P-Version Shape functions

Reference: F.E. Analysis, B.A. Szabo and I. Babuska, Wiley, 1991

Desire high order shape functions that:

- Have good numerical conditioning (Lagrange polynomials do not work well)
- Support the effective construction of element matrices
- Can be used in creating meshes with variable ~~order~~ polynomial order over the domain

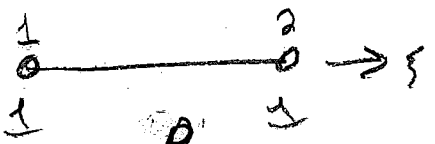
In the approach developed they

- selected polynomials with some orthogonality properties that yield better conditioned matrices.
- Support the hierarch construction of elements of increasing order (that is the matrix for the element at order p is a submatrix of the element at order p+1 - our Lagrange shape functions do not do this.)

Polynomials that have nice orthogonality properties do not yield interpolating shape functions - makes satisfying the inter element continuity requirement an Issue.

Ideas we saw with Serendipity elements can help us here.
- We will start with basic C^0 Linear Lagrangian and add things (the way its done here there is no correction stuff needed)

Consider 1-D



$$u^h = a_1 N_1 + a_2 N_2 + \sum_{i=2}^{p} a_{i+1} N_{i+1}$$

We can address the interelement continuity requirement doing the following:

Require:

$N_1(0) = 1$, $N_1(1) = 0$

$N_2(0) = 0$, $N_2(1) = 1$

$N_i(0) = 0$, $N_i(1) = 0$ for $i = 3(1)p+1$

With this:

$u^h(0) = a_1 = u_1$

$u^h(1) = a_2 = u_2$

$N_1 = \frac{1-\xi}{2}$, $N_2 = \frac{1+\xi}{2}$ work for this

the question is the remain shape functions which must be higher order – basically up to p and satisfy $N_i(0) = N_i(1) = 0$ $i = 3(1)p+1$

Shape functions that will do this are constructed from integrals of Legendre Polynomials $P_{i-1}(\xi)$

$$N_{i+1} \equiv \phi_i(\xi) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{\xi} P_{i-1}(t)\,dt \qquad i = 2(1)p$$

where the Legendre polynomials are:

$$P_0(t) = 1$$
$$P_1(t) = t$$
$$P_2(t) = \tfrac{1}{2}(3t^2 - 1)$$
$$P_3(t) = \tfrac{1}{2}(5t^3 - 3t)$$
$$P_4(t) = \tfrac{1}{8}(35t^4 - 30t^2 + 3)$$
$$P_5(t) = \tfrac{1}{8}(63t^5 - 70t^3 + 15t)$$
$$\vdots$$
$$P_n(t) = \frac{2n-1}{n} t\, P_{n-1}(t) - \frac{n-1}{n} P_{n-2}(t)$$

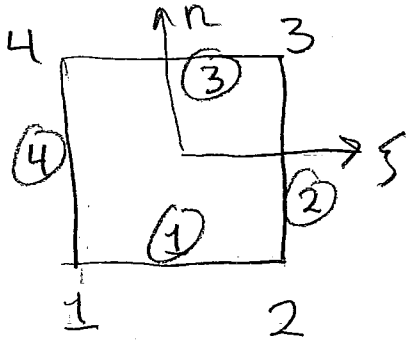
Not obvious – but you can check $\phi(-1) = \phi(1) = 0$

The $\sqrt{\frac{2i-1}{2}}$ is a normalization such that

$$\int_{-1}^{1} \phi_{i,\xi}\, \phi_{j,\xi}\, d\xi = \delta_{ij}$$ (note the orthogonality)

The basic ideas can be extended to 2-D and 3-D

Consider a quadrilateral elements



For the base linear element use our standard linear shape functions

$$N_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$$

Then much like Serendipity elements we will add modes to the sides.

Side modes: The are $4(p-1)$ side mode shape functions for $p \geq 2$

For side 1:

$$N_i^{(1)} = \frac{1}{2}(1-\eta)\,\phi_i(\xi) \qquad i = 2(1)p$$

For side 2:

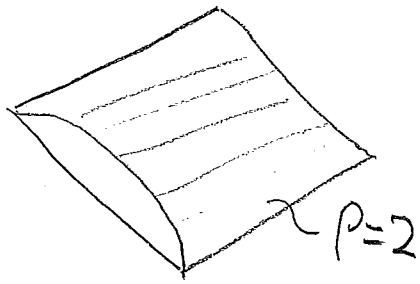
$$N_i^{(2)} = \frac{1}{2}(1+\xi)\,\phi_i(\eta) \qquad i = 2(1)p$$

For side 3:

$$N_i^{③} = \frac{1}{2}(1+\eta)\,\phi_i(\xi) \qquad i = 2(1)p$$

For side 4:

$$N_i^{④} = \frac{1}{2}(1-\xi)\,\phi_i(\eta) \qquad i = 2(1)p$$



Note that the dof a_i's associated with the side modes are not values of u at specific points - They are simple dof that are shared (with corrections for signs for odd-order terms) between elements - This gives C^0 continuity with no corrections needed.

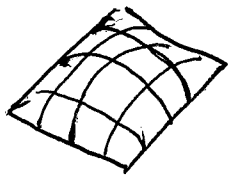
Recall that with doing these side modes like this (linear in the other direction) we never get

$$\xi^2\eta^2, \quad \xi^3\eta^2, \quad \xi^2\eta^3, \quad \xi^4\eta^2, \ldots \text{ etc.}$$

to get these we also need face modes

Face modes

There are $\frac{1}{2}(p-2)(p-3)$ face modes for $p \geq 4$

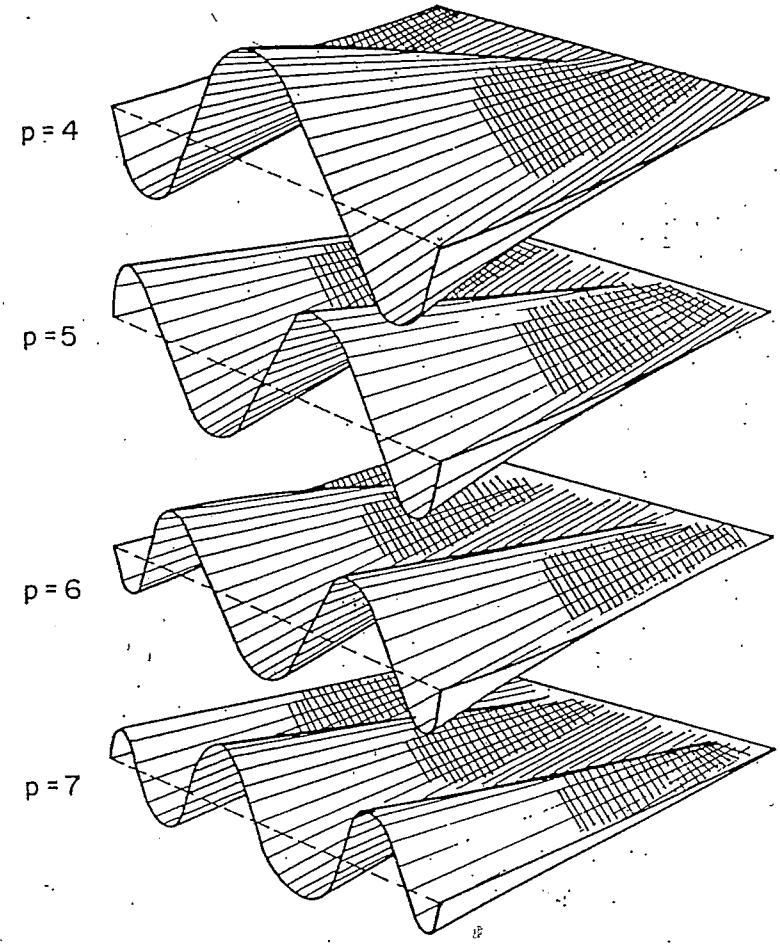


The first one is:

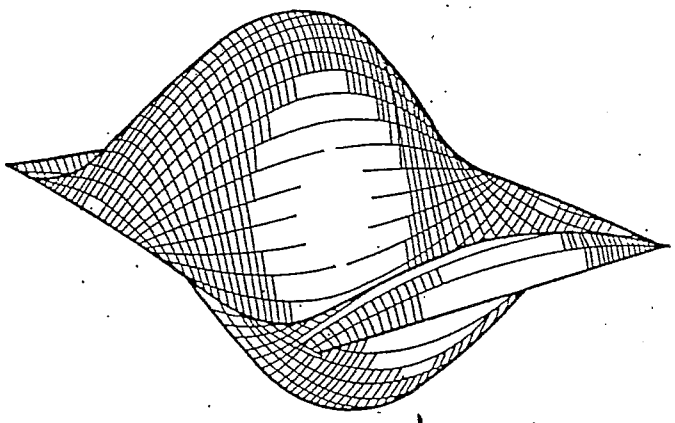
$$N_1^{(0)}(\xi,\eta) = (1-\xi^2)(1-\eta^2)$$

From there the higher modes are obtained by multiplying by Legendre polynomials

$$N_2^{(0)} = N_1^{(0)}(\xi,\eta)\, P_1(\xi)$$
$$N_3^{(0)} = N_1^{(0)}(\xi,\eta)\, P_1(\eta)$$
$$N_4^{(0)} = N_1^{(0)}(\xi,\eta)\, P_2(\xi)$$
$$N_5^{(0)} = N_1^{(0)}(\xi,\eta)\, P_1(\xi)\, P_1(\eta)$$
$$\vdots$$



Edge modes



Face mode

GEOMETRIC MAPPING

GOAL: Evaluate $\underline{x}(\underline{\xi})$ and $\frac{\partial^n \underline{x}}{\partial \underline{\xi}^n}$ "exactly", *n* up to 3 for some "shell" formulations

Approaches:

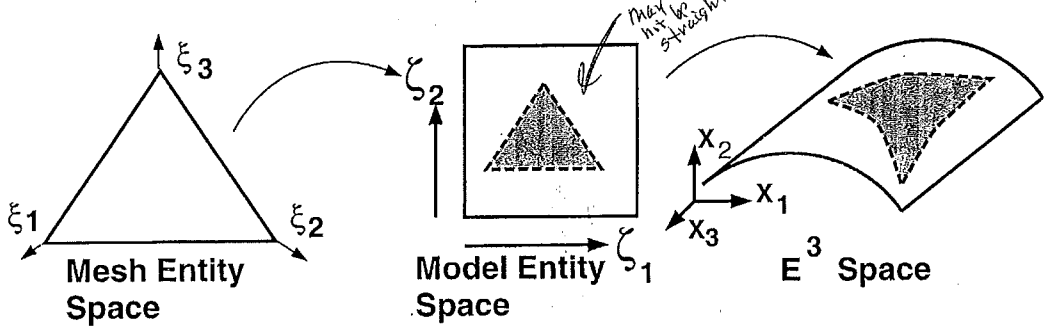
- *isoparametric*: difficult to control shape with hierarchic shape functions, inaccurate derivatives
- "fitting" approximate geometry: expensive, inaccurate derivatives
- CAD geometry: "exact" position and derivatives, must obtain needed geometry data from modeler

Geometric modelers provide $\underline{x}(\underline{\zeta})$ and $\frac{\partial^n \underline{x}}{\partial \underline{\zeta}^n}$ on a point-wise basis for curve/surface geometries

APPROACH FOR MAPPING MESH ENTITIES ON MODEL BOUNDARY

Available (from geometric modeler): $\underline{x} = \underline{x}(\underline{\zeta})$, $\frac{\partial \underline{x}}{\partial \underline{\zeta}}$, $\frac{\partial^2 \underline{x}}{\partial \underline{\zeta}^2}$, $\frac{\partial^3 \underline{x}}{\partial \underline{\zeta}^3}$

Must construct : $\underline{\zeta} = \underline{\zeta}(\underline{\xi})$ such that $\underline{x} = \underline{x}(\underline{\zeta}(\underline{\xi}))$

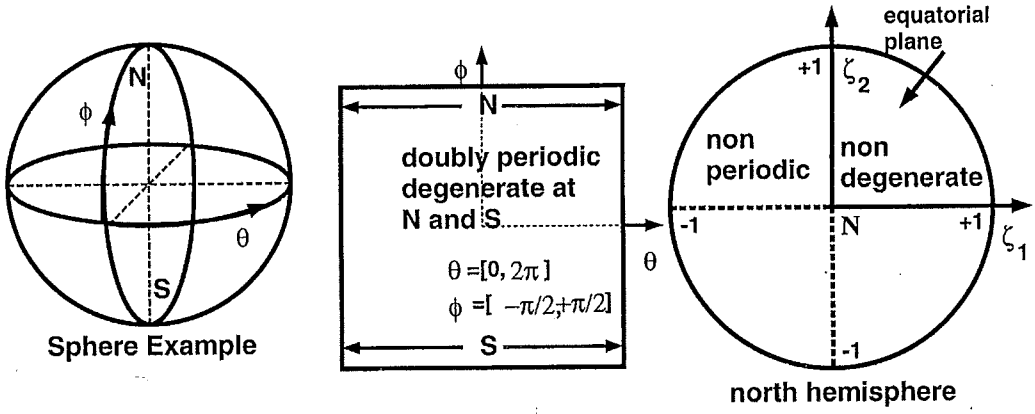


$\underline{\zeta}(\underline{\xi})$ constructed based on type of model parametric space

Derivatives through chain-rule

$$\frac{\partial()}{\partial \xi} = \left(\frac{\partial()}{\partial \zeta}\right)\frac{\partial \zeta}{\partial \xi}$$

$\underline{x}(\zeta)$ must not have degenerate points ($\frac{\partial x}{\partial \zeta}$ is undefined)
Can be solved by proper choice of $\underline{\zeta}$



Evaluating $\mathbf{x}(\xi)$

$$\mathbf{x} = \begin{bmatrix} x_1(\zeta(\xi)) \\ x_2(\zeta(\xi)) \\ x_3(\zeta(\xi)) \end{bmatrix}$$

allows us to precalculate $N(\xi)$

must find way to construct this

Evaluating $\frac{\partial \mathbf{x}}{\partial \xi}$

To evaluate these need - $\frac{\partial \zeta}{\partial \xi}^n$ and $\frac{\partial x}{\partial \xi}^n$ ← modeler provides

$$\frac{\partial x_i}{\partial \xi_j} = \left(\frac{\partial x_i}{\partial \zeta_k}\right)\left(\frac{\partial \zeta_k}{\partial \xi_j}\right) = \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial \zeta_1} & \frac{\partial x_1}{\partial \zeta_2} \\ \frac{\partial x_2}{\partial \zeta_1} & \frac{\partial x_2}{\partial \zeta_2} \\ \frac{\partial x_3}{\partial \zeta_1} & \frac{\partial x_3}{\partial \zeta_2} \end{bmatrix}}_{from\ modeler} \overbrace{\begin{bmatrix} \frac{\partial \zeta_1}{\partial \xi_1} & \frac{\partial \zeta_1}{\partial \xi_2} & \frac{\partial \zeta_1}{\partial \xi_3} \\ \frac{\partial \zeta_2}{\partial \xi_1} & \frac{\partial \zeta_2}{\partial \xi_2} & \frac{\partial \zeta_2}{\partial \xi_3} \end{bmatrix}}^{[\mathbf{J}]}$$

Evaluating $\frac{\partial^2 \mathbf{x}}{\partial \xi^2}$

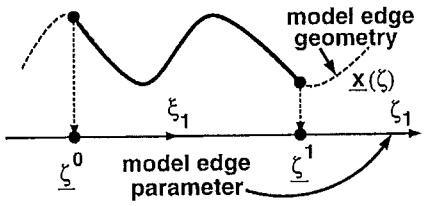
$$\frac{\partial^2 \mathbf{x}_i}{\partial \xi_j \xi_k} = \left(\frac{\partial \zeta_l}{\partial \xi_j}\right)\overbrace{\left(\frac{\partial^2 \mathbf{x}_i}{\partial \zeta_l \zeta_n}\right)}^{modeler}\left(\frac{\partial \zeta_n}{\partial \xi_k}\right) + \overbrace{\left(\frac{\partial \mathbf{x}_i}{\partial \zeta_l}\right)}^{modeler}\left(\frac{\partial^2 \zeta_l}{\partial \xi_j \xi_k}\right)$$

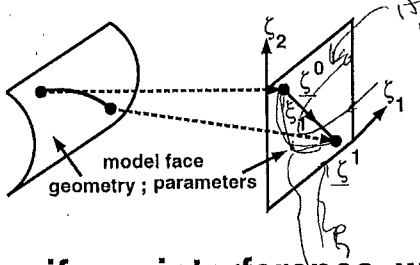
CONSTRUCTION OF $\underline{\zeta}(\underline{\xi})$ FOR MESH ENTITIES ON MODEL BOUNDARY

$\underline{\zeta}(\underline{\xi})$ constructed based on

- ☐ mesh entity classification
- ☐ type of underlying model parametric space

MESH EDGE CLASSIFIED ON MODEL BOUNDARY





if that were trimmed at the line would be a problem ↑ would have to define a higher order that gets it out of the way

$\underline{\zeta}(\underline{\xi})$ by linear interpolation if no interference with trimmed boundary

$$\zeta_i(\xi_1) = \zeta_i^0 + \xi_1 \zeta_i^1 \quad \text{(1-ξ1)}$$

& $\zeta_i(\zeta_1) = N_0 \zeta_{i_0} + N_{1/2} \zeta_{i_{1/2}} + N_1 \zeta_{i_1}$

MESH FACE CLASSIFIED ON MODEL BOUNDARY

Blend $\underline{\zeta}(\underline{\xi})$ from boundary entities for trimmed ← The blend allows you to employ the definition of the boundary that exists in the process

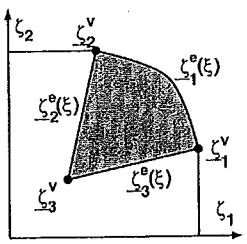
$$\underline{\zeta}(\underline{\xi}) = \left(\frac{\xi_1}{1-\xi_2}\right)\underline{\zeta}_1^e(\xi_2) + \left(\frac{\xi_3}{1-\xi_2}\right)\underline{\zeta}_2^e(1-\xi_2) + \left(\frac{\xi_2}{1-\xi_3}\right)\underline{\zeta}_2^e(\xi_3)$$
$$+ \left(\frac{\xi_1}{1-\xi_3}\right)\underline{\zeta}_3^e(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)\underline{\zeta}_3^e(\xi_1) + \left(\frac{\xi_2}{1-\xi_1}\right)\underline{\zeta}_1^e(1-\xi_1)$$
$$- \xi_1\underline{\zeta}_1^v - \xi_2\underline{\zeta}_2^v - \xi_3\underline{\zeta}_3^v$$

Triangular

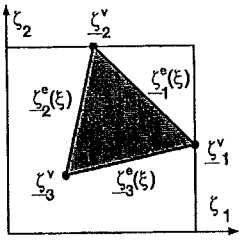
$$\underline{\zeta}(\underline{\xi}) = \sum_{i=1}^{nverts} \xi_i \underline{\zeta}_i^v \quad \text{for untrimmed model faces}$$

⇒ No geometric approximation this

Quadrilateral tensor products



Trimmed



Untrimmed

⇒ Note – do not actually have to get the functions $\zeta_i(\xi)$ from the modeler – just need to know these can be evaluated, and then to evaluate them at the integration points

MAPPING FOR INTERIOR MESH ENTITIES

Remarks:

- $\underline{x}(\underline{\zeta})$ usually unavailable for model regions
- blend from mapped $\underline{x}(\underline{\xi})$ for bounding mesh entities

INTERIOR MESH EDGE : $\underline{x}(\xi) = \underline{x}^0 + \xi_1 \underline{x}^1$

INTERIOR MESH FACE

with curved boundary

$$\begin{aligned}\underline{x}(\xi) = &\left(\frac{\xi_1}{1-\xi_2}\right)\underline{x}_1^e(\xi_2) + \left(\frac{\xi_3}{1-\xi_2}\right)\underline{x}_2^e(1-\xi_2) + \left(\frac{\xi_2}{1-\xi_3}\right)\underline{x}_2^e(\xi_3) + \\ &\left(\frac{\xi_1}{1-\xi_3}\right)\underline{x}_3^e(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)\underline{x}_3^e(\xi_1) + \left(\frac{\xi_2}{1-\xi_1}\right)\underline{x}_1^e(1-\xi_1) \\ &- \xi_1\underline{x}_1^v - \xi_2\underline{x}_1^v - \xi_3\underline{x}_3^v\end{aligned}$$

without curved boundary $\underline{x}(\underline{\xi}) = \sum_{i=1}^{3} \xi_i \underline{x}_i^v$

MAPPING FOR VOLUME MESH ENTITIES

Blend based on mapped $\underline{x}(\xi)$ of bounding mesh entities

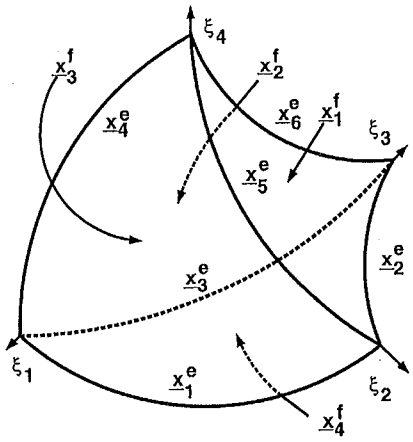
$$\begin{aligned}\underline{x}(\underline{\xi}) = &(1-\xi_1)\underline{x}_1^{\mathrm{f}}\left(\underline{\xi}'\right) + (1-\xi_2)\underline{x}_2^{\mathrm{f}}\left(\underline{\xi}'\right) \\ &+ (1-\xi_3)\underline{x}_3^{\mathrm{f}}\left(\underline{\xi}'\right) + (1-\xi_4)\underline{x}_4^{\mathrm{f}}\left(\underline{\xi}'\right) \\ &- (1-\xi_1-\xi_2)\underline{x}_6^{\mathrm{e}} - (1-\xi_1-\xi_3)\underline{x}_5^{\mathrm{e}} \\ &- (1-\xi_1-\xi_4)\underline{x}_2^{\mathrm{e}} - (1-\xi_2-\xi_3)\underline{x}_4^{\mathrm{e}} \\ &- (1-\xi_2-\xi_4)\underline{x}_3^{\mathrm{e}} - (1-\xi_3-\xi_4)\underline{x}_1^{\mathrm{e}}\end{aligned}$$

$\underline{\xi}'$ **is normalized face(edge) parameters**

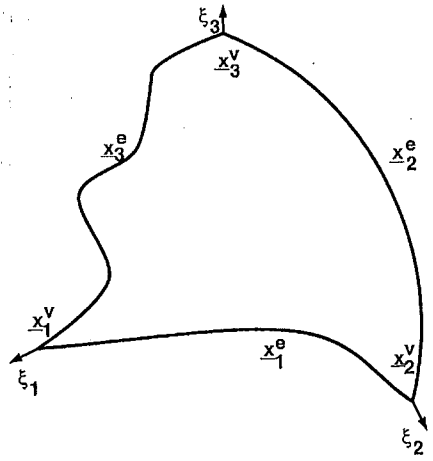
$$\sum_{i=1}^{3} \xi_i' = 1 \quad on\ face$$

$$\sum_{i=1}^{2} \xi_i' = 1 \quad on\ edge$$

BOUNDING MESH ENTITIES USED IN BLENDING



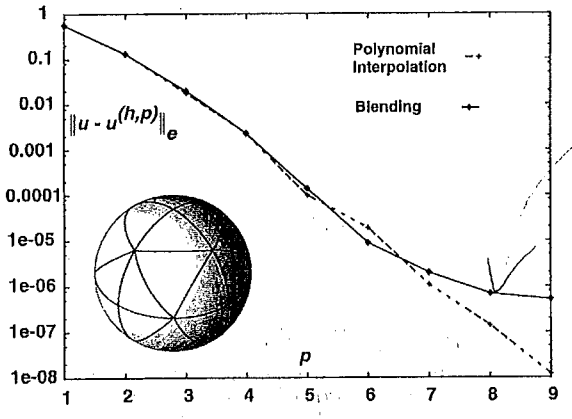
Region



Face

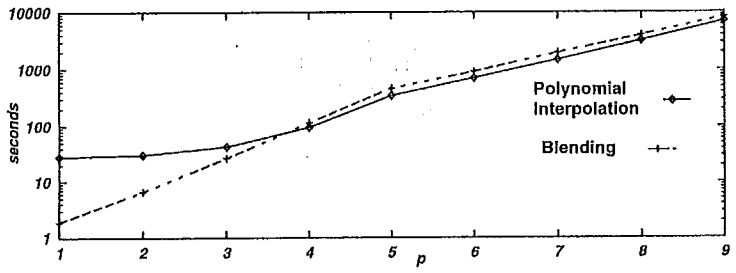
SMOOTHNESS OF RATIONAL BLENDS

- linear blends on simplices are C^0
- limits *p*-convergence
- C^k blends can be constructed; expensive



***Poisson*'s problem with smooth solution**

MAPPING COST COMPARISON WITH TENSOR-PRODUCT GAUSSIAN INTEGRATION



Total times still dominated in this example by the operation by solver

Remarks:

- uniform *p*-enrichment, no *h*-adaptation
- one time geometry interpolation of order 8
- no caching with blended mapping
- modeler queries:
  - ☐ blending: 28% of total time
  - ☐ cached interpolation: 9% of total time

actually a reasonable difference

⇒ without solving continuity problem – blends not really better –

HIGHER ORDER BLENDING SCHEME FOR TETRAHEDRA

C^k blend based on mapped $\underline{x}(\xi)$ of bounding entities

$$\begin{aligned}\underline{x}(\underline{\xi}) = &(1-\xi_1)^{k+1}\underline{x}_1^{f}\left(\underline{\xi}'\right) + (1-\xi_2)^{k+1}\underline{x}_2^{f}\left(\underline{\xi}'\right)\\ &+ (1-\xi_3)^{k+1}\underline{x}_3^{f}\left(\underline{\xi}'\right) + (1-\xi_4)^{k+1}\underline{x}_4^{f}\left(\underline{\xi}'\right)\\ &- (1-\xi_1-\xi_2)^{k+1}\underline{x}_6^{e} - (1-\xi_1-\xi_3)^{k+1}\underline{x}_5^{e}\\ &- (1-\xi_1-\xi_4)^{k+1}\underline{x}_2^{e} - (1-\xi_2-\xi_3)^{k+1}\underline{x}_4^{e}\\ &- (1-\xi_2-\xi_4)^{k+1}\underline{x}_3^{e} - (1-\xi_3-\xi_4)^{k+1}\underline{x}_1^{e}\\ &+ \xi_1^{k+1}\underline{x}_1^{v} + \xi_2^{k+1}\underline{x}_2^{v} + \xi_3^{k+1}\underline{x}_3^{v} + \xi_4^{k+1}\underline{x}_4^{v}\end{aligned}$$

$\underline{\xi}'$ is normalized face (edge) parameters

$$\sum_{i=1}^{3}\xi_i' = 1 \quad on\ face; \qquad \sum_{i=1}^{2}\xi_i' = 1 \quad on\ edge$$

EFFICIENT ELEMENT LEVEL COMPUTATIONS FOR HIGH p

Examples in PDE coordinate system:

$$\int_{\Omega^e} \left(\frac{\partial N_i}{\partial \mathbf{x}}\right) \kappa(\mathbf{x}) \left(\frac{\partial N_j}{\partial \mathbf{x}}\right)^T d\Omega; \int_{\Omega^e} N_i \rho(\mathbf{x}) N_j d\Omega; \int_{\Gamma^e} N_i h(\mathbf{x}) d\Gamma$$

Abstract parametric form: $I = \int \Theta(\xi)\Upsilon(\xi)d\xi$ ← For any of the integrals

$\Theta(\xi)$: **polynomial (exactly integrable) from** $N(\xi), \frac{\partial N}{\partial \xi}$

$\Upsilon(\xi)$: **from** $\mathrm{x}(\xi), \frac{\partial \xi}{\partial \mathrm{x}}$, **not exactly integrable**

Options:

1. **Gaussian Integration: expensive** ← on simplices with high p ← no good rules available.
2. **Precomputed tables with approximation of** $\Upsilon(\xi)$
3. **Sum-Factorization**
4. **Vector-Quadrature**

OPTIMAL FUNCTION APPROXIMATIONS

Degree q polynomial interpolation:

$$\kappa^* = \sum_{k=1}^{n_q} a_k \phi_k(\xi),\ \mathrm{x}^* = \sum_{k=1}^{n_q} b_k \phi_k(\xi)$$

Integral Error (Approximation theory):

$$\left\| \int \theta d\xi - \int \theta^* d\xi \right\|_{H^m} \leq C_\kappa h^{q+2-m} \|\theta\|_{q+1}, \quad \theta = \kappa, \mathrm{x}$$

Finite Element Error (2nd ord. elliptic BVP):

$$\left\| u - u^{(h,p)} \right\|_{H^m} \leq \frac{C h^{p+1-m} \|u\|_r}{p^{r-1}},\ r \geq p+1$$

Optimality:

$$q + 2 - m \geq p + 1 - m \ \Rightarrow q \geq p - 1$$

PRECOMPUTED INTEGRAL ENTRIES

$\Upsilon(\xi)$ approximated as degree q=p-1 polynomial interpolation

$$\Upsilon(\xi) \approx \Upsilon^*(\xi) = \sum_k^{n_q} c_k \phi_k(\xi) \quad \Rightarrow$$

$$I = \int \Theta(\xi)\Upsilon(\xi)d\xi \approx \sum_k^{n_q} c_k \underbrace{\int \Theta(\xi)\phi_k(\xi)d\xi}_{\varrho, precomputed} = \sum_k^{n_q} c_k \varrho_k$$

Cost (general curved domains):

- $O(p^3)$ $\Upsilon(\xi)$ evaluations
- $O(p^3)$ arithmetic operations
- no evaluation of $\Theta(\xi)$ at runtime
- memory- must store ϱ_k compactly

SUM-FACTORIZATION (3D EXAMPLE)

Choose Tensor product basis

$$N(\zeta) = \Psi_1(\zeta_1)\Psi_2(\zeta_2)\Psi_3(\zeta_3);\ \zeta_i \in [-1,1]$$

Reduce integrand to tensor product form

$$\int_{\Omega/\Gamma} \Theta(\xi)\Upsilon(\xi)d\xi \approx \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \sum_k a_k \Psi_1^{(k)}(\zeta_1)\Psi_2^{(k)}(\zeta_2)\Psi_3^{(k)}(\zeta_3)d\zeta_1 d\zeta_2 d\zeta_3$$

$$= \sum_k a_k \int_{-1}^{1} \Psi_1^{(k)}(\zeta_1)d\zeta_1 \int_{-1}^{1} \Psi_2^{(k)}(\zeta_2)d\zeta_2 \int_{-1}^{1} \Psi_3^{(k)}(\zeta_3)d\zeta_3$$

Cost (general curved domains):

- $O(p^3)$ integrand evaluations
- $O(p^3)$ arithmetic operations
- $\Upsilon(\xi)$ must be approximated for curved domains

VECTOR-QUADRATURE

Given orthogonal basis $\int_\Omega \varphi_i \varphi_j = \delta_{ij}$, approximate

$$\Theta(\xi) \approx \sum_{i=1}^{M} a_i \varphi_i; \quad \Upsilon(\xi) \approx \sum_{i=1}^{M} b_i \varphi_i$$

yielding

$$I = \int \Theta(\xi)\Upsilon(\xi) d\Omega \approx \mathbf{a} \odot \mathbf{b} = \sum_{i=1}^{M} a_i b_i$$

M based on maximum of Θ, Υ degree.
Cost (general curved domains):

- $O(p^3)$ integrand evaluations
- $O(p^3)$ arithmetic operations

Symmetric order decomposition of I is more efficient

INTEGRATION OPTIONS

- asymptotically all options cost $O(p^3)$ for curved domains
- all options must approximate part(s) of the integrand
- sum-factorization can be implemented using precomputed entries

Integrand approximation + precomputed integrals efficient for general curved domains:

- derive precomputed entries for elemental integrals
- identify symmetries in precomputed entries

PRECOMPUTED STIFFNESS INTEGRALS

General form (sum on repeated indices k, l, m):

$$K_{ij}^e = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x_k}\right) c(\mathrm{x}) \left(\frac{\partial N_j}{\partial x_k}\right) d\Omega$$
$$= \int_{\Omega^e} \frac{\partial N_i}{\partial \xi_l} \frac{\partial N_j}{\partial \xi_m} \underbrace{\left\{ c(\mathrm{x}(\xi)) \frac{\partial \xi_l}{\partial x_k} \frac{\partial \xi_m}{\partial x_k} J(\xi) \right\}}_{A_{lm}} d\xi, \; k, l, m = 1, \cdots, 3$$

$A_{lm}(\xi)$ interpolation (no sum on repeated n):

$$A_{lm}(\xi) = \frac{\partial \xi_l}{\partial x_k} \frac{\partial \xi_m}{\partial x_k} c(\mathrm{x}(\xi)) J(\xi) \approx \sum_{n=1}^{n_q} a_{lmn} N_n(\xi)$$

Precomputed entries: $\varrho_{ijlmn}^K = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial \xi_l} \frac{\partial N_j}{\partial \xi_m} N_n\right) d\xi$

Runtime evaluation: $K_{ij}^e = \sum_{n=1}^{n_q} a_{lmn} \varrho_{ijlmn}^K$

PRECOMPUTED MASS INTEGRALS

General form:

$$M_{ij}^e = \int_{\Omega^e} \rho(\mathrm{x}) N_i N_j d\Omega$$
$$= \int_{\Omega^e} N_i N_j \underbrace{\{\rho(\mathrm{x}) J(\xi)\}}_{B(\xi)} d\xi$$

$B(\xi)$ approximation:

$$B(\xi) = \rho(\mathrm{x}(\xi)) J(\xi) \approx \sum_{n=1}^{n_q} b_n N_n(\xi)$$

Precomputed entries: $\varrho_{ijn}^M = \int_{\Omega^e} (N_i N_j N_n) d\xi$

Runtime evaluation: $M_{ij}^e = \sum_{n=1}^{n_q} b_n \varrho_{ijn}^M$

PRECOMPUTED RHS INTEGRALS FROM Ω^e

General form:

$$f_i^e = \int_{\Omega^e} f(\mathrm{x}) N_i d\Omega$$
$$= \int_{\Omega^e} N_i \underbrace{\{f(\mathrm{x})J(\xi)\}}_{D(\xi)} d\xi$$

$D(\xi)$ approximation:

$$D(\xi) = f(\mathrm{x}(\xi))J(\xi) \approx \sum_{n=1}^{n_q} d_n N_n(\xi)$$

Precomputed entries: $\varrho_{in}^{f^\Omega} = \int_{\Omega^e} (N_i N_n) d\xi$

Runtime evaluation: $f_i^e = \sum_{n=1}^{n_q} d_n \varrho_{in}^{f^\Omega}$

PRECOMPUTED RHS INTEGRALS FROM Γ^e

General form:

$$f_i^e = \int_{\Gamma^e} h(\mathrm{x}) N_i d\Gamma$$
$$= \int_{\Gamma^e} N_i \underbrace{\{h(\mathrm{x})J(\xi)\}}_{E(\xi)} d\xi$$

$E(\xi)$ approximation:

$$E(\xi) = h(\mathrm{x}(\xi))J(\xi) \approx \sum_{n=1}^{n_q} e_n N_n(\xi)$$

Precomputed entries: $\varrho_{in}^{f^\Gamma} = \int_{\Gamma^e} (N_i N_n) d\xi$

Runtime evaluation: $f_i^e = \sum_{n=1}^{n_q} e_n \varrho_{in}^{f^\Gamma}$

ASYMPTOTIC TABLE SIZES

General form:

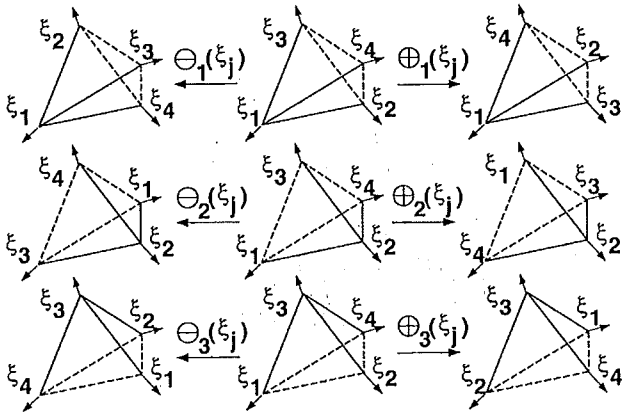
$$\int_{\Omega^e} \underbrace{\left(\frac{\partial^{n_i} N_i}{\partial \xi^{n_i}}\right)\left(\frac{\partial^{n_j} N_j}{\partial \xi^{n_j}}\right)\cdots\left(\frac{\partial^{n_k} N_k}{\partial \xi^{n_k}}\right)}_{M\ terms} N_l d\Omega$$

$$\int_{\Gamma^e} \underbrace{\left(\frac{\partial^{n_i} N_i}{\partial \xi^{n_i}}\right)\left(\frac{\partial^{n_j} N_j}{\partial \xi^{n_j}}\right)\cdots\left(\frac{\partial^{n_k} N_k}{\partial \xi^{n_k}}\right)}_{M\ terms} N_l d\Gamma,\ n_\alpha \geq 0, \alpha = 1, \cdots,$$

Size:

p	K^e, M^e		f^e	
	Entries (Million)	Storage (MB)	Entries (Million)	Storage (MB)
4	0.2	2	4e-3	0.03
5	1.9	15	0.01	0.08
6	10	80	0.05	0.40
7	40	320	0.11	0.88
8	134	1072	0.26	2

COORDINATE PERMUTATION TO IDENTIFY SYMMETRIES



$$N_i = -2\xi_1\xi_2,\ \oplus_1(N_i) = -2\xi_1\xi_3,\ \ominus_1(N_i) = -2\xi_1\xi_4$$

$$\oplus_i\left(\frac{\partial N_i}{\partial \xi_j}\right) = \frac{\partial[\oplus_i(N_i)]}{\partial[\oplus_i(\xi_j)]};\ \ominus_i\left(\frac{\partial N_i}{\partial \xi_j}\right) = \frac{\partial[\ominus_i(N_i)]}{\partial[\ominus_i(\xi_j)]}$$

PRECOMPUTED SYMMETRIES USED

Stiffness:

$$\varrho^K_{ij13n} = \varrho^K_{i\ominus_3 j\ominus_3 21n^{\ominus_3}} \qquad \varrho^K_{ij22n} = \varrho^K_{i\oplus_3 j\oplus_3 11n^{\oplus_3}} \qquad \varrho^K_{ij23n} = \varrho^K_{i\oplus_3 j\oplus_3 11n^{\oplus_3}}$$
$$\varrho^K_{ij31n} = \varrho^K_{i\ominus_3 j\ominus_3 12n^{\ominus_3}} \qquad \varrho^K_{ij32n} = \varrho^K_{i\ominus_3 j\ominus_3 21n^{\ominus_3}} \qquad \varrho^K_{ij33n} = \varrho^K_{i\ominus_3 j\ominus_3 11n^{\ominus_3}}$$
$$\varrho^K_{ij11n} = \varrho^K_{ji11n} \qquad \varrho^K_{ij12n} = \varrho^K_{ji12n} \qquad \varrho^K_{ij21n} = \varrho^K_{ji21n}$$

Mass: $\varrho^M_{ijn} = \varrho^M_{jin}$; **RHS:** $\varrho^f_{in} = \varrho^f_{ni}$

Example *p*=6:

	K^e		M^e		f^e	
	Asymp.	Comp.	Asymp.	Comp.	Asymp.	Comp.
Entries	~1e7	45839	~1e7	3659	~5e4	451
Size(KB)	~8e4	367	~8e4	30	~4e2	4

Additional symmetries may exist.

80,000 ~~1 trillion~~ 2 orders of mag. ↑ was 10 mil

NUMERICAL EXAMPLES

Model problem:

$$-\triangle u(\mathrm{x}) = f(\mathrm{x}),\ \mathrm{x} \in \Omega$$
$$u(\mathrm{x}) = 0,\ \mathrm{x} \in \partial\Omega$$

Example 1: $u = (1-x_1^2)(1-x_2^2)(1-x_3^2)$, **(Ω=cube)**

Example 2: $u = 100r(1-r)\sin(\theta)\cos(\phi)$, **(Ω=sphere)**

Timing legend:

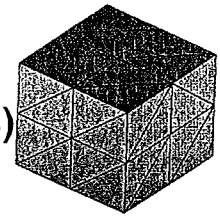
Ge: elemental computation- Gaussian Integration

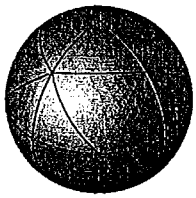
Pe: elemental computation- Precomputed entries

NUMERICAL EXAMPLES

Example 1
(48 elements)

Ge/Pe = 10





Example 2
(64 elements)

Ge/Pe = 5

Remarks:

- Relative energy error
  - same rate of convergence
  - Gauss error lower by 2 orders of magnitude in example 2 ← Note the p-1 ensures rate not value of error !!
- Needs further investigation with higher accuracy interpolation points

CLOSING REMARKS

Geometry-based issues must be carefully considered for the effective implementation of *hp*-version finite elements - can easily lose high rate of convergence

Geometry issues must be accounted for in
- mesh generation
- element mappings
- element matrix integrations

Capabilities for the automatic generation of *p*-type meshes for general 3-D geometries developed
Approaches to deal with the geometric mappings available - be careful in using rational blends on simplixes
Integrations expensive for curved elements - compact look-up tables cut cost by nearly an order of magnitude

Shape Functions Decomposition Based on the Topological Entities of $\overline{\Omega}_e$

Each topological entity has an independent specification of polynomial order. The mesh entities associated with an element are

$$\overline{\Omega}_e = \left\{ M_e^{d_e}, \partial\left(M_e^{d_e}\right) \right\} = \left\{ M_e^{d_e}, M_e^{d_e}\left\{ M_j^{d_e-1} \right\}, \ldots, M_e^{d_e}\left\{ M_j^0 \right\} \right\} \tag{1}$$

A convenient method to construct the shape functions associated with a mesh entity $M_j^{dj} \in \overline{\Omega}_e$ is

$$N_a = \psi\left(M_j^{d_j}, M_e^{d_e}\right)\phi\left(M_j^{d_j}\right) \tag{2}$$

where:

$\psi\left(M_j^{d_j}, M_e^{d_e}\right)$ is a blending function defined on the mesh entity $M_e^{d_e}$ written in its parametric coordinate system, ξ_i, which is specific to the mesh entity, $M_j^{d_j}$ (but independent of the polynomial order of the shape function)

$\phi\left(M_j^{d_j}\right)$ is a function written in the mesh entity $M_j^{d_j}$ parametric coordinate system, $\hat{\xi}_j$, is a function of the polynomial order of the desired shape function. It is independent of $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$, and thus $M_e^{d_e}$, meaning it is the same for all elements for which $M_j^{d_j}$ is part (including elements of different dimension).

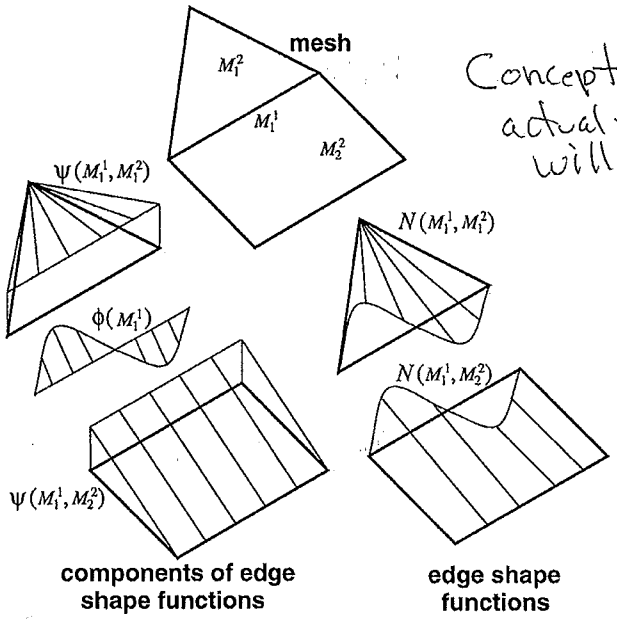


Figure 1. Construction of edge shape functions using product of blending function times edge mode.

Our focus is the definition of the proper set of shape functions over each element which:

1. Are of the polynomial order specified on each of the mesh entities included in the closure of the element, $\overline{\Omega}_e$.
2. Satisfy the C^0 interelement continuity requirement.
3. Require a minimum of numerical operations in the evaluation of the shape functions and subsequent element integrations.

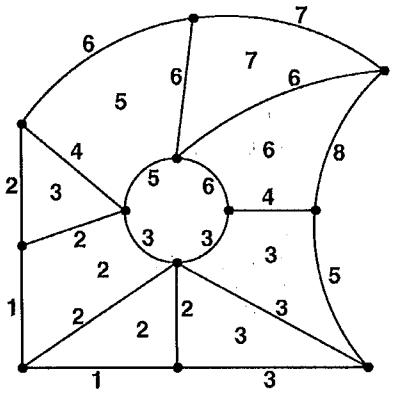


Figure 2. Example Two-dimensional variable p-order mesh.

It is possible for the face and region entities to have different p-orders specified in each of the parametric directions of the face or region.

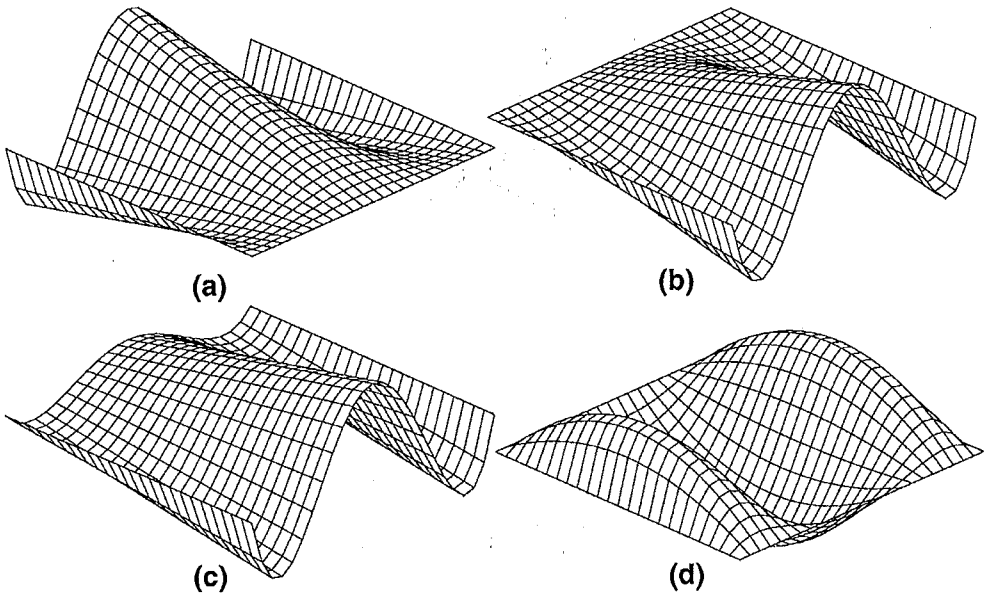


Figure 3. Directionally biased solution fields.

Parametric Coordinate Systems

The choice of parametric coordinates for the mesh entities is based on:

1. Simplicity and efficiency of the blending functions, ψ.
2. Compatibility and efficiency of use with numerical integration scheme(s).

Table 1. Entity parametric domains.

Topology		Parametric Domain
Edge	I	$\xi_1 \in [-1,1]$
	II	$\xi_1, \xi_2 \in [0,1],\ \xi_1 + \xi_2 = 1$
Triangle		$\xi_1, \xi_2, \xi_3 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 = 1$
Quadrilateral		$\xi_1, \xi_2 \in [-1,1]$
Tetrahedron		$\xi_1, \xi_2, \xi_3, \xi_4 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$
Hexahedron		$\xi_1, \xi_2, \xi_3 \in [-1,1]$
Pentahedron		$\xi_1, \xi_2, \xi_3 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 = 1,\ \xi_4 \in [-1,1]$

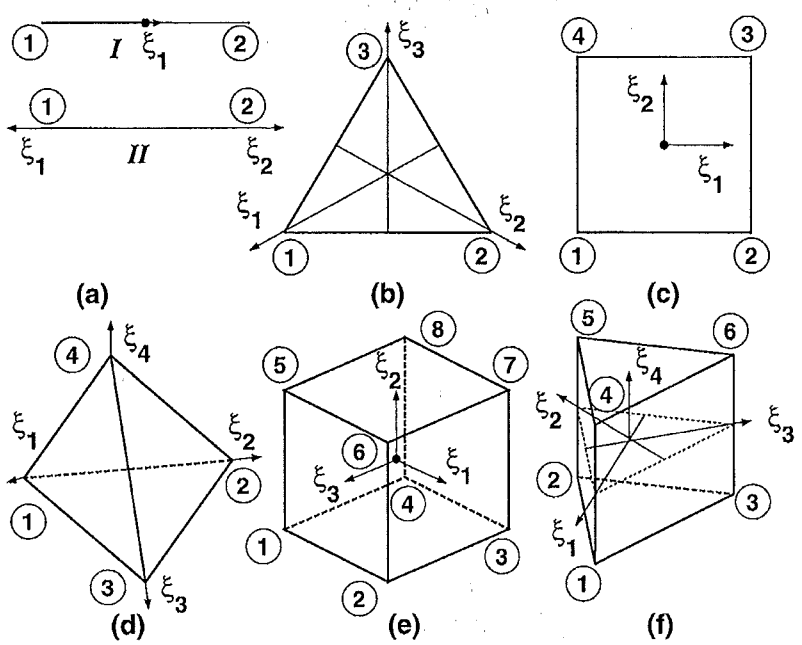


Figure 4. Entity parametric coordinates: (a) edge, (b) triangle, (c) quadrilateral, (d) tetrahedron, (e) hexahedron and (e) pentahedron.

Mappings needed to transform the element coordinates, ξ_i, to coordinates of the bounding lower order entities, $\hat{\xi}_j$. "^" used to distinguish the coordinates of the bounding lower order entities. Efficiency of mapping is dependents on the choice of possible edge parameterization.

Edge parametrization *I* leads to a trivial mapping for quadrilateral and hexahedral elements, $\hat{\xi}_1 \equiv \xi_j$. For a simplex, the mapping for an edge between vertices with index *i* and *j* (Figure 5) is

$$\hat{\xi}_1 = \xi_j - \xi_i. \tag{3}$$

Edge parameterization *II* leads to trivial mapping for simplices given the local edge index, $\hat{\xi}_i \equiv \xi_j$. For a quadrilateral or a hexahedral element (Figure 6(a,b)) the mapping is

$$\hat{\xi}_1 = \frac{1}{2}(1 - \xi_i)\,, \quad \hat{\xi}_2 = \frac{1}{2}(1 + \xi_i) \tag{4}$$

for edge directed along ξ_i parameter of the element.

For a quadrilateral face of a pentahedron as shown in Figure 6(c), the mapping between the element parametric coordinates, ξ, to the face parametric coordinates, $\hat{\xi}$, is given by

$$\hat{\xi}_1 = \xi_j - \xi_i\,, \quad \hat{\xi}_2 = \xi_4 \tag{5}$$

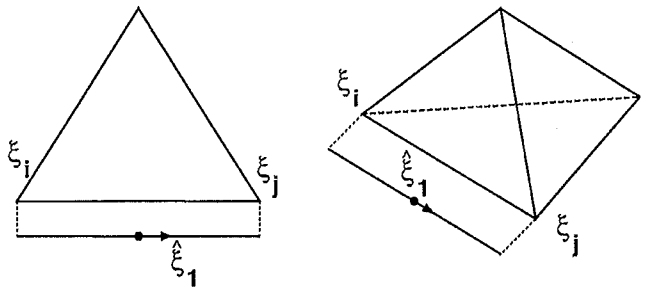


Figure 5. Parametric transformations with edge parametrization *I*.

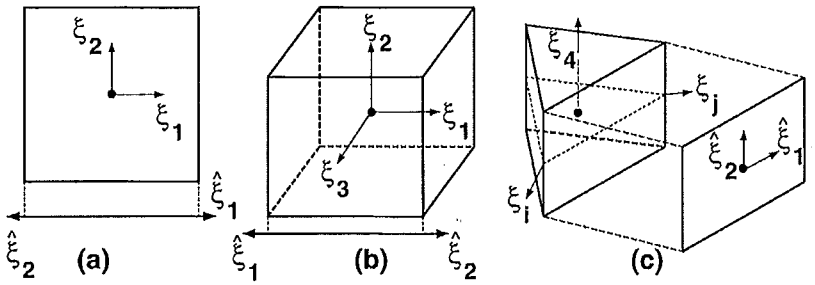


Figure 6. Parametric transformations with edge parametrization *II*.

Decomposed Shape Functions

Vertex shape functions are standard linears.

Construction of the Blending Functions ψ

The constant edge blend over a triangle is $\psi\left(M_1^1, M_1^2\right) = \frac{4\xi_i\xi_j}{1-\left(\xi_j-\xi_i\right)^2}$, and the edge function is $\phi\left(M_1^1\right) = -\frac{\sqrt{10}}{4}\left[1-\left(\xi_j-\xi_i\right)^2\right]\left(\xi_j-\xi_i\right)$. The cost of evaluation in this case is 2 adds and 7 multiplies.

The quadratic blending function is a simple polynomial, $\psi\left(M_1^1, M_1^2\right) = -2\xi_i\xi_j$, and the edge function is $\phi\left(M_1^1\right) = \left(\xi_j-\xi_i\right)$ giving a cost of shape function evaluation of 1 add and 3 multiplies.

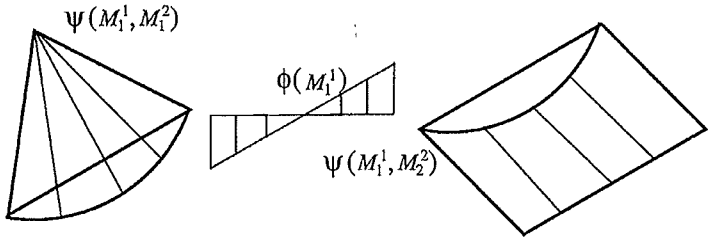


Figure 7. Alternative decomposition of cubic edge shape function using quadratic blending functions.

The blending functions, $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$, $M_j^{d_j} \in \overline{M_i^{d_e}}$, satisfy the following properties:

1. $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$ is nonzero in the interior of $M_i^{d_e}$

$$\psi\left(M_j^{d_j}, M_i^{d_e}\right) \neq 0, \quad \forall \xi \in M_i^{d_e} \tag{6}$$

2. $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$ is zero on the boundary of $\partial\left(M_i^{d_e}\right)$ except on $M_j^{d_j}$

$$\psi\left(M_j^{d_j}, M_i^{d_e}\right) = 0, \quad \forall \xi \in \left[\partial\left(M_i^{d_e}\right) - M_j^{d_j}\right] \tag{7}$$

Expressions for blending functions are based on the topology of the element and the boundary mesh entity and the parametric coordinate system of the element.

1. Line Element:
   a. Edge blend:

$$\psi\left(M_i^1, M_i^1\right) = -2\xi_1\xi_2. \tag{8}$$

2. Triangle Element:
   a. Edge blend:

$$\psi\left(M_i^1, M_j^2\right) = -2\xi_k\xi_l \tag{9}$$

   where edge M_i^1 is directed from the k-th to the l-th vertex of M_j^2.
   b. Face blend:

$$\psi\left(M_i^2, M_i^2\right) = \xi_1\xi_2\xi_3. \tag{10}$$

3. Quadrilateral Element:

a. Edge blend:

$$\psi(M_j^1, M_i^2) = \frac{1}{4}[\xi_k^2 - 1](1 \pm \xi_l) \tag{11}$$

for edges along ξ_k with $k = 1, 2;\ l = 2, 1$.

b. Face blend:

$$\psi(M_i^2, M_i^2) = \frac{1}{4}[\xi_1^2 - 1]\,[\xi_2^2 - 1]. \tag{12}$$

4. Tetrahedral Element:

a. Edge blend:

$$\psi(M_i^1, M_j^3) = -2\xi_k\xi_l \tag{13}$$

where edge M_i^1 is directed from the *k*-th to the *l*-th vertex of M_j^3.

b. Face blend:

$$\psi(M_i^2, M_j^3) = \xi_k\xi_l\xi_m \tag{14}$$

where face M_i^2 is bounded by vertices of M_j^3 with local index *k*, *l* and *m*.

5. Hexahedral Element:

a. Edge blend:

$$\psi(M_i^1, M_j^3) = \frac{1}{8}(\xi_k^2 - 1)(1 \pm \xi_l)(1 \pm \xi_m) \tag{15}$$

for edges along ξ_k with $k = 1, 2, 3;\ l = 2, 3, 1;\ m = 3, 1, 2$.

b. Face blend:

$$\psi(M_i^2, M_j^3) = \frac{1}{8}(\xi_l^2 - 1)(\xi_m^2 - 1)(1 \pm \xi_k) \tag{16}$$

for faces perpendicular to ξ_k with $k = 1, 2, 3;\ l = 2, 3, 1;\ m = 3, 1, 2$.

6. Pentahedral Element:

a. Edge blend:

$$\psi(M_i^1, M_j^3) = -\xi_k\xi_l(1 \pm \xi_4) \tag{17}$$

for edges lying in the plane perpendicular to the ξ_4 axis and defined between vertices at which $\xi_k = 1$ and $\xi_l = 1$ respectively. For edges along the ξ_4 axis and between vertices with $\xi_k = 1$

$$\psi(M_i^1, M_j^3) = \frac{1}{2}\xi_k(\xi_4^2 - 1) \tag{18}$$

b. Face blend:

$$\psi(M_i^2, M_j^3) = \frac{1}{2}\xi_1\xi_2\xi_3(1 \pm \xi_4) \tag{19}$$

for the two triangular faces.

$$\psi(M_i^2, M_j^3) = -\xi_k \xi_l (\xi_4^2 - 1) \tag{20}$$

for quadrilateral face bounded by pairs of vertices at which $\xi_k = 1$ and $\xi_l = 1$.

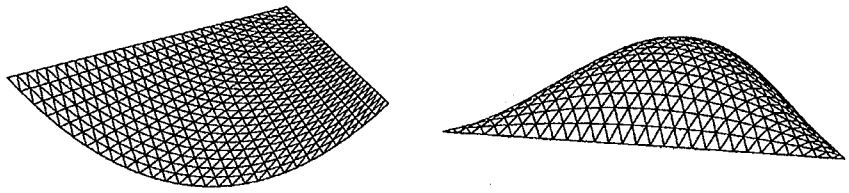


Figure 8. Triangular edge and face blending functions.

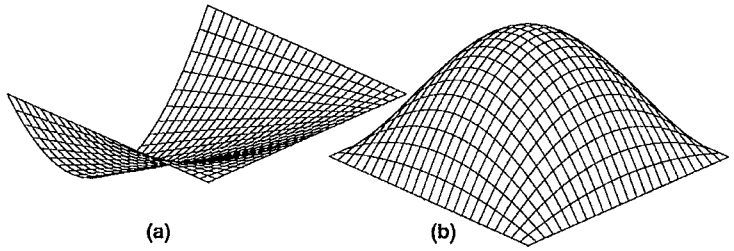


Figure 9. Quadrilateral edge and face blending functions.

Construction of the Mesh Entity Level Functions ϕ

The polynomial order of ϕ for an entity is given by $(p - q)$ where *p* is the polynomial order of the shape function and *q* is the polynomial order of the blending function used for that mesh entity.

Mesh Edge

One edge shape function of a given polynomial order *p*. For quadratic blend ψ, the edge function, $\phi(M_i^1)$, is *p-2* for shape function of polynomial order *p*. The *n*-th order Legendre polynomial, using Rodriguez formula, is given by

$$P_n(\zeta) = \left(\frac{(-1)^n}{2^n n!}\right) \frac{d^n}{d\zeta^n} (1 - \zeta^2)^n \tag{21}$$

where $\zeta \in [-1, 1]$. A useful recurrence relation is given by

$$\begin{aligned} &(n+1)P_{n+1}(\zeta) = (2n+1)P_n(\zeta) - nP_{n-1}(\zeta) \\ &P_0(\zeta) = 1, P_1(\zeta) = \zeta \end{aligned} \tag{22}$$

The edge functions based on normalized integrals of Legendre polynomials are derived from

$$\left[\frac{\zeta^2 - 1}{2}\right] * \phi(M_1^1) = \sqrt{\frac{2p-1}{2}} \int_{-1}^{\zeta} P_{p-1}(t)dt, \; p \geq 2 \tag{23}$$

where, $\zeta = \xi_1$ and $\zeta = \xi_2 - \xi_1$ for edge parameterization *I* and *II* respectively. The term $\frac{1-\zeta^2}{2}$ is the edge blend function and can always be factored out of the right hand side of equation (23).

When simplices are used in a mesh the use of the following edge functions with parameterization *II* yield better conditioning of element level matrices

$$\phi(M_1^1) = \sum_{k=0}^{p-2} (-1)^k \frac{1}{k+1} \binom{p-2}{k} \binom{p-1}{k} \xi_1^k \xi_2^{p-2-k}, \; p \geq 2 \tag{24}$$

Table 2. Edge functions.

p	ϕ	
	Equation 23	Equation 24
2	$\sqrt{\frac{3}{2}}$	1
3	$\sqrt{\frac{5}{2}}(\xi_2 - \xi_1)$	$\xi_2 - \xi_1$
4	$\sqrt{\frac{7}{2}}\left(\frac{5(\xi_2-\xi_1)^2-1}{4}\right)$	$\xi_1^2 - \xi_1\xi_2 + \xi_2^2$
5	$\sqrt{\frac{9}{2}}\left[\left(\frac{\xi_2-\xi_1}{4}\right)\left(7(\xi_2-\xi_1)^2-3\right)\right]$	$\xi_2^2 - 6\xi_1\xi_2^2 + 6\xi_1^2\xi_2 - \xi_1^2$

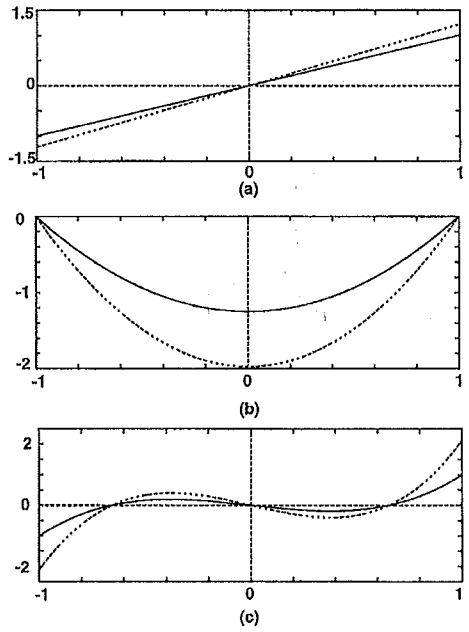


Figure 10. Plot of edge functions $\phi(M_1^1)$ for (a) *p=3*, (b) *p=4* and (c) *p=5*. The solid line is for equation (23) and the dashed lines for equation (24).

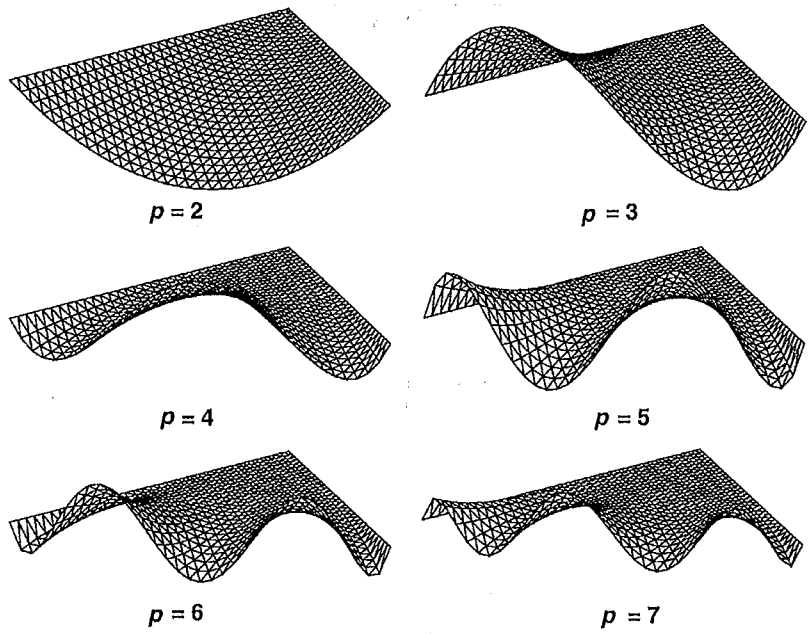


Figure 11. Triangular edge shape functions. The shapes for *p*=2 is scaled by 0.5 and those for *p*=5,6,7 are scaled by a factor of 2.0 for clarity.

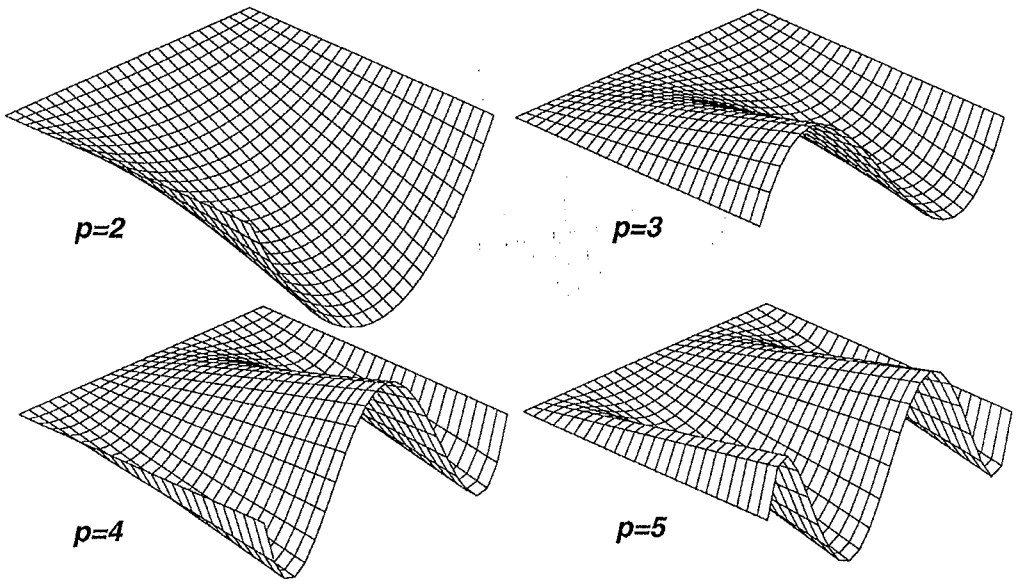


Figure 12. Quadrilateral edge shape functions.

Triangular Mesh Face

There are $(p-2)$ independent face functions for a triangular face for $p \geq 3$. For integrals of Legendre polynomials they are

$$\phi(M_i^2) = P_{\alpha_1-1}(\xi_2 - \xi_1)P_{\alpha_2-1}(2\xi_3 - 1) \tag{25}$$

where

$$\begin{aligned} &\alpha_1, \alpha_2 = 1, ..p-2 \\ &\alpha_1 + \alpha_2 + \alpha_3 = p \end{aligned} \tag{26}$$

An alternative form which yields better conditioned element matrices is

$$\begin{aligned} \phi(M_1^2) = &\sum_{i=0}^{\alpha_2-1}\sum_{j=0}^{\alpha_1-1}\left(\frac{-1}{2}\right)^{i+j} i!j!(i+j)!\binom{\alpha_1-1}{j}\binom{\alpha_1}{j}\binom{\alpha_2-1}{i}\binom{\alpha_2}{i} \\ &\times \frac{1}{\prod_{k=1}^{i+j}[k(\alpha_1+\alpha_2) - k(k-1)/2]}(\xi_1)^{\alpha_1-1-j}(\xi_2)^{\alpha_2-1-i} \end{aligned} \tag{27}$$

Symmetry and directional bias of the triangular face modes can be explained by associating the triplet, $(\alpha_1, \alpha_2, \alpha_3)$, where α_i defines the highest power of ξ_i in the face shape function. Only two of triplet parameters can be varied independently. The remaining index takes the value of 1. The sum of the triplet values equals the polynomial order of the resulting shape function.

The sets of triplet combinations based of the three possible choices of the pair of independent indices can be generated using the geometric construct shown in Figure 13. Triplets in each of the triangles ABO, ACO and BCO define the set of linearly independent face functions needed to complete a the basis of a given order. Combinations in triangle ABO, ACO and BCO correspond to choice of (ξ_1, ξ_2), (ξ_1, ξ_3) and (ξ_2, ξ_3) as the independent pair of parameter components.

Face functions generated by equation (27) correspond to the triplets in triangle ABO and are listed in Table 3. The shape functions resulting from the face functions from triangle ABO are plotted in Figure 14. Note that the missing symmetric counterpart of (2,1,1) and (1,2,1) for p=4, given by (1,1,2), can be obtained as a linear combination of (2,1,1) and (1,2,1) modes as follows

$$\left(\xi_3 - \frac{1}{3}\right) = \left(\frac{2}{3} - \xi_1 - \xi_2\right) = -\left[\left(\xi_1 - \frac{1}{3}\right) + \left(\xi_2 - \frac{1}{3}\right)\right] = -[(2,1,1) + (1,2,1)] \tag{28}$$

In those cases where the error analysis indicates the need for anisotropic p-enrichment by picking only the face shape function defined by (1,1,2), its construction by the linear combination of the (1,2,1) and (2,1,1) functions in not the most effective means to construct it; instead it can be directly obtained by choosing ξ_3 as one of the independent parameters. This is equivalent of switching from triangle ABO to triangle BCO or triangle ACO.

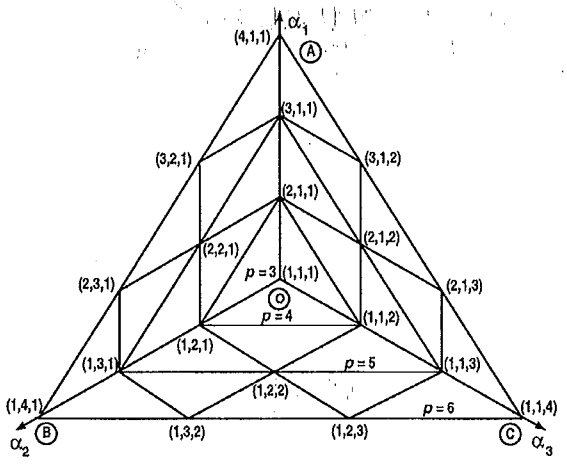


Figure 13. Face function triplet combinations.

Table 3. Triangular face functions.

p $(\alpha_1, \alpha_2, \alpha_3)$		ϕ	
		Equation 23	Equation 24
3	1,1,1	1	1
4	2,1,1	$\xi_2 - \xi_1$	$\xi_1 - \frac{1}{3}$
	1,2,1	$2\xi_3 - 1$	$\xi_2 - \frac{1}{3}$
5	3,1,1	$\frac{1}{2}\left[3(\xi_2 - \xi_2)^2 - 1\right]$	$\xi_1^2 - \frac{3}{4}\xi_1 + \frac{3}{28}$
	2,2,1	$(\xi_2 - \xi_1)(2\xi_3 - 1)$	$\xi_1\xi_2 - \frac{1}{4}(\xi_1 + \xi_2) + \frac{1}{14}$
	1,3,1	$\frac{1}{2}\left[3(2\xi_3 - 1)^2 - 1\right]$	$\xi_2^2 - \frac{3}{4}\xi_2 + \frac{3}{28}$

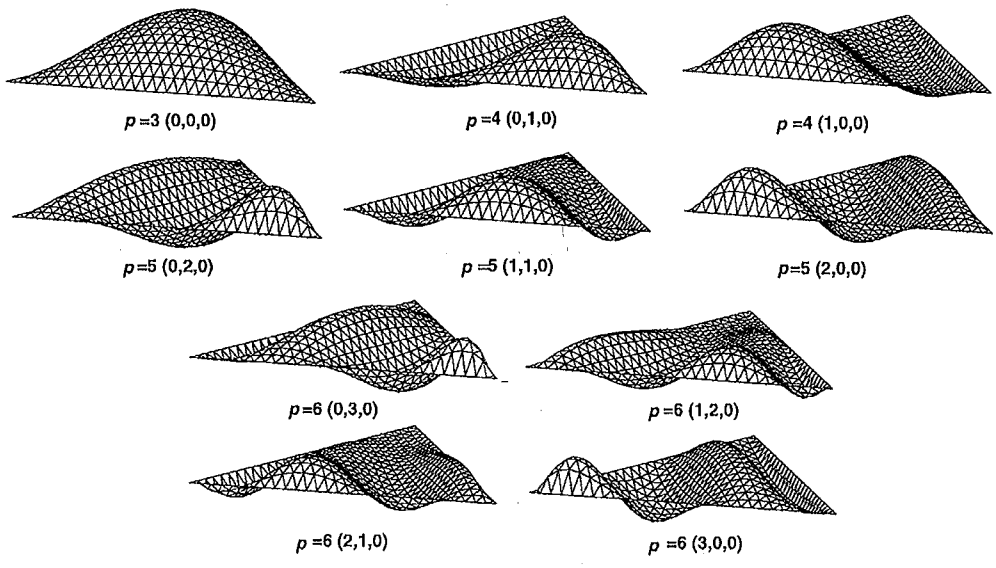


Figure 14. Triangular face shape functions. For clarity the shapes have been scaled up 5, 25, 100 and 200 times for p=3, 4, 5 and 6 respectively.

Quadrilateral Mesh Face

There are $(p-3)$ face functions for polynomial order $p \geq 4$. They are derived by

$$\frac{1}{4}\left[\xi_1^2-1\right]\left[\xi_2^2-1\right] * \phi\left(M_i^2\right) = \sqrt{\frac{2\alpha_1-1}{2}} \int_{-1}^{\xi_1} P_{\alpha_1-1}(t)dt \sqrt{\frac{2\alpha_2-1}{2}} \int_{-1}^{\xi_2} P_{\alpha_2-1}(t)dt \quad (29)$$

where $\alpha_1, \alpha_2 \geq 2$, $\quad \alpha_1 + \alpha_2 = p$
$\frac{1}{4}\left(\xi_1^2-1\right)\left(\xi_2^2-1\right)$ is the quadrilateral face blend.
(α_1, α_2) can define directional behavior where α_i is the power of ξ_i in the resulting shape function.

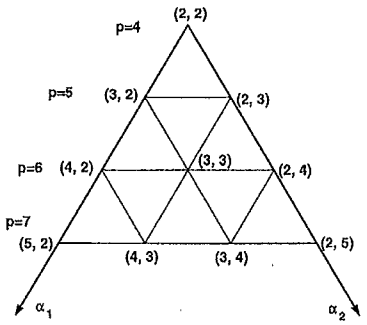


Figure 15. Construct to generate quadrilateral face modes.

Table 4. Quadrilateral face functions.

p	(α_1, α_2)	ϕ
4	2, 2	1
5	2, 3	$\frac{\sqrt{12}}{4}\xi_2$
	3, 2	$\frac{\sqrt{12}}{4}\xi_1$
6	2, 4	$\frac{\sqrt{5}}{8}\left[5\xi_2^2-1\right]$
	3, 3	$\frac{3}{2}\xi_1\xi_2$
	4, 2	$\frac{\sqrt{5}}{8}\left[5\xi_1^2-1\right]$

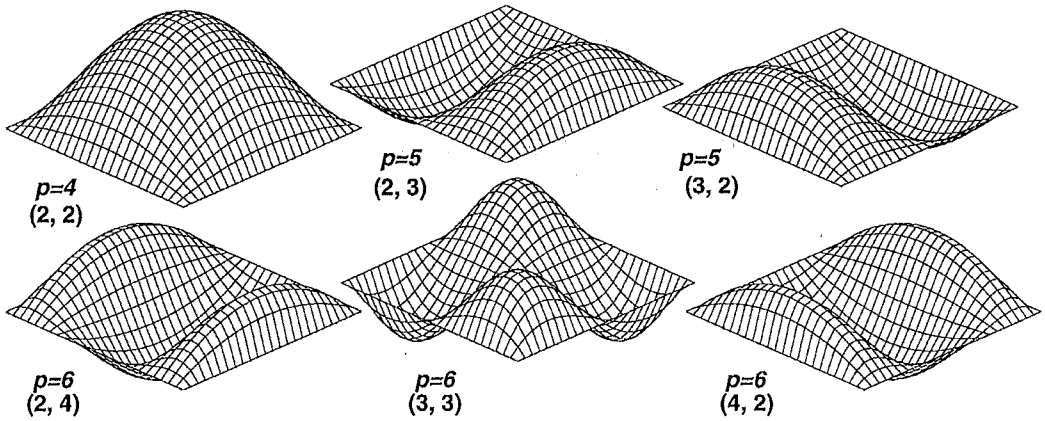


Figure 16. Quadrilateral face shape functions.

Tetrahedral Mesh Region

There are $\frac{(p-2)(p-3)}{2}$ region functions for p, $p \geq 4$.
Legendre Polynomials based functions are given by

$$\phi(M_i^3) = \xi_1\xi_2\xi_3\xi_4 P_{\alpha_1-1}(\xi_2-\xi_1)P_{\alpha_2-1}(2\xi_3-1)P_{\alpha_3-1}(2\xi_4-1) \tag{30}$$

where $\alpha_1, \alpha_2, \alpha_3 = 1, .., p-3$, $\alpha_1+\alpha_2+\alpha_3+\alpha_4 = p$
An alternate basis with better conditioning of element matrices is

$$\begin{aligned}
\phi(M_i^3) &= \xi_1\xi_2\xi_3\xi_4(A \times B \times C) \\
A &= \sum_{i=0}^{\alpha_1-1} (-1)^i i! \binom{\alpha_1-1}{i}\binom{\alpha_1}{i}\frac{(2m+5-i)!}{(2m+5)}(\xi_1)^{\alpha_1-1-i} \\
B &= \sum_{i=0}^{\alpha_2-1} i! \binom{\alpha_2-1}{i}\binom{\alpha_2}{i}\frac{(2n+3-i)!}{(2n+3)}(\xi_2)^{\alpha_2-1-i}(\xi_1-1)^i \\
C &= \sum_{i=0}^{\alpha_3-1} i! \binom{\alpha_3-1}{i}\binom{\alpha_3}{i}\frac{(2\alpha_3-i)!}{2\alpha_3}(\xi_3)^{\alpha_3-1-i}(\xi_1+\xi_2-1)^i
\end{aligned} \tag{31}$$

where $m = \alpha_1+\alpha_2+\alpha_3-3$, $n = \alpha_2+\alpha_3-2$.
Directional behavior through the barycentric coordinates quadruplet, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, where, α_i represents the power of ξ_i in the shape function.

Only three can be varied independently. Equation (31) uses ξ_1, ξ_2, ξ_3 as the independent coordinates. A geometric construct for the possible region modes is a set of four tetrahedrons each of which allow for the independent variation of three of the four indices in the quadruplet. Figure 17 shows one of the tetrahedrons that varies $\alpha_1, \alpha_2, \alpha_3$. Similar construction can be used to define remaining three tetrahedrons that define the variation of $\alpha_1, \alpha_2, \alpha_4$ or $\alpha_1, \alpha_3, \alpha_4$ or $\alpha_2, \alpha_3, \alpha_4$.

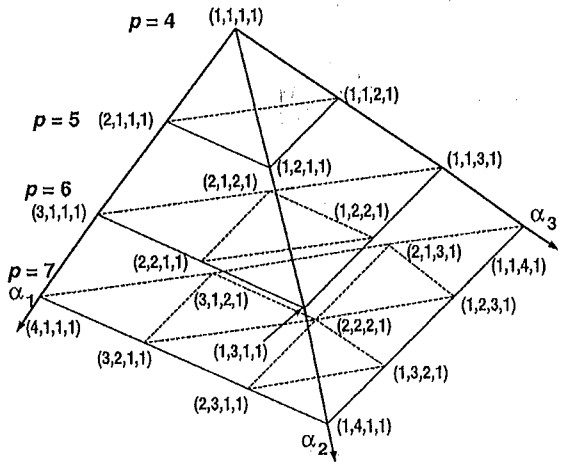


Figure 17. Geometric construct to generate tetrahedral region functions.

Table 5. Tetrahedral region functions.

p	$(\alpha_1, \ldots, \alpha_4)$	ϕ	
		Equation 30	Equation 31
4	(1,1,1,1)	1	1
5	(2,1,1,1)	$\xi_2 - \xi_1$	$\xi_1 - \frac{2}{7}$
	(1,2,1,1)	$2\xi_3 - 1$	$\xi_2 + \frac{2}{5}(\xi_1 - 1)$
	(1,1,2,1)	$1 - 2(\xi_1 + \xi_2 + \xi_3)$	$\xi_3 + \frac{2}{3}(\xi_1 + \xi_2 - 1)$
6	(3,1,1,1)	$\frac{3}{2}(\xi_2 - \xi_1)^2 - \frac{1}{2}$	$\xi_1^2 - \frac{2}{3}\xi_1 + \frac{1}{12}$
	(2,2,1,1)	$(\xi_2 - \xi_1)(2\xi_3 - 1)$	$\frac{-1}{45}(9\xi_1 - 2)(2\xi_1 + 5\xi_2 - 2)$
	(2,1,2,1)	$(\xi_2 - \xi_1)(1 - 2(\xi_1 + \xi_2 + \xi_3))$	$\frac{-1}{27}(9\xi_1 - 2)(2(\xi_1 + \xi_2 - 1) + 3\xi_3)$
	(1,3,1,1)	$\frac{3}{2}(2\xi_3 - 1)^2 - \frac{1}{2}$	$\xi_2^2 + \frac{6}{7}\xi_2(\xi_1 - 1) + \frac{1}{7}(\xi_1 - 1)^2$
	(1,2,2,1)	$(2\xi_3 - 1)(1 - 2(\xi_1 + \xi_2 + \xi_3))$	$\frac{-1}{27}(2\xi_1 + 7\xi_2 - 2)(2(\xi_1 + \xi_2 - 1) + 3\xi_3)$
	(1,1,3,1)	$\frac{3}{2}(1 - 2(\xi_1 + \xi_2 + \xi_3))^2 - \frac{1}{2}$	$\xi_3^2 + \frac{6}{5}\xi_3(\xi_1 + \xi_2 - 1) + \frac{3}{10}(\xi_1 + \xi_2 - 1)^2$

Hexahedral Mesh Region

There are $\frac{(p-4)(p-5)}{2}$ for *p, p≥6.* Legendre polynomial based product polynomial given by

$$\Psi\phi(M_i^3) = \sqrt{\frac{2\alpha_1-1}{2}}\int_{-1}^{\xi_1} P_{\alpha_1-1}(t)dt \sqrt{\frac{2\alpha_2-1}{2}}\int_{-1}^{\xi_2} P_{\alpha_2-1}(t)dt \sqrt{\frac{2\alpha_3-1}{2}}\int_{-1}^{\xi_3} P_{\alpha_3-1}(t)dt \tag{32}$$

where $\alpha_1, \alpha_2, \alpha_3 \geq 2$, $\alpha_1 + \alpha_2 + \alpha_3 = p$

Directional behavior can be understood the triplet, $(\alpha_1, \alpha_2, \alpha_3)$, where α_i is the power of ξ_i in the shape function.

Table 6. Hexahedral region functions.

p	$(\alpha_1, \alpha_2, \alpha_3)$	$\Psi\phi$
6	2, 2, 2	$\frac{3\sqrt{6}}{8}\left[\left(3\xi_1^2-1\right)\left(3\xi_2^2-1\right)\left(3\xi_3^2-1\right)\right]$
7	3, 2, 2	$\frac{3\sqrt{10}}{8}\left[\left(5\xi_1^3-3\xi_1\right)\left(3\xi_2^2-1\right)\left(3\xi_3^2-1\right)\right]$
	2, 3, 2	$\frac{3\sqrt{10}}{8}\left[\left(3\xi_1^2-1\right)\left(5\xi_2^3-3\xi_2\right)\left(3\xi_3^2-1\right)\right]$
	2, 2, 3	$\frac{3\sqrt{10}}{8}\left[\left(3\xi_1^2-1\right)\left(3\xi_2^2-1\right)\left(5\xi_3^3-3\xi_3\right)\right]$

Pentahedral Mesh Region — wedge

There $\frac{(p-3)(p-4)}{2}$ pentahedral region functions for *p≥5.* The region modes are products of the triangular face functions and an edge function along ξ_4^3 given by

$$\Psi\phi(M_i^3) = \xi_1\xi_2\xi_3\phi(M_j^2)\left(\frac{\xi_4^2-1}{2}\right)\phi(M_k^1) \tag{33}$$

where $\phi\left(M_j^2\right)$ is given by equation (27) with $\hat{\xi}_1 = \xi_1$, $\hat{\xi}_2 = \xi_2$ and $\hat{\xi}_3 = \xi_3$. $\phi(M_k^1)$ is given by equation (24) with $\hat{\xi}_1 = \xi_4$ for parameterization *I* or $\hat{\xi}_1 = \frac{1-\xi_4}{2}$ and $\hat{\xi}_2 = \frac{1+\xi_4}{2}$ for parameterization *II* and $p = \alpha_4$.

Directional properties can be interpreted by associating a quadruplet, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ where only two of ξ_1, ξ_2, ξ_3 can be varied independently. If ξ_1, ξ_2, ξ_4 are chosen as the independent coordinate parameters then the following rules generate the set of possible region functions

$$\alpha_1, \alpha_2 \geq 1\ ,\ \ \alpha_4 \geq 2\ ,\ \ \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = p \tag{34}$$

Table 7. Region functions for a pentahedron.

p $(\alpha_1\alpha_2, \alpha_3, \alpha_4)$		$\Psi\phi$
5	(1,1,1,2)	$\Psi\,\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(1-\xi_4^2)$
6	(1,1,1,3)	$\Psi\,\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3\xi_4(1-\xi_4^2)$
	(1,2,1,2)	$\Psi\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(\xi_2-\frac{1}{3})(1-\xi_4^2)$
	(2,1,1,2)	$\Psi\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(\xi_1-\frac{1}{3})(1-\xi_4^2)$

Computational Cost for Constructing Variable *p*-Order Meshes Using Decomposed Shape Functions

Operation counts required to evaluate a complete set of shape functions for a given p and CPU time to evaluate element "stiffness" matrix represented in the matrix form by

$$k_{ij} = \int_{\Omega^e} \frac{\partial N_i}{\partial \mathbf{x}} \left(\frac{\partial N_j}{\partial \mathbf{x}}\right)^T d\Omega \tag{35}$$

Implementation Comparison

Functionalities needed to compute the element level matrices for a variable p meshes

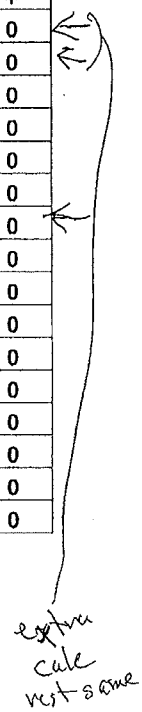
1. Access to the topological hierarchy of mesh entities for an element.
2. Ability to query (and set) the spectral order of interpolation for a given field variable over a given mesh entity.
3. Ability to query number of shape functions of a given polynomial order from a mesh entity.
4. Ability to evaluate shape function(s) of a given polynomial order from a mesh entity at a point in the domain of the element (another mesh entity).
5. Ability to evaluate derivative of shape function(s) of a given polynomial order from a mesh entity at a point in the domain of the element (another mesh entity).

Table 8. Shape functions for a triangle with uniform p=6.

Entity	p	Explicit	Decomposed	
			ψ	ϕ
vertex	0-1	ξ_i	ξ_i	1
edge	2	$-2\xi_i\xi_j$	$-2\xi_i\xi_j$	1
	3	$-2\xi_i\xi_j(\xi_j-\xi_i)$	$-2\xi_i\xi_j$	$\xi_j-\xi_i$
	4	$-2\xi_i\xi_j\left(\xi_j^2-3\xi_i\xi_j+\xi_i^2\right)$	$-2\xi_i\xi_j$	$\xi_j^2-3\xi_i\xi_j+\xi_i^2$
	5	$-2\xi_i\xi_j\left(\xi_j^3-6\xi_i\xi_j^2+6\xi_i^2\xi_j-\xi_i^3\right)$	$-2\xi_i\xi_j$	$\xi_j^3-6\xi_i\xi_j^2+6\xi_i^2\xi_j-\xi_i^3$
	6	$-2\xi_i\xi_j\left(\xi_j^4-10\left(\xi_i\xi_j^3-2\xi_i^2\xi_j^2+\xi_i^3\xi_j\right)+\xi_i^4\right)$	$-2\xi_i\xi_j$	$\xi_j^4-10\left(\xi_i\xi_j^3-2\xi_i^2\xi_j^2+\xi_i^3\xi_j\right)+\xi_i^4$
face	3	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2\xi_3$	1
	4	$\xi_1\xi_2\xi_3(\xi_1-\frac{1}{3})$	$\xi_1\xi_2\xi_3$	$\xi_1-\frac{1}{3}$
		$\xi_1\xi_2\xi_3(\xi_2-\frac{1}{3})$	$\xi_1\xi_2\xi_3$	$\xi_2-\frac{1}{3}$
	5	$\xi_1\xi_2\xi_3(\xi_1^2-\frac{3}{4}\xi_1+\frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_1^2-\frac{3}{4}\xi_1+\frac{3}{28}$
		$\xi_1\xi_2\xi_3(\xi_1\xi_2-\frac{1}{4}(\xi_1+\xi_2)+\frac{1}{14})$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2-\frac{1}{4}(\xi_1+\xi_2)+\frac{1}{14}$
		$\xi_1\xi_2\xi_3(\xi_2^2-\frac{3}{4}\xi_2+\frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_2^2-\frac{3}{4}\xi_2+\frac{3}{28}$
	6	$\xi_1\xi_2\xi_3(\xi_2^3-\frac{1}{5}(6\xi_2^2-2\xi_2+\frac{1}{6}))$	$\xi_1\xi_2\xi_3$	$\xi_2^3-\frac{1}{5}(6\xi_2^2-2\xi_2+\frac{1}{6})$
		$\xi_1\xi_2\xi_3\left(\xi_1\xi_2(\xi_2-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_1}{3}-\xi_2^2+\frac{2\xi_2}{3}-\frac{1}{12}\right)\right)$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2(\xi_2-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_1}{3}-\xi_2^2+\frac{2\xi_2}{3}-\frac{1}{12}\right)$
		$\xi_1\xi_2\xi_3\left(\xi_1\xi_2(\xi_1-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_2}{3}-\xi_1^2+\frac{2\xi_1}{3}-\frac{1}{12}\right)\right)$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2(\xi_1-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_2}{3}-\xi_1^2+\frac{2\xi_1}{3}-\frac{1}{12}\right)$
		$\xi_1\xi_2\xi_3(\xi_1^3-\frac{1}{5}(6\xi_1^2-2\xi_1+\frac{1}{6}))$	$\xi_1\xi_2\xi_3$	$\xi_1^3-\frac{1}{5}(6\xi_1^2-2\xi_1+\frac{1}{6})$

Table 9. Operation count for shape functions for a triangle with uniform p=6.

Entity	p	Explicit		Decomposed					
				ψ		ϕ		$\psi*\phi$	
		*	+	*	+	*	+	*	+
vertex	0-1	0	0	0	0	0	0	1	0
edge	2	2	0	2	0	0	0	1	0
	3	3	1	2	0	0	1	1	0
	4	9	2	2	0	6	2	1	0
	5	13	3	2	0	10	3	1	0
	6	20	4	2	0	17	4	1	0
face	3	2	0	2	0	0	0	1	0
	4	3	1	2	0	0	1	1	0
		3	1	2	0	0	1	1	0
	5	5	2	2	0	2	2	1	0
		5	3	2	0	2	3	1	0
		5	2	2	0	2	2	1	0
	6	10	3	2	0	7	3	1	0
		9	5	2	0	6	5	1	0
		9	5	2	0	6	5	1	0
		10	3	2	0	7	3	1	0



Example problem with exact solution given by

$$u(x,y) = \frac{4}{\pi^2} \sum_{n=1}^{\infty} \frac{\sinh(n\pi x)\sin(n\pi y)}{n^2 \cosh(n\pi)};\ n\ odd \tag{36}$$

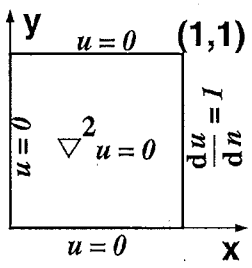


Figure 18. Example problem.

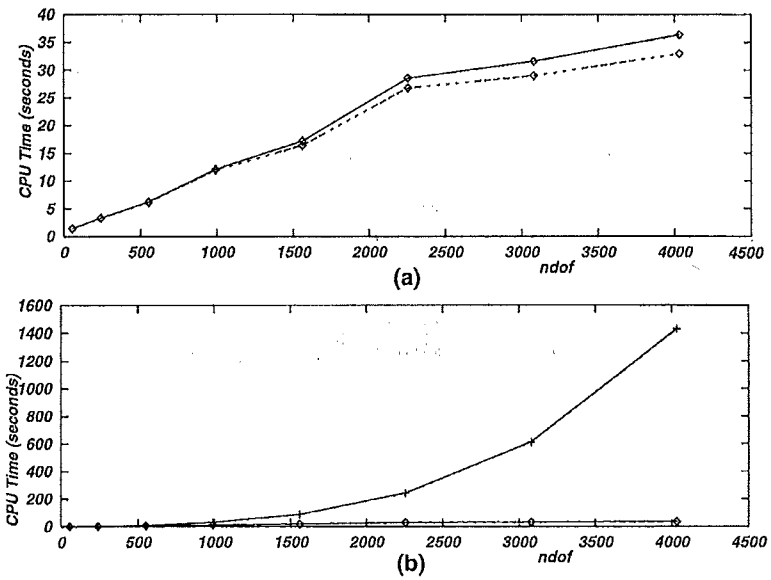


Figure 19. CPU time to compute (a) element level matrices and (b) total solution for 128 elements with uniform p from 1 to 8. Solid and broken lines represent decomposed and explicitly evaluated shape functions respectively. $\diamond$ and + represent time to compute element matrices and total solution time respectively.

[17]Flaherty, J.E., Loy, R.M., Ozturan, C., Shephard, M.S., Szymanski, B.K., Teresco, J.D. and Ziantz, L.H., "Parallel structures and dynamic load balancing for adaptive finite element computations", *Applied Numerical Mathematics,* 26(1-2):241-263, 1998.

[18]Bottasso, C.L., Klaas, O. and Shephard, M.S., "Data structures and mesh modifications for unstructured multigrid adaptive techniques", *Engineering with Computers,* 14(3):235-247, 1998.

A couple of References

1997

[19]Adjerid, S., Aiffa, M., Flaherty, J.E., Hudson, J.B. and Shephard, M.S., "Modeling and adaptive numerical techniques for oxidation of ceramic composites", *Ceramic Engng. and Scientific Proc.,* 18:315-322, 1997.

[20]Bottasso, C.L. and Shephard, M.S., "A parallel adaptive finite element flow solver for rotary wing aerodynamics", *AIAA,* 35(6):1-8, 1997.

[21]Givoli, D., Flaherty, J.E. and Shephard, M.S., "Parallel adaptive finite element analysis of viscous flows", *Int. J. Num. Meth. Heat Fluid Flow,* 7(8):880-906, 1997.

[22]Shephard, M.S., Flaherty, J.E., Bottasso, C.L., de Cougny, H.L., Ozturan, C. and Simone, M.L., "Parallel automated adaptive analysis", *Parallel Computing*, 23:1327-1347, 1997.

[23]Shephard, M.S., Dey, S. and Flaherty, J.E., "A straight forward structure to construct shape functions for variable p-order meshes", *Comp. Meth. Appl Mech. Engng.,* 147:209-233, 1997.

[24]Beall, M.W. and Shephard, M.S., "A general topology-based mesh data structure", *Int. J. Numer. Meth. Engng.,* 40(9):1573-1596, 1997.

[25]Fish, J., Shek, K.-L., Pandheeradi, M., and Shephard, M.S., "Computational plasticity for composite structures based on mathematical homogenization: Theory and practice", *Comp. Meth. Appl. Mech. Engng.,* 148-53-73, 1997.

[26]Flaherty, J.E., Loy, R.M., Shephard, M.S., Szymanski, B.K., Teresco, J.D. and Ziantz, L.H., "Adaptive local refinement with octree load balancing for the parallel solution of three-dimensional conservation laws", *J. of Parallel and Distributed Computing*, 47:139-152, 1997.

[27]Dey, S., Shephard, M.S. and Flaherty, J.E., "Geometry-based issues associated with p-version finite element computations", *Comp. Meth. Appl. Mech. Engng.,* 150:39-55, 1997.

[28]Givoli, D., Flaherty, J.E. and Shephard, M.S., "Parallel adaptive 3D finite element analysis of CZ melt flows", *J. Crystal Growth,* 174:1-6, 1997.

[29]Givoli, D., Flaherty, J.E. and Shephard, M.S., "Analysis of InP LEC melt flows using a parallel adaptive finite element scheme", *J. Crystal Growth,* 180:510-516, 1997.

[30]Dey, S., Shephard, M.S. and Georges, M.K., "Elimination of the adverse effects of small model features by local modifications of automatically generated meshes", *Eng. With Computers,* 13(3):134-152, 1997.

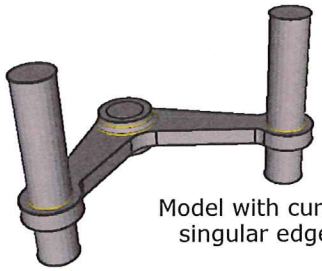
Implementation Requirements

- The influence of model geometry on the character of the solution
  - Solution can be decomposed as edge, vertex, edge-vertex singular

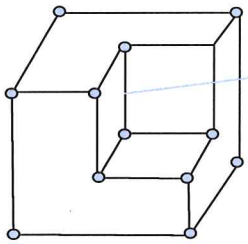
Edge singularity: $u(r,\theta,z) = \sum_{k=1}^{K} c_k(z) r^{\alpha_k} f_k(\theta) + v(r,\theta,z)$

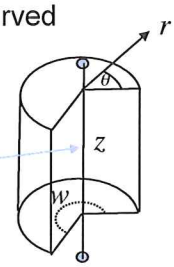
$c_k(z)$ edge flux intensity factor, $f_k(\theta)$ eigenfunctions analytic in θ
$v(r,\theta,z)$ smooth functions, $\alpha_k = k\pi / w$

- A model edge is singular if the interior dihedral angle $\omega > \pi$
- Require cylindrical mesh with geometric gradation 0.15
- In general 3D curved model, singular model edges may be curved
  - Require curved mesh gradation layout



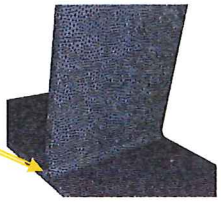
Model with curved singular edges





Implementation Requirements

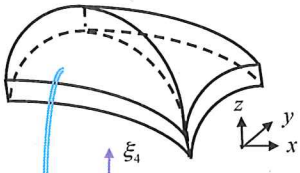
- Thin sections in 3D curved domains
  - Apply 2D plate or shell models
  - Can not deal with fillets, bosses, stiffeners etc.
  - Need special finite elements for the 2D to 3D transition



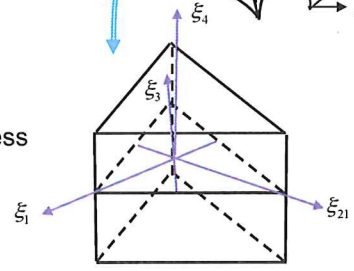
- Apply fully 3D p-version method for thin sections
  - Based on the p-version hierarchic plate and shell models

$$u_x = \sum_{i=0}^{n} z^i u_{xi}(x,y) \quad u_y = \sum_{i=0}^{n} z^i u_{yi}(x,y) \quad u_z = \sum_{i=0}^{m} z^i u_{zi}(x,y)$$

    - Converge to three dimensional model as $n, m \to \infty$



  - Choose polynomial order differently for in-plane and thickness directions
  - Require one layer prismatic elements through thickness direction

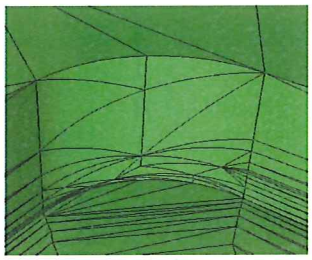


$$\xi_1, \xi_2, \xi_3 \to x, y$$
$$\xi_4 \to z$$

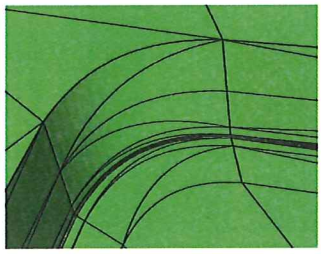
Simmetrix Inc.
Enabling Simulation-Based Design

Construction of p-Version Meshes

- Complexities for 3D curved domains
  - Shape of mesh entities on curved model boundary
  - Procedures to construct proper graded curved meshes
  - Structured prismatic meshes for thin section structures
- Key steps and techniques
  - Isolate and mesh singular edges and thin sections
  - Curve the mesh entities on the boundaries in properly order
  - Curved local mesh modifications to correct invalid elements
- Approaches being taken
  - Boundary layer mesh generation
  - Linear volume mesh generation



Curving without order



Curving with order

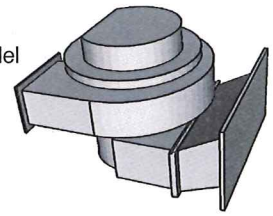
Simmetrix Inc. Enabling Simulation-Based Design

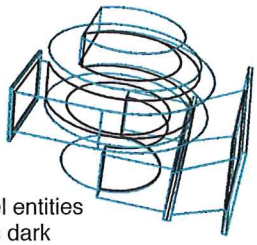
SCOREC

Identifying and Meshing Singular Edges

- Examine the interior dihedral angle variation for the model edges

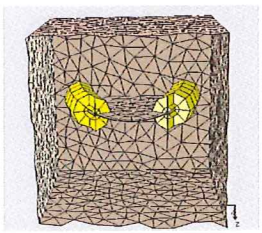
Geometric model

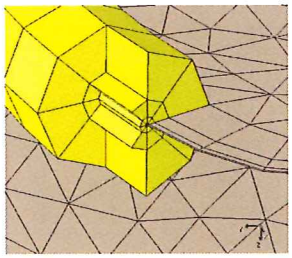




Singular model entities marked as dark

- Apply advance layer mesh generation to construct straight sided layer mesh

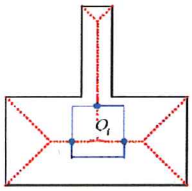




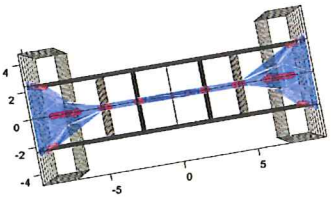
Simmetrix Inc. Enabling Simulation-Based Design

SCOREC

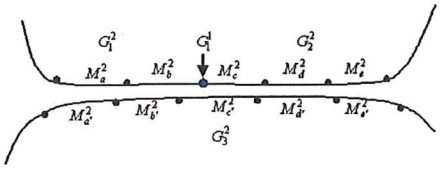
Identifying Thin Sections

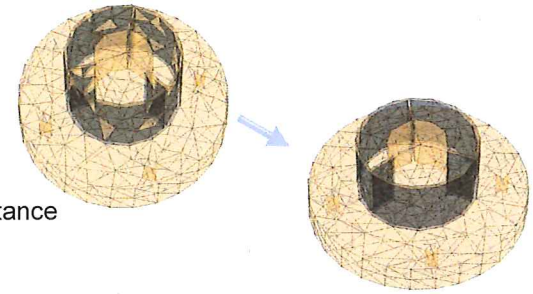


- Compute a set of medial surface points
  - Do not need a fully accurate medial surface
  - Intersection of octant edges and medial surface



- Collect set of opposite surface triangles based on classification on model face
  - This will not final all of them





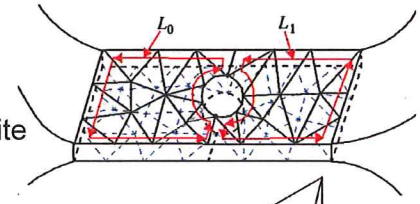
- Determine missing triangles
  - Use adjacencies, classifications and distance

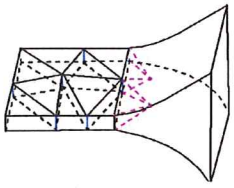
Simmetrix Inc.
Enabling Simulation-Based Design

SCOREC

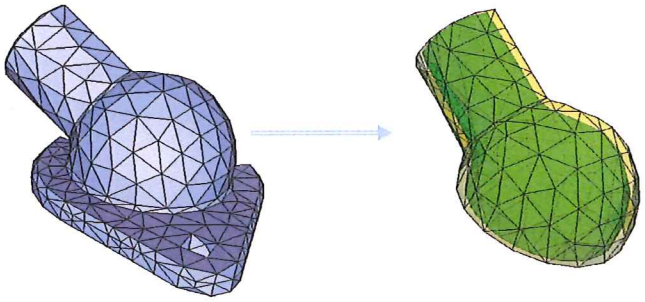
Meshing Thin Sections

- Build loops of opposite surface patches
  - Edge and face adjacencies





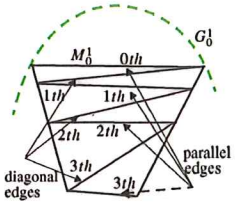
- Modify surface triangulations so loops of opposite thin section patches have "equivalent" edge discretization
- Copy triangulation from one opposite patch to the other and create prismatic mesh
- Construct transition to allow automatic tet meshing of the rest of the domain



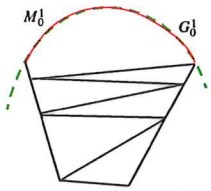
SCOREC

Mesh Curving

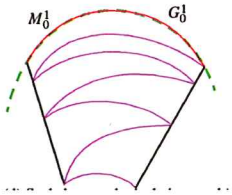
- Curve mesh entities in the boundary layer using curving consistent with that required to curve the entities classified on the curved boundaries



Layer elements

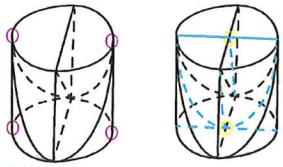


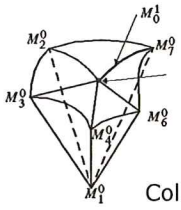
Curve M_0^1 G_0^1



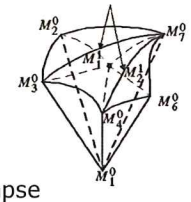
Scale curve

- Curved local mesh modification to correct invalid mesh entities
  - An element is invalidity if its minimum determinant of Jacobian less than or equal to zero
  - Reshape, split, collapse, swap and refinement


Split



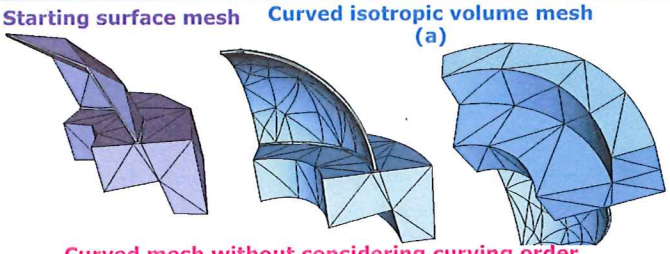
Collapse



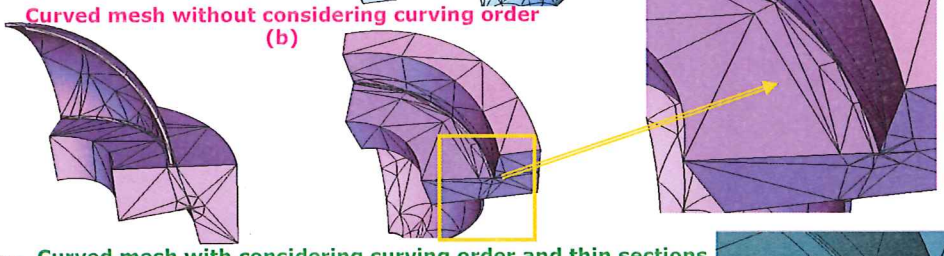
Simmetrix Inc.
Enabling Simulation-Based Design

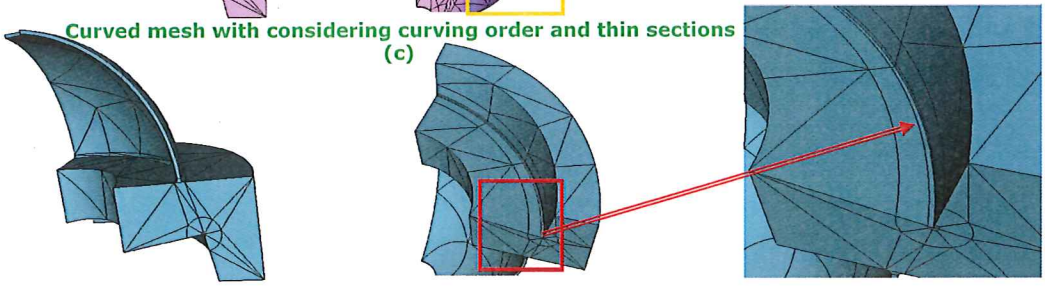
SCOREC

p-Version Mesh Results



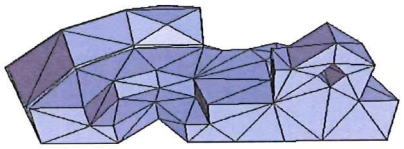
Mesh	Regions	Gradation
a	305	0
b	706	0.15
c	350	0.15



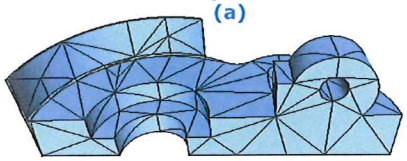


p-Version Mesh Results

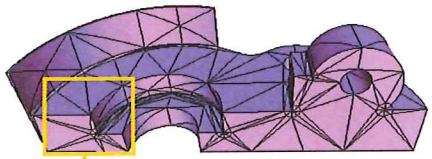
Starting surface mesh

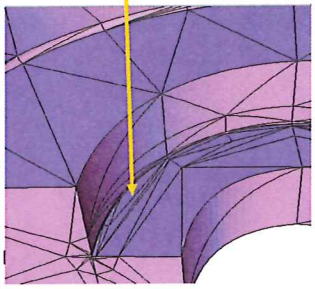


Curved isotropic volume mesh (a)

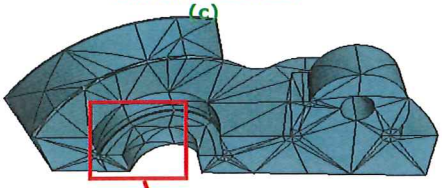


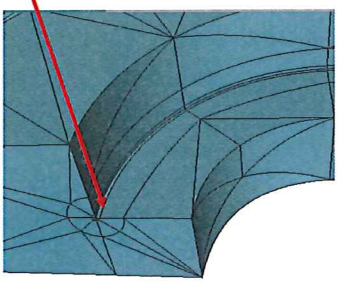
Curved mesh without considering curving order (b)





Curved mesh with considering curving order and thin sections (c)





Mesh	Regions	Gradation
a	260	0
b	996	0.15
c	756	0.15

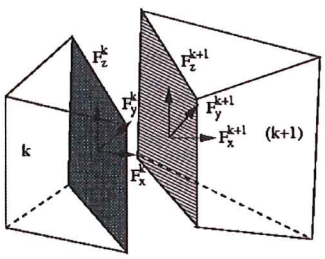
The Application of p-Version Meshes in StressCheck

- Objective
  - Compare the performance between p-version meshes and conventional meshes
  - Error measured in energy norm is of the most interest
- StressCheck is a p-version finite element analysis software
  - Founded by Prof. Babuska, Szabo etc in 1980's
  - Support uniform p-, adaptive p- and q-version method
  - Support tetrahedral, hexahedral and prismatic elements
- Extrapolation error estimation[1] in StressCheck

$$\|e\|^2 = \Pi(u^h) - \Pi(u) \approx \frac{C}{N^{2\beta}} \qquad e_r = \sqrt{\frac{\|e\|^2}{\Pi(u)}}\%$$

$\Pi(u^h)$, $\Pi(u)$ finite element and exact potent ional energy

- Error indicator[2] in StressCheck
  - Jumps in the stress resultants across any shared faces



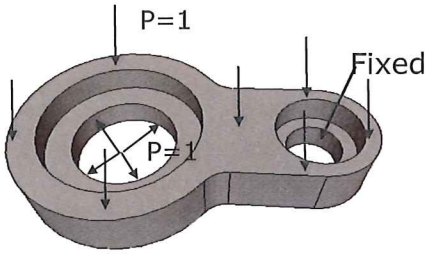
1 B.A. Szabo and I. Babuska (1991) *Finite Element Analysis*
2 StressCheck user's manual (2005)

Simmetrix Inc.
Enabling Simulation-Based Design

SCOREC

Problem 1

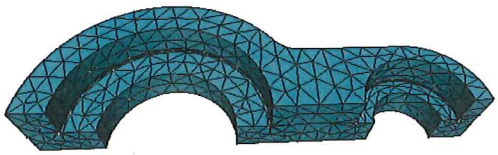
- Problem 1
  - Linear isotropic material
    - Young's Module E=3e7 psi
    - Poisson 's ratio $\upsilon = 0.3$
- Extension strategy
  - Uniform p=2 to p=8
  - Adaptive p – satisfy prescribe tolerance 0.1% or reach maximum allowable p order 8



Geometric model

- Selection of the exact potential energy $\Pi(u)$

$\Pi(u^h)(\times 10^{-1})$			$\Pi(u)(\times 10^{-1})$
p=6	p=7	p=8	(8 digits)
-3.73765	-3.73841	-3.73881	-3.73954411



Finer curved graded mesh
5481 regions and 4 graded layer

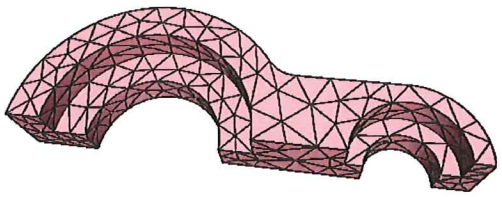
$$e_r = \sqrt{\left|\frac{\Pi(u) - \Pi(u^h)}{\Pi(u)}\right|} = 0.019\%$$

Simmetrix Inc. Enabling Simulation-Based Design

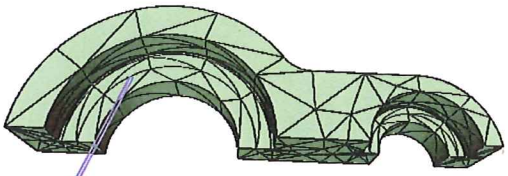
SCOREC

Meshes for Example 1

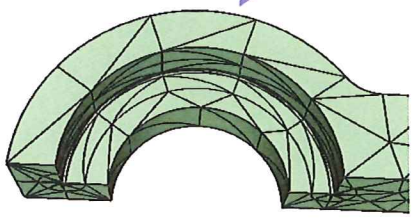
- Finite element meshes



Mesh a
1014 regions



Mesh b
888 regions
2 layers gradation

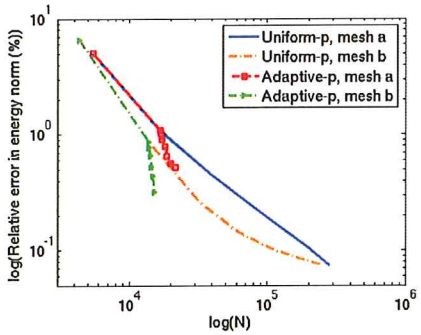


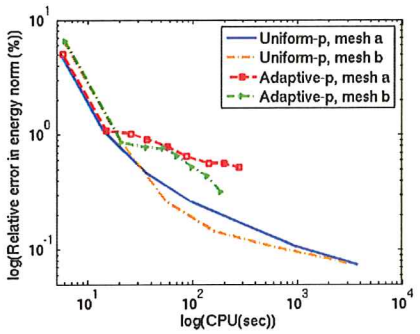
Simmetrix Inc. Enabling Simulation-Based Design

SCOREC

p-Version Analysis Results for Problem 1

- Relative error in energy norm with respect to Dof and CPU time





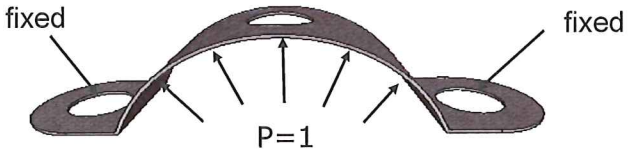
- p-version mesh b converges faster than mesh a
  - Uniform p: mesh b uses **15%** less dof comparing mesh a at p=8
  - Adaptive p: mesh b uses **30%** less dof comparing mesh a at last adaptive step
- CPU time
  - Uniform p: mesh b uses **16%** less CPU comparing mesh a at p=8
  - Adaptive p: mesh b uses **33%** less CPU comparing mesh a at last adaptive step

Simmetrix Inc. Enabling Simulation-Based Design

SCOREC

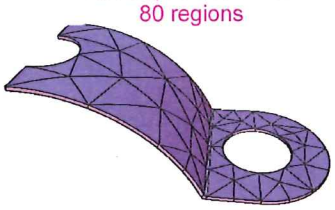
Thin Solid Model in StressCheck

- Model Problem
  - Linear isotropic material

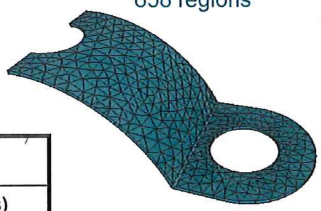


- Extension strategy
  - Uniform p analysis from p=2 to p=8
  - Fixed the thickness polynomial order q=1,2,3 and p=2 to p=8 for the in-plane direction for curved prismatic mesh

Curved prismatic mesh 80 regions



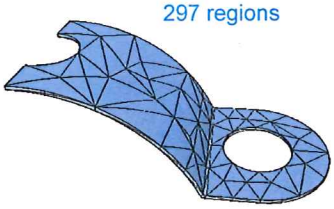
Curved prismatic mesh 658 regions



- Exact potential energy $\Pi(u)$

$\Pi(u^h)$			$\Pi(u)$
p=6	p=7	p=8	(8 digits)
-1.33677	-1.33773	-1.33822	-1.33894387

Curved tet mesh 297 regions



Simmetrix Inc. Enabling Simulation-Based Design

$$e_r = \sqrt{\left|\frac{\Pi(u) - \Pi(u^h)}{\Pi(u)}\right|} = 0.016\%$$