Structure Theoretical Overview

1 Introduction

This document describes the theoretical basis of the Structure computer program, along with some implementational aspects. The information presented is based on release 2.0.

1.1 Overview of Functionality

Applied Structure is implemented as two separate executable programs: the analysis engine, and the graphical user interface. This manual is primarily concerned with the analysis engine. The Ap- plied Structure analysis engine is a finite element analysis code which models three-dimensional deformable structures as an assembly of elements. Version 2.0 is restricted to small deflection lin- ear elastic analysis and isotropic material behavior. The basic features of Structure 2.0 are outlined below.

Basic analysis features:

- Three-dimensional, plane stress, plane strain, and axisymmetric models.
- Beam, plate, shell, solid, spring, and mass elements.
- The use of high-order p-elements to obtain a converged solution by means of a locally adap- tive edge-based polynomial order update algorithm.
- General element geometry which allows exact representation of underlying curves, surfaces, and volumes in the geometry terms of stiffness, mass and load calculations.
- The capability to "link" abutting elements without requiring that an entire edge or face be shared.
- Linear static and modal analysis.
- Loads and constraints relative to cartesian, cylindrical, or spherical coordinate systems.
- Gravitational, centrifugal, and thermal loading.
- Modal time marching for transient dynamic analysis.
- Frequency marching for harmonic excitation analysis.

Design features:

- Design variables.
- Parametric sensitivity.
- Local perturbation sensitivity.
- Optimization with multiple optimization limits which can reference multiple analyses.

1.2 The p-Method

Applied Structure is based on the p-version of the finite element method, or p-method, which has been treated in research literature for about 20 years, with notable contributions by I. Babuska, B. Szabo, A. Peano, and others. Theoretical foundations of the p-method are discussed by Babuska, Szabo, and Katz (1981) and by Babuska and Szabo (1982).

Conventional finite element codes rely on the h-method, in which a deformable mechanical struc- ture is analyzed by subdividing it into elements within which the field variables are approximated by low order polynomial functions, often linear, and sometimes quadratic or cubic. The accuracy of the analysis is increased by refining the mesh of elements. In the p-method, the basis functions for representing the field variables are high-order polynomials, and convergence is obtained by in- creasing the order of approximation within each element. Mesh refinement is therefore not re- quired for convergence.

In practice, it is not possible to increase the order of the elements indefinitely. The elements in Applied Structure use polynomial basis functions up to ninth order, which is sufficient for accurate representation of the physical behavior of most structures in regions where displacements and stresses are smoothly varying. In the neighborhood of cracks, re-entrant corners, and concentrated loads, however, stresses can be singular (theoretically infinite); elements must be further subdivid- ed near such local zones in order to obtain more accurate results.

2 Implementation of the p-Method in Structure

2.1 Formulation and Basis Functions

Applied Structure uses a displacement formulation, in which stiffness matrices and load vectors are derived by minimizing the structure's potential energy expressed in terms of displacements and rotations. As in other finite element programs, displacements within each element are represented by a linear combination of basis functions, or shape functions.

In the p-method, however, the primary variables are not nodal displacements, but are coefficients of linear, quadratic, cubic, and higher order polynomial basis functions. Once the finite element equations are solved for these coefficients, physical displacement and stress results are calculated over a grid as a postprocessing step.

The shape functions are based on the integrals of the Legendre polynomials. These functions are recommended by Babuska, Szabo, and Katz (1981) because they have some orthogonality proper- ties which result in well conditioned stiffness matrices. Structure uses polynomials from degree one through nine.

The shape functions are generated by taking products of the polynomials in the different natural coordinate directions, and can be separated into four types:

1. Linear nodal functions, which vanish at all but one node in an element. These are identical to the conventional shape functions in h-elements.
2. Edge functions, which are nonzero along a particular element edge and adjacent element faces and volumes.
3. Face functions, which are nonzero along a particular element face and adjacent element vol- umes.
4. Volume functions, which are nonzero within a solid element, but vanish on the faces.

Conventional h-element formulations usually rely exclusively on functions of type (1.) above. Oc- casionally second order functions of type (2.) are employed to transition between shell elements of different order. Second order functions of type (3.) or (4.) are sometimes included as "bubble" modes for shells or solids.

2.2 Geometric Mapping of Elements

The element formulations require that the underlying geometry be continuous and differentiable, but no other geometric restrictions are imposed. During element and load calculations, geometric information is obtained from a geometry module, which computes the mappings from element pa- rameter space to the underlying curve, surface, or solid geometry. The geometry module computes positions, jacobians, and local basis vectors for an expandable library of geometric shapes, which includes all the geometric entitites supported by the Structure user interface. The elements are therefore "geometrically exact," in the sense that no approximations (e.g. planar facets or iso- parametric maps) are made in representing the original geometry.

The Structure elements are superparametric, because the field variables are represented by functions up to ninth order, while the geometry may be of arbitrary high order. For example, an infinite number of nonrational polynomials are required to represent a cylindrical surface. For most geometric entities, the higher order components of the geometry intoduce small and smooth contributions to the solution, and convergence to the correct solution, including rigid body motion, is obtained at moderate polynomial order. If convergence is not obtained for an element which is associated with complex underlying geometry, such as multiple inflection points, the element should be subdivided. In order to prevent the construction of elements with poor convergence characteristics, the Structure user interface imposes some limitations during element cre- ation (e.g., one inflection point per edge, maximum subtended edge angle of ninety degrees).

2.3 Data Structures and Bookkeeping

When the model is read in by the analysis engine, a number of data structures are set up to permit continuity of the displacements between elements as the polynomial order changes.

The following types of data and associations are stored:

- Global sense of element edges and faces. Each edge and face has a global sense and orien- tation which determines the global sense of functions associated with the edge or face.
- Local sense of element edges and faces relative to global. Since elements may share edges or faces, the edge or face functions over a shared edge or face may have a different local sense (i.e. relationship to the global sense) for different elements. This local sense information is important for evaluating the shape functions within a particular element.
- Polynomial order of edges. The maximum polynomial order along each edge is treated as an independent quantity which can be set arbitrarily from one to nine. For example, if the polynomial order of an edge is five, displacements along the

edge will be represented by a series summation of edge functions up to order five.

- Polynomial order of faces in each natural coordinate direction. The polynomial order of the face functions for a particular face in each of the two natural coordinate directions is determined from the orders of the edges that enclose the face.
- Polynomial order of volumes in each natural coordinate direction. The polynomial order of the volume functions for a particular solid element in each of the three natural coordinate directions is determined from the orders of the edges of the solid element.

The "global/local sense" data allows the shape functions to be evaluated correctly within each el- ement of an assembly of elements, and facilitates the assembly of a global stiffness matrix based on the coefficients of the shape functions.

The polynomial order data structure allows the shape functions to be evaluated with arbitrary max- imum polynomial orders along each edge of the model. This permits continuous transitions be- tween elements of different polynomial order, with the requirement that the shared edges have the same polynomial order. For example, a quad element with edge orders 3 3 3 4 can be joined to a quad element with edge orders 4 5 5 5.

2.4 Global Solution Algorithm

Applied Structure uses an out-of-core LUD skyline block decomposition algorithm as the basis of the global solution procedure. For static analysis, one factorization is performed, and results for each load case are obtained by back substitution. For modal analysis, a block subspace iteration algorithm is employed, with shifting for unconstrained vibration problems. See Section 4 for ad- ditional information on solution algorithims.

2.5 Adaptive Convergence Algorithm

The advantage of a p-version finite element code is that convergence can be obtained by increasing the polynomial order of the elements without remeshing. Structure uses a locally adaptive polynomial order update scheme which takes advantage of the basis functions and data structure described above, in which a polynomial order (one through nine) can be assigned independently to each element edge.

The convergence algorithm is based on an iterative procedure in which the analysis is solved for successively higher polynomial orders. The convergence check which drives the polynomial order update is based on edge displacements and element strain energies.

The convergence algorithm consists of the following steps:

(0) The polynomial order of all edges is set to the minimum polynomial order specified by the user. The default minimum polynomial order is 1.

(1) A finite element solution is carried out.

(2) Point displacements and rotations over a grid of sample points along the element edges are cal- culated and stored. The total strain energy of each element is calculated and stored.

(3) If this is the first iteration, all the edges are updated by one polynomial order, and steps (1) and (2) are repeated. A minimum of two successive iterations are required in order to proceed with the convergence check described in the subsequent steps.

(4) The total strain energy of the model is calculated and stored, and the error in the global energy norm (an RMS stress measure) is estimated from the three point extrapolation formula given by Szabo and Babuska (1990). If this is the second iteration, the error is estimated from the strain en- ergy change relative to the first iteration, without extrapolating.

(5) The local edge displacements and element strain energies are compared with the values from the previous iteration. If the local displacement and strain energy percent differences and the esti- mated error in the global energy norm are less than the user input tolerance, then the results are considered to be converged and the iteration stops. The iteration also stops if the user has selected a particular response quantity for convergence control, and the change in that quantity is less than the tolerance. For modal analysis, the default convergence check is on frequency only, and the it- eration stops if the change in frequency for all modes is less than the tolerance.

(6) If the percent differences between the current results and the results of the previous iteration are greater than the user input tolerance, the polynomial orders of the edges are updated.

(7) The following rules determine which edges are updated:

(a) Edges for which the differences in displacements exceed the tolerance are updated by one poly- nomial order. If the coefficients of all odd or all even symmetry functions along a particular edge are neglibible, the edge is updated by two polynomial orders to account for symmetry. The coefficients of the higher-order shape functions are also examined. If it is found that the higher-order coefficients are small relative to the lower-order coefficients, the update is not performed. This prevents unnecessary updates due to rigid motion.

(b) If the square root of the difference in strain energy of an element relative to the previous itera- tion exceeds the tolerance, all edges of the element which have not already been updated in con- junction with (a) are updated by one polynomial order. The energy content of the higher-order shape functions is also examined. If it is found that the higher-order functions contribute negligi- bly relative to the lower-order functions, the update is not performed. This prevents unnecessary updates due to changing strain energy contributions of lower-order shape functions.

(c) For multiple load sets or modes, the update patterns for each load set or mode are merged with an "or" operation. The final update pattern therefore reflects the combined effects of all the load sets or modes. Different update patterns, and hence slightly different results, are obtained if two load sets are run in the same analysis or in two separate analyses.

(8) If the polynomial order update requires that the polynomial order of an edge exceed the maxi- mum allowable order, the results are considered to be not converged and the iteration stops. Oth- erwise, another "p-loop" iteration is carried out by returning to step (1) and performing a new anal- ysis with the updated polynomial orders.

In the above iterative procedure, it is possible that the displacements will converge more rapidly along some edges than others, and that the strain energies of some elements will converge more rapidly than others. Once convergence is reached for an edge or element, the local polynomial or- der update stops, although the polynomial order may continue to be updated for edges in other parts of the model where stress and displacement gradients are higher. This often leads to an inhomo- geneous distribution of edge orders, which is computationally efficient, since high-order shape functions are employed only where they are needed.

The above convergence algorithm estimates error by the size of "the last term in the series," and does not predict the exact error. Also, the convergence algorithm is driven by displacements and strain energy, which are known to converge faster than stresses. For stress convergence, a stress quantity should be selected as a response quantity to control convergence. In interpreting the qual- ity of the results, it is always a good idea to examine convergence curves for specific quantities of interest.

3 Element Formulations

Element stiffness matrices are based on a displacement formulation and are fully integrated. Re- duced integration of transverse shear terms is used for beam and plate elements only if the mini- mum and maximum polynomial orders are set to one. Consistent mass matrices (non-lumped) are computed for modal analysis. The element formulations require continuous and differentiable parametrically defined edges, faces, and volumes, but no further assumptions are made regarding the underlying element geometry.

For beam and shell elements, the choice of which element formulation to use (straight or curved beam, flat or curved shell) is made automatically by the engine. A straight beam formulation is used if a beam is straight to within one part in 10000, otherwise a curved beam formulation is used. Similarly, a flat plate formulation is used if a shell is flat to within one part in 10000, otherwise a curved shell formulation is used.

The displacement and rotation field variables are expressed in terms of a polynomial series:

$$\boldsymbol{u} = \sum_{i=1}^{n} \begin{bmatrix} u_x^{(i)} \\ u_y^{(i)} \\ u_z^{(i)} \end{bmatrix} \ell^{(i)} \qquad \beta = \sum_{i=1}^{n} \begin{bmatrix} \beta_x^{(i)} \\ \beta_y^{(i)} \\ \beta_z^{(i)} \end{bmatrix} \ell^{(i)} \qquad \text{(1a,b)}$$

where $\boldsymbol{u}$ and β are the displacement and rotation vectors, the functions $\ell^{(i)}$ are the hierarchical polynomial basis functions given by Babuska, Szabo, and Katz (1981),

$u_x^{(i)}$, $u_y^{(i)}$, $u_z^{(i)}$, $\beta_x^{(i)}$, $\beta_y^{(i)}$, and $\beta_z^{(i)}$ are the coefficients in the expansion for the cartesian components, and n is the number of terms in the series, which include polynomials up to order p. For the implementation of the elements, polynomial basis functions up to ninth order (p = 9) are used. Further details regarding the formulations for the different element types are given below.

3.1 Solid, 2D Quadrilateral, and 2D Triangular Elements

These elements are based on a straightforward extension of the conventional continuum formula- tion (see, e,g, Bathe (1982)), using the basis functions (1).

3.2 Straight Beam Element

The straight beam element is a Timoshenko beam element, which includes the effects of transverse shear deformation and rotary inertia. It can be viewed as a special case of the curved beam element, discussed in more detail below.

3.3 Curved Beam Element

The formulation for the curved beam element assumes a general space curve axis, but the imple- mentation in Structure is currently limited to planar curved beams. The basis functions are the hierarchical polynomial functions given by Babuska, Szabo, and Katz (1981). The beam strain energy includes the contributions of extension, biaxial bending, torsion, and transverse shear, and is written in a convenient vector form from which stiffness matrices are readily obtained in terms of global cartesian displacements and rotations.

The curved beam axis is defined by the function

$$\boldsymbol{X}(\xi) = (x(\xi), y(\xi), z(\xi)) \qquad -1 \le \xi \le 1 \tag{2}$$

which represents a space curve with mapping parameter ξ. The local basis vector $\boldsymbol{e}_1$ is tangent to the beam axis, and the local basis vectors $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ are the principal bending axes, which are also functions of ξ for a twisted curved beam. The differential arc length *ds* of the beam axis is given by

$$ds = |\boldsymbol{X}_{,\xi}| d\xi \tag{3}$$

where the comma denotes differentiation with respect to ξ.

The strain energy Π of the beam can be written

$$\begin{aligned}\Pi = \int \Big\{ \frac{1}{2} [& EA(\boldsymbol{u}_{,s} \bullet \boldsymbol{e}_1)^2 + EI_2(\beta_{,s} \bullet \boldsymbol{e}_2)^2 + EI_3(\beta_{,s} \bullet \boldsymbol{e}_3)^2 \\ & + k_2 GA(\boldsymbol{u}_{,s} \bullet \boldsymbol{e}_2 - \beta \bullet \boldsymbol{e}_3)^2 + k_3 GA(\boldsymbol{u}_{,s} \bullet \boldsymbol{e}_3 + \beta \bullet \boldsymbol{e}_2)^2 \\ & + GJ(\beta_{,s} \bullet \boldsymbol{e}_1)^2] - (\boldsymbol{F} \bullet \boldsymbol{u} + \boldsymbol{M} \bullet \beta) \Big\} ds \end{aligned} \tag{4}$$

where $\boldsymbol{u}$ is the displacement of the beam axis, β is the rotation vector of the beam section, *E* is Young's modulus, *G* is the shear modulus, *A* is the section area, I_2 and I_3 are the moments of area about the principal axes, k_2 and k_3 are the transverse shear factors for the principal directions, J is the torsion constant for the section, and $\boldsymbol{F}$ and $\boldsymbol{M}$ are force and moment load vectors. The terms of Eq. (4) can be readily identified with extension, bending, transverse shear, and torsion. It can be verified that the Euler-Lagrange equations associated with the strain energy expression (4) are the equilibrium equations derived by Love (1927).

The corresponding vector form for the kinetic energy Θ of the beam is

$$\Theta = \frac{1}{2} \int \rho [A(\dot{\boldsymbol{u}} \bullet \dot{\boldsymbol{u}}) + I_2(\dot{\beta} \bullet \boldsymbol{e}_2)^2 + I_3(\dot{\beta} \bullet \boldsymbol{e}_3)^2 + (I_2 + I_3)(\dot{\beta} \bullet \boldsymbol{e}_1)^2] ds \tag{5}$$

where ρ is the density and the dot denotes differentiation with respect to time. The terms of Eq. (5) can be identified with

translational kinetic energy and rotary inertia about the three local axes.

The vector forms (4,5) permit the direct substitution of the series expansions (1) in order to obtain stiffness and mass matrices in terms of the polynomial coefficients of the cartesian displacement and rotation components. For example, substituting Eq. (1a)

$$\frac{1}{2}\int EA(\boldsymbol{u}_{,s}\bullet\boldsymbol{e}_1)^2 ds = \sum_{i,j=1}^{n}\left[u_x^{(i)}\ u_y^{(i)}\ u_z^{(i)}\right]\left\{\int EA\,l_{,s}^{(i)}l_{,s}^{(j)}\begin{bmatrix}e_{1x}\\ e_{1y}\\ e_{1z}\end{bmatrix}\left[e_{1x}\ e_{1y}\ e_{1z}\right]ds\right\}\begin{bmatrix}u_x^{(j)}\\ u_x^{(j)}\\ u_x^{(j)}\end{bmatrix}$$

into the first term of Eq. (4) yields:

(6)

where the term in braces on the right-hand side is a 3*n* by 3*n* stiffness matrix for the extensional response of the beam. For the beam element, the *l(i)* are functions of ξ only, and the derivatives $l^{(i)}$ are evaluated from Eq. (3) and the chain rule.

3.4 Flat Plate Element

The flat plate element is a Reissner-Mindlin element, which includes the effects of transverse shear deformation and rotary inertia. It can be viewed as a special case of the curved shell element, dis- cussed in more detail below.

3.5 Curved Shell Element

The formulation utilizes the degenerate continuum approach of Ahmad, Irons, and Zienkiewicz (1970) to obtain a six degree-of-freedom transverse shear deformable shell element. The basis functions are the hierarchical polynomials used by Szabo and Sahrmann (1988). To express the shell stiffness in terms of all six global degrees of freedom, the variationally consistent treatment of the "drilling" rotation suggested by Hughes and Brezzi (1989) is employed.

The shell midsurface is defined by the function

$$\bar{\boldsymbol{X}}(\xi,\eta) = (x(\xi,\eta), y(\xi,\eta), z(\xi,\eta)) \qquad (7)$$

which represents a surface with mapping parameters ξ> and η. For a quadrilateral element, the mapping parameters have the ranges $-1\le\xi\le 1, -1\le\eta\le 1$. The lines of constant ξ and η are not, in general, locally orthogonal, but a local orthogonal basis ($\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$) can be defined as follows:

$$\boldsymbol{e}_1 = \frac{\bar{\boldsymbol{X}}_{,\xi}}{|\bar{\boldsymbol{X}}_{,\xi}|} \quad ; \quad \boldsymbol{e}_3 = \frac{\bar{\boldsymbol{X}}_{,\xi}\times\bar{\boldsymbol{X}}_{,\eta}}{|\bar{\boldsymbol{X}}_{,\xi}\times\bar{\boldsymbol{X}}_{,\eta}|} \quad ; \quad \boldsymbol{e}_2 = \boldsymbol{e}_3\times\boldsymbol{e}_1 \qquad (8)$$

where the basis vector $\boldsymbol{e}_3$ is the normal to the shell midsurface.

The shell of thickness *h* can therefore be represented by the mapping

$$\boldsymbol{X}(\xi,\eta,\zeta) = \bar{\boldsymbol{X}}(\xi,\eta) + \boldsymbol{e}_3\frac{h}{2}\zeta \quad ; \quad -1\le\zeta\le 1 \qquad (9)$$

where the upper and lower surfaces are defined by $\zeta = \pm 1$.

With the assumption that points which lie along a normal to the midsurface before deformation lie in a straight line after deformation (Reissner-Mindlin kinematics), the displacement vector ***U*** at any point of the shell can be written

$$\boldsymbol{U}(\xi,\eta,\zeta) = \boldsymbol{u}(\xi,\eta) + \frac{\zeta h}{2}(\beta(\xi,\eta)\times\boldsymbol{e}_3) \qquad (10)$$

where ***u*** is the displacement of a point on the shell midsurface, and β is the rotation of the shell normal.

Substitution of Eqs. (1) with basis functions $\phi^{(i)}(\xi,\eta)$ into Eq. (10) then yields the polynomial expansions of the three displacement degrees of freedom. The shell stiffness and mass matrices are obtained by the degenerate continuum approach of Ahmad, Irons, and Zienkiewicz (1970), using the kinematic equation (10), which results in stiffness and mass matrices in terms of the polynomial coefficients of the global cartesian displacement and rotation components.

Note that Eq. (10) assumes a general rotation vector β which can have a "drilling" component normal to the shell. In order to remove the spurious drilling mode associated with the kinematic expression (10), a penalty term of the following form is added to the element strain energy expression:

$$\Pi_d = \frac{1}{2}\alpha G \int\int h \left[\frac{1}{2}(\nabla \times \boldsymbol{U})|_{\zeta=0} \bullet \boldsymbol{e}_3 - \beta \bullet \boldsymbol{e}_3\right]^2 |\bar{\boldsymbol{X}}_{,\xi} \times \bar{\boldsymbol{X}}_{,\eta}| d\xi d\eta \qquad (11)$$

where α is a dimensionless constant and the integral is carried out over the shell midsurface. The form of Eq. (11) follows the suggestions of Hughes and Brezzi (1989) for the variationally consis- tent treatment of the drilling degree of freedom within the context of a displacement formulation.

4 Solution Techniques

For static analysis, a direct (out-of-core skyline) decomposition solver is employed. The basic algorithm is that of Mondkar and Powell (1974), but with input-output modifications to improve efficiency. After decomposition, multiple load cases may be solved.

For modal analysis, the eigenvalue/eigenvector solution is carried out using subspace iteration (Bathe, 1982). Solutions with rigid body modes are calculated using shifting. A Sturm sequence check is performed to ensure that no modes requested by the user have been missed. If there are any missing modes, the subspace iteration analysis is repeated with more Ritz vectors.

If "mode tracking" of the N-th mode is requested during a sensitivity or optimization study, the mode shape of the N-th mode of the first analysis performed is stored. During subsequent analyses all modes but one will be approximately mass-orthogonal with this stored mode shape. The remaining mode is closest in shape to the original mode, so it is "tracked".

Time and frequency marching is done with modal superposition. The exact solution in modal space is used for impulse loads; for arbitrary loads an explicit numerical scheme (central differences) is used with automatic selection of a stable step size. In Strucutre 2.0, only impulse loading is available through the graphical user interface.

Sensitivity and optimization studies require sensitivity derivatives , which are calculated using the "semi-analytic" method (Haug, *et al*, 1986).

5 Linking

Linking is the term used in Structure for the automatic generation of multi-function constraints to facilitate model transitioning. To assure accurate results with the finite element method, displacement compatibility must exist across element boundaries; this is assured by the basis functions employed in Structure's elements whenever elements are joined in a conventional manner. However, in a situation such as the "t-vertex" model transition in Fig. 1, compatibility is not assured across the joint between the left and right elements. Compatibility must be enforced explicitly be generating constraint relations that force the elements on one side of the joint to follow the element on the other side.

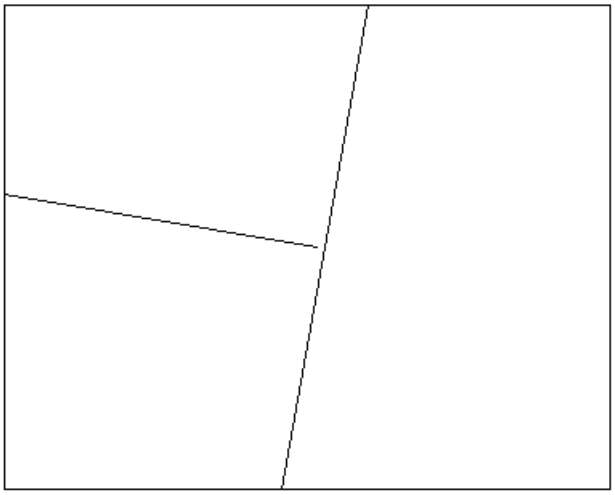

Figure 1. Three element model requiring linking.

5.1 Multi-Function Constraints For p-Version Elements

Model transitioning is often performed in conventional h-version finite element analysis by applying multi-point constraints among various nodes in the model. With higher-order (p-version) finite elements, multi-point constraints generalize to "multi-function constraints" because the shape functions are not nodally based. In Fig. 1, constraint equations are necessary to force the right edge of the lower left-hand element (edge 1) to follow the left edge of the right element (edge 2). The displacements of edge 1 are

$$u_i(\mathrm{r1}) = \sum_{j}^{N} U_j^1 h_j(\mathrm{r1}) \qquad (1)$$

where r1 is the natural coordinate along edge 1, *Uj* are the coefficients, and *hj* are the basis functions along the edge. By displacement we refer to any global component of generalized displacement (translations and rotations). Similarly, the displacements along edge 2 are given by

$$u_i(\mathrm{r2}) = \sum_{j}^{M} U_j^2 h_j(\mathrm{r2}) \qquad (2)$$

where r2 is the natural coordinate along edge 2. If the number of coefficients *M* along edge 2 is greater than or equal to the number *N* along edge 1, then edge 2 has the freedom to follow edge 1 exactly if the coefficents *U*2 are correcly related to *U*1. This requires constraint equations of the form:

$$U_i^2 = \sum_{j}^{N} \alpha_j U_j^1 \qquad (3)$$

This represents a multifunction constraint between the coefficients of the basis functions of edge 2 and those of edge 1. Linking automatically chooses the coefficients, α_j to enforce displacement compatibility.

5.2 Edge-to-Edge Linking

When the user specifies shell-shell linking and the elements involved meet along common edges, edge-to-edge links are generated. To derive the constraint relation between the basis functions of one edge and another, compatibility is enforced by minimizing the least-square error between the displacements of edge 2 and edge 1. The displacement error is

$$E = \int (u_1 - u_2)^2 ds \qquad (4)$$

where s is the arc length along edge 2. Substituting the for u_1 and u_2 from Eqs. (1,2), we then set

$$\frac{\partial E}{\partial U_j^2} = 0 \qquad j = 1 \ldots M \qquad (5)$$

resulting in the linear system

$$[K_2]\{U_2\} = [K_1]\{U_1\} \qquad (6)$$

where

$$K_2 = \int (h_i^2 h_j^2) ds \quad ; \quad K_1 = \int (h_j^2 h_i^1) ds \qquad (7)$$

Equations (7) are solved for U_2 in terms of U_1, which results in the desired multi-function constraint relations.

5.3 Face-to-Face Linking

Face-to-face linking is performed in a similar manner, but here we must ensure compatibility between a smaller face, face 2, that touches a larger face, face 1, and the least-squares integrals are performed over face 2.

5.4 Shell-Edge to Solid-Face Transitioning

Constraints for shell-edge to solid-face transitioning are generated by minimizing the least-square error between the displacements of the edge of the shell and a face of the solid. This involves integrating the error along the arc length of the shell and through its thickness.

5.5 Limitations

Linking will always assure compatibility as long as the polynomial order of the"dependent" edge or face (e.g. edge 2) is greater than or equal to that of the "independent" edge or face it is constrained to follow. Limitations arise from the ability of the P-elements to correctly represent the displacement of the constrained situation.

The polynomial basis functions of the elements can theoretically represent any solution, but an infinite number of polynomial functions might be required. A practical limitation of 9 has been applied on the polynomial order in Structure. It is difficult to represent sharply varying displacements along an edge. This does not occur in the "T-vertex" situation in Fig. 1 as long as the elements are all the of the same thickness and material property. However, for example, if element 2 were 10 times thicker than the other two elements, a stiffness "jump" would occur at the top of edge 2, causing a "kink" in the slope of the solution along edge 1, and it would be difficult for polynomials to follow this exactly. Another situation that causes a discontinuity is shown in Fig. 2. Here the left edge of an element is only continuous with another element along a portion of its length.

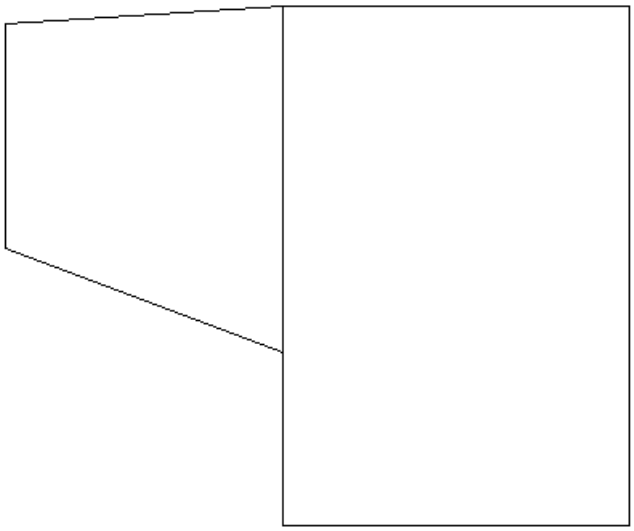

Figure 2. Model with link introducing displacement-gradient discontinuity.

6 Design Studies

Applied Structure can perform parametric design studies which require modification of the original model's geometry and properties, and multiple p-version finite element analyses The program architecture consists of three separate modules: A design study control module interprets user run requests from the database. As needed, it requests model updates from the geometry module, and it requests finite element solutions and interprets results from the finite element module.

6.1 Finite Element Module

The finite element module performs individual p-adaptive finite element solutions. It can also calculate sensitivity derivatives as described in section 4 if needed in the design study. In addition to conventional postprocessing, the finite element module extracts one or more scalar result quantities that are needed for convergence plotting, sensitivity analysis, and optimization. The quantities of interest (such as maximum von Mises stress in the model, stress in a particular element, maximum displacement magnitude in the model, etc.) are specified by the user and are referred to as "response quantities".

6.2 Parameterized Geometry Module

The parameterized geometry module is designed to allow the user to specify design variables that can change the shape and properties of the model in a general and physically meaningful way. This module accesses the model database created in the user interface. The user has created one or more "design variables" in the model, (radius of a circle, point position, curve shape, element properties, etc). An arbitrary number of design variables may be grouped together into a single set. A nondimensional parameter governs the set, to which the individual members are related linearly. As the parameter varies from 0 to 1, the design variables vary from their start to end values. After setting all design variables to new values as requested by the design study control module, the associative database is updated to modify the geometry, and the geometry and properties of all the elements are updated to the final configuration.

6.3 Design Study Control Module

This module controls the interaction of the finite element module and the geometry module to perform a "design study" requested by the user. Design studies include analysis, sensitivity studies, and optimization studies as described below. The two core operations the design study module performs are: 1. send updated values of design variable set parameters to the geometry module and request that the finite element model be modified accordingly and 2. request a finite element solution folowed by extraction of the values of response quantities.

The description of all finite element analysis requests defined by the user resides in the database; the control module need only refer to the analysis number. Several flags govern the analysis request: whether graphical postprocessing output is required, whether this is a new "base" analysis or a sensitivity derivative analysis, and what response quantities must be evaluated. Given

these operations a series of study types can be performed:

6.4 Standard Study

A single p-adaptive finite element solution is performed, and data are generated for graphical post-processing and convergence plots of the error norms used by the convergence loop as well as convergence plots of any response quantities specified by the user. For transient analyses plots of scalar results vs. time or frequency are produced. Multiple analyses of the same model may be "stacked" in the same run, which may be of different types (static, modal, or transient) and have different boundary conditions.

6.5 Offset Sensitivity Study

In this type of study, the user may specify the values of a series of design variable set parameters. The geometry module updates the model to correspond to those values, then a standard study is performed. The user can thus quickly set up studies to examine multiple points in design space without remeshing.

6.6 Local Sensitivity Study

Local sensitivity studies calculate the derivative of the user's response quantities with respect to one or more design variable set parameters. A p-adaptive base solution is first performed, followed by multiple perturbation solutions with each design variable set parameter perturbed in turn to permit forward difference approximation of derivatives by the semi-analytic method. The user may optionally request that the model be offset to an arbitrary point in design space before the derivative calculation is performed.

6.7 Global Sensitivity Study

Global sensitivity studies perform an arbitrary "line search" in design space, by varying N design variable set parameters from initial to final values in M steps. At each step the geometry module updates the model given new design variable set parameter values, then a p-adaptive finite element analysis is performed.

6.8 Optimization

The ability to offset the model to a point in design space, and to perform finite element and sensitivity-derivative calculations at that point, is the basis for a general optimization capability. The user defines a series of design variable set parameters, and gives starting values for them. (Most typically, the values corresponding to the model's "base" configuration are used.) A goal function and multiple limit functions are also specified. The goal and limits may refer to model mass or to any response quantity, and need not refer to the same analysis. For example the user may request that frequency of a certain mode in a modal analysis be maximized, with a limit on the model mass, one or more stress limits on a static analysis, a second set of stress limits on another analysis with different boundary conditions, etc. The lower and upper bounds on the design variable set parameters are also treated as limits. During an optimization, the design study control module requests the minimum number of finite element analyses needed to evaluate the goal and limits, by keeping track of which analyses refer to the goal and limits.

The variable metric method (Vanderplaats, 1984) is used to determine search directions at feasible points in design space (points where no limits are violated). When one or more limits are "active" (equal to their limiting values within a tolerance), the modified method of feasible directions is used to determine the search direction (Vanderplaats, 1984). Polynomial fitting is used to perform line searches along the search direction. If one or more limits is violated, a "feasibility search" algorithm is executed: The search direction that will most quickly make the violated limits active is found by examining the derivatives of the limits at the current design point. A line search is then performed in that direction to determine a feasible design point. If one or more limits are active while others are violated, the search direction to correct the violated limits will be projected along the active limits. The user's initial model need not be feasible.

Optimization convergence conditions include sensing small derivatives of the goal and sensing that a limit vertex has been encountered. After optimization has terminated by a convergence condition, the feasible point in design space that gives the best value of the goal is reported. (If the optimizer is unable to find a feasible point it reports the "least infeasible" point found.) Full results for graphical postprocessing are prepared at the optimum point so the user may compare the results of the initial and the optimium solution. If the optimum point was not the last design point investigated during the optimization, then the solution at the best design point found is repeated so postprocessing results may be prepared. In addition, plotting data are prepared for the goal, each of the limits, and any other response quantities specified by the user, as a function of optimization step number.

References

Ahmad, S., Irons, B. M., and Zienkiewicz, O. C. (1970), "Analysis of Thick and Thin Shell Structures by Curved Finite Elements," *Int. J. for Num. Meth. in Eng.* **2**, 419-451.

Babuska, I. (1988). "The p- and hp- versions of the finite element method: the state of the art," in *Finite Elements: Theory and Applications*, D. L. Dwoyer, M. Y. Hussaini, and R. G. Voigt, Eds., Springer-Verlag, New York..

Babuska, I., Szabo, B. A., and Katz, I. N. (1981). "The p-Version of the Finite Element Method," *SIAM J. of Numer. Anal.* **18**, 515-545.

Babuska, I., and Szabo, B. (1982). "On the Rates of Convergence of the Finite Element Method," *Int. J. for Num. Meth. in Eng.* **18**, 323-341.

Bathe, K. (1982). *Finite Element Procedures in Engineering Analysis.* Prentice Hall, New Jersey.

Haug, E., Choi, K., and Komkov, V. (1986). *Design Sensitivity Analysis of Structural Systems*, Academic Press, Orlando, Florida.

Hughes, T. J. R., and Brezzi, F. (1989). "On Drilling Degrees of Freedom," *Comp. Meth. in Appl. Mech. and Eng.* **72**, 105-121.

Issa, M. S., Wang, T. M., and Hsiao, B. T. (1987). "Extensional Vibrations of Continuous Circular Curved Beams with Rotary Inertia and Shear Deformation," *Journal of Sound and Vibration* **114**, 297-308.

Love, A. E. H. (1927). *A Treatise on the Mathematical Theory of Elasticity, Fourth Edition*. (Cambridge University Press, Cambridge, reprinted by Dover Publications, Inc., New York.)

Mondkar, D., and Powell, G. (1974). "Large Capacity Equation Solver for Structural Analysis", Computers and Structures, 4, 699-728.

Mottershead, J. E. (1980). "Finite Elements for Dynamical Analysis of Helical Rods," *Int. J. of Mech. Sci.* **22**, 267-283.

Naganarayana, B. P., and Prathap, G. (1990). "Consistency Aspects of Out-of-plane Bending, Torsion, and Shear in a Quadratic Curved Beam Element," *Int. J. for Num. Meth. in Eng.* **30**, 431-443.

Peano, A. G. (1976). "Hierarchies of conforming finite elements for plane elasticity and plate bending," *Comp. & Maths. with Appls.* **2**, 211-24.

Prathap, G., and Bhashyam, G. R. (1982). "Reduced Integration and the Shear-Flexible Beam Element," *Int. J. for Num. Meth. in Eng.* **18**, 195-210.

Prathap, G., and Naganarayana, B. P. (1990). "Analysis of Locking and Stress Oscillations in a General Curved Beam Element," *Int. J. for Num. Meth. in Eng.* **30**, 177-200.

Szabo, B. A. (1989). "Hierarchic Plate and Shell Models Based on p-Extension," in *Analytical and Computational Models of Shells, CED-Vol. 3,* presented at the Winter Annual Meeting of the ASME, Dec. 10-15, 1989, 317-331.

Szabo, B. A., and Babuska, I. (1990). *Finite Element Analysis,* John Wiley & Sons, New York.

Szabo, B. A., and Sahrmann, G. J. (1988). "Hierarchic Plate and Shell Models Based on p-Extension," *Int. J. for Num. Meth. in Eng.* **26**, 1855-1881.

Tabarrok, B., Farshad, M., and Yi, H. (1988). "Finite Element Formulation of Spatially Curved and Twisted Rods," *Comp. Meth. in Appl. Mech. and Eng.* **70**, 275-299.

Vanderplaats, G. (1984). *Numerical Optimization Techniques for Engineering Design,* McGraw- Hill, New York.

Wang, T. M., Laskey, A. J., and Ahmad, M. F. (1984). "Natural Frequencies for Out-of-plane Vibrations of Continuous Curved Beams Considering Shear and Rotary Inertia," *Int. J. of Solids and Structures* **20**, 257-265.