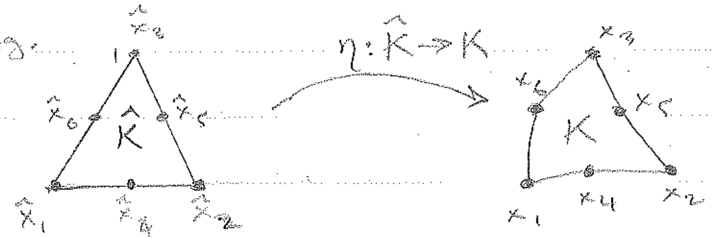
ISOPARAMETRIC ELEMENTS

Features :

(i) Provide a standard domain for integration

(ii) Allow for curved simplex edges (higher order elements)

e.g.



$\underline{x} = \eta(\underline{\lambda})$

$\underline{\lambda} = \eta^{-1}(\underline{x})$

K = "actual" domain of element

$\hat{K}$ = "standard" domain

η = 1-1 mapping of pts between $\hat{K}$ and K

$\{\hat{N}_a,\ a = 1, \ldots, n\}$ = "Standard" shape functions on $\hat{K}$

Ex $d=2$ $k=2$ Quadratic 6-node triangle

$$\left.\begin{aligned}
\hat{N}_1 &= \lambda_1(2\lambda_1 - 1) \\
\hat{N}_2 &= \lambda_2(2\lambda_2 - 1) \\
\hat{N}_3 &= \lambda_3(2\lambda_3 - 1) \\
\hat{N}_4 &= \lambda_1\lambda_2 \\
\hat{N}_5 &= \lambda_2\lambda_3 \\
\hat{N}_6 &= \lambda_3\lambda_1
\end{aligned}\right\} \quad \frac{\partial \hat{N}_a}{\partial \lambda_\alpha} \text{ easy to compute.}$$

"Actual" Shape functions $N_a = \hat{N}_a \circ \eta^{-1}$

i.e. $$\left.\begin{aligned} N_a(x) &= \hat{N}_a(\eta^{-1}(x)) \\ \hat{N}_a(\lambda) &= N_a(\eta(\lambda)) \end{aligned}\right\} \quad x_i = \eta_i(\lambda) \quad \lambda_\alpha = \eta_\alpha^{-1}(x)$$

Isoparametric $\quad \eta_i(\lambda) = \sum_{a=1}^{n} x_{ia} \hat{N}_a(\lambda)$

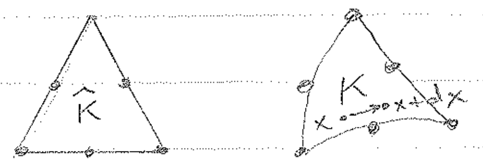
Properties:

(i) $\sum_{a=1}^{n} N_a(x) = \sum_{a=1}^{n} \hat{N}_a(\lambda) = 1$ Shape functions interpolate constants exactly

(ii) At nodes $x_a = \eta(\hat{x}_a)$ $a=1,\ldots,n$

Then $N_a(x_b) = \hat{N}_a(\eta^{-1}(x_b)) = \hat{N}_a(\hat{x}_b) = \delta_{ab}$

<u>Computation of Derivatives</u>



$N_a(x+dx) \sim N_a(x) + N_{a,i}(x)\,dx_i$

$dN_a(\lambda) = N_{a,i}(x)\,\underbrace{dx_i}_{\text{how related to } d\lambda_\alpha?}$

$x_i = \eta_i(\lambda)$

$dx_i = \eta_{i,\alpha}(\lambda)\,d\lambda_\alpha$

and

$1 = \sum_\alpha \lambda_\alpha \Rightarrow 0 = \sum_\alpha d\lambda_\alpha$

and

$\eta_{i,\alpha}(\lambda) = \sum_{a=1}^{n} x_{ia} \hat{N}_{a,\alpha}(\lambda)$

together,

$$\begin{Bmatrix} dx_i \\ \hline 0 \end{Bmatrix} = \begin{bmatrix} \eta_{i,\alpha} \\ 1 \cdots 1 \end{bmatrix} \begin{Bmatrix} d\lambda_1 \\ \vdots \\ d\lambda_{d+1} \end{Bmatrix}$$

$$\underline{dx} = \underbrace{\underline{J}(\lambda)}_{\text{Jacobian matrix}} \; \underline{d\lambda}$$

$\therefore \; d\lambda_\alpha = J^{-1}_{\alpha i}(\lambda)\,dx_i \qquad \alpha = 1,\ldots,d+1;\ i=1,\ldots,d$

$\therefore \; dN_a = \hat{N}_{a,\alpha}\,d\lambda_\alpha = \hat{N}_{a,\alpha} J^{-1}_{\alpha i}\,dx_i$

$\equiv N_{a,i}\,dx_i$

$$\Rightarrow \boxed{N_{a,i} = \hat{N}_{a,\alpha} J^{-1}_{\alpha i}}$$

Also, to compute integrals over K we need relation between dV and $d\hat{V}$

$$dV = \det(\underset{\sim}{J})\, d\hat{V} \qquad \hat{V} \equiv |\hat{K}| = \text{measure of } \hat{K}$$
$$= J\, d\hat{V}$$

BREAK.

<u>NUMERICAL QUADRATURE</u> (INTEGRATION)

Integrals we have to compute

$$K^{K}_{iakb} = \int_K c_{ijkl}(\underset{\sim}{x})\, N_{a,j}(\underset{\sim}{x}) N_{b,l}(\underset{\sim}{x})\, dV$$

$$f^{ext,K}_{ia} = \int_K f_i(\underset{\sim}{x}) N_a(\underset{\sim}{x})\, dV + \int_{\partial K \cap S_2} \bar{t}_i(\underset{\sim}{x}) N_a(\underset{\sim}{x})\, dV$$

Generic problem: $I = \int_K f\, dV$

But f is usually defined over $\hat{K}$ for isoparametric elements, e.g. $N(\underset{\sim}{x}) = \hat{N}(\eta^{-1}(\underset{\sim}{x})) = \hat{N}(\lambda)$:

$N = \hat{N} \circ \eta^{-1} \rightarrow N \circ \eta = \hat{N} \quad N(\underset{\sim}{x}(\lambda)) = \hat{N}(\lambda)$

so

$$I = \int_{\hat{K}} f \circ \eta \, \frac{dV}{d\hat{V}}\, d\hat{V} = \int_{\hat{K}} f \circ \eta \; J\, d\hat{V} = \int_{\hat{K}} \hat{f}\, d\hat{V}$$

The idea behind numerical quadrature is to replace integrals with sums

$$I \approx \tilde{I} = \left(\sum_{p=1}^{Q} (f \circ \eta)(\lambda^p)\, \hat{w}_p\, J(\lambda^p) \right) \hat{V}$$

$$= \sum_{p=1}^{Q} (f \circ \eta)(\lambda^p)\, w_p \qquad w_p = \hat{w}_p\, J(\lambda^p)\, \hat{V}$$

Q = no. of quadrature points

$\{\lambda^p,\ p = 1, \dots, Q\}$ = quadrature points (sampling pts)

$\{w^p,\ p = 1, \dots, Q\}$ = quadrature weights

Alternative Perspective on Isoparametric Simplicial Elements:
Curvilinear ~~Coor~~ Parametrization

Barycentric coords: $\sum_{i=1}^{d+1} \lambda_i = 1$

$\Rightarrow$ 1 of λ's is redundant. Choose d of the λ's as curvilinear coords $\{s_i,\ i=1,\dots,d\}$.

E.g., $s_i = \lambda_i,\ i=1,\dots,d$ and $\lambda_{d+1} = 1-\sum_{i=1}^{d} s_i$

$$\hat N(\underline{\lambda}) \longrightarrow \hat N(\underline{s})$$

$$dx_i = \eta_{i,j}\, ds_j \qquad \eta_i = \sum_a x_{ia} \hat N_a(\underline{s}) \qquad \eta_{i,j} = \sum_a x_{ia} \hat N_{a,j}(\underline{s})$$

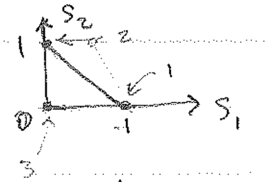
$$\{dx_i\} = \underbrace{[\eta_{i,j}]}_{\underline{J}\ \leftarrow\ d\times d \text{ matrix}}\{ds_j\}$$

$$ds_i = J^{-1}_{ij}\, dx_j \longrightarrow dN_a = \hat N_{a,i}\, ds_i = \hat N_{a,i} J^{-1}_{ij}\, dx_j$$

$$\Rightarrow \boxed{N_{a,j} = \hat N_{a,i} J^{-1}_{ij}}$$

$$\hat N_{a,j}(\underline{s}) = \hat N_{a,\alpha} \frac{\partial \lambda_\alpha}{\partial s_j}$$

Ex $d=2$ $k=1$



$\lambda_1 = s_1$ $\lambda_2 = s_2$ $\lambda_3 = 1-s_1-s_2$

$$N_a = \lambda_a \Rightarrow \begin{aligned} \hat N_1 &= \lambda_1 = s_1 \\ \hat N_2 &= \lambda_2 = s_2 \\ \hat N_3 &= \lambda_3 = 1-s_1-s_2 \end{aligned}$$

$\hat N_{1,1} = 1$

$\hat N_{1,2} = 0$ $\qquad \hat N_{3,1} = -1$ $\qquad [\hat N_{a,i}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$

$\hat N_{2,1} = 0$ $\qquad \hat N_{3,2} = -1$

$\hat N_{2,2} = 1$

Shape Function Tests:

Test 1: Consistency:

(i) Compute $N^E_{a,j}(x)$ at random $x \in E$

(ii) Compute $N^E_{a,j}$ numerically

(iii) compare

Test 2: C^0 completeness (can interpolate a linear polynomial exactly)

(i) Choose random linear polynomial

$$p(s) = \sum_{|\alpha| \leq 1} \underset{\substack{\uparrow \\ \text{random}}}{a_\alpha} s^\alpha$$

(ii) sample $p(s)$ at nodes $p_a = p(s_a)$ $a = 1, \ldots, N$

(iii) evaluate p^* @ random s^* $p^* = p(s^*)$

(iv) interpolate $\tilde{p} = \sum_{a=1}^{N} N_a(s^*) p_a$

(v) compare $\tilde{p} \stackrel{?}{=} p^*$

Test ensures that U_h is subspace of U.

BACK TO QUADRATURE

$$\therefore \quad K^E_{iakb} = \sum_{p=1}^{Q} c_{ijkl}(\lambda^{(p)})\, N^E_{a,j}(\eta(\lambda^{(p)}))\, N^E_{b,l}(\eta(\lambda^{(p)}))\; \hat{w}_p\, J(\lambda^{(p)})\, \hat{V}$$

$$f^{ext,E}_{ia} = \sum_{p=1}^{Q} f_i(\eta(\lambda^{(p)}))\, N^E_a(\eta(\lambda^{(p)}))\; \hat{w}_p\, J(\lambda^{(p)})\, \hat{V}$$

\# pts for surf. integration $\rightarrow$

$$+ \sum_{q=1}^{Q_s} \bar{t}_i(\eta(\lambda^{(q)}))\, N^E_a(\eta(\lambda^{(q)}))\; \hat{w}_q\, J(\lambda^{(q)})\, \hat{A}$$

Computation of Element Arrays

Recall Voigt Notation

$$\underline{\sigma} = [\sigma_{11}\ \sigma_{22}\ \sigma_{33}\ \sigma_{12}\ \sigma_{13}\ \sigma_{23}]^T$$

$$\underline{\epsilon} = [\epsilon_{11}\ \epsilon_{22}\ \epsilon_{33}\ 2\epsilon_{12}\ 2\epsilon_{13}\ 2\epsilon_{23}]^T$$

$$W = \tfrac{1}{2} c_{ijkl}\, \epsilon_{ij}\, \epsilon_{kl} = \tfrac{1}{2} c_{\alpha\beta}\, \epsilon_\alpha\, \epsilon_\beta$$

$$\sigma_\alpha = \frac{\partial W}{\partial \epsilon_\alpha} = c_{\alpha\beta}\, \epsilon_\beta$$

$$\epsilon^E_{ij}(x) = \tfrac{1}{2}(u_{i,j} + u_{j,i}) = \tfrac{1}{2}\sum_{a=1}^{N^E}(u^E_{ia} N^E_{a,j} + u^E_{ja} N^E_{a,i})$$

$$\hookrightarrow \quad \epsilon^E_\alpha = \sum_{a=1}^{N^E} B^E_{\alpha i a}\, u^E_{ia}$$

or "Strain-Displacement operator"

$$\underline{B}^E = \text{"B-matrix"}, \quad \longrightarrow \quad \underline{\epsilon}^E(x) = \sum_{a=1}^{N} \underline{B}^E_a(x)\, \underline{u}^E_a = \underline{B}^E \underline{u}^E$$

$$\underline{B}^E = [\,\underline{B}^E_1 \mid \ldots \mid \underline{B}^E_N\,] \qquad \underline{u}^E = \begin{Bmatrix} \underline{u}^E_1 \\ \vdots \\ \underline{u}^E_N \end{Bmatrix}$$

$$\underline{B}^E_a = \begin{bmatrix} N^E_{a,1} & 0 & 0 \\ 0 & N^E_{a,2} & 0 \\ 0 & 0 & N^E_{a,3} \\ N^E_{a,2} & N^E_{a,1} & 0 \\ N^E_{a,3} & 0 & N^E_{a,1} \\ 0 & N^E_{a,3} & N^E_{a,2} \end{bmatrix}$$

$$\epsilon_{11} = \sum_a N_{a,1}\, u_{1a}$$

$$\epsilon_{22} = \ldots$$

$$2\epsilon_{12} = \sum_a N_{a,1}\, u_{2a} + N_{a,2}\, u_{1a}$$

$$\ldots$$

Implementing &
Testing Quadrature Rules

Recall $$I = \int_{\hat{K}} \hat{f}\, d\hat{V} = \sum_{p=1}^{Q} \hat{f}(x^{(p)})\, \hat{w}_p \quad \hat{V}$$

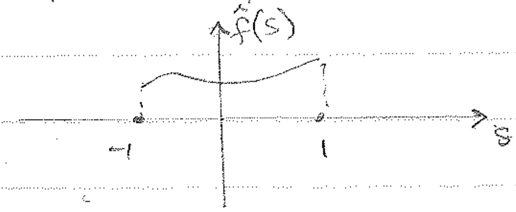
Implement in C++

```
Class Quadrature {
  ~~std::vector<~~
  struct { Point {
    tvmet::Vector< dim_n, double > coords;
    double weight;
  }

  std::vector< Point > _points;
  void compute ( int order ); // where you code
}                              // point locations & weights

int main() {
  Quadrature quad;
  quad.compute(2);
  double I = 0;
  ~~for( p= ; p< Q ; p++ )~~
  for( p= quad.points().begin(); p!= quad.point().end();
                                                        p++)
    I+= f( p -> coords ) * p -> weight * V

}
```

Gauss(-Legendre) Quadrature

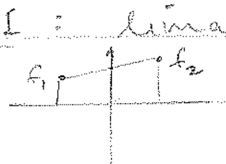
Choose locations of sampling points so that polynomial functions can be integrated exactly w/ fewest possible pts.

1-D:



$$I = \int_{-1}^{1} \hat{f}(s)\,ds = \left(\sum_{p=1}^{Q} \hat{f}(s_p)\,\hat{\omega}_p\right)\hat{V}$$

Ex 1: linear $\Rightarrow$ must integrate $\int_{-1}^{1} 1\,ds$ and $\int_{-1}^{1} s\,ds$ exactly

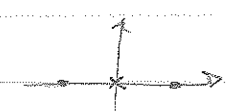


$$\int_{-1}^{1} 1\,ds = \sum_{p=1}^{Q} \cancel{\hat{f}(s_p)}^{\,1}\,\hat{\omega}_p\,\cancel{\hat{V}}^{\,2} = 2\sum_{p=1}^{Q}\hat{\omega}_p$$

$$= 2 \Rightarrow \boxed{\sum_{p=1}^{Q}\hat{\omega}_p = 1}$$

$$\int_{-1}^{1} s\,ds = 0 = \sum_{p=1}^{Q} s_p\,\hat{\omega}_p\,2 \Rightarrow \boxed{\sum_{p=1}^{Q} s_p\,\hat{\omega}_p = 0}$$

Fewest pts? $Q=1 \Rightarrow \hat{\omega}_1 = 1 \quad s_1^* = 0$



In general pts. are located symmetrically about $s=0$ and a polynomial of degree $2n-1$ can be integrated exactly by an n-pt rule.

Check $2n-1 = 1$ (linear) $\Rightarrow$ $n=1$ (1-pt rule) ✓.

Ex 2: $n=2 \rightarrow 2n-1 = 3$ (2-pt rule $\rightarrow$ cubic)

i.e., integrate $\{1, s, s^2, s^3\}$ exactly

$$\int_{-1}^{1} 1\, ds = 2 = 2(\hat{w}_1 + \hat{w}_2) \qquad (1)$$

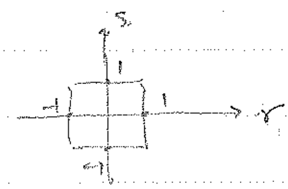
$$\int_{-1}^{1} s\, ds = 0 = 2(s_1 \hat{w}_1 + s_2 \hat{w}_2) \qquad (2)$$

$$\int_{-1}^{1} s^2 ds = \frac{2}{3} = 2(s_1^2 \hat{w}_1 + s_2^2 \hat{w}_2) \qquad (3)$$

$$\int_{-1}^{1} s^3 ds = 0 = 2(s_1^3 \hat{w}_1 + s_2^3 \hat{w}_2) \qquad (4)$$

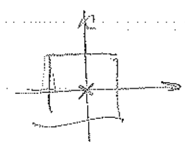
$$\Rightarrow \quad \hat{w}_1 = \hat{w}_2 = \frac{1}{2} \qquad s_1 = -\frac{1}{\sqrt{3}} \quad s_2 = \frac{1}{\sqrt{3}}$$

<u>2-D</u> : Quadralateral elements

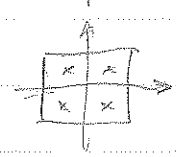


$$I = \int_{-1}^{1}\int_{-1}^{1} \hat{f}(r,s)\, dr\, ds$$

$$= \sum_{p=1}^{n}\left(\sum_{q=1}^{n} \hat{f}(r_p, s_q)\, \hat{w}_q\, 2\right) \hat{w}_p\, 2$$

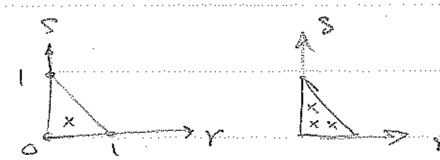


$$= \sum_{p=1}^{n}\sum_{q=1}^{n} f(r_p, s_q)\, \hat{w}_q \hat{w}_p \underbrace{2 \cdot 2}_{\hat{V}}$$



$$= \sum_{p=1}^{n \times n} f(r_p, s_p)\, \hat{w}_p\, \hat{V}$$

Triangular Elements



$$\lambda_2 = r, \ \lambda_3 = s, \ \lambda_1 = 1 - r - s$$

$$I = \int_{\hat{A}} \hat{f}(\lambda)\, d\hat{A} = \left(\sum_{p=1}^{Q} \hat{f}(\lambda_p)\, \hat{w}_p\right) \hat{V}$$

$$\hat{A} = \frac{1}{2}|K| = \frac{1}{2}$$

3-D : Hex $\quad I = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \hat{f}(r,s,t)\, dr\, ds\, dt = \sum_{p=1}^{n \times n \times n} \hat{f}(r_p, s_p, t_p)\, \hat{w}_p\, \hat{V}$

Tet $\quad I = \int_{\hat{V}} \hat{f}(\lambda)\, d\hat{V} = \sum_{p=1}^{Q} \hat{f}(\lambda_p)\, \hat{w}_p\, \hat{V} \qquad \hat{V} = \frac{1}{6}$