, more points: n ≈ m**

Node Elimination Algorithm

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

Node Elimination Algorithm

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

[Xiao and Gimbutas, Comp. Math. App., 2010]

Node Elimination Algorithm

$$\begin{pmatrix} \int_\Omega w(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega w(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega w(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

[Xiao and Gimbutas, Comp. Math. App., 2010]

- **Expected number of integration points:** m=**3**
  (in two dimensions)

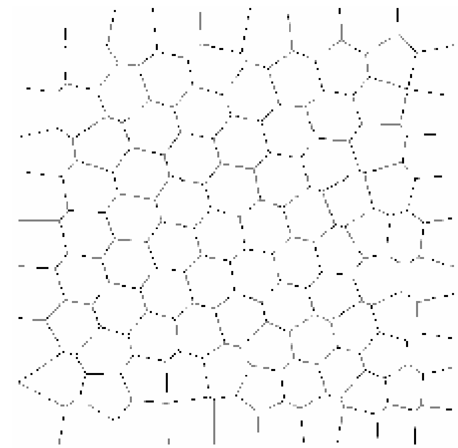
- **Start from a quadrature over the partitions**

- **Significance factor:** $s_j = \omega_j \sum_{i=1}^{m} \phi_i^2(\mathbf{x}_j)$

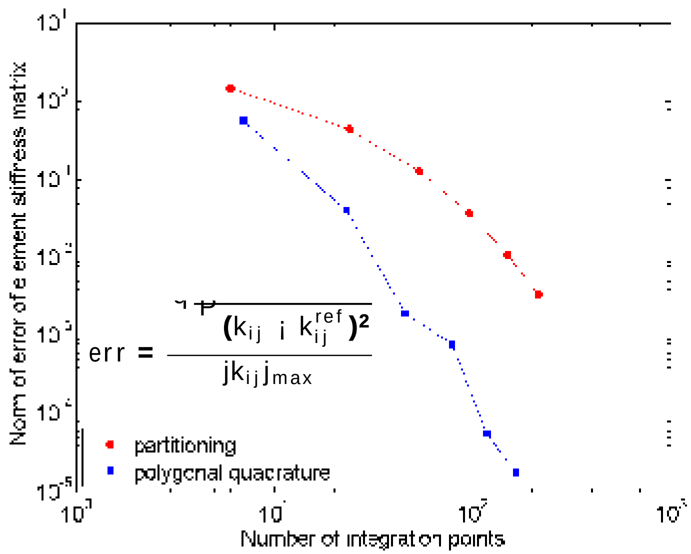
Polygonal Quadratures: Accuracy

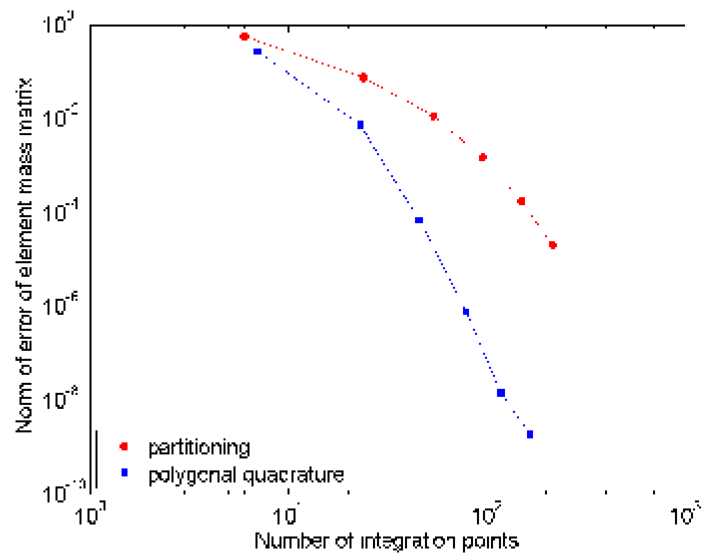
Number of integration points for different n-gons

order	Expected	Obtained			
		$n=5$	$n=6$	$n=7$	$n=8$
5	7	7	7	7	7
10	22	23	23	23	24
20	77	79	79	81	80
30	166	166	166	166	166

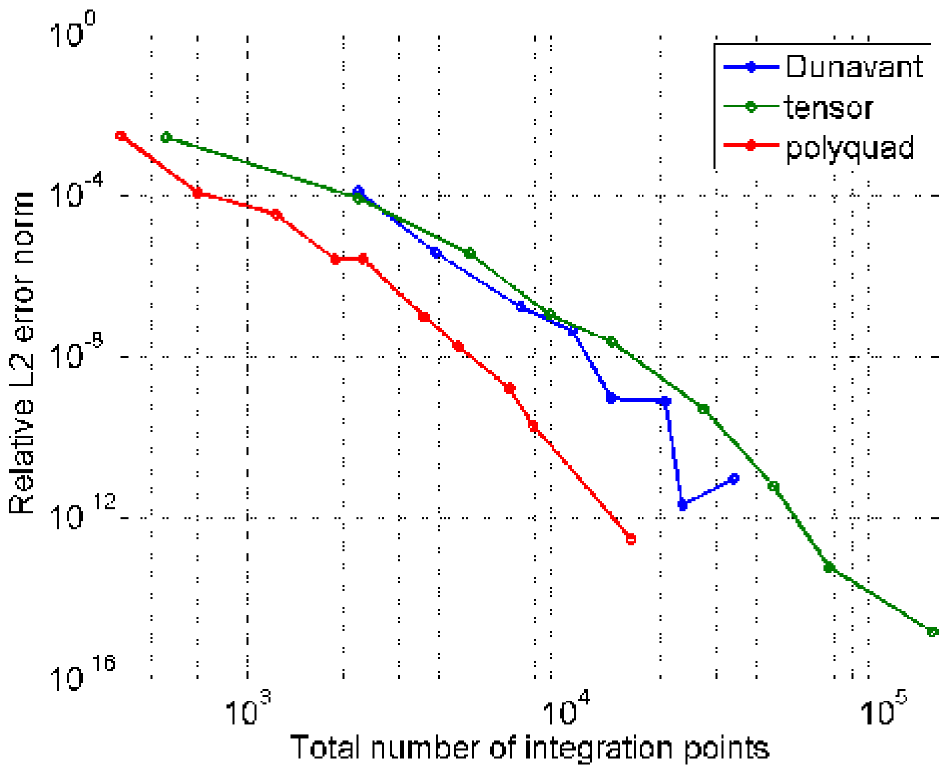


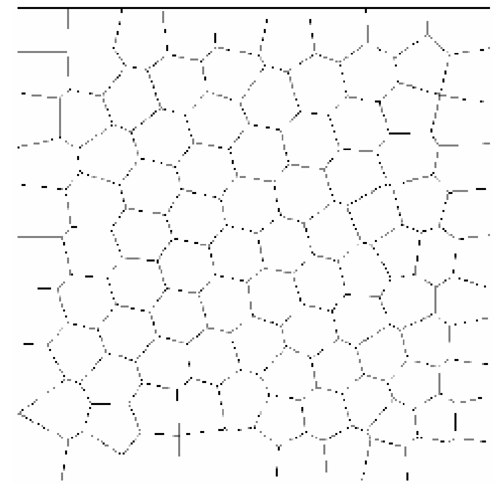
[PolyMesher: Talischi, Paulino, et al., 2012]



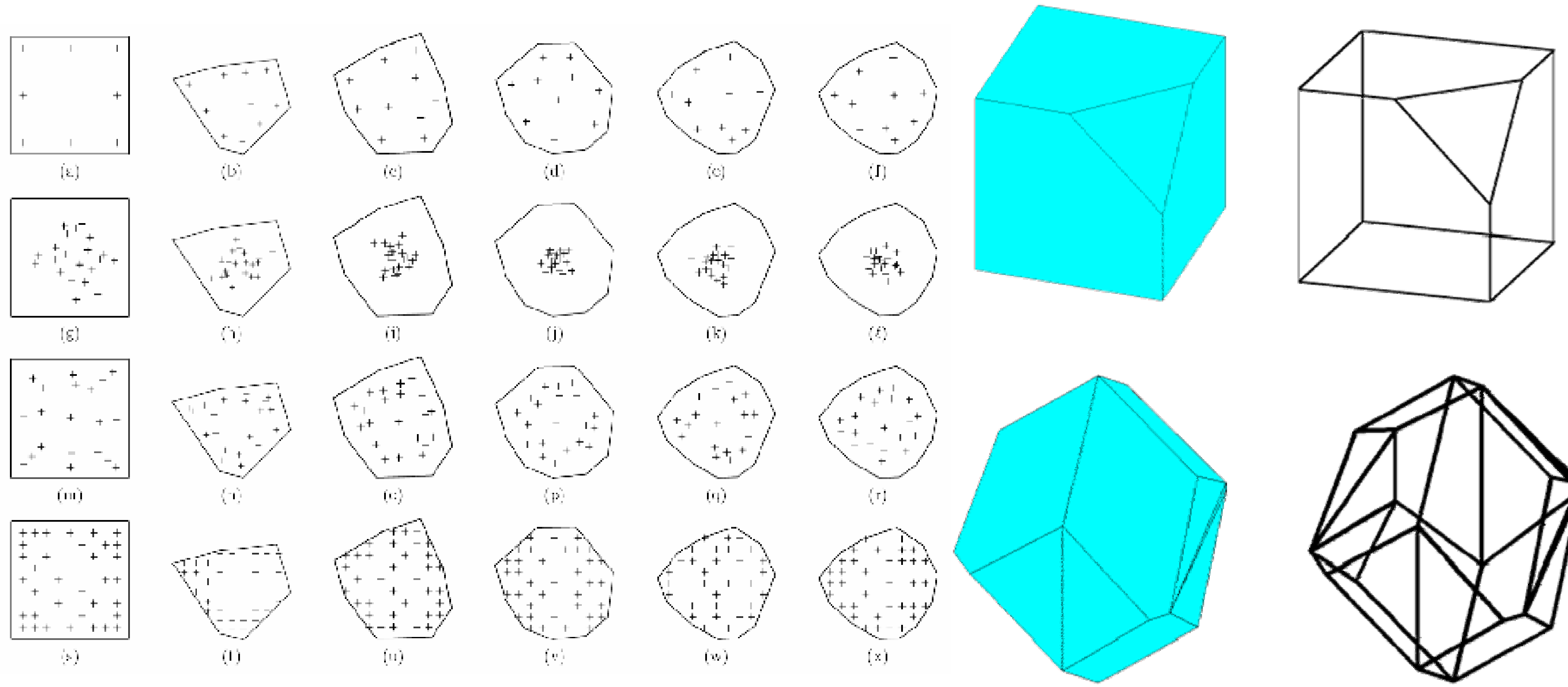


Displacement Patch Test





Quadratures on the Fly

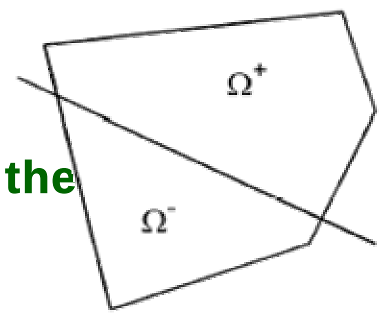


Quadratures on the Fly

$$\begin{pmatrix} \int_\Omega \omega(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\ \int_\Omega \omega(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\ \vdots \\ \int_\Omega \omega(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

Quadratures on the Fly

$$
\begin{pmatrix}
\int_\Omega w(\mathbf{x})\phi_1(\mathbf{x})\,d\mathbf{x} \\
\int_\Omega w(\mathbf{x})\phi_2(\mathbf{x})\,d\mathbf{x} \\
\vdots \\
\int_\Omega w(\mathbf{x})\phi_m(\mathbf{x})\,d\mathbf{x}
\end{pmatrix}
=
\begin{pmatrix}
\phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\
\phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\
\vdots & \vdots & \vdots & \vdots \\
\phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n)
\end{pmatrix}
\begin{pmatrix}
w_1 \\ w_2 \\ \vdots \\ w_n
\end{pmatrix}
$$



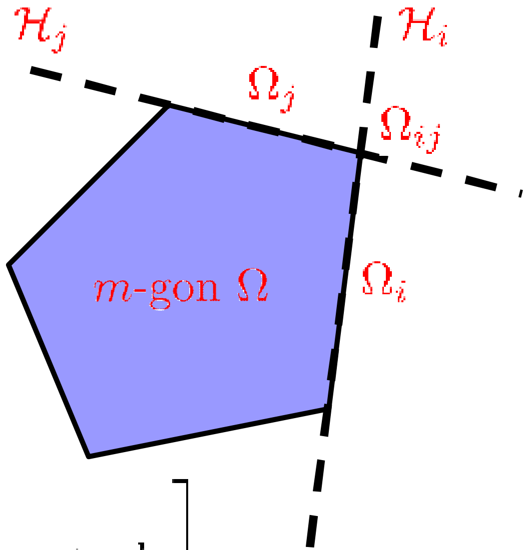
q For strong discontinuity: replace the weight function with the generalized Heaviside function

q For weak discontinuity: construct two quadratures on the two sides of the interface

Homogeneous Quadratures

For $f : \mathbb{R}^n \to \mathbb{R}$ **a real, q-homogeneous function,**

$f(\lambda \mathbf{x}) = \lambda^q f(\mathbf{x})$ for $\lambda > 0$, $\mathbf{x} \in \mathbb{R}^n$, **we have:**



$$\int_\Omega f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{m} \frac{d(\mathbf{o}, \mathcal{H}_i)}{n+q} \int_{\Omega_i} f d\mu$$

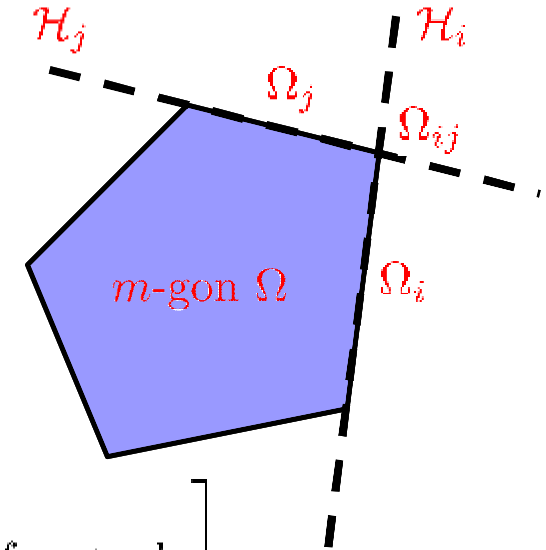
$$\int_{\Omega_i} f d\mu = \frac{1}{n+q-1} \left[\sum_{j \neq i} d_i(\mathbf{x}_0, \mathcal{H}_{ij}) \int_{\Omega_{ij}} f d\nu + \int_{\Omega_i} < \nabla f, \mathbf{x}_0 > d\mu \right]$$

[Lasserre, Proc. AMS, 1998]

Homogeneous Quadratures

For $f : \mathbb{R}^n \to \mathbb{R}$ a real, q-homogeneous function,

$f(\lambda \mathbf{x}) = \lambda^q f(\mathbf{x})$ for $\lambda > 0$, $\mathbf{x} \in \mathbb{R}^n$, **we have:**

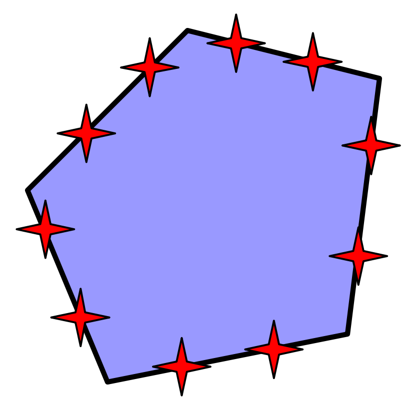


$$\int_\Omega f(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{m} \frac{d(\mathbf{o}, \mathcal{H}_i)}{n+q} \int_{\Omega_i} f \, d\mu$$

$$\int_{\Omega_i} f \, d\mu = \frac{1}{n+q-1} \left[\sum_{j \neq i} d_i(\mathbf{x}_0, \mathcal{H}_{ij}) \int_{\Omega_{ij}} f \, d\nu + \int_{\Omega_i} < \nabla f, \mathbf{x}_0 > d\mu \right]$$

[Lasserre, Proc. AMS, 1998]

$$\Longrightarrow \mathcal{Q}(f) = \sum_{a=1}^{nsp} w_a f(\mathbf{x}_a) = (n+q) \int_\Omega f(\mathbf{x}) d\mathbf{x}$$



(convex domains)

Discontinuous Quadratures

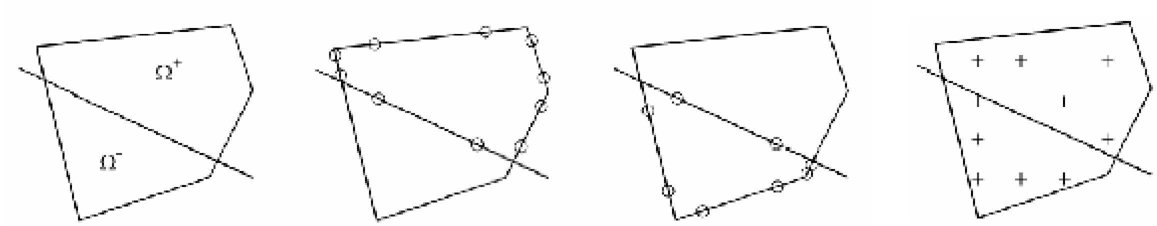
$$\mathcal{Q}_d(f(\mathbf{x})) = \sum_{i=1}^{n} w_i f(\mathbf{x}_i) = \int_{\Omega} f(\mathbf{x}) H(\mathbf{x}) d\mathbf{x} = \int_{\Omega^+} f(\mathbf{x}) d\mathbf{x} - \int_{\Omega^-} f(\mathbf{x}) d\mathbf{x}$$

$$\begin{pmatrix} \int_{\Omega} H(\mathbf{x}) \phi_1(\mathbf{x})\, d\mathbf{x} \\ \int_{\Omega} H(\mathbf{x}) \phi_2(\mathbf{x})\, d\mathbf{x} \\ \vdots \\ \int_{\Omega} H(\mathbf{x}) \phi_m(\mathbf{x})\, d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

Discontinuous Quadratures

$$\mathcal{Q}_d(f(\mathbf{x})) = \sum_{i=1}^{n} w_i f(\mathbf{x}_i) = \int_{\Omega} f(\mathbf{x}) H(\mathbf{x}) d\mathbf{x} = \int_{\Omega^+} f(\mathbf{x}) d\mathbf{x} - \int_{\Omega^-} f(\mathbf{x}) d\mathbf{x}$$

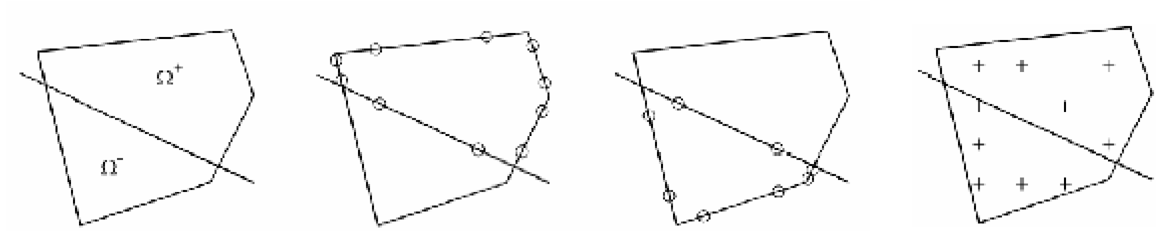
$$\begin{pmatrix} \int_\Omega H(\mathbf{x})\phi_1(\mathbf{x})\, d\mathbf{x} \\ \int_\Omega H(\mathbf{x})\phi_2(\mathbf{x})\, d\mathbf{x} \\ \vdots \\ \int_\Omega H(\mathbf{x})\phi_m(\mathbf{x})\, d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$



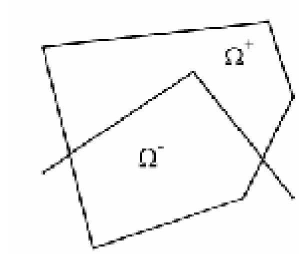
Discontinuous Quadratures

$$\mathcal{Q}_d(f(\mathbf{x})) = \sum_{i=1}^{n} w_i f(\mathbf{x}_i) = \int_\Omega f(\mathbf{x}) H(\mathbf{x}) d\mathbf{x} = \int_{\Omega^+} f(\mathbf{x}) d\mathbf{x} - \int_{\Omega^-} f(\mathbf{x}) d\mathbf{x}$$

$$\begin{pmatrix} \int_\Omega H(\mathbf{x})\phi_1(\mathbf{x})\, d\mathbf{x} \\ \int_\Omega H(\mathbf{x})\phi_2(\mathbf{x})\, d\mathbf{x} \\ \vdots \\ \int_\Omega H(\mathbf{x})\phi_m(\mathbf{x})\, d\mathbf{x} \end{pmatrix} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_n) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_n) \\ \vdots & \vdots & \vdots & \vdots \\ \phi_m(\mathbf{x}_1) & \phi_m(\mathbf{x}_2) & \dots & \phi_m(\mathbf{x}_n) \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}$$

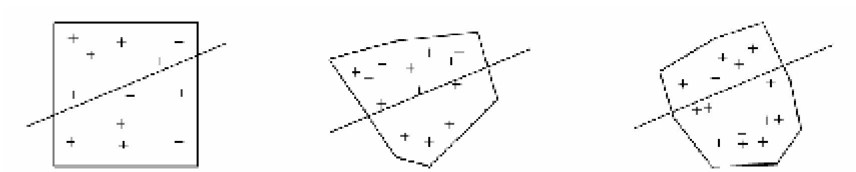


$$\int_\Omega f(\mathbf{x}) H(\mathbf{x}) d\mathbf{x} = \begin{cases} 2\int_{\Omega+} f(\mathbf{x}) d\mathbf{x} - \int_\Omega f(\mathbf{x}) d\mathbf{x} & \text{if } \Omega^+ \text{ is convex,} \\ \int_\Omega f(\mathbf{x}) d\mathbf{x} - 2\int_{\Omega-} f(\mathbf{x}) d\mathbf{x} & \text{if } \Omega^- \text{ is convex.} \end{cases}$$

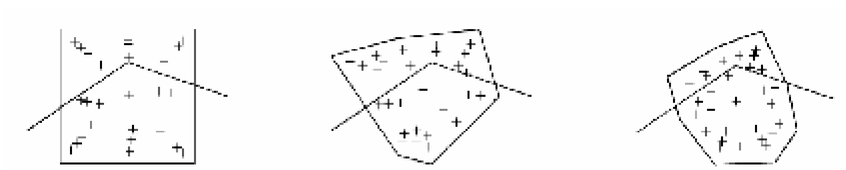


[Mousavi and Sukumar, Comp. Mech., 2011]

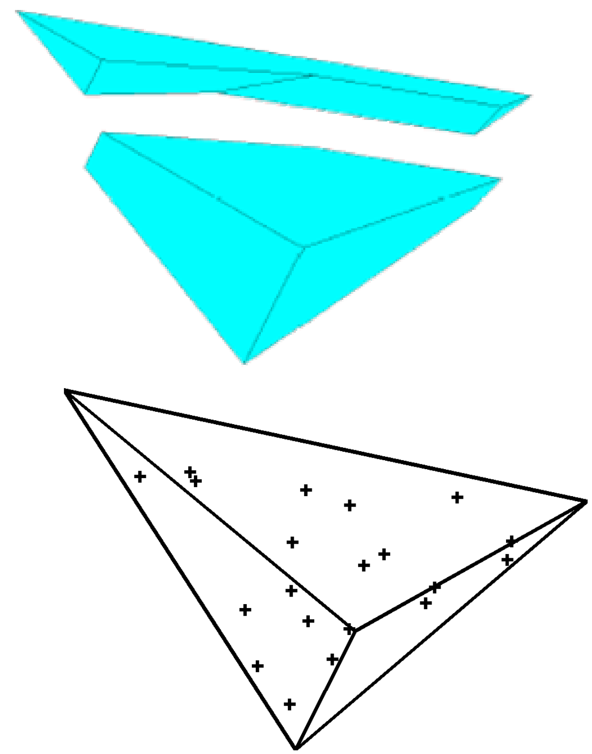
Discontinuous Quadratures



$order = 4,\ \ numx = 15,\ \ \epsilon = 10^{-14}$



$order = 6,\ \ numx = 28,\ \ \epsilon = 10^{-14}$



$order = 3,\ \ numx = 20,\ \ \epsilon = 10^{-15}$

[Mousavi and Sukumar, Comp. Mech., 2011]

Weighted Quadratures: mass matrix

Typical element mass matrix $$\mathbf{M}^e = \int_{\Omega_0} \mathbf{N}(\xi,\eta)^T \mathbf{N}(\xi,\eta) \det(\mathbf{J})\, d\xi\, d\eta$$

Jacobian of the transformation $$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

$$x = \sum_{i=1}^{n} N_i(\xi,\eta) x_i; \quad y = \sum_{i=1}^{n} N_i(\xi,\eta) y_i$$

Polygonal basis functions $$N_i = \frac{p(\xi,\eta)}{Q}$$ **(e.g., for hexagon:** $Q = 3 - \xi^2 - \eta^2$**)**

$$\Longrightarrow \det(\mathbf{J}) \propto \frac{1}{Q^3}$$

$$\Longrightarrow M^e_{IJ} = \int_{\Omega_0} \frac{p(\xi,\eta)}{Q^5} d\xi\, d\eta$$

Weighted Quadratures: stiffness matrix

Element stiffness matrix

$$\mathbf{K}^e = \int_{\Omega} \mathbf{B}(x,y)^T \mathbf{B}(x,y)\,dx\,dy$$

$$= \int_{\Omega_0} (\mathbf{J}^{-1}\mathbf{B}(\xi,\eta))^T (\mathbf{J}^{-1}\mathbf{B}(\xi,\eta)) \det(\mathbf{J})\,d\xi\,d\eta$$

Inverse of the Jacobian

$$\det(\mathbf{J}) = \frac{P(\xi,\eta)}{Q^3}$$

$$\Rightarrow \mathbf{J}^{-1} = \frac{Q^3}{P(\xi,\eta)}\left[\frac{p(\xi,\eta)}{Q^2}\right] = \frac{Q}{P(\xi,\eta)}[p(\xi,\eta)] \quad (2 \times 2)$$

Basis function derivative

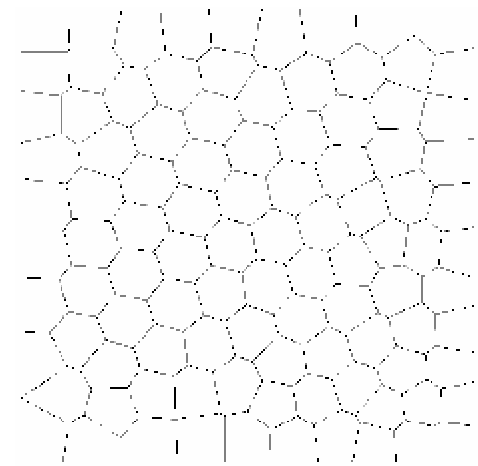
$$B_i = \frac{\partial N_i}{\partial \xi} = \frac{\frac{\partial p}{\partial \xi} Q - \frac{\partial Q}{\partial \xi} p}{Q^2}$$

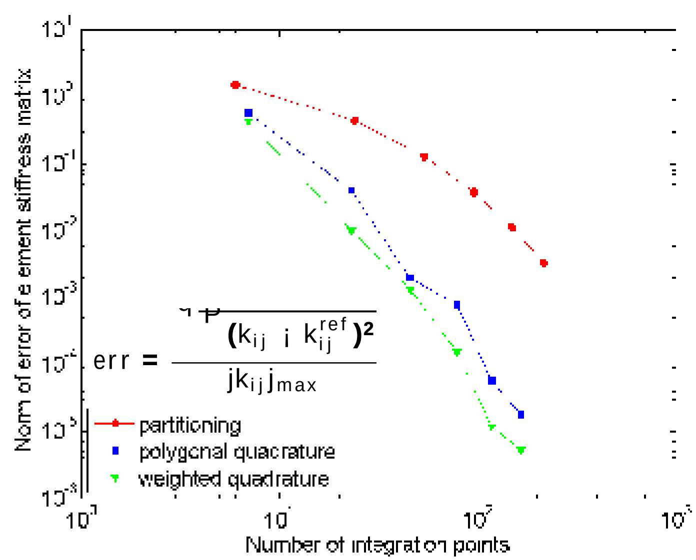
$$\Rightarrow K^e_{IJ} = \int_{\Omega_0} \frac{p(\xi,\eta)}{P(\xi,\eta)Q^5}\,d\xi\,d\eta$$

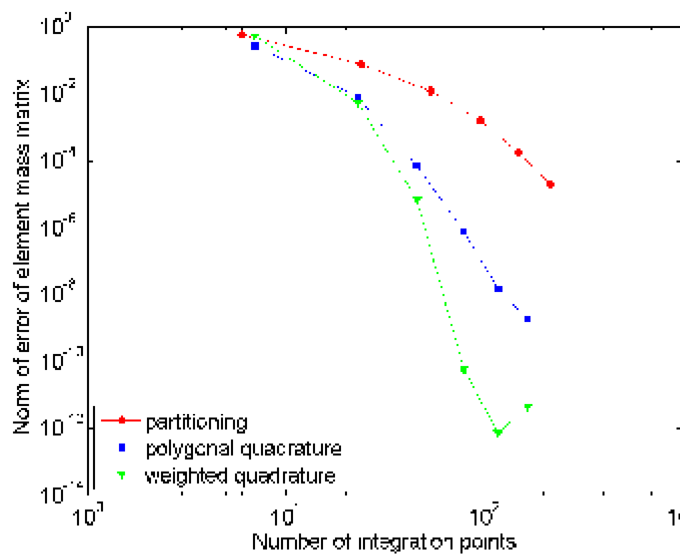
Weighted Quadratures: Accuracy

Number of integration points for different n-gons

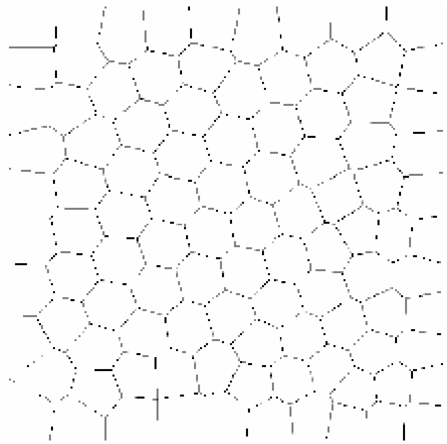
order	Expected	Obtained			
		$n = 5$	$n = 6$	$n = 7$	$n = 8$
5	7	8	8	7	8
10	22	23	23	24	24
20	77	81	81	80	82
30	166	165	165	185	166

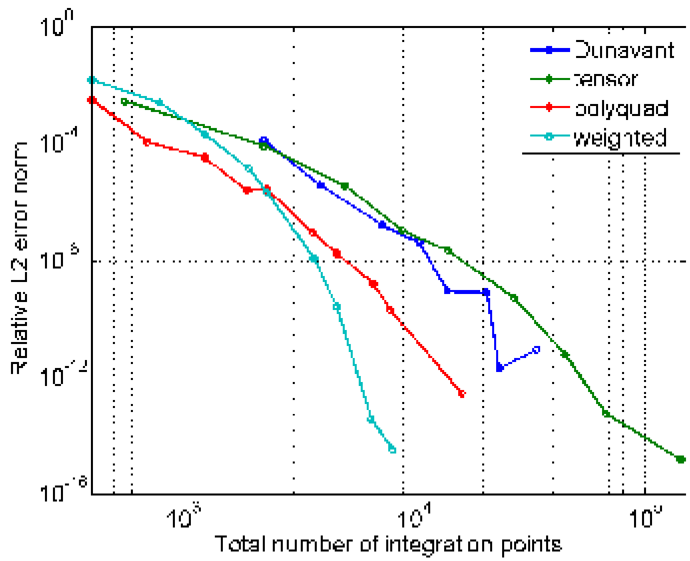




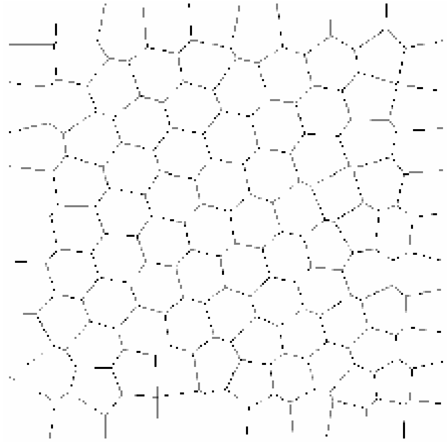


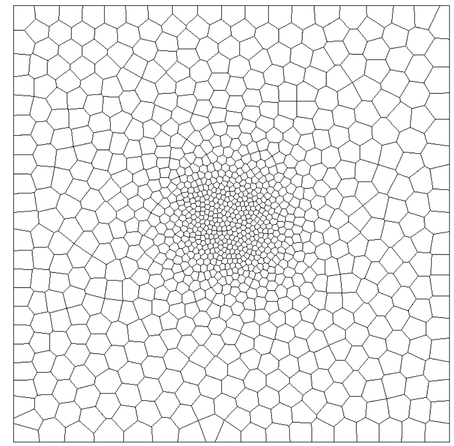
Displacement Patch Test

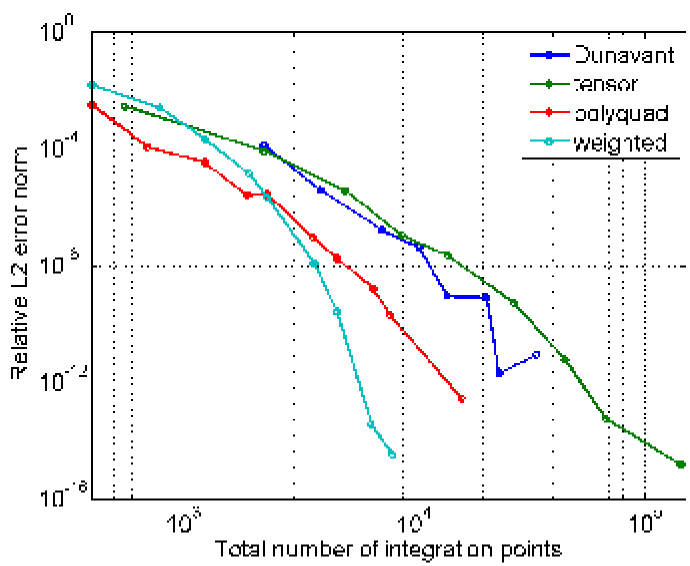


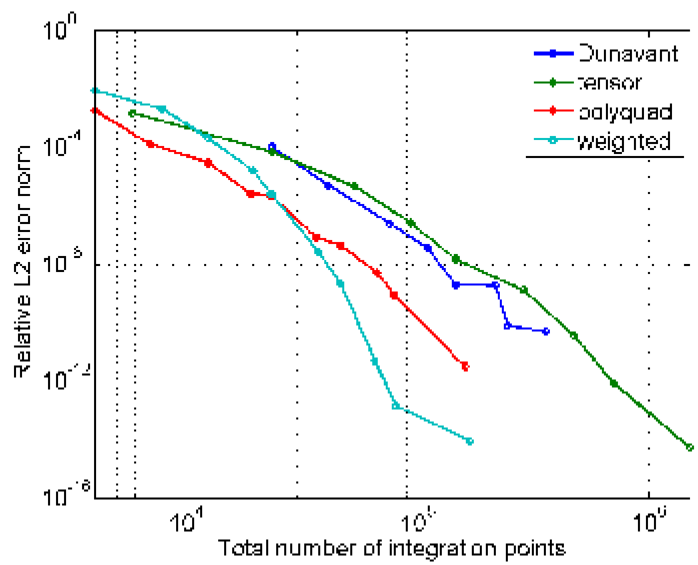


Displacement Patch Test









Conclusions

- Polygonal quadratures are more efficient than triangulation
- Polygonal quadratures need one level of mapping, whereas partitioning requires two levels of mapping
- For higher accuracies, weighted quadratures are more efficient than polynomial precision polygonal quadratures