WL-TR-91-3006

EXPLICIT FINITE ELEMENT METHOD FOR TRANSPARENCY IMPACT ANALYSIS

R. A. Brockman
T. W. Held

University of Dayton Research Institute
Dayton, Ohio 45469-0110

June 1991

Final Report for Period September 1988 - September 1990

Approved for public release; distribution unlimited

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6553

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Robert E. McCarty

RICHARD A. SMITH
Aerospace Engineer

ROBERT E. MCCARTY, Supervisor
Aircrew Protection Branch

FOR THE COMMANDER

Richard E. Colclough, Jr.

RICHARD E. COLCLOUGH, JR.
Chief
Vehicle Subsystems Division

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/FIVR, WPAFB, OH 45433-6553 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UDR-TR-90-114	5. MONITORING ORGANIZATION REPORT NUMBER(S) WL-TR-91-3006
6a. NAME OF PERFORMING ORGANIZATION University of Dayton Research Institute 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Flight Dynamics Directorate (WL/FIVR) Wright Laboratory
6c. ADDRESS (City, State, and ZIP Code) 300 College Park Avenue Dayton, OH 45469-0110	7b. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433-6553
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Flight Dynamics Directorate 8b. OFFICE SYMBOL (If applicable) WL/FIVR	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33600-88-D-0414
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433-6553	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.
62201F	2402	03	71

11. TITLE (Include Security Classification)
EXPLICIT FINITE ELEMENT METHOD FOR TRANSPARENCY IMPACT ANALYSIS

12. PERSONAL AUTHOR(S)
R. A. Brockman, T. W. Held

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM SEP88 TO SEP90	1991 June	170

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

structural analysis	soft-body impact	X3D
finite element	birdstrike	PATRAN
explicit methods	aircraft transparency	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report documents new analytical methods for the solution of soft-body impact problems, including birdstrike. The approach is based on explicit finite element techniques which allow for very detailed material modeling and make efficient use of the current generation of supercomputers. The most important improvements in the present work over previous efforts are in the areas of soft-body impact loading, material modeling, and treatment of layered wall constructions. Impact loads are obtained directly by including the impacting body in the finite element model, eliminating the need for ad hoc loading models. Both the transparency and the soft body materials may be rate-sensitive, exhibit nonlinear bulk behavior, and may fail during the finite element solution. Methods have been developed to make commercial graphics software (PATRAN) deal correctly with failed elements for results output. An improved model of laminated shells is included in the plate and shell element so that each layer of a multilayered transparency need not be modeled with distinct elements, even

(continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Richard A. Smith	(513) 255 - 2516	WL/FIVR

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

BLOCK 19 (continued)

when soft interlayers are present. The methods described have been implemented in an explicit finite element code X3D. User and programmer information are included in the report as appendices. Applications described in the report include two standard test problems and an F-16 birdstrike simulation.

FOREWORD

This report documents the development and application of new analytical methods for the solution of soft-body impact problems, including birdstrike. The effort was conducted between September 1988 and September 1990, by the Structures Group of the Aerospace Mechanics Division, University of Dayton Research Institute (UDRI), Dayton, Ohio, under Air Force Contract F33600-88-D-0414. The program was sponsored by the Air Force Aeronautical Systems Division, Scientific Applications Division, Wright-Patterson Air Force Base, Ohio. Air Force administrative direction was provided by Mr. Les Whitford, ASD/SCEA, and technical direction and support was provided by Mr. Richard A. Smith, WL/FIVR.

University of Dayton project supervision was provided by Mr. Dale H. Whitford, Supervisor, Aerospace Mechanics Division, and Mr. Blaine S. West, Head, Structures Group. The principal investigators were Dr. Robert A. Brockman and Mr. Thomas W. Held. The authors wish to acknowledge the contribution of Mr. Michael P. Bouchard, also of the UDRI Structures Group, for his assistance in bird model development and verification, and Mr. Michael Gran of WL/FIVR for his useful technical and editorial suggestions.

TABLE OF CONTENTS

Section		Page
	FOREWORD	iii
	LIST OF FIGURES	vii
1	INTRODUCTION	1
	1.1 TRANSPARENCY IMPACT PROBLEMS	1
	1.2 COMPUTER CODE X3D	2
2	THEORETICAL BACKGROUND	3
	2.1 EXPLICIT FINITE ELEMENT SOLUTION	3
	Principle of Virtual Displacements	3
	Finite Element Discretization	4
	Implicit Solution	6
	Explicit Solution	7
	2.2 FINITE ELEMENT TYPES	9
	Solid Hexahedral Element	9
	Solid Tetrahedral Element	11
	Plate and Shell Element	11
	2.3 MATERIAL MODELS	17
	Equation of State	17
	Elastic-Plastic Deviatoric Stress Model	18
	Plane Stress Elastic-Plastic Model	20
	Newtonian Fluid Model	21
	2.4 LAYERED PLATE AND SHELL MODEL	21
3	SAMPLE ANALYSES	25
	3.1 TAYLOR CYLINDER	25
	3.2 EXPLOSIVELY LOADED CYLINDRICAL SHEL	36
	3.3 F-16 BIRDSTRIKE SIMULATIONS	42
4	SUMMARY AND CONCLUSIONS	61
	REFERENCES	63
Appendix A.	X3D INPUT DATA	A.1
Appendix B.	PRE- AND POST-PROCESSING	B.1
Appendix C.	PROGRAMMING INFORMATION	C.1
Appendix D.	MATERIAL PROPERTY TABLES	D.1

LIST OF FIGURES

Figure	Title	Page
2.1	Explicit Solution Procedure	8
2.2	Eight-Node Solid Hexahedron Element	10
2.3	Hourglass Deformation Patterns for Solid Element	12
2.4	Four-Node Solid Tetrahedron Element	13
2.5	Four-Node Quadrilateral Plate Element	14
3.1	Finite Element Model of Taylor Cylinder Problem	26
3.2	Deformed Geometry of Taylor Cylinder, View #1	27
3.3	Deformed Geometry of Taylor Cylinder, View #2	28
3.4	Cylinder Length versus Time	29
3.5	Cylinder End Velocity versus Time	30
3.6	Cylinder Radius versus Time at Impact Surface	31
3.7	Cylinder Radial Velocity versus Time	32
3.8	Plastic Strains on Symmetry Plane at 80 μs	33
3.9	Plastic Strains on Impact Surface at 80 μs	34
3.10	Geometry of Explosively Loaded Cylindrical Shell	37
3.11	Final Deformed Shape of Cylindrical Shell	38
3.12	Plastic Strains in Explosively Loaded Shell	39
3.13	Mid-span Displacement History	40
3.14	Quarter-span Displacement History	41
3.15	Geometry of F-16 Transparency and Bird	43
3.16	Contact Element Grid for Bird Impact Problem	44
3.17	Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals)	47
3.18	Centerline Profiles of Deformed Canopy for 350-Knot Impact (4-ms Intervals)	52
3.19	Carpet Plot of Centerline Shape versus Time Step for Low-Strength Bird 350-Knot Impact	53
3.20	Deformed Geometry of F-16 Canopy for High-Strength Bird 350-Knot Impact (2-ms Intervals)	54
3.21	Deformed Geometry of Failed Canopy at 450-Knot Impact	56
B.1	Transparency Impact Deformed-Shape Plot With and Without Failed Elements Removed from the Display	B.11

SECTION 1
INTRODUCTION

This report documents the development of new computational techniques for analyzing structural response to high-speed impact. While much of the work is applicable to a variety of impact problems, the focus of the present study is the problem of *soft body impact* on aircraft transparencies, which presents a number of unique challenges (Brockman, 1988).

In what follows, we describe the elements of a practical method of analyzing soft body impact problems. The key improvements in the present technique include:

- *Soft-body impact loads:* the bird appears explicitly in the finite element model, so that ad hoc estimates of the impact loading distribution are unnecessary;

- *Material modeling:* the material models include strain rate sensitivity and failure; and

- *Layered shells:* multilayered constructions, including those with soft interlayers, can be modeled using a single layer of surface elements.

The results obtained thus far are encouraging, although more work is needed in the areas of model validation (versus experimental data) and materials characterization.

1.1 TRANSPARENCY IMPACT PROBLEMS

Impact phenomena encompass a broad range of structural behavior and response times, which depend upon the stiffness, strength, mass, geometry, velocities, and failure characteristics of the bodies involved. Soft body impacts, such as transparency birdstrikes, are distinctive among impact problems: while the response is often highly nonlinear, critical features of the response may occur either at early times or long (milliseconds) after the impact is finished.

The current generation of birdstrike analysis software (Brockman, 1982) has been in use for about ten years, and has been employed in a number of successful applications (McCarty, 1983, 1989a, 1989b; Bouchard, 1989). However, practical transparency analysis remains a time-consuming and laborious process, and in some circumstances the present inventory of analysis tools may not be optimal. For instance, an impact solution may be dominated by complicated contact conditions which preclude the use of large time steps, so that the advantages of an implicit solution are lost. Much of the material in this document reflects our continuing search for new methods which apply to unusual or more highly nonlinear situations.

The analytical technique discussed in this report is an *explicit* finite element method, of the type used widely for the numerical solution of shock and wave propagation problems. The explicit family of time integration algorithms is attractive because it is readily adapted to high performance on the current generation of supercomputers, which combine parallel or pipelined processors, moderate amounts of high-speed memory, and relatively slow disk performance. An added benefit is the ability to implement more detailed material and failure models. In subsequent sections, we present the underlying theory of the explicit technique, compare the characteristics of implicit and explicit integration algorithms, and address some of the problems which remain in the application of these methods to soft body impact simulation.

1.2 COMPUTER CODE X3D

The analytical methods described in the body of this report are implemented in a computer code called X3D. X3D is an explicit, three-dimensional finite element program intended for use in solving impact, wave propagation, and other short-duration problems in structural dynamics. A number of utility programs also exist for exchanging data between X3D and commercial modeling software, and for certain specialized data reduction tasks. The appendices of this document summarize the data preparation requirements for X3D, and the operation of the supporting software for pre- and postprocessing.

X3D contains two types of finite elements, solids and plates. The solid elements are an eight-node hexahedron, based on a mean-stress approximation with anti-hourglass stabilization, as described by Flanagan and Belytschko (Flanagan, 1981); and a four-node tetrahedron. The material model used for solids consists of a polynomial equation of state coupled with a von Mises plasticity model, a simple power-law correction for strain rate sensitivity, and a failure criterion based upon the ultimate stress.

The plate element in X3D is a four-node quadrilateral based upon a Mindlin-Reissner type thick-plate theory. A corotational axis system, which rotates with the element but does not deform, is used to simplify the element kinematics. The plate element uses a reduced (one-point) Gaussian quadrature, in conjunction with anti-hourglass stabilization techniques. An approximate model for layered media is implemented for the plate element, so that plates and shells having layers with large differences in stiffness can be represented effectively using a single element in the thickness direction. For each layer the material is elastic, perfectly plastic, and obeys plane stress assumptions. Transverse shear stresses in the plate element are uncoupled from the tangential stresses, and follow an elastic constitutive relation.

Applied forces in X3D may consist of body forces and surface pressure. Kinematic boundary conditions may include prescribed nodal displacements, rigid-wall constraints, and contact between specified surfaces within the mesh. Initial conditions may be specified for the translational velocity components for all or part of the finite element model.

Output from X3D consists of a printed solution log, a "trace" output file, and one or more restart files. A restart file is used to resume an analysis from the point of interruption; during a restart, selected data (e.g., external forces) may be replaced as needed to define the remainder of the problem. The restart file also is used for generating printed output and graphics. The trace output file contains results for a selected set of node points or elements. This form of output is useful for quick reviews of results at key points in a model, and for producing time-history plots.

SECTION 2
THEORETICAL BACKGROUND

This Section describes the underlying theory for each component of the impact solution: the explicit time integration procedure, the solid and plate finite element approximations, and the material models. For the plate element, the treatment of layered constructions is described separately in some detail.

2.1 EXPLICIT FINITE ELEMENT SOLUTION

This section presents a brief introduction to the theoretical aspects of the explicit finite element solution. The reader familiar with the more traditional implicit finite element solutions implemented in general purpose analysis codes will recognize most of the steps in this development. We begin with the principle of virtual work, and outline the process of spatial discretization common to all transient finite element solution techniques. The result of this step is a semidiscrete system which may be analyzed by numerous time integration algorithms (Belytschko, 1983). The remainder of the discussion shows how implicit and explicit time integration schemes derive from the same semi-discretized system of equations.

Principle of Virtual Displacements

The starting point for the explicit finite element solution consists of the momentum equations in spatial coordinates (Malvern, 1969):

$$\sigma_{ji,j} + \rho b_i = \rho a_i$$

Here σ_{ij} is the Cauchy stress (true stress), ρ is the current density, b_i is the prescribed body force per unit mass, and a_i are the acceleration components. A comma is used to denote partial differentiation with respect to the spatial coordinates x_i; for example, $f_{,i} = \partial f/\partial x_i$. A superimposed dot denotes a material time derivative. The boundary conditions state that either the displacement assumes a prescribed value, or that the traction condition

$$n_j \sigma_{ji} = t_i$$

is satisfied. The prescribed tractions are denoted by t_i, and n_i are the components of the outward normal to the boundary. One and only one of these conditions applies at every point of the boundary, in each of three linearly independent directions. On contacting surfaces (which we identify by superscripts 'α' and 'β'), the velocities satisfy the unilateral constraint:

$$\left(v_i^{\beta} - v_i^{\alpha}\right) n_i^{\alpha} \geq 0$$

Whenever the geometric constraint is an equality, the normal component of the tractions on the opposing surfaces must be equal and opposite:

$$\sigma_{ij}^{\alpha} n_i^{\alpha} n_j^{\alpha} - \sigma_{ij}^{\beta} n_i^{\beta} n_j^{\beta} = 0$$

We will assume that the kinematic boundary conditions are satisfied, including the unilateral contact constraints. The weak form of the system above is (Reddy, 1984b):

$$\int_V \left[\sigma_{ji,j} + \rho b_i - \rho a_i\right] \delta v_i \, dV + \int_{A_\sigma} \left[t_i - n_j \sigma_{ji}\right] \delta v_i \, dA - \int_{A_\alpha} \left[\sigma_{ij}^{\alpha} n_i^{\alpha} n_j^{\alpha} - \sigma_{ij}^{\beta} n_i^{\beta} n_j^{\beta}\right] \delta v_k^{\alpha} n_k^{\alpha} \, dA = 0$$

The test functions δv_i must vanish where the displacements are prescribed but are arbitrary and independent elsewhere. If δv_i are interpreted as a virtual velocity field, the weak form above represents the rate at which existing forces do work and corresponds to the principle of virtual work. Applying Gauss' divergence theorem and noting that every point of the boundary must belong to A_σ, A_α, the segment A_β corresponding to A_α, or the remainder of the boundary A_u on which δv_i vanishes, one obtains:

$$\int_V \sigma_{ji} \delta v_{i,j} \, dV + \int_V \rho a_i \delta v_i \, dV - \int_V \rho b_i \delta v_i \, dV - \int_{A_\sigma} t_i \delta v_i \, dA = 0$$

The unknown quantities in the virtual work equation are the velocity components v_i, which are the fundamental unknowns, and the stresses σ_{ij}, which are determined from v_i using the appropriate constitutive relations. The interpretation of the virtual work principle $\delta W = 0$ is as follows: among all kinematically admissible velocity fields $v_i(\mathbf{x},t)$, the velocity field which makes $\delta W = 0$ vanish for arbitrarily chosen test functions δv_i is the true velocity field.

Finite Element Discretization

Constructing a finite element approximation of the principle involves the following steps:

- express δW as a sum of contributions from individual elements:

$$\delta W = \sum_{e=1}^{N} \delta W^{(e)}$$

- approximate the velocity field within each element in terms of discrete nodal values:

$$v_i^{(e)}(x,t) = N_J(x) \, V_{iJ}(t)$$

- approximate the arbitrary test functions δv_i within an element in terms of discrete nodal values:

$$\delta v_i^{(e)}(x,t) = N_J(x) \, \delta V_{iJ}(t)$$

in which δV_{iJ} are arbitrary and independent.

The $N_J(x)$ are known functions of position, called *shape functions;* with this approximation, the unknowns are reduced to the set of nodal values V_{iJ}, which are functions solely of time. Therefore the resulting system will consist of *ordinary differential equations,* with time the independent variable. The division of the virtual work expression into contributions from disjoint elements requires that either v_i be continuous across the element boundaries, or that jump conditions be introduced to account for the unboundedness of the derivatives $v_{i,j}$. We choose to select the shape functions $N_J(x)$ so that the velocity field remains continuous everywhere.

Introducing the finite element approximations in the virtual work expression gives the semidiscrete equation:

$$\sum_{e=1}^{N}\left[\int_V \sigma_{ji} N_{K,j}\, dV + \int_V \rho N_K N_L \dot{V}_{iL}\, dV - \int_V \rho b_i N_K\, dV - \int_{A_\sigma} t_i N_K\, dA\right]\delta V_{iK} = 0$$

Since the nodal test function values δV_{iK} are arbitrary and independent, the coefficient of δV_{iK} (consisting of the sum of all bracketed terms) must vanish. This condition yields one equation corresponding to each unconstrained nodal value δV_{iK} (or V_{iK}).

We define the internal nodal force vector,

$$F_{iK}^{int} = \sum_{e=1}^{N} \int_V \sigma_{ji} N_{K,j}\, dV$$

the mass matrix,

$$M_{KL} = \sum_{e=1}^{N} \int_V \rho N_K N_L\, dV$$

and the external force vector,

$$F_{iK}^{ext} = \sum_{e=1}^{N} \int_V \rho b_i N_K\, dV + \int_{A_\sigma} t_i N_K\, dA$$

Then the semidiscrete equations of motion for the finite element model are simply:

$$M_{KL} \dot{V}_{iL} = F_{iK}^{ext} - F_{iK}^{int}$$

subject to the initial conditions and the prescribed displacement/velocity boundary conditions. To this point, the formulation of both implicit and explicit solution techniques is the same, although the best choice for the element shape functions often is different for the two classes of temporal solutions.

Implicit Solution

The most common approach to solving the semidiscrete equations of motion in structural dynamic (as opposed to wave propagation) problems is to adopt an implicit finite difference approximation in time for each degree of freedom. We present a brief summary of this approach below, for comparison with the explicit technique.

A common choice for the implicit integration method is the trapezoidal rule (Newmark, 1959; Hughes, 1983):

$$\frac{V_{iK}^{(t+\Delta t)} + V_{iK}^{(t)}}{2} \approx \frac{U_{iK}^{(t+\Delta t)} - U_{iK}^{(t)}}{\Delta t}$$

$$\frac{\dot{V}_{iK}^{(t+\Delta t)} + \dot{V}_{iK}^{(t)}}{2} \approx \frac{V_{iK}^{(t+\Delta t)} - V_{iK}^{(t)}}{\Delta t}$$

When the equation of motion is applied at time t+Δt (assuming the state at time t is known), the difference formulas can be used to eliminate all of the unknowns except the nodal displacements at the end of the time step:

$$\frac{4}{\Delta t^2} M_{KL} U_{iL}^{(t+\Delta t)} + F_{iK}^{int}\left[U^{(t+\Delta t)}\right] = F_{iK}^{ext} + M_{KL}\left[\dot{V}_{iL}^{(t)} + \frac{4}{\Delta t} V_{iL}^{(t)} + \frac{4}{\Delta t^2} U_{iL}^{(t)}\right]$$

If the internal forces depend linearly on the nodal displacements at time t+Δt, that is,

$$F_{iK}^{int}\left[U^{(t+\Delta t)}\right] = K_{iKjL} U_{jL}^{(t+\Delta t)}$$

then the system resembles an equilibrium system with the "effective" stiffness coefficients

$$K^{eff} = K + \frac{4}{\Delta t^2} M$$

which remain constant unless properties or constraints change or the time step is modified. If the system is nonlinear, the algebraic system must be solved iteratively (usually by some variant of Newton's method).

Since the balance equations at time t+Δt determine the state of the system at t+Δt, the stability properties of the implicit technique are quite good For linear systems, the trapezoidal rule used above is known to be *unconditionally stable;* that is, the numerical integration remains stable for any choice of the time step. The method is accurate for frequency components whose period is much larger than Δt; when the period and time step are similar in magnitude, accuracy degrades quickly. This "low-pass filter" behavior is responsible for the favorable stability properties of the implicit family of integration methods (Hughes, 1983).

A transparency model of moderate size may contain upwards of 10,000 degrees of freedom, and the stiffness K(stored in sparse format) occupies several million words of computer storage. While the computational effort required to form, triangularize, and solve the matrix at each iteration is considerable, the most serious problems with efficiency involve input and output. Present-day supercomputers are still too small to cope with the system in high-speed memory, so that disk transfers become the limiting factor in problem throughput.

Explicit Solution

The explicit solution approach is preferred in the solution of problems of wave propagation and short-duration impact response. The temporal discretization uses the central difference formulas:

$$V_{iK}^{(t+\frac{1}{2}\Delta t)} \approx \frac{U_{iK}^{(t+\Delta t)} - U_{iK}^{(t)}}{\Delta t}$$

$$\dot{V}_{iK}^{(t)} \approx \frac{V_{iK}^{(t+\frac{1}{2}\Delta t)} - V_{iK}^{(t-\frac{1}{2}\Delta t)}}{\Delta t}$$

A typical step in the solution consists of applying the equations of motion at time t to determine the accelerations:

$$M_{KL}\dot{V}_{iL}^{(t)} = F_{iK}^{ext} - F_{iK}^{int}$$

and then using the finite difference formulas to obtain new velocities

$$V_{iK}^{(t+\frac{1}{2}\Delta t)} = V_{iK}^{(t-\frac{1}{2}\Delta t)} + \Delta t\,\dot{V}_{iK}^{(t)}$$

and positions:

$$U_{iK}^{(t+\Delta t)} = U_{iK}^{(t)} + \Delta t\,V_{iK}^{(t+\frac{1}{2}\Delta t)}$$

The new displacement and velocity solutions are used to update the element stresses and internal forces in preparation for the next time step (see Figure 2.1).

In practice, the mass matrix M_{KL} is *lumped* (made diagonal), so that no equation-solving is required, and in fact the system matrices need never be formed. Krieg and Key (Krieg, 1973) show that mass lumping is desirable in explicit solutions, as it tends to counteract the temporal discretization error.

The explicit integration process is *conditionally stable;* that is, stability is assured only if the time step is less than a critical value. For the central difference method:

$$\Delta t \leq \frac{2}{\omega_{max}}$$

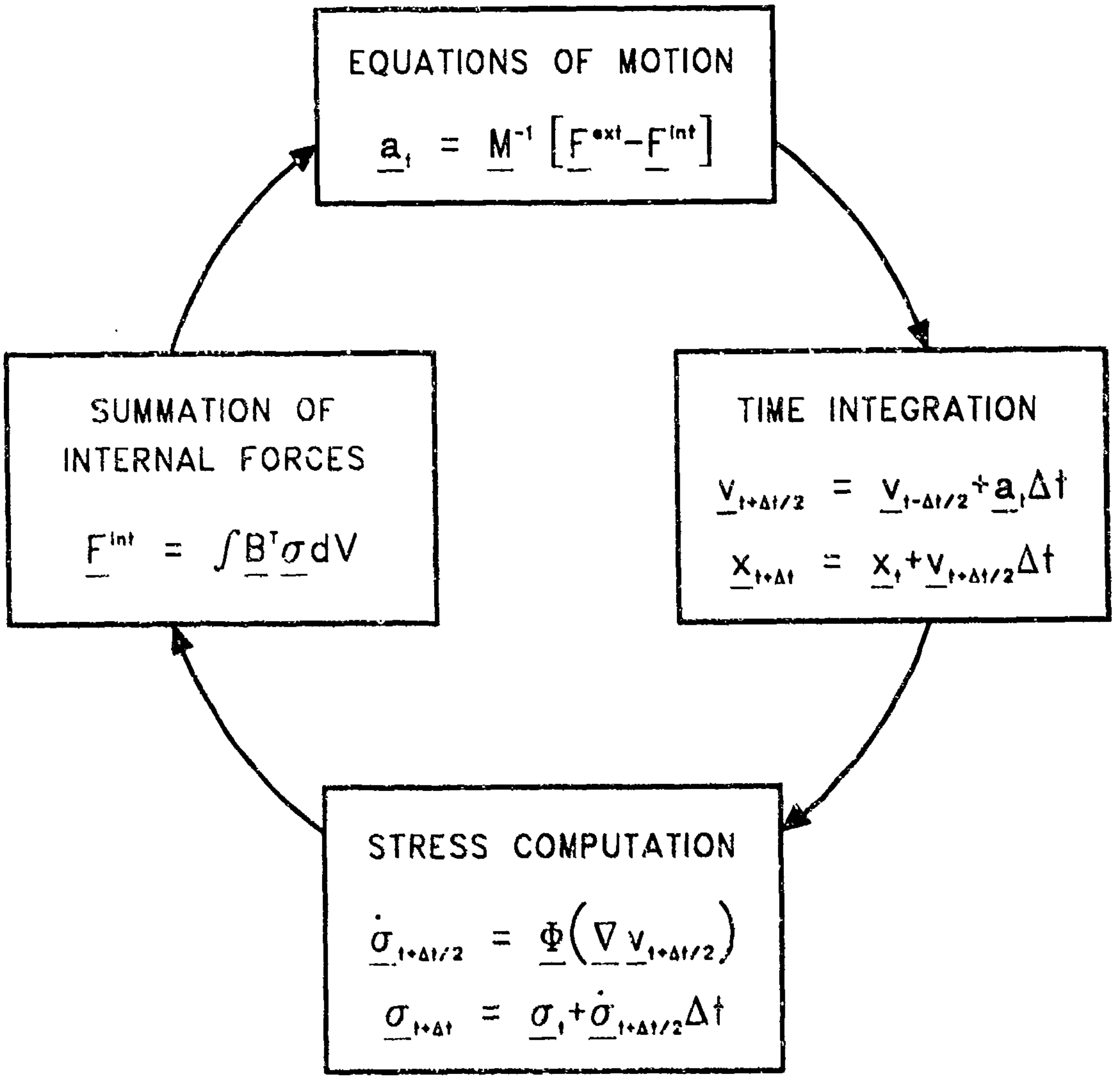

EQUATIONS OF MOTION
$\underline{a}_t = \underline{M}^{-1}\left[\underline{F}^{ext}-\underline{F}^{int}\right]$
TIME INTEGRATION
$\underline{v}_{t+\Delta t/2} = \underline{v}_{t-\Delta t/2}+\underline{a}_t\Delta t$
$\underline{x}_{t+\Delta t} = \underline{x}_t+\underline{v}_{t+\Delta t/2}\Delta t$
STRESS COMPUTATION
$\dot{\underline{\sigma}}_{t+\Delta t/2} = \underline{\Phi}\left(\underline{\nabla}\,\underline{v}_{t+\Delta t/2}\right)$
$\underline{\sigma}_{t+\Delta t} = \underline{\sigma}_t+\dot{\underline{\sigma}}_{t+\Delta t/2}\Delta t$
SUMMATION OF INTERNAL FORCES
$\underline{F}^{int} = \int\underline{B}^T\underline{\sigma}\,dV$

Figure 2.1. Explicit Solution Procedure.

where ω_{max} is the highest natural frequency of the discrete system. In practice we bound the highest frequency by determining the maximum *element* frequency (Belytschko, 1983), a calculation which can be performed very quickly at each time step.

Because explicit techniques are best suited for solving short-duration dynamic problems, which typically involve nonlinear response, they are less familiar to most engineers than the implicit methods used most commonly for statics and structural dynamics. Explicit finite element methods were first used in the early 1970's (Chiapetta, 1973; Key, 1974) and now are used extensively for analyzing ballistic and hypervelocity impact, detonation physics, crash response, forging and high-speed forming, and other highly nonlinear dynamic events (Johnson, 1979; Goudreaux, 1982; Zukas, 1982; Anderson, 1987).

2.2 FINITE ELEMENT TYPES

Three finite elements are implemented in X3D: an eight-node solid hexahedron, a four-node solid tetrahedron, and a four-node plate or shell element. In the explicit solution, the distinguishing features of a finite element lie in the way its mass properties and internal forces are computed. We describe each of the three elements in some detail below.

Solid Hexahedral Element

The solid hexahedral finite element (Figure 2.2) uses a displacement and velocity approximation based upon trilinear polynomials; that is, the element's displacement and velocity components each vary linearly along each edge of the element. In addition, we compute the stress components from a *mean stress approximation*, using only the mean velocity gradient for the element (Flanagan, 1981). This measure is desirable to maintain good element performance, and also reduces the effort required for element computations. However, the resulting element is a *constant stress element*, and therefore a generous number may be necessary for accurate modeling. In particular, a single layer of these solids is incapable of developing a bending moment.

Besides the stress relation, the key computation performed for the solid element involves the shape function derivatives needed to form the mean velocity gradient and the internal forces. The mean velocity gradient is defined as:

$$\bar{v}_{i,j} = \frac{1}{V}\int_V v_{i,j}\,dV \approx \frac{1}{V}V_{iK}\int_V N_{K,j}\,dV = \frac{1}{V}B_{jK}V_{iK}$$

The "B matrix" B_{jK} also is important in evaluating the internal force vector, since

$$F_{iK}^{int} = \int_V \bar{\sigma}_{ji}N_{K,j}\,dV = B_{jK}\bar{\sigma}_{ji}$$

Because of the mean stress approximation, certain modes of deformation exist for the hexahedron which are stress-free but do not represent rigid body motions. These *hourglassing deformation*

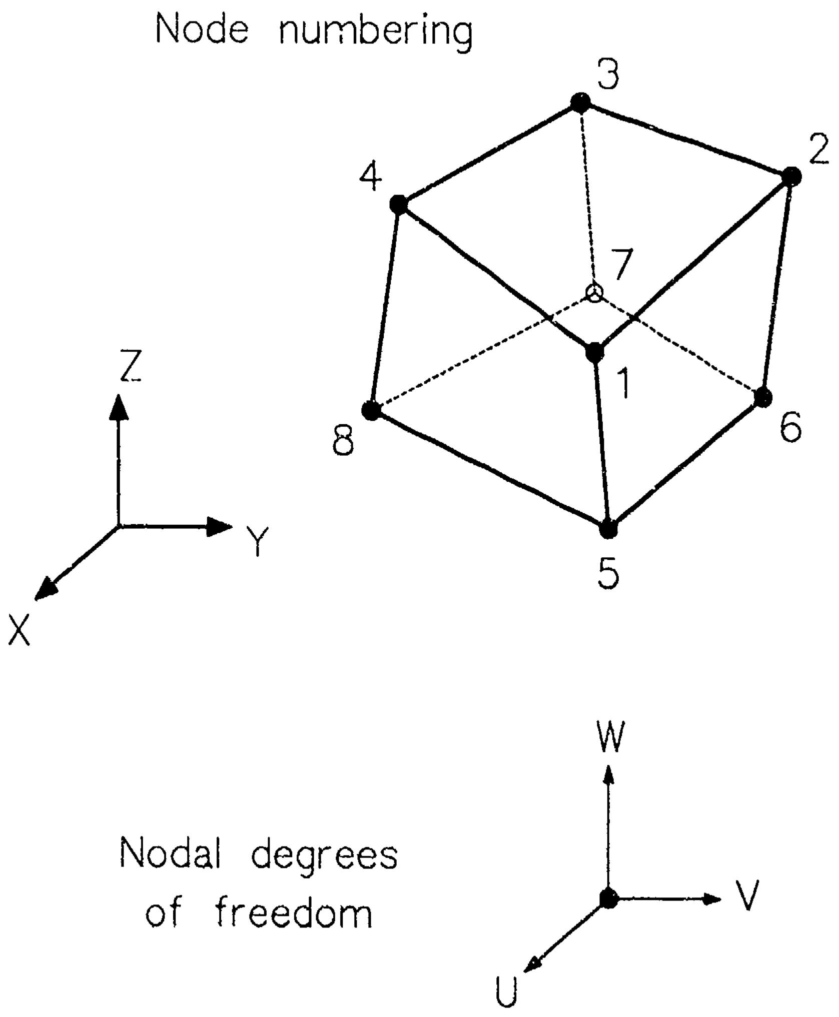


Figure 2.2. Eight-Node Solid Hexahedron Element.

modes correspond to linearly varying stress patterns which are not detected by the mean stress approximation (Figure 2.3). To stabilize these potentially unstable motions, we employ an *anti-hourglass viscosity* to resist the hourglass motions through internal damping forces, as described by Flanagan (1981) and Brockman (1984).

Solid Tetrahedral Element

The tetrahedral solid (Figure 2.4) is a constant-strain, constant-stress element based upon a fully linear displacement and velocity field approximation. The element is quite similar to the hexahedron of the previous subsection, but does not use an anti-hourglass viscosity. The twelve degrees of freedom for the element capture the six rigid-body motions and the six uniform strain/stress modes, so that no unstable deformation patterns exist for individual elements.

The tetrahedron is included in X3D for its utility in soft-body impact modeling. Since the element has no unstable modes, it can be used to follow very large distortions without causing numerical problems. We use the tetrahedron to model the bird in bird impact simulations, using an equation of state typical of water, a very low strength deviatoric model, and an ultimate failure strain of about 5 (500%).

The tetrahedron usually is inconvenient for modeling because of its basic shape: even filling a rectangular region with tetrahedra can be a confusing process. However, it is possible to generate tetrahedral elements automatically with PATRAN and other modeling packages.

In X3D, the tetrahedron is implemented as a *five-node* element, the fifth node coinciding with the first. This artifice serves to distinguish the four-node tetrahedron from the four-node quadrilateral plate element during input. Ordering of the nodes of the tetrahedron is not crucial. If the element is defined "inside out", X3D reverses the connections to define a valid element.

Plate and Shell Element

The plate and shell element in X3D is a four-node quadrilateral with six degrees of freedom per node (Figure 2.5). The displacement and rotation components each are interpolated separately, using bilinear polynomials. The resulting element is quite similar to that described in (Belytschko, 1984).

Unlike the solid elements, the plate element must be formulated in a local axis system because of the differing treatment of the plate thickness from that of the planform directions. We employ a *corotational coordinate system* which rotates with the element (Belytschko, 1973), and therefore is constructed anew at each time step of the solution based upon the current element geometry. The element local coordinate system is shown in Figure 2.5. We summarize the construction of the local axis system below, since the axis directions are potentially useful in interpreting the finite element results.

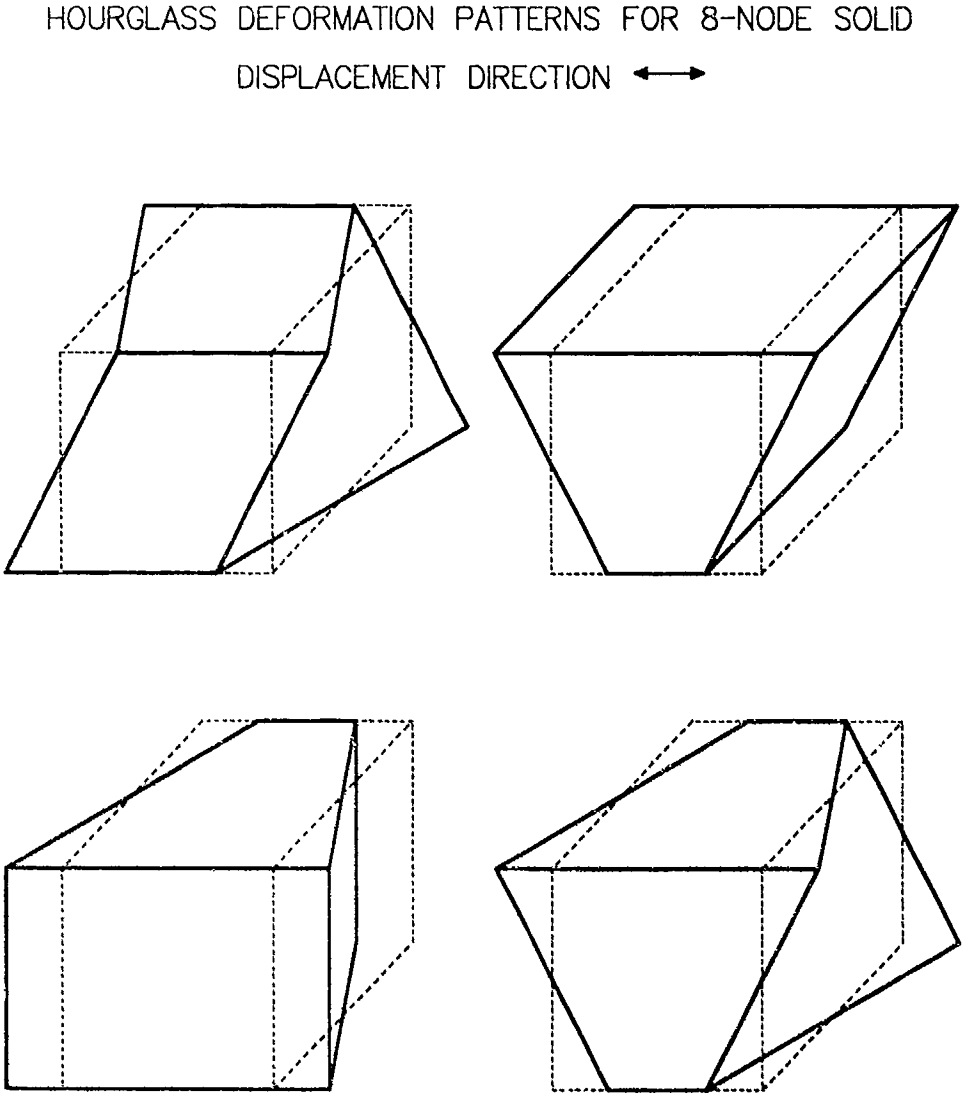

HOURGLASS DEFORMATION PATTERNS FOR 8-NODE SOLID
DISPLACEMENT DIRECTION ←→

Figure 2.3. Hourglass Deformation Patterns for Solid Element.

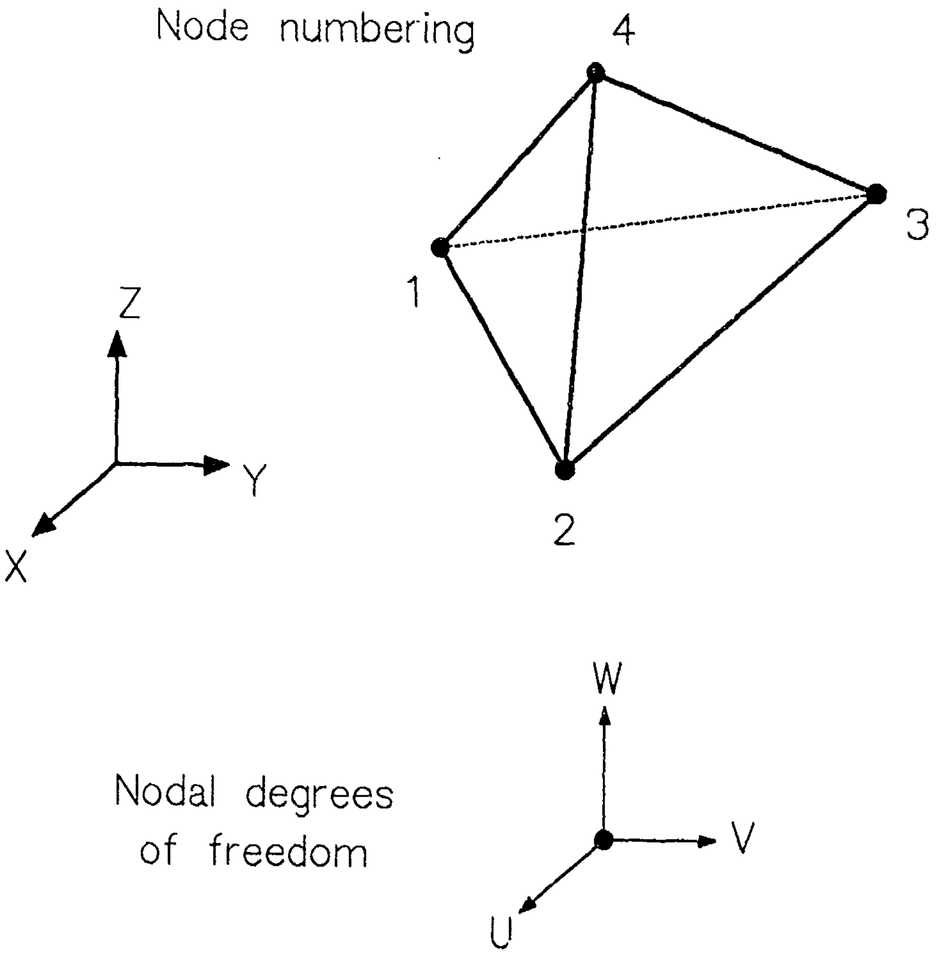


Figure 2.4. Four-Node Solid Tetrahedron Element.

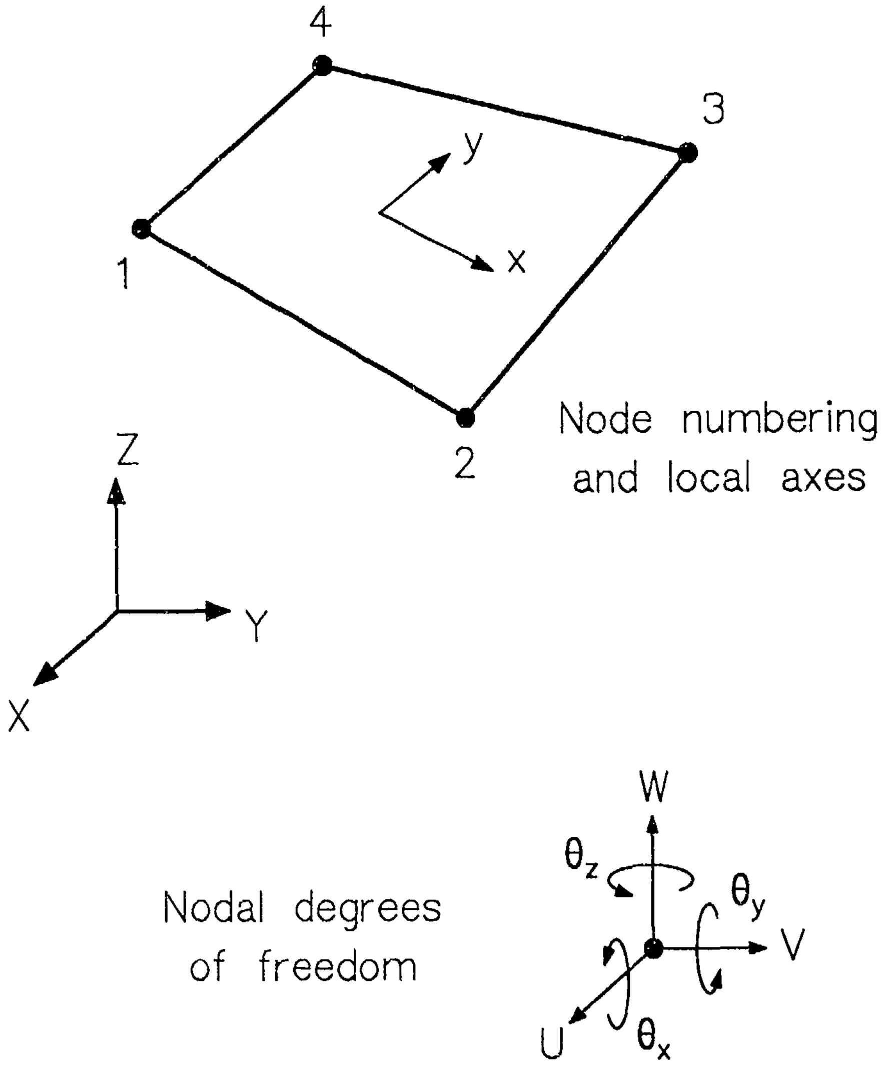


Figure 2.5. Four-Node Quadrilateral Plate Element.

Let X_i, Y_i, and Z_i denote the global coordinates at the i^{th} element corner. We will denote the differences in nodal coordinates by double-subscripted quantities; for instance, for two nodes 'i' and 'j' within an element:

$$X_{ij} = X_i - X_j \qquad Y_{ij} = Y_i - Y_j \qquad Z_{ij} = Z_i - Z_j$$

We represent a vector from the i^{th} node to the j^{th} node by:

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j = X_{ij}\hat{i} + Y_{ij}\hat{j} + Z_{ij}\hat{k}$$

The local axis transformation for the plate element is computed as follows:

$$\vec{s}_3 = \vec{r}_{31} \times \vec{r}_{42}$$

$$\vec{e}_3 = \frac{\vec{s}_3}{\| \vec{s}_3 \|}$$

$$\vec{s}_1 = \vec{r}_{21} - \left(\vec{r}_{21} \cdot \vec{e}_3 \right) \vec{e}_3$$

$$\vec{e}_1 = \frac{\vec{s}_1}{\| \vec{s}_1 \|}$$

$$\vec{e}_2 = \vec{e}_3 \times \vec{e}_2$$

$$[A]^T = \left[\vec{e}_1 \mid \vec{e}_2 \mid \vec{e}_3 \right]$$

The rows of **A** are the basis vectors for the local coordinates; that is:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{local} = [A] \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}_{global}$$

The point (X_0, Y_0, Z_0) is the origin of the local coordinate system, which is placed at the element center.

The plate element shape functions are formulated entirely in local coordinates. The element calculations are performed with respect to the "mean plane" of the element, and corrected as necessary to account for out of plane warping of the reference surface. Denoting the local coordinates by lower-case symbols (x_i, y_i, z_i), the shape function derivatives ior the bilinear four-node element are:

$$N_{,x} = \frac{1}{2A}\begin{bmatrix} y_{24} \\ y_{31} \\ y_{42} \\ y_{13} \end{bmatrix} \qquad N_{,y} = \frac{1}{2A}\begin{bmatrix} x_{42} \\ x_{13} \\ x_{24} \\ x_{31} \end{bmatrix}$$

Here A is the element planform area,

$$A = ½\left(x_{31}y_{42} + x_{24}y_{13}\right)$$

If a vector f contains the nodal values $[f_1, f_2, f_3, f_4]$ of a function f(x,y) defined on the element, then:

$$\left.\frac{\partial f}{\partial x}\right|^{x=0,y=0} = N_{,x}^{T} f \qquad \left.\frac{\partial f}{\partial y}\right|^{x=0,y=0} = N_{,y}^{T} f$$

The plate uses a mean-stress approximation for its inplane directions, similar to the solid hexahedron. At any thickness station, the velocity gradient is evaluated at the centroid of the element, and assumed to be constant throughout the element (except through the thickness). To resist unstable motions resulting from the assumption of a uniform velocity gradient, the X3D plate element uses a stiffness hourglass control scheme (Flanagan, 1981).

Integration through the shell thickness uses Simpson's Rule, with the number of points controlled by user input. If h is the thickness of a plate element (or a layer of a laminated element), and N is the number of integration points, the integration points and weights for an N-point Simpson's rule (N odd) are defined by:

$$z_k = \left[\frac{k-1}{N-1} - ½\right]h$$

$$W_k = \frac{h}{3(N-1)}[\ 1,\ 4,\ 2,\ 4,\ 2, \ldots, 4,\ 1\ ]$$

The integral of an arbitrary function f(z) is evaluated using:

$$\int_{-h/2}^{h/2} f(z)\,dz = \sum_{k=1}^{N} f(z_k)\,W_k$$

Notice that an integration point always is present at the extreme surfaces of any layer, so that yielding in any layer is captured immediately as it occurs.

The integration rule shown above applies *within each layer* of a plate element. For an element with 'L' layers, the force and moment resultants are:

$$N_{ij} = \int_{-h/2}^{h/2} \sigma_{ij}\, dz = \sum_{l=1}^{L} \sum_{k=1}^{N} W_k^{(l)} \sigma_{ij}\left(z_k^{(l)}\right)$$

$$M_{ij} = \int_{-h/2}^{h/2} z\sigma_{ij}\, dz = \sum_{l=1}^{L} \sum_{k=1}^{N} W_k^{(l)} z_k^{(l)} \sigma_{ij}\left(z_k^{(l)}\right)$$

Since point stress relations are used in the constitutive models (rather than, say, moment-curvature relationships), each layer may be a different material, and even use a different material model. Layered constructions in which the stiffness characteristics vary dramatically from layer to layer require special treatment; this question is addressed in subsection 2.4.

Formulation of the lumped mass coefficients for a plate element is an intriguing issue. Calculation of the rotational inertias by Lobatto rules (a common method of lumping) leads to very strict time step restrictions which are not justified by the accuracy obtained. Hughes, Cohen, and Haroun (Hughes, 1978) have described an alternative scheme for computing the rotational lumped masses which relieves the stringent time step restriction without upsetting convergence, and which is used in X3D. The lumped mass coefficients are, for a homogeneous element, $M_{\underline{\alpha\alpha}}$= ρAh/4 for the translational degrees of freedom, and $M_{\underline{\alpha\alpha}}$= ρAh/48 for the bending rotations. No inertia is assigned to the "drilling" rotation θ_z in the local coordinate system.

2.3 MATERIAL MODELS

The material constitutive relationships used for both the solid and plate finite elements consist of a *deviatoric* (shear) relation and a *bulk* (pressure-volume) model. The hydrostatic pressure (mean compressive stress) is defined by:

$$p = -\frac{1}{3}\sigma_{kk}$$

We take the stress tensor to be composed of a hydrostatic, or pressure, stress, and a deviatoric stress tensor σ' which is independent of the pressure:

$$\sigma_{ij} = -p\delta_{ij} + \sigma'_{ij}$$

These two contributions to the stress tensor are determined independently in the material model by a *deviatoric stress model* and a *mechanical equation of state.*

Equation of State

The equation of state, which describes pressure-volume behavior, relates the hydrostatic pressure p to the compression ratio η as follows:

$$-p = K_1\eta \qquad (tension, \eta < 0)$$

$$-p = K_1\eta + K_2\eta^2 + K_3\eta^3 \qquad (compression, \eta > 0)$$

in which

$$\eta = \frac{\rho - \rho_0}{\rho_0}$$

so that $\eta > 0$ implies compression of the element. In the material model subroutines, the corresponding rate forms are used:

$$\frac{d\eta}{dt} = -(1+\eta)d_{kk} \qquad \frac{dp}{dt} = -K_{eff}d_{kk}$$

Note that the effective bulk modulus (which is used in computing allowable time steps) is

$$K_{eff} = \rho_0\frac{dp}{d\rho}$$

or

$$K_{eff} = K_1 + 2K_2\eta + 3K_3\eta^2$$

The input parameters which define the pressure-volume behavior of the material are:

ρ_0	reference state density
K_1, K_2, K_3	linear, quadratic, and cubic bulk coefficients

Elastic-Plastic Deviatoric Stress Model

The deviatoric material model used for solids is a rate-dependent, isotropic, hypoelastic theory appropriate for moderate to large deformations. A brief summary of the model is given below.

The rate of deformation tensor **D** and spin tensor **W** are the primary kinematic quantities used in the model. They are defined in terms of the velocity gradient in current coordinates as:

$$d_{ij} = ½\left(v_{i,j} + v_{j,i}\right) \qquad w_{ij} = ½\left(v_{i,j} - v_{j,i}\right)$$

The elastic part of the model assumes a linear relationship between the deviatoric part of the rate of deformation at a point and the time derivative of the deviatoric Cauchy (true) stress, as seen by an observer which rotates with the material:

$$\overset{\nabla\prime}{\sigma_{ij}} = 2Gd_{ij}^{\prime}$$

A stress rate of this type is said to be a *corotational* rate. The particular corotational rate used here is the *Jaumann rate;* it is related to the material time derivative of the true stress by:

$$\frac{d\sigma'_{ij}}{dt} = \overset{\nabla\prime}{\sigma}_{ij} + \sigma'_{ik}w_{kj} - w_{ik}\sigma'_{kj}$$

Notice that, if no rotation occurs at a point, the spin tensor **W** vanishes and the two rates are the same; otherwise, the state of stress must be "rotated" at each step as indicated above.

The severity of the current stress state is measured in terms of the *von Mises effective stress* σ_e:

$$\sigma_e = \sqrt{\frac{3}{2}\sigma'_{ij}\sigma'_{ij}}$$

The effective stress is a scalar quantity which may be compared with the yield stress in simple tension to determine when yielding occurs. The current yield stress is a rate-dependent quantity which is assumed to vary with strain rate and effective plastic strain according to:

$$\sigma_y = \sigma_y^0\left[1 + \left(\frac{\dot{e}}{D}\right)^{1/p}\right] + H\varepsilon_p$$

The effective strain rate is defined by:

$$\dot{e} = \sqrt{\frac{2}{3}d'_{ij}d'_{ij}}$$

The *hardening modulus* H, also used in the definition of the current yield stress, is defined by

$$H = \frac{EE'}{E - E'}$$

Here E is the elastic modulus, and E' is the slope of the uniaxial stress-strain curve past the point of yielding. Note that in the case of unaxial stress, both the von Mises stress and the effective strain rate reduce to the corresponding axial components.

The elastic model and yield criterion above are applied at each integration point in the model for each time step. The elastic stress, sometimes called a *trial stress*, is correct if $\sigma_e \leq \sigma_y$; that is, provided the effective stress is within the current yield surface. Plastic flow is assumed to occur whenever $\sigma_e > \sigma_y$. In this case the computed stresses are returned to the yield surface using the radial return correction:

$$\sigma'_{ij} \leftarrow \sigma'_{ij}\frac{\sigma_y}{\sigma_e}$$

When plastic flow occurs, the updated effective plastic strain and the effective shear modulus (used in determining the allowable time step) are evaluated using:

$$\varepsilon_p \leftarrow \varepsilon_p + \frac{\sigma_e - \sigma_y}{3G+H} \qquad G_{eff} = \frac{GH}{3G+H}$$

The parameters which define the material's deviatoric behavior with this model are:

G	linear shear modulus
σ_y	quasi-static yield stress
1/D	rate sensitivity scale factor
1/p	rate sensitivity exponent
H	hardening modulus
σ_{ult}	ultimate stress

Notice that H = 0 for materials which exhibit no work hardening, and that setting 1/D = 0 and 1/p = 1 causes the yield stress to be independent of the strain rate.

Plane Stress Elastic-Plastic Model

In plate elements, the elastic-plastic material model is slightly more complicated than for three-dimensional solids because of the *zero normal stress constraint*, $\sigma_{zz} = 0$. During the plasticity calculation, it is necessary to determine a final state of stress which not only lies on the yield surface, but which satisfies the condition $\sigma_{zz} = 0$. In terms of the deviatoric stress, we require:

$$\sigma'_{zz} = p$$

This means that the deviatoric model and the bulk model (equation of state) are not entirely independent, and must be solved simultaneously with the normal stress constraint.

Since the volumetric strain in plate and shell structures tends to be relatively small, we employ a linear equation of state in the plane stress case,

$$p = -K\varepsilon_{kk}$$

with

$$K = \frac{E}{3(1-2\nu)}$$

being the bulk modulus. If the time step involves plastic flow, the pressure obtained in this way must be determined iteratively.

When plastic flow occurs, the three-dimensional plasticity equations are solved, using an assumed value of the normal strain rate. The starting estimate is the elastic value

$$\frac{d\varepsilon_{zz}}{dt} = -\nu\left(\frac{d\varepsilon_{xx}}{dt} + \frac{d\varepsilon_{yy}}{dt}\right)$$

Notice that the value assumed for the normal strain rate also affects the pressure. The value of ε_{zz} which gives a final stress state with $\sigma_{zz} = 0$ is determined using a Newton-Raphson iteration:

$$\dot{\varepsilon}_{zz} \leftarrow \dot{\varepsilon}_{zz} - \frac{\sigma_{zz}}{\partial\sigma_{zz}/\partial\dot{\varepsilon}_{zz}}$$

The derivative is estimated by differencing about the current estimate of ε_{zz}. Each cycle of the iteration requires an application of the three-dimensional plasticity model, the (linear) equation of state, and the radial return correction. Typically, the point stress solution requires two or three iterations to converge, even for strain steps on the order of several percent.

Newtonian Fluid Model

The isotropic Newtonian fluid model provided for the three-dimensional solid elements is an experimental feature intended for potential use in hydrodynamic impact modeling. The bulk behavior is described in polynomial form as for a solid, while the deviatoric stress is related linearly to the rate of deformation:

$$\sigma_{ij} = -p\delta_{ij} + 2\mu d'_{ij}$$

Here p is determined from the polynomial equation of state. The constant μ is the *dynamic viscosity* of the fluid, in force length-time units (i.e., FT/L^2).

2.4 LAYERED PLATE AND SHELL MODEL

This section presents a method of approximation for plates and shells having large stiffness variations from layer to layer. Layered structures of this type often require detailed and expensive models, since conventional plate and shell finite elements do not reproduce the correct transverse shear strain distributions through the wall thickness. The present approach requires a single layer of elements having six engineering degrees of freedom per node, regardless of the number of layers.

The approximation discussed here uses closed-form elasticity solutions to develop transverse shear flexibility corrections, which bring this contribution to the energy into line with that caused by pure bending, twisting, and extension. For large displacement problems, the technique is applied in corotational coordinates. Changes in stiffness caused by plasticity can be accounted for by recomputing the flexibility corrections based upon instantaneous moduli.

The Mindlin-Reissner theory of plates (Mindlin, 1951) is based upon kinematic assumptions through which the displacement components (U,V,W) at a generic point in the plate are determined by the midsurface components (u,v,w) and two cross-sectional rotations (θ_x,θ_y). If z is the direction normal to the plate midsurface, then:

$$U(x,y,z) = u(x,y) + z\theta_y(x,y)$$

$$V(x,y,z) = v(x,y) - z\theta_x(x,y)$$

$$W(x,y,z) = w(x,y)$$

The assumption that the tangential displacements (U,V) vary linearly through the plate thickness provides an extremely crude representation of the transverse shear strain field. For monolithic, isotropic elements, a uniform reduction factor often is applied to the shear strain energy to obtain more realistic behavior. Equating the transverse shear strain energy consistent with the assumed displacements to that of a parabolic strain field satisfying the equilibrium conditions yields the correction factor of 5/6, which is used commonly for isotropic plates and shells.

In the present development, we rely on a generalization of this idea first proposed by Whitney (Whitney, 1973) for arbitrary wall constructions. The correction is approximate, but proves sufficient to bring the shear strain energy into line with that of other modes of deformation, in a way which reflects the relative flexibility of these modes for a given material layup. Several investigators have employed a similar concept for fiber-reinforced composite laminates (Chow 1971, 1975; Noor 1975, 1989; Chatterjee, 1979).

Consider first a layered construction for which the shear strains and resultant forces are related by:

$$\begin{bmatrix} Q_{xz} \\ Q_{yz} \end{bmatrix} = \begin{bmatrix} k_1 A_{44} & 0 \\ 0 & k_2 A_{55} \end{bmatrix} \begin{bmatrix} \gamma_{xz} \\ \gamma_{yz} \end{bmatrix}$$

Based solely on the elastic stress-strain relationship of the material, factors k_1 and k_2 both should equal one. However, due to the excessive constraint imposed by the kinematic assumptions of the plate theory, the strains γ_{iz} produced by given shear forces Q_{iz} are too large over much of the plate thickness. Accordingly, the total strain energy predicted is too large, and the approximation appears "too stiff." This error does not respond to mesh refinement, since the displacement field approximation through the thickness remains linear. Our intent is to select values for k_1 and k_2 which lead to stored energies of a more reasonable magnitude, and thus yield better element behavior.

Since the shear resultants are uncoupled for the case noted above, the basic aspects of the method can be outlined within a single plane. Below, we discuss the determination of k_1, the shear correction factor for the (x,z) plane.

The shear corrections suggested by Whitney (Whitney, 1973) depend upon the assumption of cylindrical bending, for which an analytical relationship may be established between the local bending stress and the transverse shear force resultant (Whitney, 1970):

$$\sigma_{x,x}^{(m)} = \frac{-Q_{11}^{(m)}}{D}\left(B_{11} - A_{11}z\right)Q_{xz}$$

The superscript (m) refers to a particular layer within the laminate cross-section, and parameter D is defined by:

$$D = D_{11}A_{11} - B_{11}^{2}$$

When combined with the analytic solution, the equilibrium equation

$$\sigma_{x,x}^{(m)} + \sigma_{xz,z}^{(m)} = 0$$

can be integrated through the plate thickness to obtain the shear stress within a layer:

$$\sigma_{xz}^{(m)} = \frac{1}{2D}\left[a^{(m)} + Q_{11}^{(m)}z\left(2B_{11} - A_{11}z\right)\right]Q_{xz}$$

The constants of integration $a^{(m)}$ are determined by the condition that σ_{xz} be continuous at the layer interfaces, and from the free surface boundary condition at either the upper or lower surface. From the condition that $\sigma_{xz} = 0$ at $z = -t/2$, we obtain:

$$a^{(1)} = \frac{1}{4}Q_{11}^{(1)}t\left(A_{11}t + 4B_{11}\right)$$

in which m = 1 refers to the bottom layer of the laminate. Letting $z_l^{(m)}$ be the lower surface of layer m, the interface continuity conditions for m = 2, 3, ... give:

$$a^{(m)} = a^{(m-1)} + \left[Q_{11}^{(m)} - Q_{11}^{(m-1)}\right]\left(A_{11}z_l^{(m)} - 2B_{11}\right)z_l^{(m)}$$

With the above definitions, the strain energy density in any layer may be written in the form:

$$V^{(m)} = \frac{1}{2}g^{(m)}(z)Q_{xz}$$

with

$$g^{(m)}(z) = \frac{1}{G_{xz}^{(m)}}\left[\frac{a^{(m)}}{2D} + \frac{Q_{11}^{(m)} z}{2D}\left(2B_{11} - A_{11} z\right)\right]^2$$

Integrating the shear strain energy through the laminate thickness, and equating the result to the total strain energy per unit area obtained from the plate theory resultants,

$$V = \frac{Q_{xz}^2}{2k_1 A_{55}}$$

we obtain for the shear correction factor:

$$k_1 = \left[A_{44} + \int_{-t/2}^{t/2} g^{(m)}(z)\, dz\right]^{-1}$$

The remaining factor k_2 may be determined similarly, using the appropriate elastic constants for the (y,z) plane.

For layered elements composed solely of elastic, orthotropic material, the foregoing approximation of the transverse shear stress distribution is used to predict delamination based on a maximum resolved shear stress criterion. The transverse shear stress components σ_{xz} and σ_{yz} are recovered at each integration point, and the magnitude of the maximum interlaminar shear stress compared with a critical *delamination stress*, σ_d. Delamination is predicted at a point when:

$$\sqrt{\sigma_{xz}^2 + \sigma_{yz}^2} = \sigma_d$$

When delamination is predicted at a point, the stress field is adjusted to reflect *only the midsurface strain state.* Therefore, as delamination progresses through the thickness of a laminate, the ability to resist bending deformation gradually disappears.

SECTION 3
SAMPLE ANALYSES

This section presents selected problems which demonstrate the analysis techniques described in Section 2. The first two are common benchmark problems with well-known solutions, and illustrate the accuracy of the method for both thin shell and three-dimensional continuum problems. The final example is a simulation of a birdstrike on a monolithic aircraft transparency under conditions which are near the maximum capacity of the structure.

3.1 TAYLOR CYLINDER

The Taylor cylinder experiment, which is used to estimate the mechanical properties of metals at high strain rates, involves the normal impact of a cylinder onto a rigid surface. Although the event is axisymmetric, we will use a three-dimensional model of one quarter of the cylinder. The X3D model (Figure 3.1) consists of 1350 8-node solids. The geometry and material properties for the case considered are:

E	= 117,000 MPa	R_0	= 3.2 mm
ν	= 0.350	L_0	= 32.4 mm
σ_y	= 400 MPa	V_0	= 227,000. mm/sec
E_t	= 100 MPa		
ρ	= 8.93×10^{-9} N-sec^2/mm^4 (8930 kg/m^3)		

These material constants are typical of copper. Purely isotropic strain hardening is assumed, and no ultimate stress is specified (i.e., elements may not fail during the solution).

Virtually all of the kinetic energy of the cylinder is dissipated through plastic deformation within about 80 μs. Figures 3.2 and 3.3 show deformed mesh plots of the cylinder in its final state. Figures 3.4-3.7 show time histories of the cylinder's length and the radius at the impact surface, as well as the corresponding velocities. Figures 3.8 and 3.9 are contour line plots of effective plastic strain after 80 μs. The analysis was performed in 8886 time steps, and required 6 hours, 36 minutes on a VAX 8650 computer (about 0.00198 CPU seconds per element time step). The same analysis runs in about 40 minutes on A CRAY X-MP (0.0002 seconds per element time step).

Results from the X3D solution compare very well with analyses using the DYNA and NIKE codes, as shown in the table below.

QUANTITY	X3D	DYNA2D	DYNA3D	NIKE2D
Final length, mm	21.47	21.47	21.47	21.47
Maximum radius, mm	7.081	7.127	7.034	7.068
Maximum ε_p at center	2.95	3.05	2.95	2.97

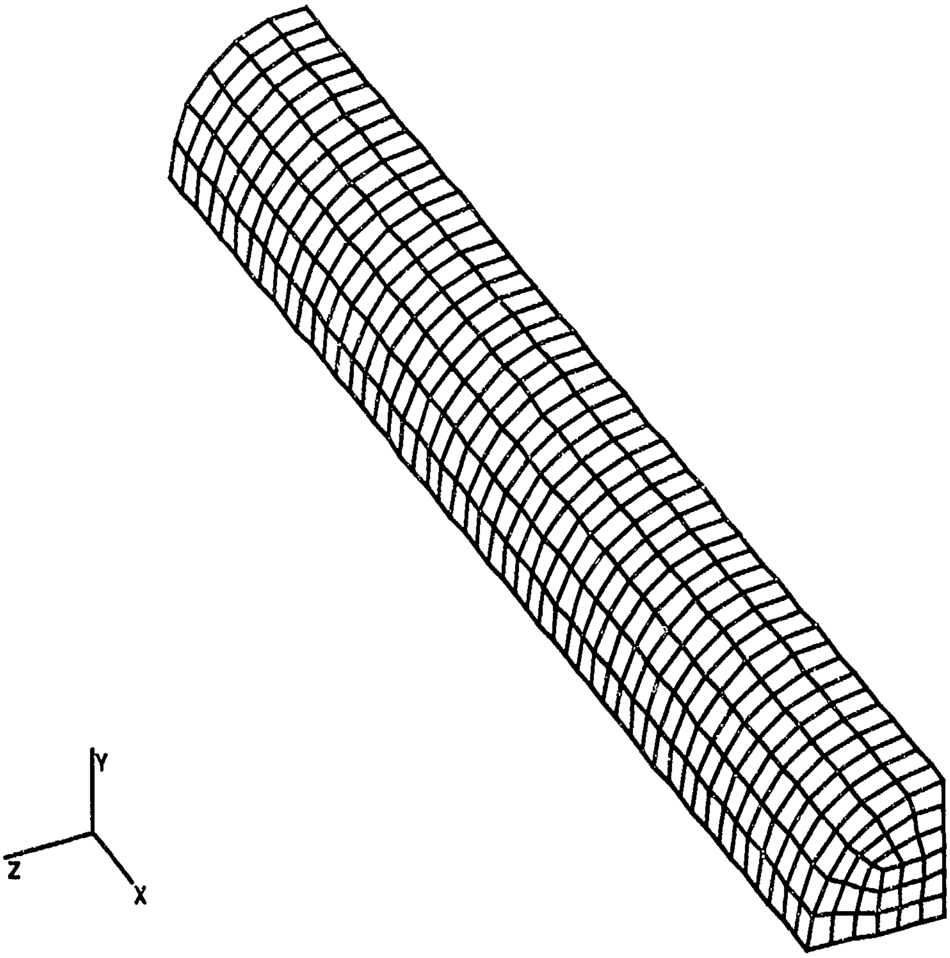


Figure 3.1. Finite Element Model of Taylor Cylinder Problem.

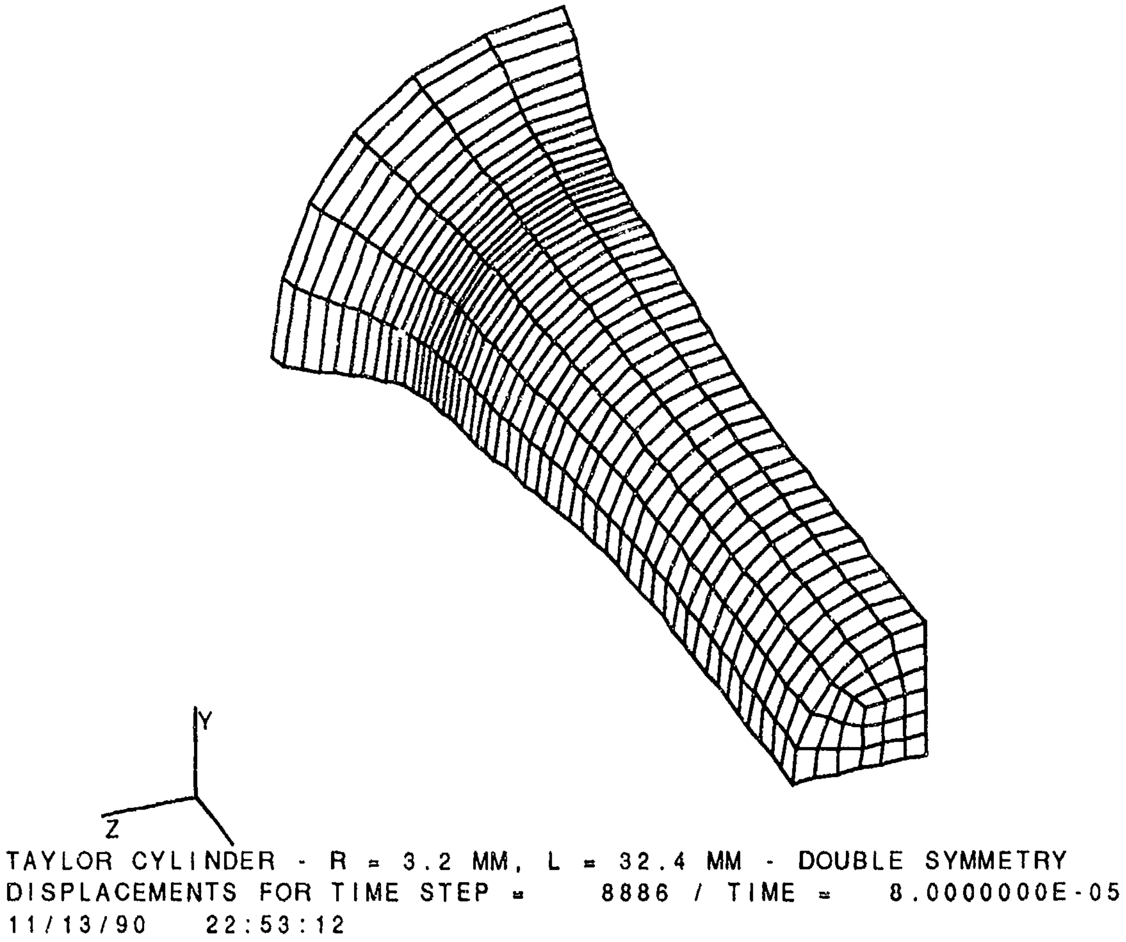
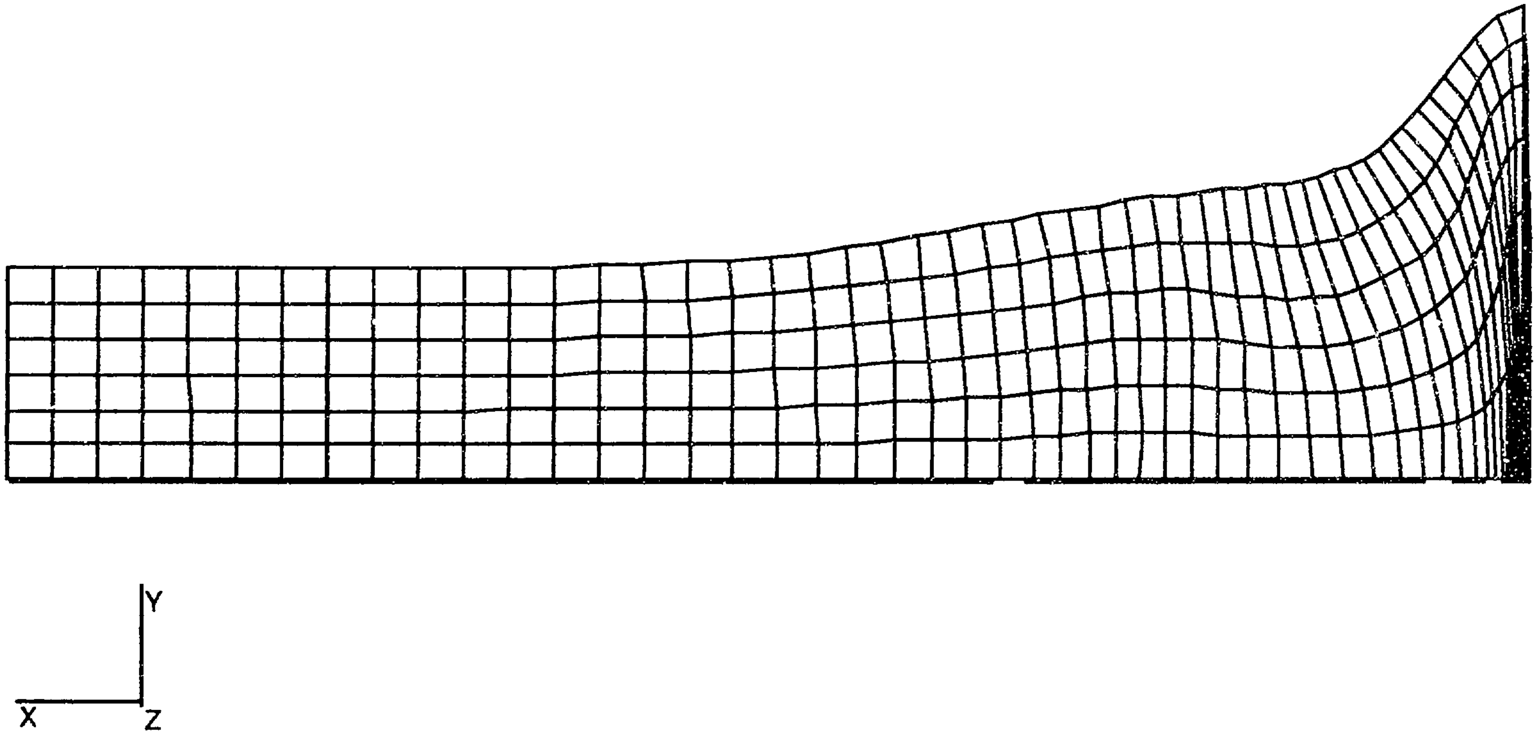

Y
Z
TAYLOR CYLINDER - R = 3.2 MM, L = 32.4 MM - DOUBLE SYMMETRY
DISPLACEMENTS FOR TIME STEP = 8886 / TIME = 8.0000000E-05
11/13/90 22:53:12

Figure 3.2. Deformed Geometry of Taylor Cylinder, View #1.



```
TAYLOR CYLINDER - R = 3.2 MM, L = 32.4 MM - DOUBLE SYMMETRY
DISPLACEMENTS FOR TIME STEP =     8886 / TIME =    8.0000000E-05
11/13/90    22:53:12
```

Figure 3.3. Deformed Geometry of Taylor Cylinder, View #2.

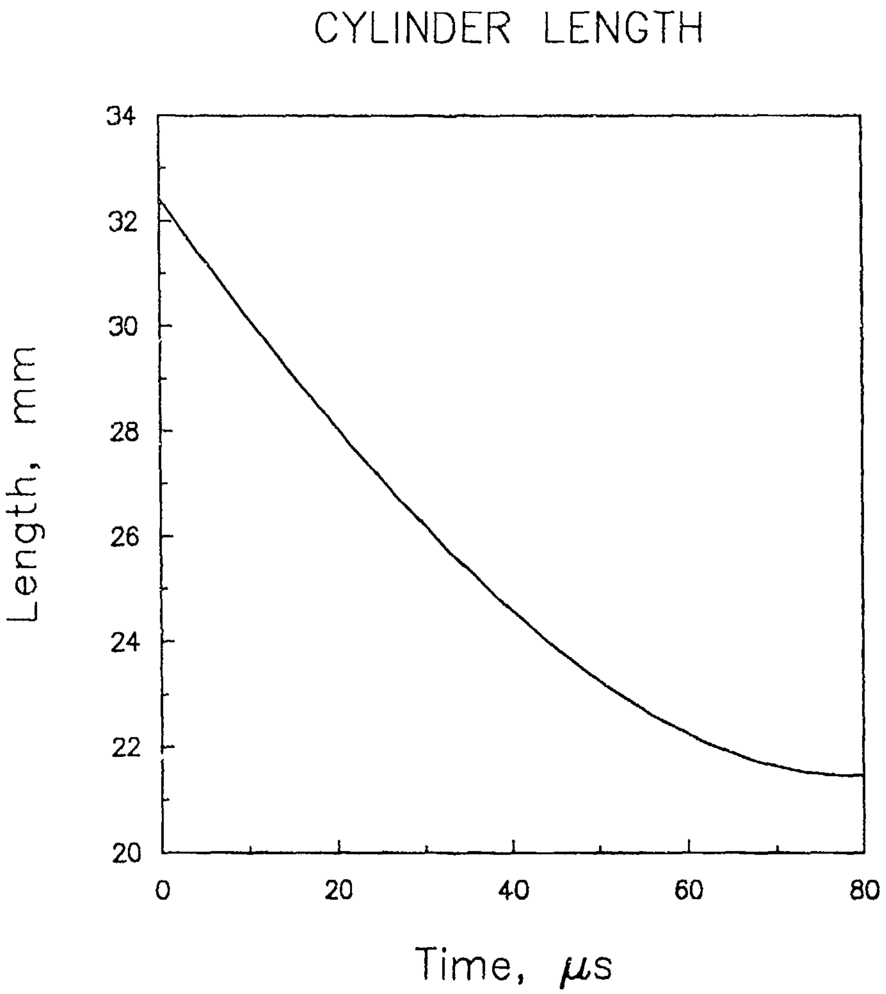


Figure 3.4. Cylinder Length versus Time.

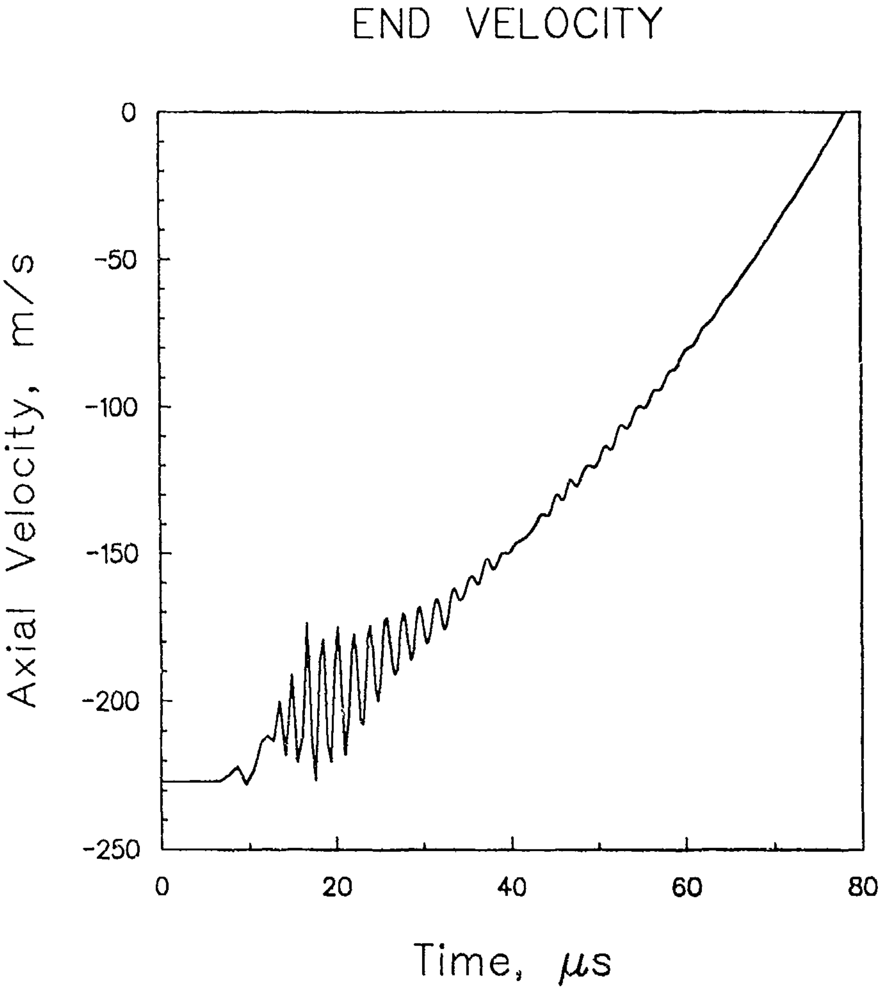


Figure 3.5. Cylinder End Velocity versus Time.

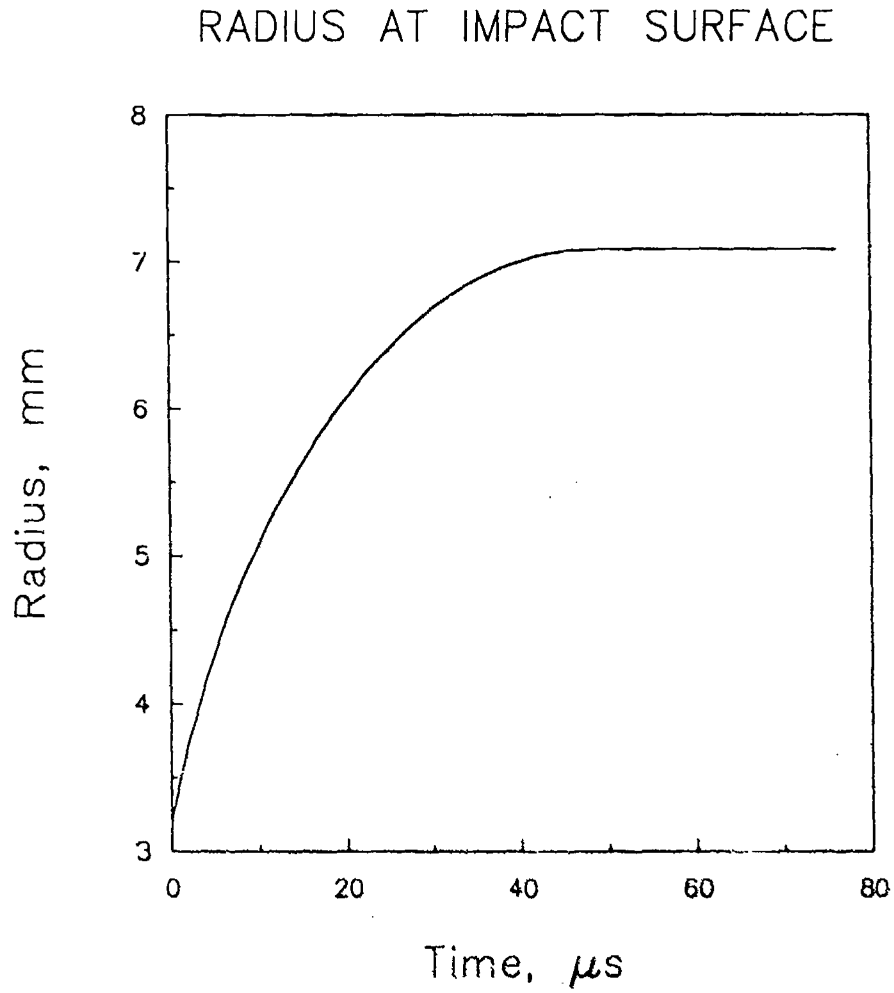


Figure 3.6. Cylinder Radius versus Time at Impact Surface.

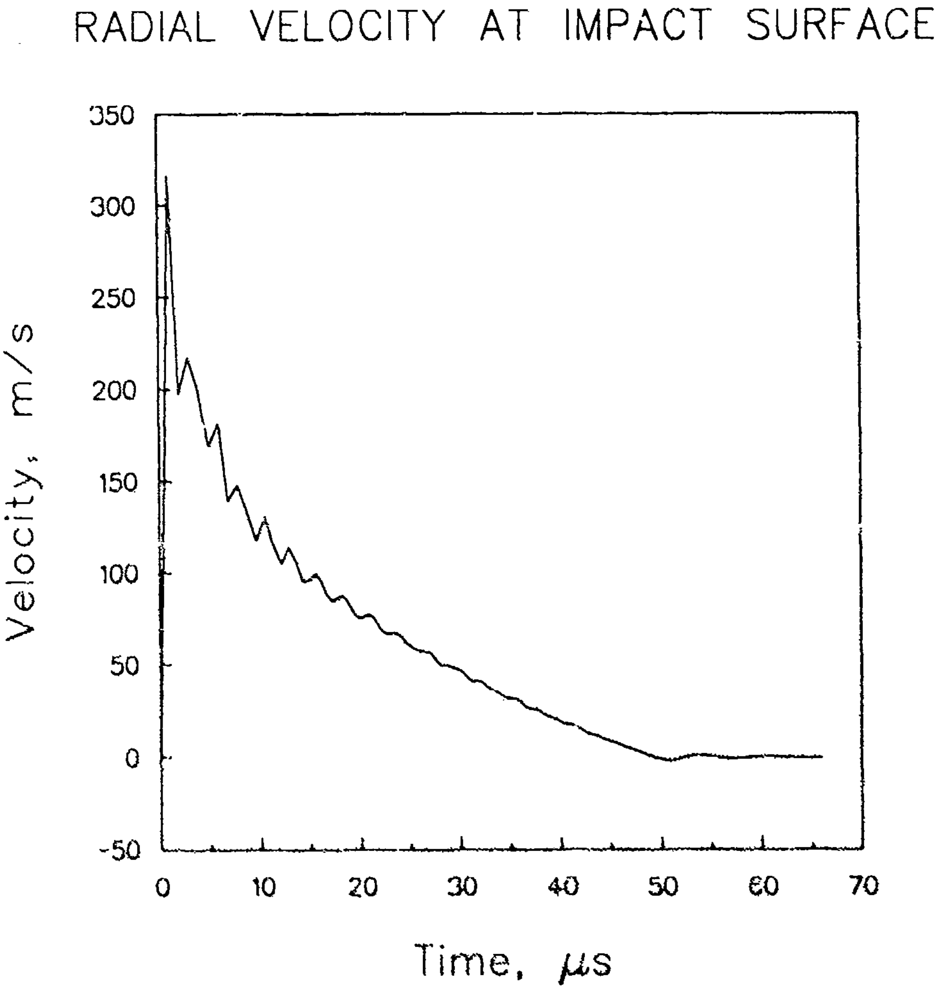
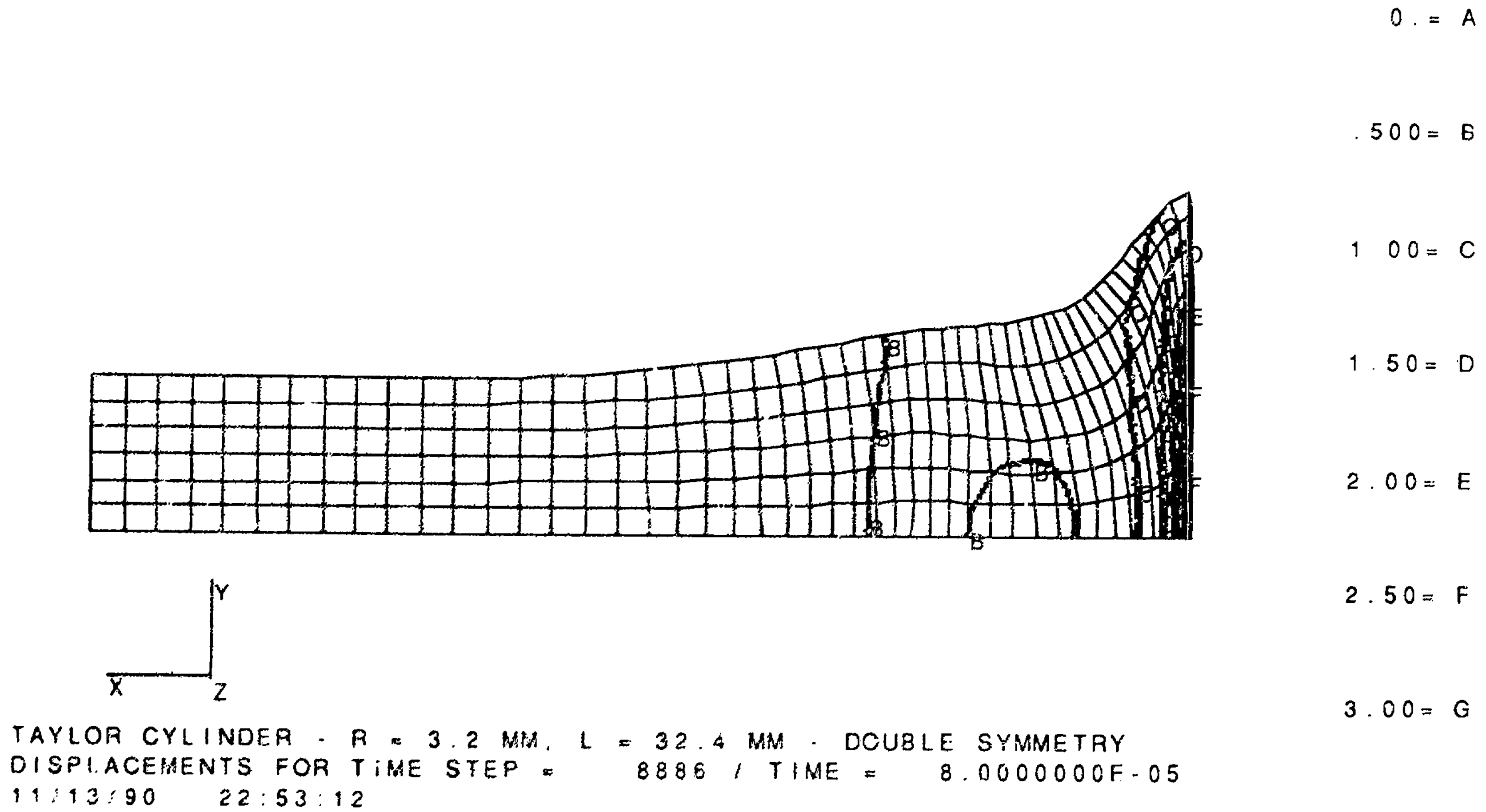


Figure 3.7. Cylinder Radial Velocity versus Time.



0 . = A
. 500 = B
1 00 = C
1 . 50 = D
2 . 00 = E
2 . 50 = F
3 . 00 = G
B
B
B
B
B
C
D
E
F
Y
X
Z
TAYLOR CYLINDER - R = 3.2 MM, L = 32.4 MM - DOUBLE SYMMETRY
DISPLACEMENTS FOR TIME STEP = 8886 / TIME = 8.0000000E-05
11/13/90 22:53:12

Figure 3.8. Plastic Strains on Symmetry Plane at 80 μs.

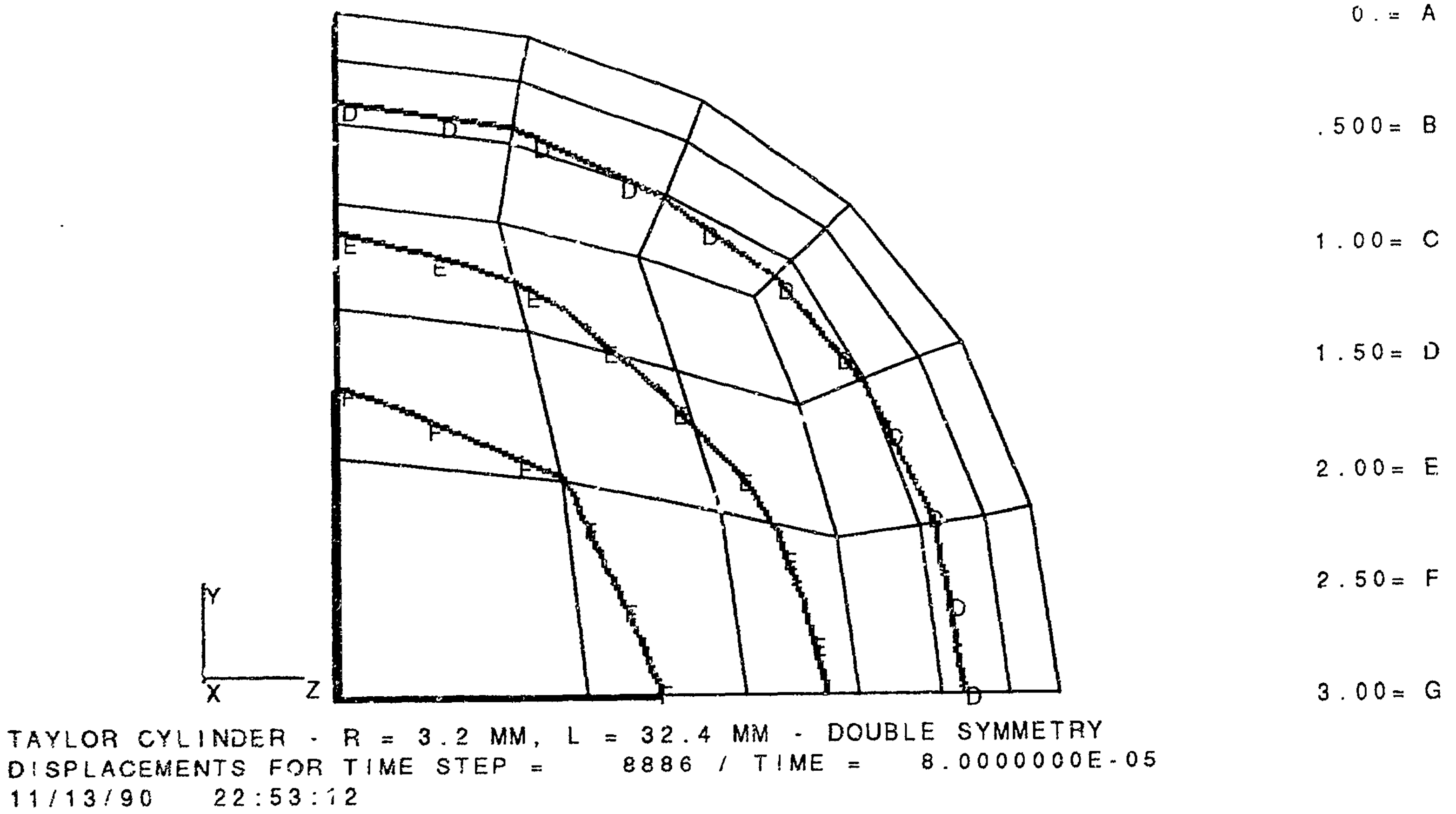


Figure 3.9. Plastic Strains on Impact Surface at 80 µs.

The listing below shows the PATRAN modeling directives needed to construct a geometric model of the cylinder and generate the finite element mesh (note that the resulting model must then be "equivalenced" to merge coincident node points).

```
GO
1
1
GR,1,,0/0/0
GR,2,,0/3.2/0
LI,1,ARC,0/0/0/1/0/0/45,2
LI,2,ARC,0/0/0/1/0/0/45,3
GR,5,,0/1.6/0
GR,6,,0/0/1.6
GR,7,,0/1.46/1.46
LI,3,2G,,1,5
LI,4,2G,,1,6
LI,5,2G,,5,7
LI,6,2G,,6,7
PATCH,1,2L,,1,5
PATCH,2,2L,,2,6
PATCH,3,2L,,4,5
HPAT,1T3,EXT,32.4/0/0,1T3
GFEG,HP1T3,,4/4/51
CFEG,HP1T3,HEX,,1
DFEG,HP1,DISP,//0/0/0/0,1,F1
DFEG,HP2,DISP,/0//0/0/0,1,F2
DFEG,HP3,DISP,/0//0/0/0,1,F1
DFEG,HP3,DISP,//0/0/0/0,1,F3
```

Additional data needed to complete the model for X3D are shown below. These items normally would be added to the X3D input file generated by PATX3D (see Appendix B).

```
INITIAL CONDITIONS
 -227000 0 0     17    816  1
 -227000 0 0    829   1428  1
 -227000 0 0   1438   1887  1
END

MAT3D
 1 1 8.93E-9 130000. 0. 0. 43333. 400.   0. 1.  100.
END

PARAMETERS
 0. 80.E-6 10.E-6 0. 0. 0 15000 1 5 0 20
END

TOLERANCES
 0.90 0.1
END

REFERENCE
 -0.01 32.5 -0.1 15. -0.1 15.
END

TRACE NODES
 1 819 1879
END
```

3.2 EXPLOSIVELY LOADED CYLINDRICAL SHELL

This example is taken from Marchertas and Belytschko (1974), who present both computational and experimental results for the problem. A 120° cylindrical panel (Figure 3.10) is loaded by igniting a charge spread over most of the surface. In the numerical solution, we represent this impulsive loading by a uniform initial velocity along the radius of the shell. The geometric and material parameters for the shell are:

$R = 2.9375$ in.	$E = 10.5 / 10$ psi
$t = 0.125$ in.	$\rho = 0.0965$ lb/cu.in
$L = 12.56$ in.	$\sigma_y = 44{,}000$ psi
$V_0 = 5{,}650$ in./sec.	

A three-point integration through the thickness of the shell has been used in the present case. This is the minimum thickness integration order, and may give a solution which is slightly too flexible.

Results of the X3D solution, which was performed in 886 time steps, are shown in Figures 3.11 through 3.14. The response consists mainly of a flattening of the inner portion of the shell, consisting mostly of permanent deformation. The displacements peak at around 0.4 ms, with the largest inward displacements approaching half the radius. After this point, there is some elastic recovery (lasting about another 0.1-0.2 ms), but only very small vibration, since most of the energy has been dissipated through plastic flow. Displacement histories at selected points (Figures 3.13 and 3.14) agree quite well with experimental results.

The listing below shows a PATRAN session file which creates the geometric and finite element models for the cylinder problem.

```
GO
1
1
GR,1,,0/2.9375/0
LI,1,ARC,5(0)/-1/60,1
LI,2,TR,0/0/12.56,1
VIEW
1
27.1,-31,-14.8
PLOT
PATCH,1,2L,,1,2
SET,LI,0
PLOT
PATCH,1,SHOW
GFEG,P1,G,9/17
SET,LABN,OFF
PLOT
CFEG,P1,QUAD,,1
```

Additional data needed to complete the model for analysis with X3D are listed below. Note that the initial velocity components are directed radially inward, and have a magnitude at each node of 5.650 in/sec. Points on the edges of the loaded region are assigned half the nominal initial velocity to provide the correct impulse to the shell.

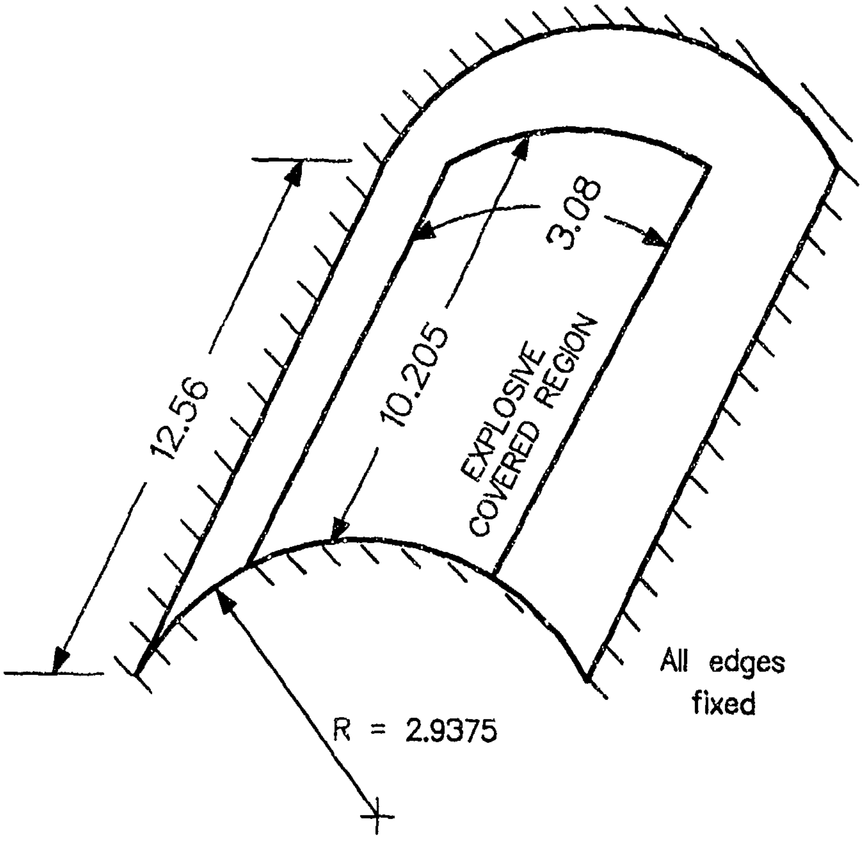


Figure 3.10. Geometry of Explosively Loaded Cylindrical Shell.

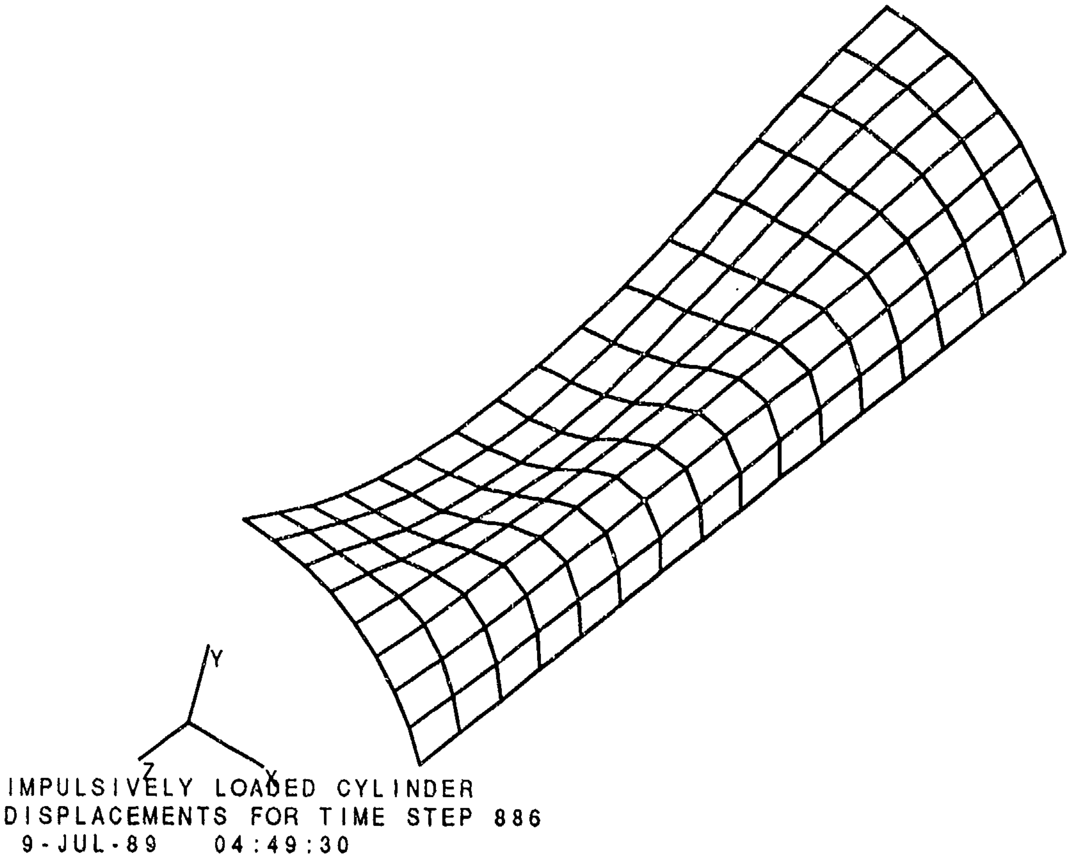


Figure 3.11. Final Deformed Shape of Cylindrical Shell.

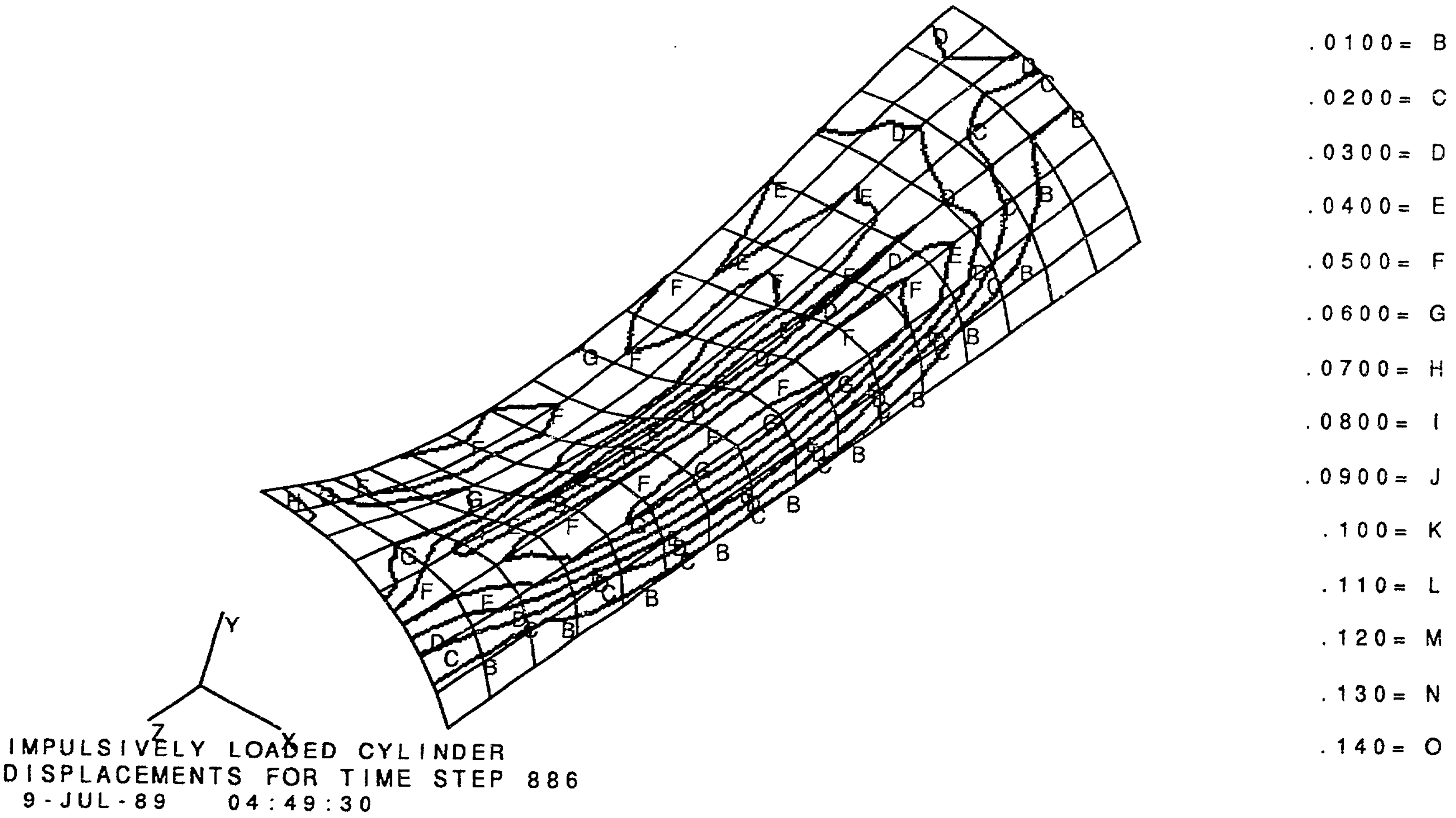


Figure 3.12. Plastic Strains in Explosively Loaded Shell.

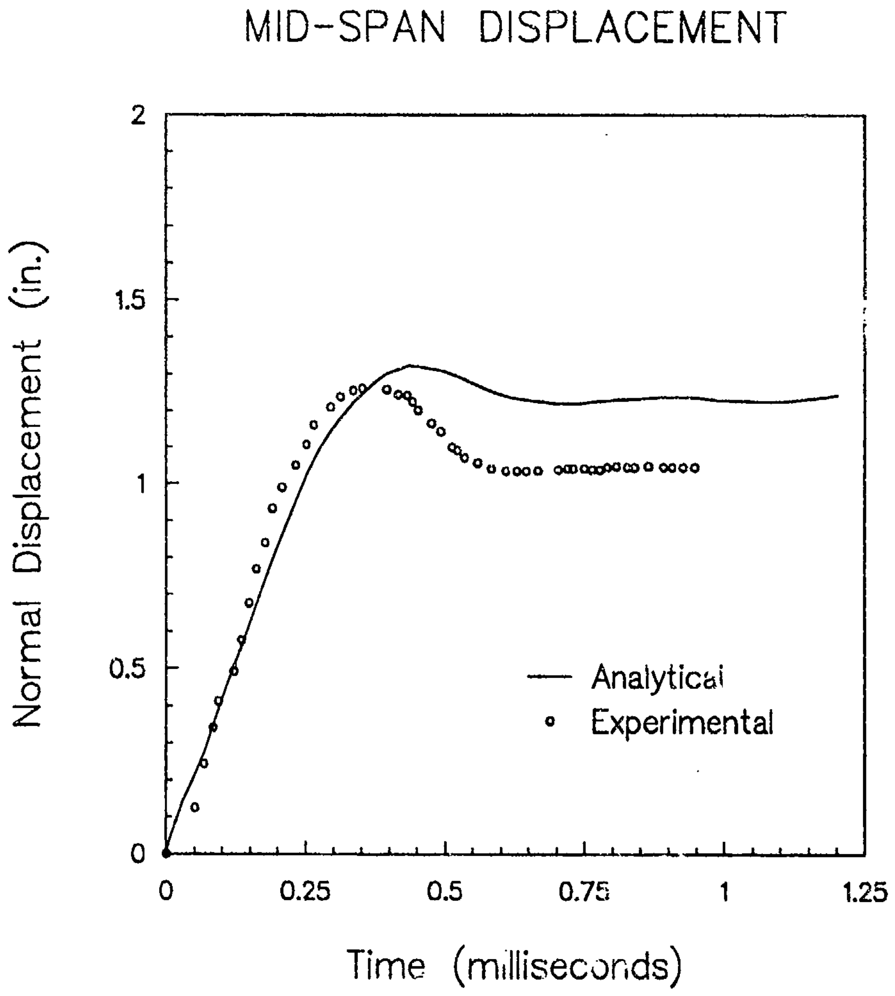


Figure 3.13. Mid-span Displacement History.

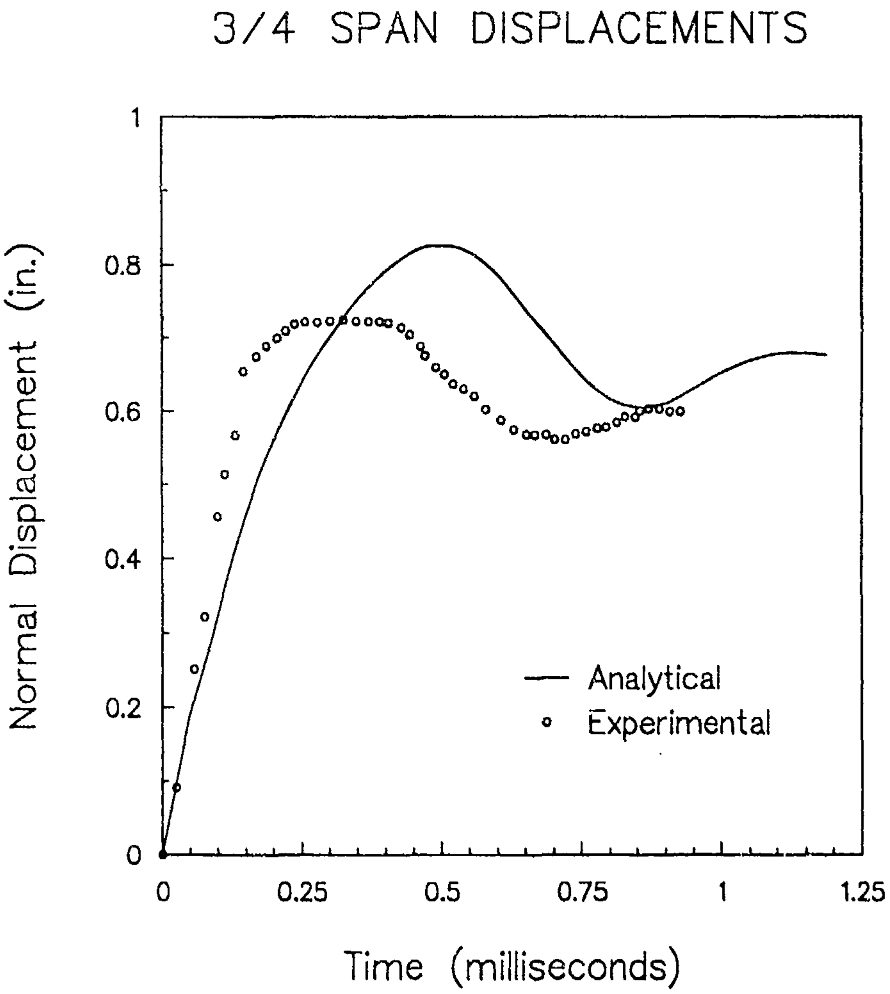


Figure 3.14. Quarter-span Displacement History.

```
BOUNDARY CONDITIONS
  1   9    1    1 2 3                    !  Rear end pinned
145 153    1    1 2 3                    !  Front end pinned
  9 153    9    1 2 3                    !  Hinge along side
  1 145    9    1 6                      !  Symmetry on x=0
END

INITIAL VELOCITY
     0.     -2825.0   0.   28   28   1
     0.     -5650.0   0.   37  145   9     ! Theta = 0.
   -368.8   -2800.9   0.   29   29   1
   -737.5   -5601.7   0.   38  146   9     ! Theta = 7.5
   -731.2   -2728.8   0.   30   30   1
  -1462.3   -5457.5   0.   39  147   9     ! Theta = 15.
  -1081.1   -2610.0   0.   31   31   1
  -2162.2   -5219.9   0.   40  148   9     ! Theta = 22.5
   -706.3   -1223.3   0.   32   32   1     ! (the corner)
  -1412.5   -2446.5   0.   41  149   9     ! Theta = 30.
END

LAMINATES
  1, 1, 1, -0.0625, 0.0625
END

MAT2D
  1, 1, 0.000250, 10.5E6, 0.3, 0., 44000., 0.
END

PARAMETERS
  0. 0.0012 0.0002 0. 0. 0 2500 1 3 0 5
END

TOLERANCES
  0.9  0.03  0,03  0.03  0.03
END

TRACE NODES
  37    &                    !  Node 37 is 9.42" from end
  73                         !  Node 73 is 6,28" from end (center)
END
```

3.3 F-16 BIRDSTRIKE SIMULATIONS

The F-16 bubble transparency provides a useful example for validation since the impact response involves very large motions, and the coupling between the load distribution and the deformation is strong. As a first step in validating the X3D code for birdstrike simulation, we have carried out several analyses of centerline impacts on the original production canopy, a ½ inch thick monolithic polycarbonate design. This part is capable of defeating birdstrikes up to about 350 knots.

Figure 3.15 shows the geometry of the transparency and of the projectile, a four-pound bird idealized as a circular cylinder. The patch outlined around the crown of the transparency, as well as the entire bird, are covered with contact elements (Figure 3.16).

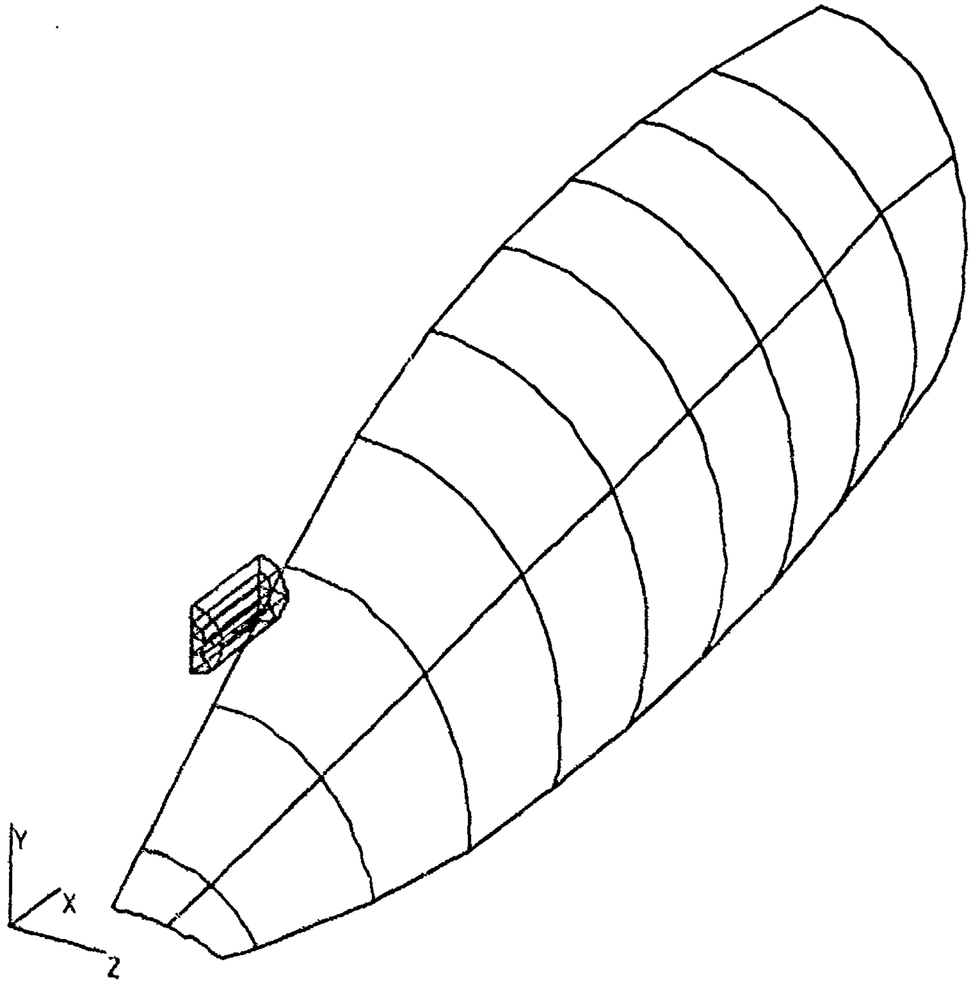


Figure 3.15. Geometry of F-16 Transparency and Bird.

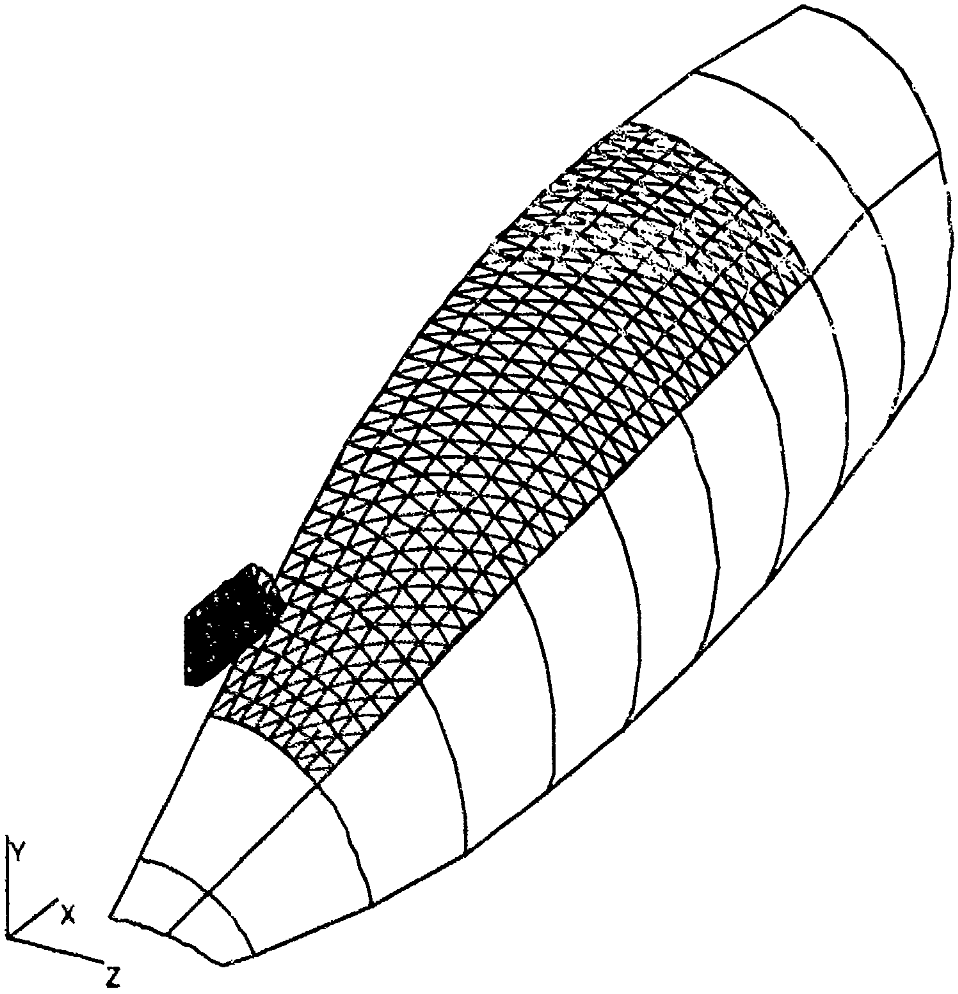


Figure 3.16. Contact Element Grid for Bird Impact Problem.

The transparency model consists of 928 quadrilateral plate elements. The polycarbonate material is represented by the following material properties:

$E = 325{,}000$ psi	$H = 36{,}100$ psi
$v = 0.36$	$D = 196{,}560$ /sec
$\rho = 0.0001113$ lb-sec^2/in^4	$p = 12$
$\sigma_y = 7{,}140$ psi	$\sigma_{ult} = 13{,}000 - 16{,}000$ psi

The bird is represented by 960 tetrahedral solids with equation-of-state coefficients typical of water, and very small shear stiffness and strength. The *low-strength* bird model, used in about half of our simulations, uses the properties:

$\rho = 0.0000888$ lb-sec^2/in^4	$G = 30{,}000$ psi
$K_1 = 337{,}000$ psi	$\sigma_y = 3{,}000$ psi
$K_2 = 729{,}000$ psi	$H = 300$ psi
$K_3 = 2{,}020{,}000$ psi	$\sigma_{ult} = 3{,}000$ psi
$K_t = 1{,}000$ psi (tension)	

This model of the bird material produces a pressure-volume response similar to water, and a "brittle" shear behavior: the ultimate and yield stresses coincide, so that element failure occurs at relatively small strains. A *high-strength* bird model has been used as well, with $\sigma_{ult} = 4{,}500$ psi. This model permits roughly 500% plastic (deviatoric) strain before the material is declared failed. Note that when elements of the bird model fail due to large shear distortion, their mass is retained in the problem, and the corresponding nodes continue to be used in contact calculations. Therefore, portions of the impacting body which have "failed" continue to transfer momentum to the target, but do not contribute to the summation of internal forces. In the deformed plots which follow, nodes attached to failed elements in the bird model are shown as small circles representing the center of mass positions.

For the cases considered, the center of impact is at fuselage station 112 (measured in inches), which is about two feet aft of the forward edge of the transparency. The initial velocity of the bird is horizontal, and equal to 350 knots (7,094 in/sec) at all nodes.

Figure 3.17 shows a sequence of deformed shapes at two millisecond intervals for a 350-knot impact. This solution employs the low-strength bird model, and a relatively high (16,000 psi) ultimate strength for the polycarbonate. The impact response is elastic, although the von Mises stresses near the crown (where the maximum curvature occurs at 10-12 ms) are well above the static yield stress. The displacement data are similar to experimentally observed values (Figure 3.18), although the computed deformed shape exhibits larger displacements in the forward region. Minor differences in the exact center of impact from that assumed in the calculation may be responsible for this discrepancy. Figure 3.19 shows a carpet plot obtained from the deformed centerline shapes at closer intervals during the solution. The time axis moves from right to left in the figure (that is, the right boundary corresponds to the undeformed geometry). The finer time resolution in this plot reveals additional detail, including the reflection of bending waves from the aft support.

Results from a similar calculation using the higher-strength bird model are shown in Figure 3.20. This solution predicts a predominantly elastic response as well. The impact loading computed with the stronger bird model obviously is quite different from that resulting from the weaker model, and generally provides somewhat greater energy transfer to the target. The question of which bird model is more realistic cannot be resolved without additional experimental data.

Sensitivity studies based upon perturbations of these basic models reveal very promising results. In particular, a modest reduction in ultimate stress for the polycarbonate (13,000 psi, probably a more realistic value) results in localized failure (two plate elements) at 350 knots, indicating that this impact velocity is close to the maximum capability of the transparency. Increasing the impact speed to 450 knots and decreasing the polycarbonate ultimate stress to 9,150 psi results in fairly dramatic failure (Figure 3.21).

In experiments performed at AEDC (Watt, 1978), massive failures were observed for impacts in excess of 350 knots for the ½ inch part. No attempt has been made to iterate for a critical velocity in this study, since so many details are unknown with regard to the material properties, precise support conditions, and center of impact location.

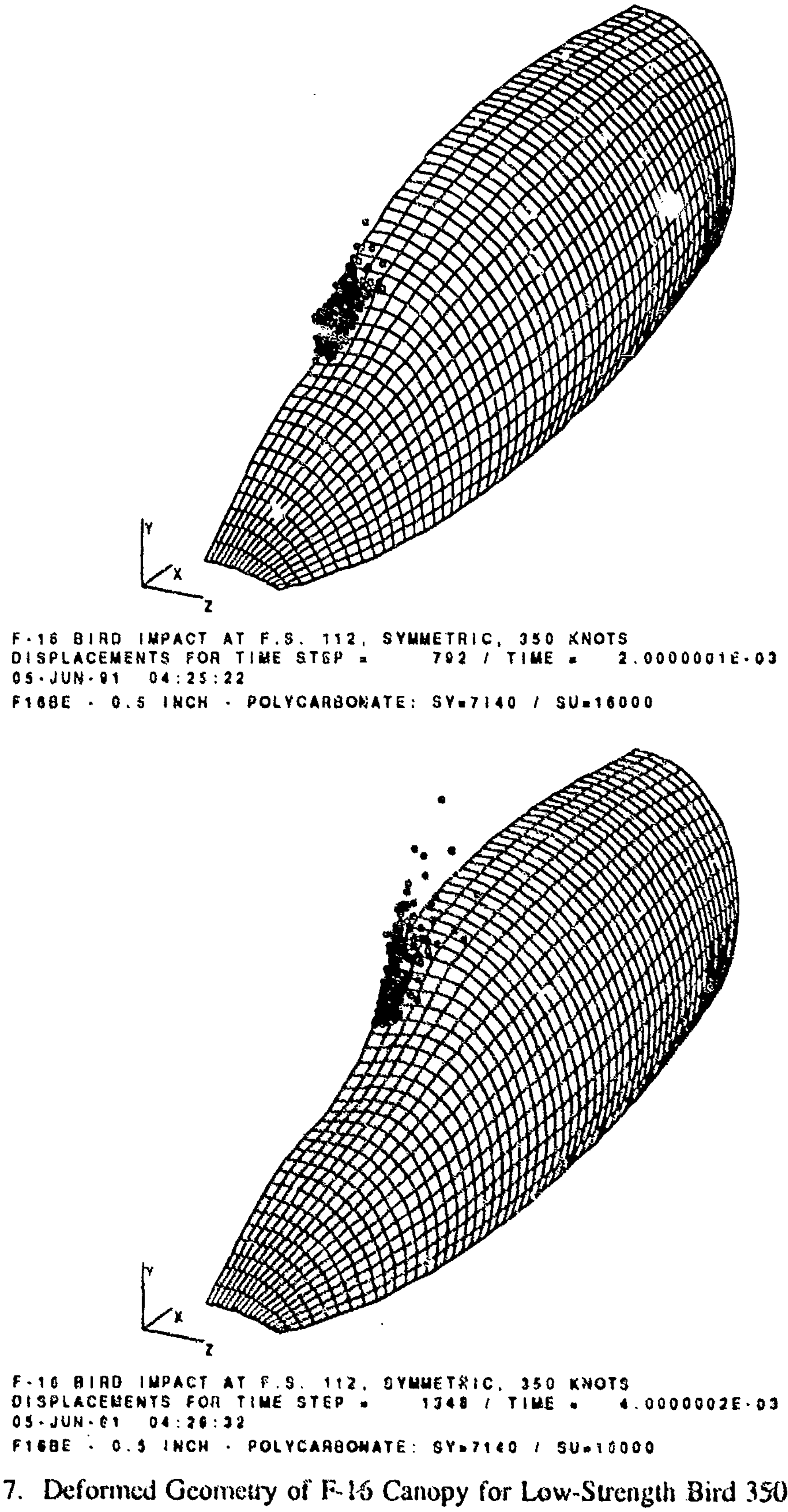

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 792 / TIME = 2.0000001E-03
05-JUN-91 04:25:22
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 1348 / TIME = 4.0000002E-03
05-JUN-91 04:26:32
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.17. Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals).

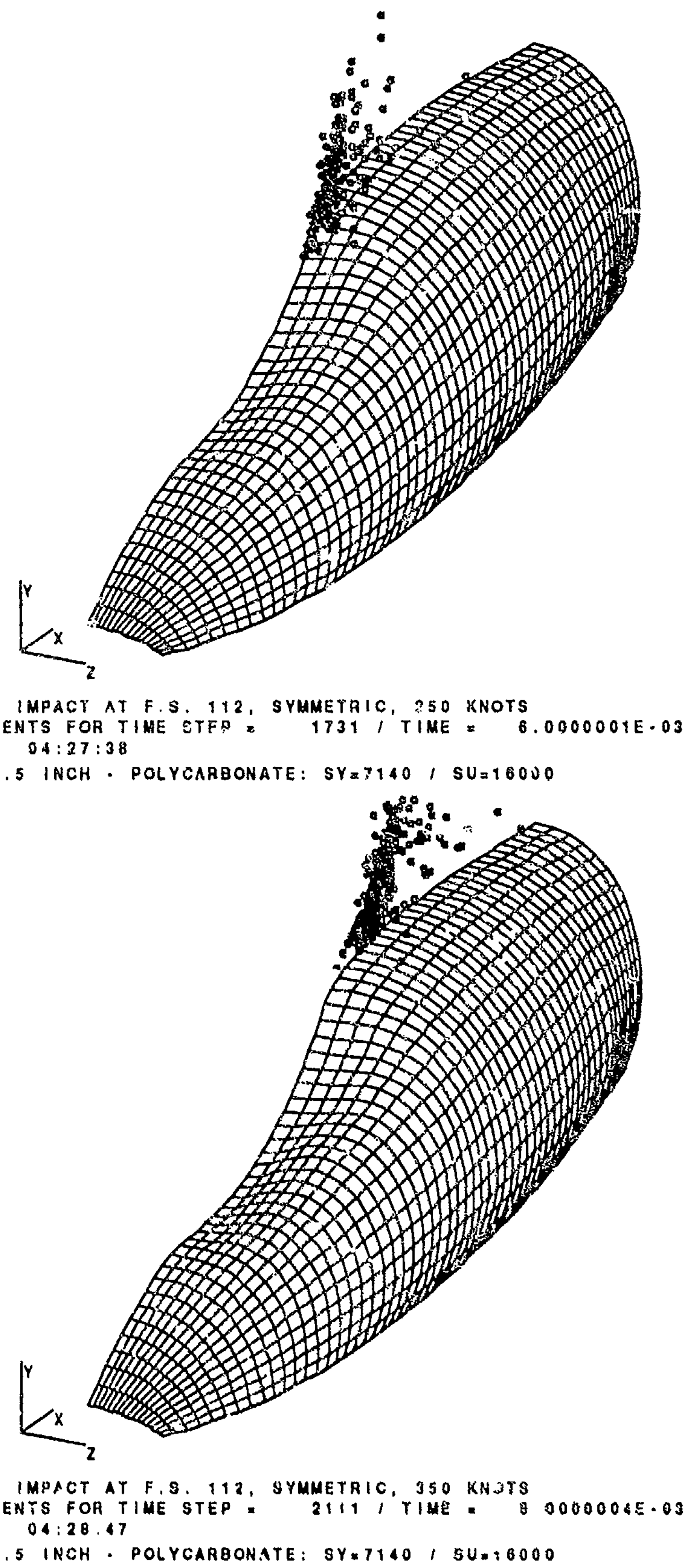

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 1731 / TIME = 6.0000001E-03
05-JUN-91 04:27:38
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 2111 / TIME = 8.0000004E-03
05-JUN-91 04:28:47
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.17. Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals). (Continued)

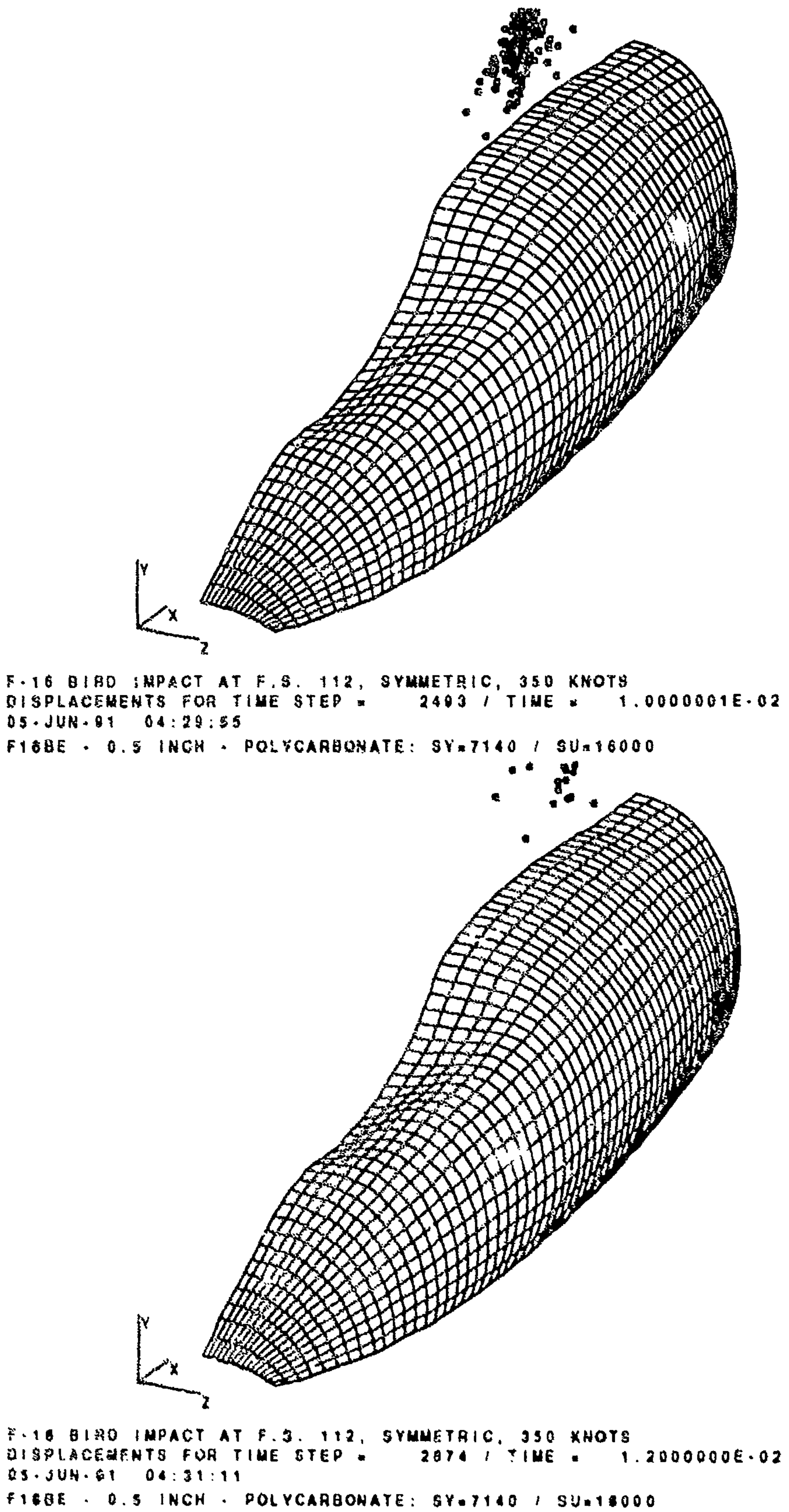
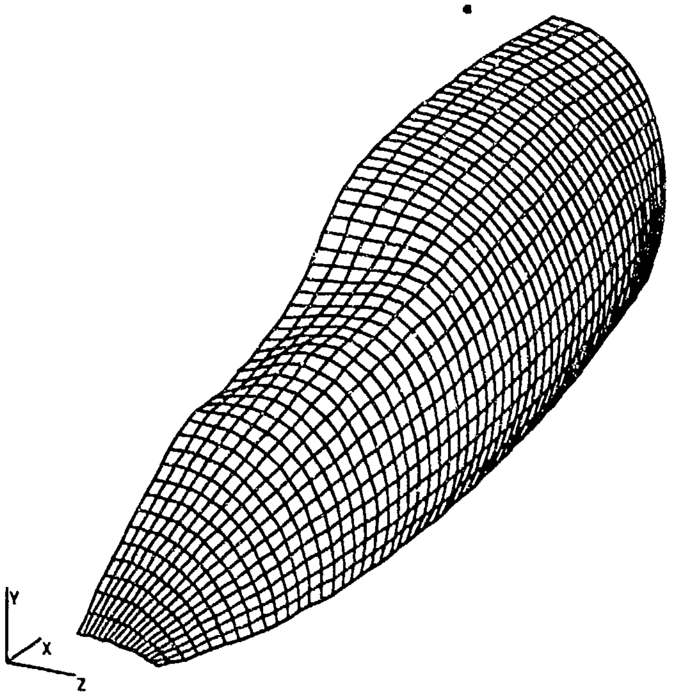
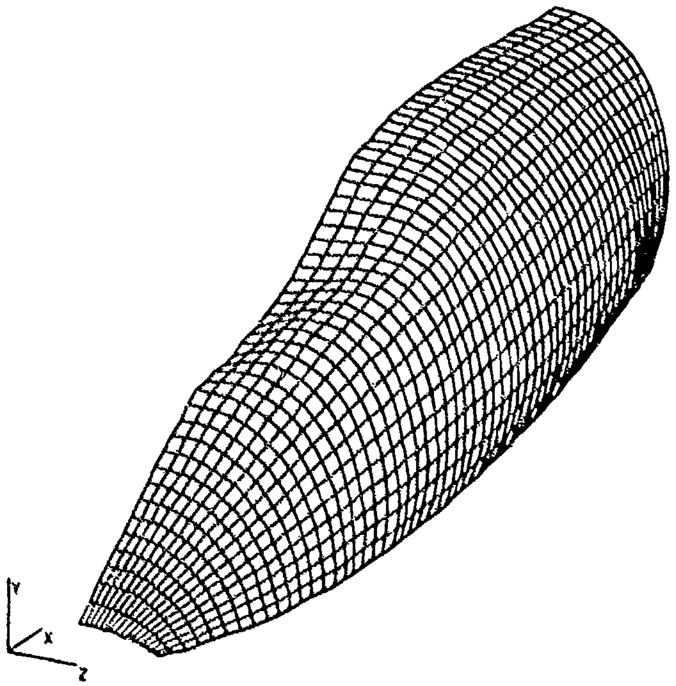

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 2493 / TIME = 1.0000001E-02
05-JUN-91 04:29:55
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 2874 / TIME = 1.2000000E-02
05-JUN-91 04:31:11
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.17. Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals). (Continued)

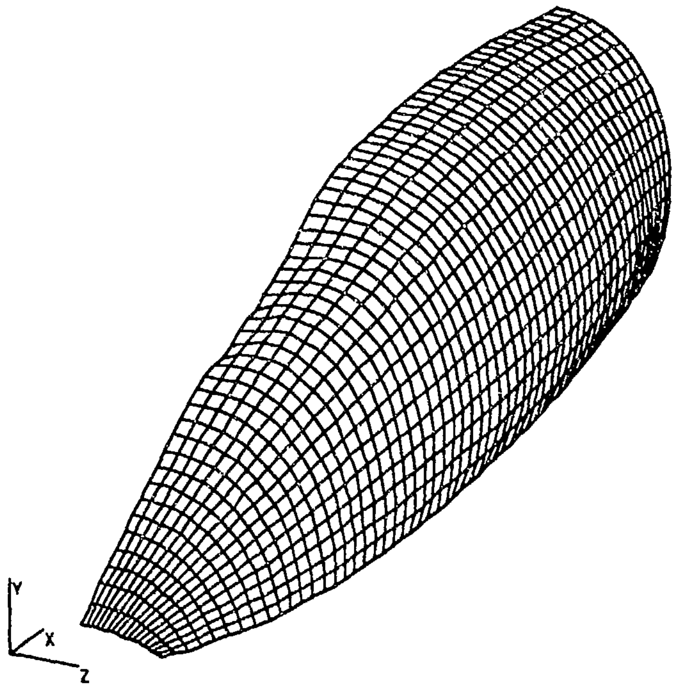


F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 3256 / TIME = 1.4000000E-02
05-JUN-91 04:32:25
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

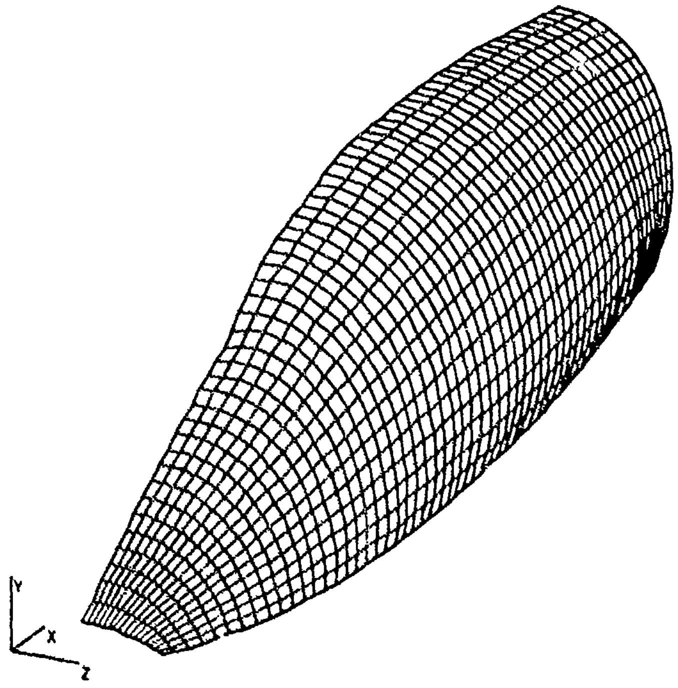


F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 3637 / TIME = 1.6000001E-02
05-JUN-91 04:33:41
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.17. Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals). (Continued)



F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 4017 / TIME = 1.8000001E-02
05-JUN-91 04:34:59
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000



F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 4307 / TIME = 2.0000001E-02
05-JUN-91 04:36:21
F16BE - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.17. Deformed Geometry of F-16 Canopy for Low-Strength Bird 350-Knot Impact (2-ms Intervals). (Concluded)

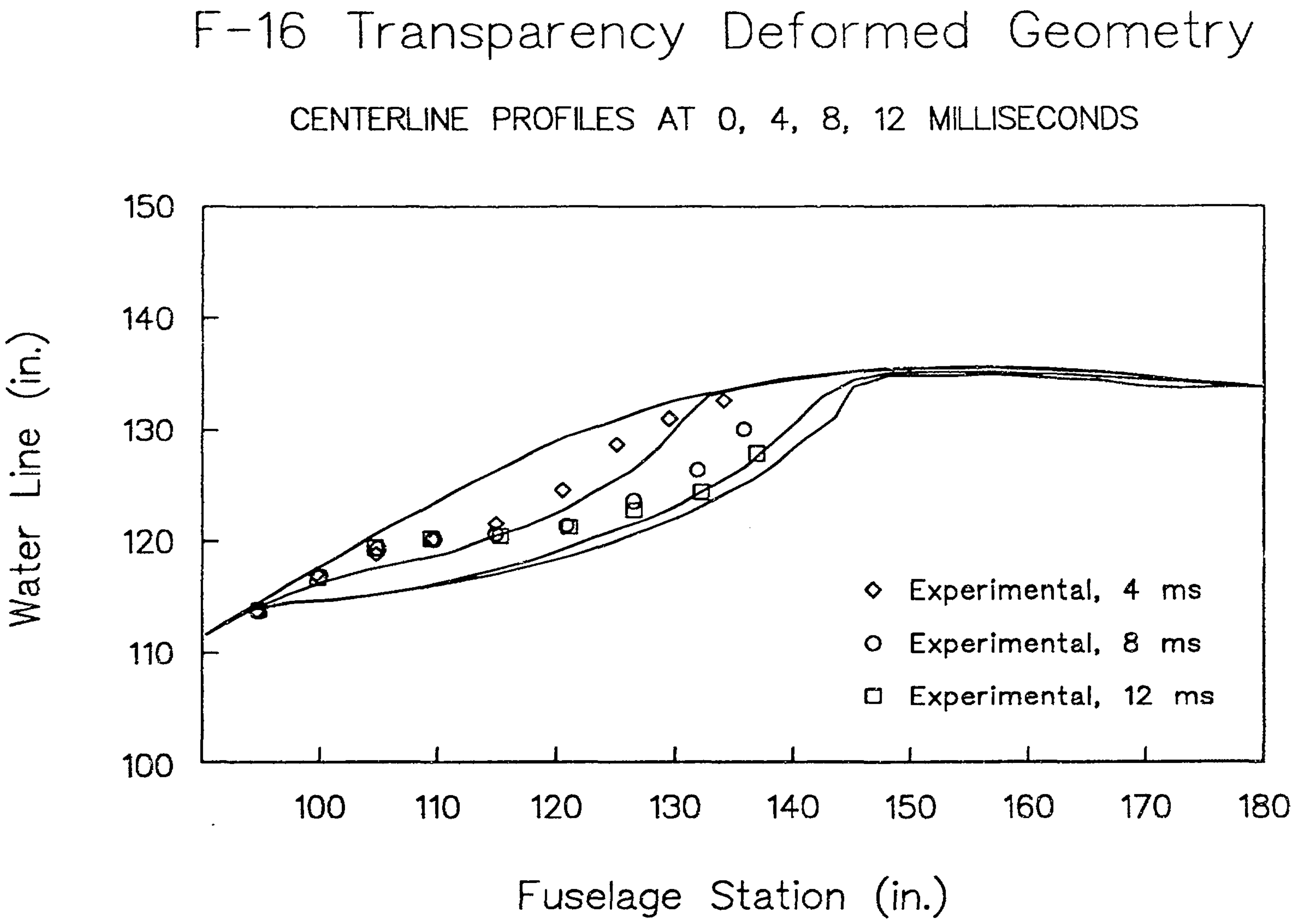


Figure 3.18. Centerline Profiles of Deformed Canopy for 350-Knot Impact (4-ms Intervals).

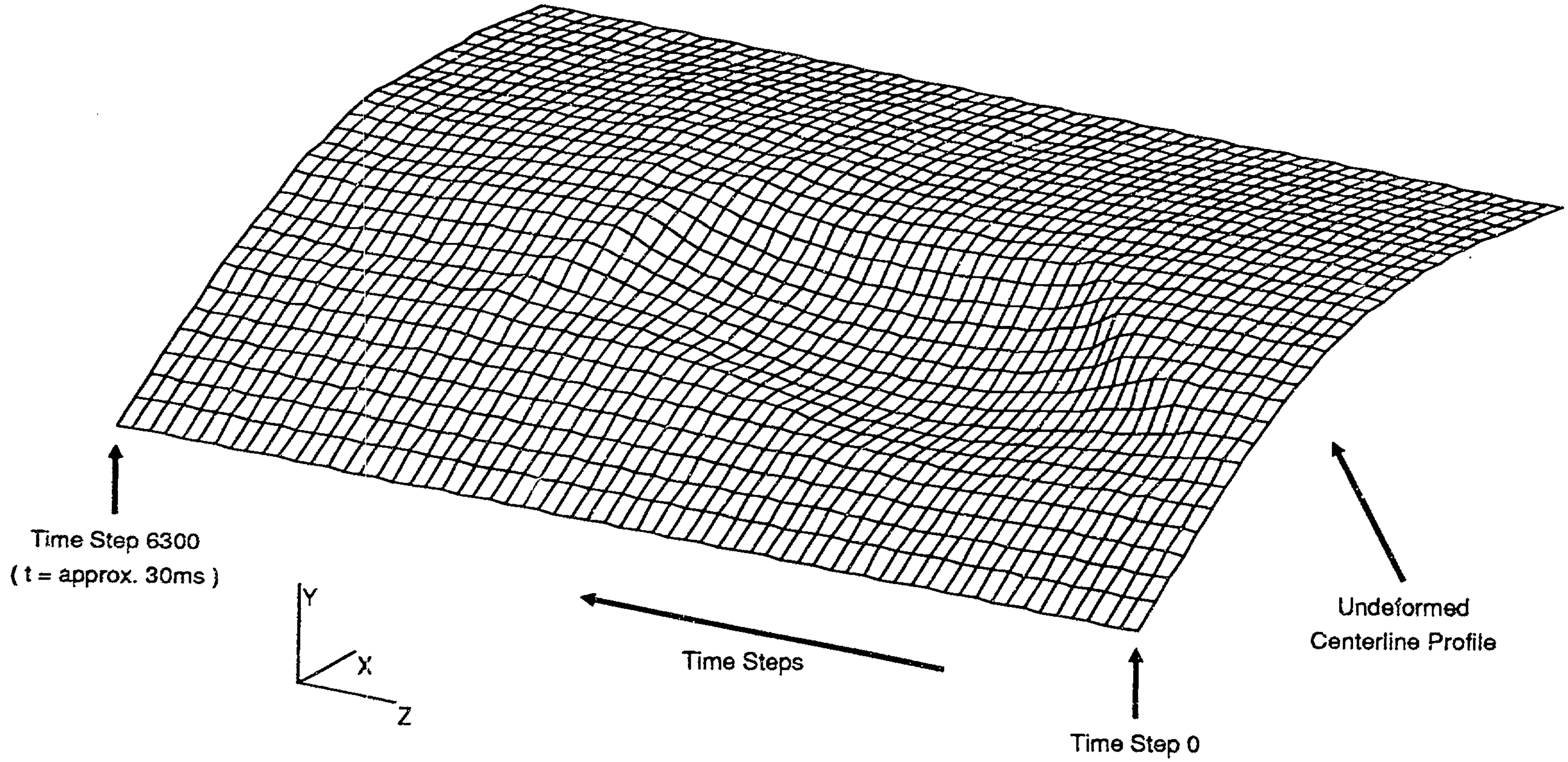


Figure 3.19. Carpet Plot of Centerline Shape versus Time Step for Low-Strength Bird 350-Knot Impact.

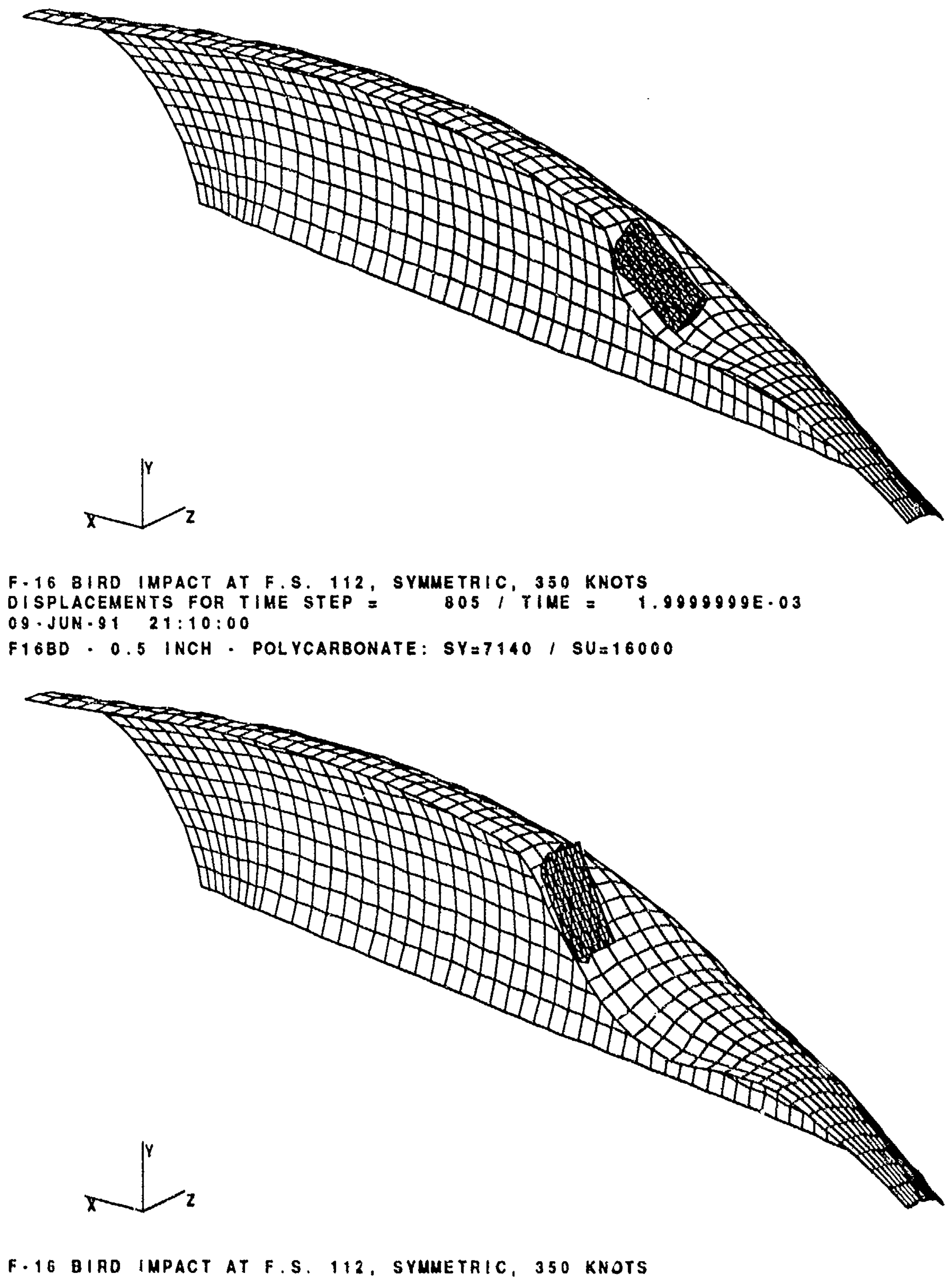

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 805 / TIME = 1.9999999E-03
09-JUN-91 21:10:00
F16BD - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 1682 / TIME = 3.9999997E-03
09-JUN-91 21:12:02
F16BD - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.20. Deformed Geometry of F-16 Canopy for High-Strength Bird 350-Knot Impact (2-ms Intervals).

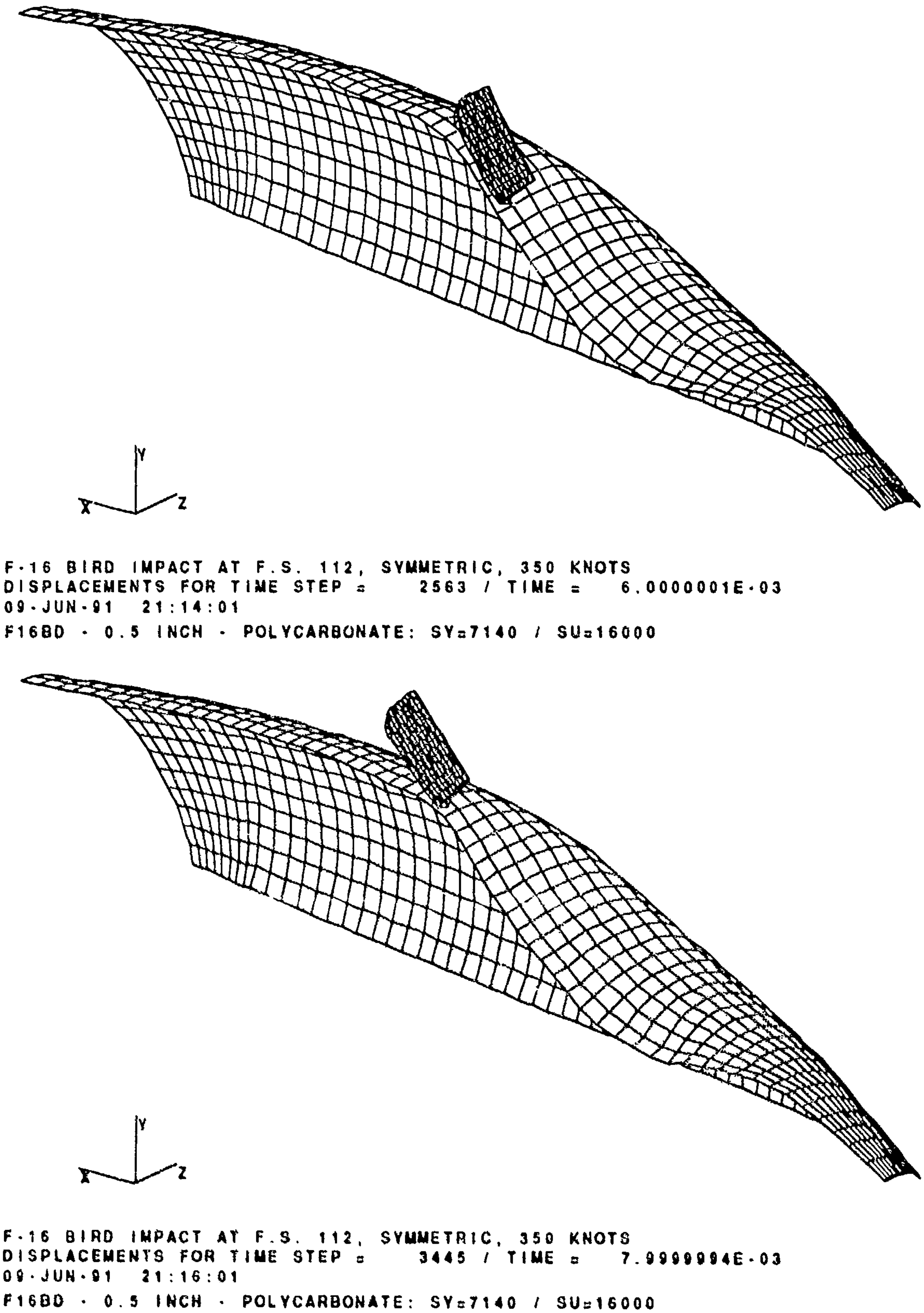

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 2563 / TIME = 6.0000001E-03
09-JUN-91 21:14:01
F16BD - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
DISPLACEMENTS FOR TIME STEP = 3445 / TIME = 7.9999994E-03
09-JUN-91 21:16:01
F16BD - 0.5 INCH - POLYCARBONATE: SY=7140 / SU=16000

Figure 3.20. Deformed Geometry of F-16 Canopy for High-Strength Bird 350-Knot Impact (2-ms Intervals). (Concluded)

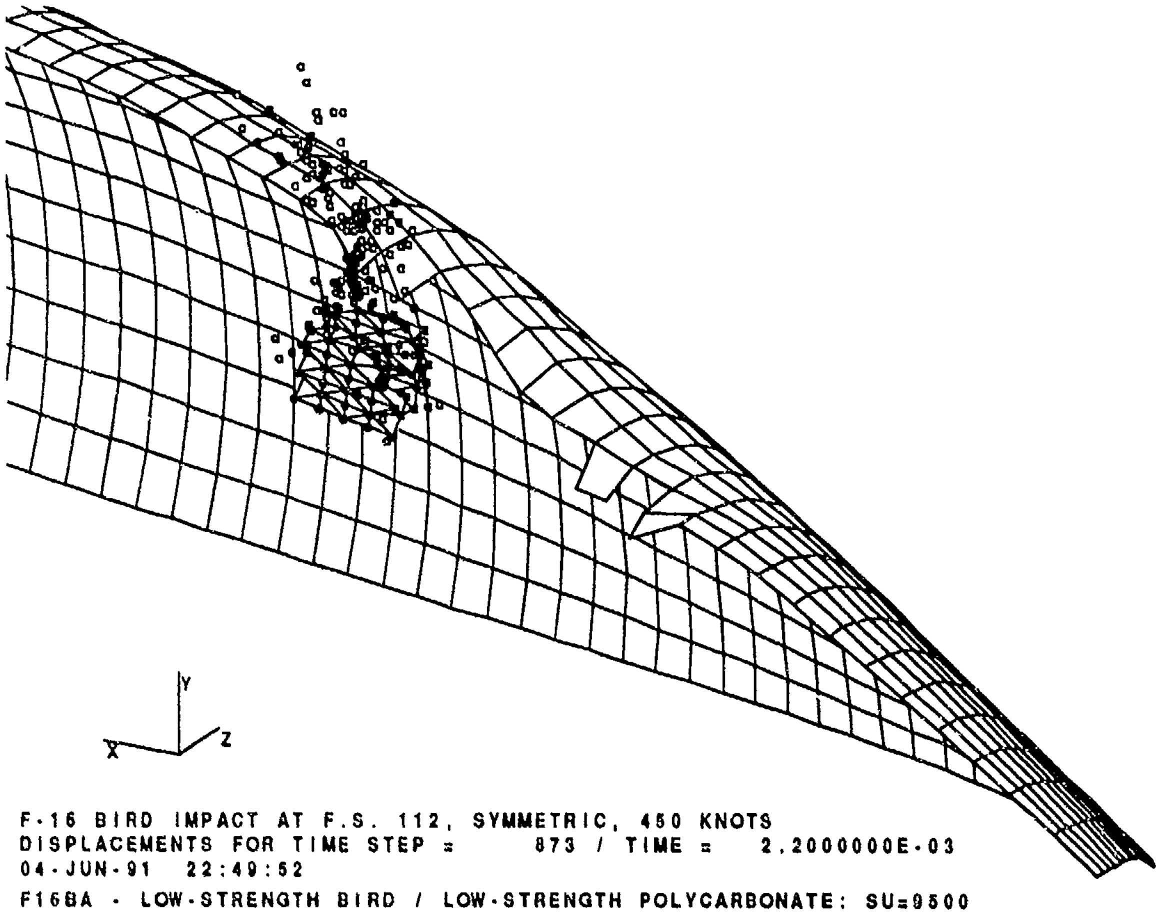

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 450 KNOTS
DISPLACEMENTS FOR TIME STEP = 873 / TIME = 2.2000000E-03
04-JUN-91 22:49:52
F16BA - LOW-STRENGTH BIRD / LOW-STRENGTH POLYCARBONATE: SU=9500

Figure 3.21. Deformed Geometry of Failed Canopy at 450-Knot Impact.

Below, we summarize the PATRAN modeling directives needed to generate both the transparency and the bird geometry. Note that the contact elements (TRI elements in PATRAN) must be generated after the "real" finite elements, since those corresponding to slave surfaces will be retained only in the form of slave *nodes* in the final model. It should be mentioned also that the model generated below must be EQUIVALENCED in PATRAN to merge coincident nodes; these commands are not reproduced in the listing since they consist chiefly of menu responses.

Bird Geometry

```
GR,1,,0/0/0
GR,2,,2.1/0/0
LI,1,ARC,0/0/0/0/0/1/45,2
LI,2,ARC,0/0/0/0/0/1/45,3
GR,5,,0.8/0.8/0.
GR,6,,1.05/0/0
GR,7,,0/1.05/0
LI,10,ARC,0/0/0/0/0/-1/45,2
LI,11,ARC,0/0/0/0/0/-1/45,8
GR,10,,0.8/-0.8/0
GR,11,,0/-1.05/0
LI,3,2G,,1,6
LI,4,2G,,6,2
LI,5,2G,,1,7
LI,6,2G,,7,4
LI,7,2G,,5,6
LI,8,2G,,5,7
LI,9,2G,,3,5
LI,12,2G,,1,11
LI,13,2G,,9,11
LI,14,2G,,6,10
LI,15,2G,,10,11
LI,16,2G,,8,10
PAT,1,2L,,3,8
PAT,2,2L,,2,8
PAT,3,2L,,1,7
PAT,4,2L,,3,15
PAT,5,2L,,11,15
PAT,6,2L,,10,14
HPAT,1T6,EXT,0/0/8.4,1T6
HPAT,11T16,ROTATE,0/0/0/0/1/0/-90,1T6
HPAT,1T6,DEL
HPAT,1T6,TRANS,8.295/24.865/0,11T16
HPAT,11T16,DEL
```

Transparency Geometry

```
VIEW
6
-30,50,40
SET,LAB1,OFF
SET,LAB2,OFF
PLOT
GRID,500,,80/32/15
GRID,401,,-9.6466217/11.733902/0.0
GRID,402,,-8.1981583/9.657295/6.80838490
GRID,403,,-6./14.042999/0.0
GRID,404,,-6./9.652824/8.06249428
GRID,405,,3./19.615303/0.0
GRID,406,,2.999992/9.735832/11.2527800
GRID,407,,12./24.868202/0.0
```

```
GRID,408,,12./10.000626/12.9540672
GRID,409,,21./29.506805/0.0
GRID,410,,20.999985/10.451752/13.8953772
GRID,411,,30./32.737701/0.0
GRID,412,,30./11.093910/14.3166628
GRID,413,,39./34.642700/0.0
GRID,414,,39./11.932182/14.4222097
GRID,415,,48./35.502502/0.0
GRID,416,,48./12.972084/14.4203587
GRID,417,,57./35.627899/0.0
GRID,418,,57./14.219543/13.9449654
GRID,419,,66./35.238403/0.0
GRID,420,,66./15.681000/13.2272902
GRID,421,,79.991043/33.902252/0.
GRID,422,,71.540497/16.690796/12.7192183
GRID,423,,-9.5191574/11.551163/2.47858191
GRID,424,,-9.0153732/10.828903/4.78343964
GRID,425,,-6./13.552261/3.25448656
GRID,426,,-6./12.001816/5.97303820
GRID,427,,3./18.333733/5.30707264
GRID,428,,2.999992/14.722885/9.17022038
GRID,429,,12./22.767708/7.00296783
GRID,430,,12./17.187294/11.5525970
GRID,431,,20.999985/26.665382/8.44793892
GRID,432,,20.999969/19.360451/13.4951038
GRID,433,,30./29.437210/9.42669487
GRID,434,,30./21.036125/14.8301888
GRID,435,,39./31.164444/9.86274910
GRID,436,,39./22.312538/15.4056635
GRID,437,,48./32.064651/9.80831242
GRID,438,,48./23.295860/15.3382416
GRID,439,,57./32.385376/9.37167931
GRID,440,,57./24.063339/14.7139320
GRID,441,,66./32.319046/8.65612888
GRID,442,,66./24.737915/13.7020693
GRID,443,,78.652328/31.175598/8.00284386
GRID,444,,75.374023/24.498642/12.6071825
GRID,500,DEL
LINE,401,4G,,401/423/424/402
LINE,402,4G,,403/425/426/404
LINE,403,4G,,405/427/428/406
LINE,404,4G,,407/429/430/408
LINE,405,4G,,409/431/432/410
LINE,406,4G,,411/433/434/412
LINE,407,4G,,413/435/436/414
LINE,408,4G,,415/437/438/416
LINE,409,4G,,417/439/440/418
LINE,410,4G,,419/441/442/420
LINE,411,4G,,421/443/444/422
PATCH,501T510,2L,,401T410,402T411
PATCH,401T420,BREAK,0.5,501T510
PATCH,501T510,DEL
```

Finite Element Mesh

```
GFEG,P401T402,,9/3
GFEG,P403T418,,9/7
GFEG,P419T420,,9/9
CFEG,P401T420,QUAD,,1
GFEG,HP1T6,,3/3/9
CFEG,HP1T6,TET,,1
```

Contact Elements and Boundary Conditions

```
SET,LINES,0
ERASE
HP,1T6,PLOT
PAT,401T420,PLOT
CFEG,P405T415B2,TRI/3/1,T1
1
1
1
1
1
1
CFEG,HP1T6,TRI/3/2,T11
1
1
1
1
1
1
DFEG,HP1,DISP,//0/0/0/0,1,F1
DFEG,HP2,DISP,//0/0/0/0,1,F2
DFEG,HP4,DISP,//0/0/0/0,1,F3
DFEG,HP5,DISP,//0/0/0/0,1,F4
DFEG,P401T419B2,DISP,//0///,1,EDGE1
DFEG,P401T402,DISP,0/0/0///,1,EDGE4
1
DFEG,P402T420B2,DISP,0/0/0///,1,EDGE3
DFEG,P419T420,DISP,0/0/0///,1,EDGE2
1
```

Additional data needed to complete the model consists of material and laminate definitions, initial velocities for the bird nodes, and solution control parameters. Typical input for these items is shown in the listing below, which describes parameters for a 350-knot impact against a ½ inch polycarbonate transparency. The example uses material properties for the high-strength bird shown in previous figures, and uses a conservative ultimate stress of 13,000 psi for the polycarbonate. This model's analysis results predicted a small failure in the transparency (2 plate elements failed), with no bird elements marked as failed.

```
TITLE
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 350 KNOTS
$  Data translated by PATX3D - 26-OCT-90   03:26:56
$  Last mod 17-May-91 for X3D 2.5 - this is F16BI.DAT:
$    1. High-strength bird - should be no failure in TET elements.
$    2. Lower-strength polycarbonate: SU=13000 psi.
END

PARAMETERS
$ TIME TMAX   TREST   DTMIN  DTMAX  INCR  INCMAX  IREST  NINTPL  ISTAT ITRAC
  0.   0.018  0.00020 0.     0.     0     170000  1      5       0     20
END

INITIAL VELOCITY
   7093.65  0.  0.  1004 1300 1      ! 350 KNOTS
$  8107.03  0.  0.  1004 1300 1      ! 400 KNOTS
END
```

```
RIGID
$ Use a rigid wall at z=0; all nodes must remain positive z.
$ This will constrain bird nodes which may be set loose from
$ failed tet elements.
$ 15-Apr-91
 0.0, 0.0, 1.0, 0.0
END

TOLERANCES
  0.90  0.1  0.01 0.01 0.01
END

LAMINATES
 1   1   1   -0.5   0.0
END

TANG
$ F-16 transparency symmetry-plane rotational constraints.  22-Apr-91
    1    2    3   52   53   54   55   56   57  154  155  156  157  158  &
  159  256  257  258  259  260  261  358  359  360  361  362  363  460  &
  461  462  463  464  465  562  563  564  565  566  567  664  665  666  &
  667  668  669  766  767  768  769  770  771  868  877  886  895  904  &
  913  922  931
END

MAT2D
$ Polycarbonate. -- 23-Oct-90
$ Mod for alternate curve fit; lower SU to 13,000 -- 12-Apr-91
  1 0 0.0001113 325000.  0.36  1.0  7140.  36100.  5.088E-6  0.083333  13000.
$ Initial curve fit values are below...
$ 1 0 0.0001113 325000.  0.36  1.0  9400.  36100.  13.       1.        16000.
END

MAT3D
$ Water bird -- Mat type 3 -- 10-Apr-91
$ MATLNO  RHO          BULK2                SHEAR         RSCALE  H         Q2   Q1
$  ITYPE        BULK1          BULK3  BULKT          YIELD     REXP    SULT
1 3 0.0000888 3.37E5 7.29E5 20.2E5 1000. 30000. 3000. 0. 1. 300. 4500. 1.5 .06
END

TRACE
$ Track some of the F-16 transparency midline nodes (z=0).
    1    3   53   55   57  155  157  159  257  259  261  359  361  &
  363  461  463  465  563  565  567  665  667  669  767  769  771  &
  877  895  913  931
END

NODE COORDINATES
        1       -9.646622         11.73390        0.0000000E+00
        2       -7.823311         12.88845        0.0000000E+00
        3       -6.000000         14.04300        0.0000000E+00

      ...
```

End of the example. All remaining data blocks are generated by PATX3D.

SECTION 4
SUMMARY AND CONCLUSIONS

This report describes an improved analytical capability for soft-body impact simulation. Important advances have been made in the modeling of impact loads, nonlinear materials, and layered wall constructions. The explicit approach adopted in the present work exploits the strengths of the current generation of supercomputer hardware, so that analysis cost and turnaround times are reduced significantly over the previous generation of birdstrike analysis software. Our experiences thus far in performing applications indicates that the solution methodology is very reliable, requiring minimal user intervention to avoid or correct problems with the solution. The implicit methods used in our earlier work on these problems demand a great deal of user attention for stable, accurate, and convergent results, while the explicit technique is relatively trouble-free.

We feel that the work reported herein is a significant step toward a reliable capability for design screening and parametric investigations. The most important work which is needed to complete such a capability includes:

> *Model Validation.* Additional comparisons of analytical predictions with full-scale impact test data are needed to develop confidence in the accuracy of the analysis and our knowledge of its limitations.
>
> *Materials Characterization.* The transparency materials in wide use are high-polymer compounds with very complex characteristics. Much more experimental and analytical work is needed to understand these materials adequately and model their behavior faithfully.

With a modest effort in these areas, the techniques and software described herein can become a truly useful and reliable tool for the design and evaluation of a new generation of impact-resistant transparency systems.

REFERENCES

The references listed below describe much of the theoretical basis of the X3D computer program. Some include instructive sample problems which have been used in the validation of the code. All of them contain valuable background information on dynamic analysis and are highly recommended.

Anderson, C. E. Jr. (1987), An Overview of the Theory of Hydrocodes, *Int. J. Impact Engng.* **5**, 33-59.

Barber, J. R., H. R. Taylor, and J. M. Wilbeck (1978), *Bird Impact Forces and Pressures on Rigid and Compliant Targets*, AFFDL-TR-77-60, Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio.

Belytschko, T. and B. J. Hsieh (1973), Nonlinear Transient Finite Element Analysis with Convected Coordinates, *Int. J. Num. Meth. Engng.* **7**, 255-271.

Belytschko, T. (1983), An Overview of Semidiscretization Procedures, in T. Belytschko and T. J. R. Hughes (eds.), *Computational Methods for Transient Analysis*, Chapter 1, North-Holland Co., Amsterdam.

Belytschko, T., J. I. Lin, and C.-S. Tsay (1984), Explicit Algorithms for the Nonlinear Dynamics of Shells, *Comp. Meths. Appl. Mech. Engng.* **42**, 225-251.

Belytschko, T. and J. I. Lin (1986), Eigenvalues and Stable Time Steps for the Bilinear Mindlin Plate Element, Preprint, Northwestern University.

Brockman, R. A. (1981), MAGNA: A Finite Element System for Three-Dimensional Static and Dynamic Structural Analysis, *Comp. Struct.* **13**, 415-423.

Brockman, R. A. (1982), *MAGNA (Materially And Geometrically Nonlinear Analysis), Part I - Finite Element Analysis Manual*, AFWAL-TR-82-3098, Part I, Air Force Wright Aeronautical Laboratories, Wright Patterson Air Force Base, Ohio.

Brockman, R. A. (1983), Current Problems and Progress in Aircraft Transparency Impact Analysis, in S. A. Morolo (ed.), *Proceedings of the 14th Conference on Aerospace Transparent Materials and Enclosures*, AFWAL-TR-83-4154, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio, 1058-1082.

Brockman, R. A. (1984), *Finite Element Analysis of Soft-Body Impact*, AFWAL-TR-84-3035, Wright-Patterson Air Force Base, Ohio.

Brockman, R. A. (1987), *A Simple Finite Element Approximation for Multilayered Plates and Shells*, UDR-TR-87-113, University of Dayton.

R. A. Brockman (1988), Computational Methods for Soft-Body Impact Problems, in S. Atluri and G. Yagawa (eds.), *Computational Mechanics 1988*, Volume 1, pp. 10.vi.1-4, Springer-Verlag.

Bouchard, M. P. (1989), Finite Element Analysis of the B-1B Windshield System, *Proceedings of the 15th Conference on Aerospace Transparent Materials and Enclosures*, Monterey, CA, January 1990.

Chatterjee, S. N. and S. V. Kulkarni (1979), Shear Correction Factors for Laminated Plates, *AIAA J.* **17**(5), 498-499.

Chiapetta, R., T. Belytschko, and J. Rouse (1973), *A Computer Code for Dynamic Stress Analysis of Media-structure Problem with Nonlinearities (SAMSON)*, AFWL-TR-72-104, Air Force Weapons Laboratory, Kirtland AFB, New Mexico.

Chow, T. S. (1971), On the Propagation of Flexural Waves in an Orthotropic Laminated Plate and its Response to an Impulsive Load, *J. Comp. Matls.* **5**, 306-319.

Chow, T. S. (1975), Theory of Unsymmetric Laminated Plates, *J. Appl. Physics* **46**, 219-221.

Dodds, R. H. Jr. (1987), Numerical Techniques for Plasticity Computations in Finite Element Analysis, *Comp. Struct.* **26**(5), 767-779.

Flanagan, D. P. and T. Belytschko (1981), A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control, *Int. J. Num. Meth. Engng.* **17**, 679-706.

Goudreaux, G. L. and J. O. Hallquist (1982), Recent Developments in Large-Scale Finite Element Lagrangian Hydrocode Technology, *Comp. Meth. Appl. Mech. Engng.* **33**, 725-757.

Hughes, T. J. R., M. Cohen, and M. Haroun (1978), Reduced and Selective Integration Techniques in the Finite Element Analysis of Shells, *Nuc. Engng. Des.* **46**, 203-222.

Hughes, T. J. R. (1983), Analysis of Transient Algorithms with Particular Reference to Stability Behavior, in T. Belytschko and T. J. R. Hughes (eds.), *Computational Methods for Transient Analysis*, Chapter 2, North-Holland Co., Amsterdam.

Johnson, G. R., D. D. Colby, and D. J. Vavrick (1979), Three Dimensional Computer Code for Dynamic Response of Solids to Intense Impulsive Loads, *Int. J. Num. Meth. Engng.* **14**, 1865-1871.

Jones, R. M. (1975), *Mechanics of Composite Materials*, Scripta Book Co., Washington, D. C.

Key, S. W. (1974), *HONDO - A Finite Element Computer Program for the Large Deformation Dynamic Response of Axisymmetric Solids*, SLA-74-0039, Sandia National Laboratories, Albuquerque, New Mexico.

Krieg, R. D. and S. W. Key (1973), Transient Shell Response by Numerical Time Integration, *Int. J. Num. Meth. Engng.* **17**, 273-286.

Malvern, L. E. (1969), *An Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall Co., Englewood Cliffs, N.J.

Marchertas, A. H. and T. B. Belytschko (1974), *Nonlinear Finite Element Formulation for Transient Analysis of Three-Dimensional Thin Structures*, Report ANL-8104, Argonne National Laboratory.

McCarty, R. E. (1983), MAGNA Computer Simulation of Bird Impact on the TF-15 Aircraft Canopy, in S. A. Morolo (ed.), *Proceedings of the 14th Conference on Aerospace Transparent Materials and Enclosures*, AFWAL-TR-83-4154, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio, 973-1008.

McCarty, R. E. (1985), Three Dimensional Nonlinear Dynamic Finite Element Analysis for the Response of a Thick laminated Shell to Impact Loads, *Proc. AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics, and Materials Conference*, Orlando, Florida.

McCarty, R. E. (1989a), Analytic Assessment of Bird Impact Resistant T-46A Aircraft Windshield System Designs using MAGNA, *Proc. 15th Conference on Aerospace Transparent Materials and Enclosures*, Monterey, CA.

McCarty, R. E. (1989b), MAGNA Analysis of Space Shuttle Orbiter Windshield System Bird Impact Protection, *Proc. 15th Conference on Aerospace Transparent Materials and Enclosures*, Monterey, CA.

Mindlin, R. D. (1951), Influence of Rotatory Inertia and Shear on the Bending of Elastic Plates, *J. Appl. Mech.* **18**, 1031-1036.

Newmark, N. M. (1959), A Method of Computation for Structural Dynamics, *J. Eng. Mech. Div.* **85**, 67-94.

Noor, A. K. (1975), Stability of Multilayered Composite Plates, *Fibre Sci. Tech.* **8**, 81-89.

Noor, A. K. and W. S. Burton (1989), Stress and Free Vibration Analyses of Multilayered Composite Plates, *Compos. Struc.* **11**(2).

Pagano, N. J. (1970), Exact Solutions for Rectangular Bidirectional Composites and Sandwich Plates, *J. Comp. Matls.* **4**, 20-34.

Palazotto, A. N. and W. P. Witt (1985), Formulation of a Nonlinear Compatible Finite Element for the Analysis of Laminated Composites, *Comput. Struct.* **21**, 1213-1234.

Reddy, J. N. (1984a), *Introduction to the Finite Element Method,* McGraw-Hill Co., New York.

Reddy, J. N. (1984b), *Energy and Variational Methods in Applied Mechanics,* John Wiley & Sons, New York.

Watt, R. M. (1978), *AFFDL F-16 Canopy Bird Impact Test,* Test Report, Project V41S-18A, von Karman Gas Dynamics Facility, Arnold Engineering Development Center, Arnold AF Station, Tennessee.

West, B. S. and R. A. Brockman (1980), *Evaluation of Bird Load Models for Dynamic Analysis of Aircraft Transparencies,* AFWAL-TR-80-3092, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio.

Whitney, J. M. and N. J. Pagano (1970), Shear Deformation in Heterogeneous Anisotropic Plates, *J. Appl. Mech.* **37**, 1031-1036.

Whitney, J. M. (1973), Shear Correction Factors for Orthotropic Laminates under Static Load, *J. Appl. Mech.* **40**, 302-304.

Zukas, J. A., T. Nicholas, H. F. Swift, L. B. Greszczuk, and D. R. Curran (1982), *Impact Dynamics,* John Wiley & Sons, New York.

APPENDIX A
X3D INPUT DATA

OVERVIEW

Input data are supplied to X3D on a formatted file containing a series of *input data blocks.* Each block begins with a descriptive header, lists the data, and ends with a record containing the string "END". Data in an input block consists of one or more records; typically there is one record for each element, node, or other model entity. Within a record, data may be arranged in free format, with data items separated by commas or blanks. Data blocks may appear in the input file in any order; however, large models usually are processed more efficiently if the shorter blocks are placed near the beginning of the file. The Table below summarizes all of the input block types recognized by X3D.

TABLE A.1
SUMMARY OF X3D INPUT BLOCKS

INPUT BLOCK NAME	DESCRIPTION
BODY forces	Body forces for entire model
BOUNdary conditions	Nodal displacement constraints
CONTact	Surface contact data
EDIT	Edit model entities
ELEMents	Element connections and properties
FORCe	Nodal point force data
HISTory functions	Time history functions
INITial conditions	Initial velocity conditions
LAMInates	Plate cross-section data
LINKed elements	Element failure links
MASS	Concentrated mass data
MAT2d	Plate (2-D) material properties
MAT3d	Solid (3-D) material properties
MPOSt	Request MPOST format output file
NODE coordinates	Initial nodal positions
PARAmeters	Solution control parameters
PRESsure	Element pressure loading data
REFErence volume	Coordinate limits for plotting
RESTart	Request restart of a previous analysis
RIGId walls	Rigid barrier constraints
SOFT body	Soft body impact data
TANGential rotation	Tangential rotation BC's for shells
TITLe	Descriptive problem title
TOLErances	Additional solution parameters
TRACe nodes	Trace nodes for summary output
@DEBug	Diagnostic output requests

For new analyses (as opposed to restarts) several input data blocks, such as the **NODE** and **ELEMent** blocks, are required; others, like **MASS** and **CONTact**, are needed only when the problem calls for them. In a *restart* run, an analysis can be resumed from a point of interruption, possibly with modified data. When a restart is performed, normally only the modified data blocks are required. Table A.2 shows the input block requirements and options for both new and restarted analyses with X3D.

TABLE A.2
SUMMARY OF INPUT DATA BLOCK REQUIREMENTS
FOR NEW AND RESTARTED ANALYSES

INPUT DATA BLOCK	STATUS FOR NEW ANALYSIS	STATUS FOR RESTART RUN	ACTION IF PRESENT AT RESTART
BODY	Optional	N/A	
BOUN	**REQUIRED**	N/A	
CONT	Optional	Optional	Replace old data
EDIT	Optional	Optional	Modify old data
ELEM	**REQUIRED**	N/A	
FORC	Optional	Optional	Replace old data
HIST	Optional	Optional	Replace old data
INIT	Optional	N/A	
LAMI	Optional	N/A	
LINK	Optional	N/A	
MASS	Optional	N/A	
MAT2	Optional	N/A	
MAT3	Optional	N/A	
MPOS	Optional	Optional	Replace old data
NODE	**REQUIRED**	N/A	
PARA	**REQUIRED**	**REQUIRED**	Replace old data
PRES	Optional	Optional	Replace old data
REFE	Optional	N/A	
REST	N/A	**REQUIRED**	Read restart file
RIGI	Optional	N/A	
SOFT	Optional	N/A	
TANG	Optional	N/A	
TITL	**REQUIRED**	Optional	Replace old data
TOLE	Optional	Optional	Replace old data
TRAC	Optional	Optional	Replace old data
@DEB	Optional	Optional	Replace old data

INPUT FORMAT CONVENTIONS

Input formatting in X3D is relatively flexible. Problem data may be entered in free format, and may contain omitted items, continuation lines, and embedded comments. Data must be entered in columns 1 through 80 in order to be processed correctly. Important special characters in the X3D input include:

!	Comment character; the remainder of the current line will be ignored
$	Comment character; equivalent to !
&	Continuation character; the current record continues on the next line
,	Separator for data items; a string of commas (,,,,) may be used to skip a series of data items

The examples below illustrate the use of these special characters, as well as the conventions used in interpreting input to X3D.

All X3D input items follow FORTRAN naming conventions. That is, names beginning with I-N denote integer values, while names beginning with A-H or O-Z are floating-point (real) data. The only exceptions are alphanumeric (character) inputs, which consist only of the block headers, END lines, and problem title.

As an example, suppose that a particular input record calls for six data items: N1, N2, N3, A, B, and C. By the FORTRAN convention, N1, N2, and N3 are integer data, and A, B, and C define real-valued parameters.

X3D will read all six items as floating-point numbers, and truncate them to integer values as necessary. Omitting an item of data is equivalent to entering a zero value (although X3D may supply non-zero defaults for some data items).

Table A.3 shows several possible data records corresponding to the data items of the previous example: N1, N2, N3, A, B, C. The center column of the Table lists the values which X3D actually assigns to these six variables. The right-hand column contains comments explaining the rules or conventions which have been used in interpreting the input data.

TABLE A.3
EXAMPLES OF X3D INPUT FORMAT CONVENTIONS

INPUT RECORD	VALUES ASSIGNED	EXPLANATION & COMMENTS

1 2 3 4 5.5 6	N1 = 1
	N2 = 2
	N3 = 3
	A = 4.0
	B = 5.5
	C = 6.0
		Multiple spaces are the same as a single space
		Integer values are converted to reals as necessary

1.5, 2,, 4.4, 5 6	N1 = 1
	N2 = 2
	N3 = 0
	A = 4.4
	B = 5.0
	C = 6.0
		Real values are truncated to integers as necessary
		Repeated commas signify an omitted value

1.E4,,,,, 27	N1 = 10000
	N2 = 0
	N3 = 0
	A = 0.0
	B = 0.0
	C = 27.0
		Exponents may be used anywhere, and are converted to integers where appropriate
		A data item is assumed between each pair of commas

2,, 7 1 &
8.5
	N1 = 2
	N2 = 0
	N3 = 7
	A = 1.0
	B = 8.5
	C = 0.0
		& indicates that the record is continued on the next line
		When trailing items are omitted, zero values are assigned

$ a comment line
1 -0.2, &! note
! another comment
3,,,   ! note
-1
	N1 = 1
	N2 = 0
	N3 = 3
	A = 0.0
	B = 0.0
	C = 0.0
		Comments after ! or $ ignored
		-0.2 is truncated to zero
		Last line (-1) is not read since the preceding line contains no &

X3D FILE SUMMARY

Although X3D works entirely in memory, operating the code involves several input and output files. They are:

FILE NAME	FORM	DESCRIPTION
(logical unit 5)	Formatted	Input data
(logical unit 6)	Formatted	Printed output
NREST	Binary	Restart output
OREST	Binary	Restart input
TRACE	Formatted	Trace node output
MPOST	Formatted	MAGNA-compatible results file

All files use the system default naming conventions. For instance, the TRACE file would be named TRACE on a CRAY/UNICOS system, where file extensions are not used; on a VAX/VMS system, where the default file type for FORTRAN-generated files is .DAT, the same file would be called TRACE.DAT. The NREST, OREST, TRACE, and MPOST files are opened using the basic file names only.

The input and output files are not opened explicitly by the program, due to differences in the ways various systems treat the standard FORTRAN input and output units. On most UNIX-based machines, these files correspond to the system's standard input and standard output. For other systems, the user may need to take additional measures. For example, on a VAX/VMS system, input must either be stored on the file FOR005.DAT, or assigned to the logical name FOR005 (e.g., ASSIGN MYDATA.DAT FOR005).

Two external files are used for restart functions. NREST is the "new" restart file; that is, the restart output file, and is created whenever restart output is requested. OREST, the "old" file, supplies restart input to the program. OREST must exist whenever an analysis restart is requested. Note also that the restart file is used by postprocessing utilities to recover results for plotting. Therefore, one usually requests restart file output even if the analysis will not be resumed at a later time.

In general, restart files are not portable between machines of different architectures. For example, a VAX/VMS user could not copy a restart file from a Sun to the VAX and then run X3POST using that file. There are some exceptions to this restriction (CRAY machines, in particular, using special system utilities), but these are not discussed here. This means that a user must run X3POST on the machine that created the restart file, or on a machine like it.

Two additional external files, TRACE and MPOST, contain formatted results suitable for listing, plotting, or other postprocessing. The TRACE file contains formatted output for a user-specified list of nodes. The MPOST file contains model data and results in a form compatible with output from the MAGNA finite element program. Both of these files are ASCII files and may be transferred between computer systems without difficulty.

X3D INPUT SUMMARY

The remainder of this Appendix contains detailed descriptions of the individual X3D input blocks. Table A.4 below gives a summary of the data items required in each block.

TABLE A.4
SUMMARY OF X3D INPUT PARAMETERS

BODY	BX, BY, BZ
BOUN	IBEG, IEND, ISTEP, IFIX1, IFIX2, IFIX3, IFIX4, IFIX5, IFIX6
CONT	ISET, NODEA, NODEB, NODEC
	END
	JSET, NODE1, NODE2, NODE3, ..., NODEn
	END
EDIT	/DEL N1, N2, N3, ...
ELEM	IEL, MATL, N1, N2, N3, N4, N5, N6, N7, N8, IEGEN, INGEN
FORC	NODE, IDIR, IFUNC, SCALE
HIST	ICURV, NPT, t_1, F_1, t_2, F_2, ..., t_{NPT}, F_{NPT}
INIT	V_x, V_y, V_z, NFIRST, NLAST, NINCR
LAMI	LAMNO, NLAYER, $MATL_1$, $MATL_2$, ..., $MATL_{NLAYER}$, Z_1, Z_2, ..., $Z_{NLAYER+1}$
LINK	M_1, NS_1, M_2, NS_2, ..., M_n, NS_n
MASS	NODE, TRMASS, XRI, YRI, ZRI
MAT2	MATL, *1*, RHO, E, XNU, ALPHA, YIELD, H, RSCALE, REXP, SULT
or	MATL, *2*, RHO, E1, E2, XNU12, G12, G13, G23, SF, SD
MAT3	MATL, *1*, RHO, BULK1, BULK2, BULK3, SHEAR, YIELD, RSCALE, REXP, H, SULT, Q2, Q1
or	MATL, *2*, RHO, BULK1, BULK2, BULK3, VISC, Q2, Q1
or	MATL, *3*, RHO, BULK1, BULK2, BULK3, BULKT, SHEAR, YIELD, RSCALE, REXP, H, SULT, Q2, Q1
MPOST	(none)
NODE	IPT, X, Y, Z, IGEN
PARA	TIME, TMAX, TREST, DTMIN, DTMAX, INCR, INCMAX, IREST, NINTPL, ISTAT, ITRACE
PRES	IBEG, IEND, IFACE, IFUNC, SCALE
REFE	XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
REST	INCOLD
RIGI	A, B, C, D
TANG	N_1, N_2, N_3, ... N_n
TITL	ITITLE
TOLE	DTFRAC, HGDAMP, HGCONS(1), HGCONS(2), HGCONS(3), FRATIO, VRATIO
TRAC	N_1, N_2, N_3, ..., N_n
@DEB	KEYWORD = value

BODY FORCES
INPUT DATA BLOCK

Defines uniform body forces (such as gravity) which act on the model.

STATUS: Optional (ignored when restarting)

FORMAT:

```
BODY
BX, BY, BZ
END
```

EXAMPLE:

```
BODY
0., 0., -386.4
END
```

VARIABLES:

BX	Body force per unit mass in the X direction
BY	Body force per unit mass in the Y direction
BZ	Body force per unit mass in the Z direction

NOTES:

1. Body forces per unit mass have units of acceleration.

2. Positive values act in the positive global coordinate directions.

3. BX, BY, BZ are constant with respect to time, and act on all finite elements in the model.

BOUNDARY CONDITIONS

INPUT DATA BLOCK

Defines velocity constraints at the nodes of the model.

STATUS: Optional (ignored when restarting)

FORMAT:

```
BOUN
IBEG, IEND, ISTEP, IFIX1, IFIX2, IFIX3, IFIX4, IFIX5, IFIX6
<repeat as needed>
END
```

EXAMPLE:

BOUN

1, 15, 2, 1, 5, 6 ! Nodes 1,3,5,...,15 fixed in U_x, R_y, R_z

7, 10,, 3 ! Nodes 7-10 fixed in U_z

20,,,1,2,3 ! Node 20 fixed in U_x, U_y, U_z

END

VARIABLES:

IBEG	First node in the series of points to be constrained
IEND	Last node in the series of points to be constrained
ISTEP	Node number increment
IFIXn	Values of nodal degree of freedom components to be constrained. Component numbers 1, 2, 3 correspond to translational DOFs U_x, U_y, and U_z, respectively; components 4, 5, 6 are rotational DOFs R_x, R_y, and R_z.

NOTES:

1. IBEG, IEND, and ISTEP define a sequence of nodes to which a common set of velocity constraints will be applied. The sequence consists of nodes IBEG, IBEG+ISTEP, IBEG+2*ISTEP, ..., IEND.

2. If IEND is omitted, only node IBEG is constrained.

3. If ISTEP is omitted, an increment of 1 is assumed. If present, ISTEP must be positive.

4. IFIX values may be entered in any order, omitting component numbers which are unnecessary.

CONTACT SURFACES
INPUT DATA BLOCK

Defines opposing surfaces which may interact through contact constraints.

STATUS: Optional (replaced if present when restarting)

FORMAT:

```
CONT
ISET, NODEA, NODEB, NODEC
<repeat as needed to define all master surfaces>
END
JSET, NODE1, NODE2, NODE3, ..., NODEn
<repeat as needed to define all slave nodes>
END
```

EXAMPLE:

```
CONT
1, 400, 401, 402          ! 3-D master surface defined by 3 nodes
2, 10, 12, -1             ! 2-D master surface in Y-Z plane
END                       ! (End of master surface data)
1, 20, 512, 316 &         ! Slave nodes for set 1, continued on next line
      18, 358
2, 15, 19, 45             ! Slave nodes for set 2
END                       ! (End of slave node data)
```

VARIABLES:

ISET	Contact constraint set number for master surface
NODEA	First connected node defining a master surface segment
NODEB	Second connected node defining a master surface segment
NODEC	Third connected node defining a master surface segment
JSET	Contact constraint set number for current list of slave nodes
NODEn	Slave node number(s)

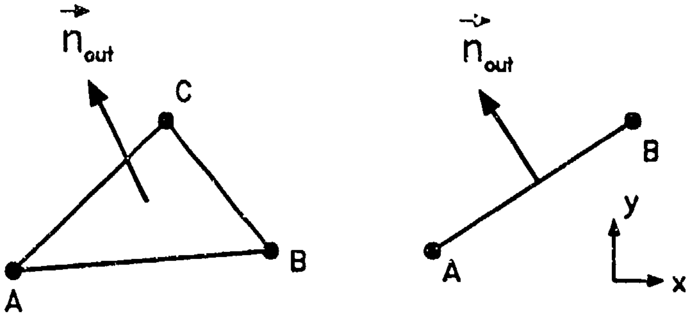
NOTES:

1. Contact constraints will be enforced between all master surface segments and slave nodes which have the same constraint set number (ISET=JSET).

2. A master surface segment normally is a triangle defined by three nodes. It may correspond to a portion of the surface of a solid element, or of a quadrilateral plate element. Boundary surface segments (e.g., those which lie on symmetry planes) have a different form (see Note 4).

NOTES (Continued):

3. The ordering of the three nodes defining a master surface defines its orientation. When viewed from the "outside", the three nodes NODEA, NODEB, NODEC appear in *counterclockwise* order.



4. On symmetry planes or partially constrained boundary lines, a two-node master surface is appropriate. Usually this occurs when the slave nodes must move in a plane. For a 2-D master segment, NODEA and NODEB specify the nodes at the two endpoints of the segment; *NODEC must be set to -1, -2, or -3 to define the global axis (X, Y, or Z) normal to the plane containing the master segment and the slave nodes.* The outward direction for a 2-D master surface segment is defined by the cross product of a vector along the given normal axis (X, Y,or Z) with a vector directed from NODEA to NODEB. Therefore, the outward direction of the segment can be reversed simply by interchanging NODEA and NODEB.

5. Master surface segments may be defined in any order. Set numbers (ISET) should be numbered from 1 through the maximum number of constraint sets.

6. Two- and three-node master surfaces may appear in the same set; however, we have not yet encountered a situation in which this is appropriate.

7. NODE1, ..., NODEn is a list of nodes to be included in a particular constraint set. Nodes may be listed in any order.

NOTES (Continued):

8. The contact solution is most efficient when a given number of master surfaces and slave nodes are segregated into as many separate sets as possible, since the number of potential contact pairs (master surface, slave node) is minimized. *A surface segment or slave node may appear in more than one set,* so potential contact constraints in adjacent areas can be overlapped. It is more efficient to duplicate slave nodes than surface segments.

EDIT
INPUT DATA BLOCK

Defines editing operations to be performed on the model (usually when restarting).

STATUS: Optional

FORMAT:

```
EDIT
KEYWORD, N1, N2, N3, ...
<repeat as needed to define all edits>
END
```

EXAMPLE:

```
EDIT
/DEL 10, 22, 312            ! Mark elements 10, 22, and 312 as failed
/DEL 502, 26      &         ! Specify more elements with either another
    362, 1004               ! 'delete' command, or continuation line(s)
END
```

VARIABLES:

KEYWORD	Editing command
Ni	List of entities to be operated upon

NOTES:

1. Currently, the only valid editing keyword is /DEL (delete elements). This command may be used to remove inactive or unimportant elements from the solution, usually to reduce computation time or achieve a larger time step in the remaining solution. Elements modified with /DEL are deleted *permanently* from the solution.

ELEMENTS
INPUT DATA BLOCK

Defines connections and materials for all finite elements in the model.

STATUS: Required (ignored when restarting)

FORMAT:

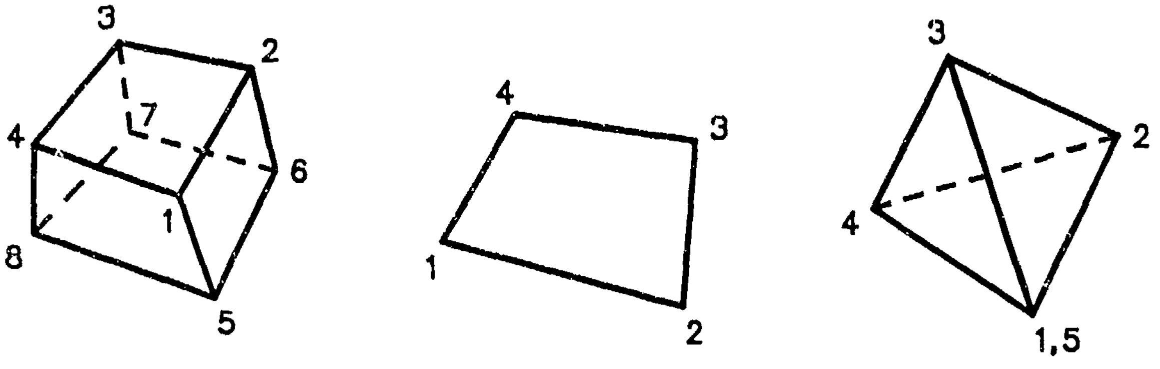
```
ELEM
IEL, MATL, N1, N2, N3, N4, N5, N6, N7, N8, IEGEN, INGEN
<repeat as needed to define all elements>
END
```

EXAMPLE:

```
ELEM
25, 1, 14,12,13,16,20,22,28,27        ! 3-D HEX element, material no. 1
30, 1,,,,,,,,, 1, 6                   ! Generate elements 26-30
4, 3, 7, 9, 12, 15                    ! Plate element, laminate no. 3
9, 3,,,,,,,,, 1, 10                   ! Generate elements 5-9
END
```

VARIABLES:

IEL	Element number to be defined
MATL	Material property number (solids) or laminate number (plates)
Ni	Nodes defining the element, listed in the order shown below
IEGEN	Element number increment for successive elements in a generation sequence
INGEN	Node number increment for successive elements in a generation sequence



ELEMENTS
INPUT DATA BLOCK (CONTINUED)

NOTES:

1. Finite elements must be numbered sequentially from 1 to N_e, the number of elements in the model. Elements may be defined in any order.

2. MATL refers to a material property ID for 3-D solid elements (see input block **MAT3D**). For plate/shell elements, MATL is a *laminate number*, as defined in the **LAMI** input block.

3. X3D identifies the type of element by the number of nodes connected to it. 3-D solid hexahedra (bricks) have eight nodes; tetrahedra are defined by five nodes, with $N_5 = N_1$. Plate/shell elements are defined by four corner nodes.

4. To generate elements, first enter the beginning element in the series, with IEGEN and INGEN omitted. On the next line, enter the last element in the series, with appropriate values for the generation parameters. For example, the data

   ```
   5, 1, 4, 6, 8, 9, 2, 10, 15, 12
   25, 1,,,,,,,,, 5, 4
   ```

   generates a sequence of elements with numbers 5, 10, 15, 20, and 25, in which the node numbers for each element are 4 greater than for the previous element. That is, the input above is equivalent to:

   ```
   5, 1, 4, 6, 8, 9, 2, 10, 15, 12
   10, 1, 8, 10, 12, 13, 6, 14, 19, 16
   15, 1, 12, 14, 16, 17, 10, 18, 23, 20
   20, 1, 16, 18, 20, 21, 14, 22, 27, 24
   25, 1, 20, 22, 24, 25, 18, 26, 31, 28
   ```

   Only nonzero node numbers are incremented during element generation.

5. Node numbers need not be entered for the second line of a generation sequence (see the example above).

6. The material property number for all elements in a generation sequence is the same as the property number for the last element input.

FORCES
INPUT DATA BLOCK

Defines time-dependent concentrated forces and moments at selected nodes.

STATUS: Optional

FORMAT:

FORC
NODE, IDIR, IFUNC, SCALE
<repeat as needed to define all forces>
END

EXAMPLE:

```
FORC
125, 3, 2, -100.           ! Force at node 125, acting in the Z direction.
                           ! The magnitude at any time is interpolated from
                           ! time history curve #2, and multiplied by -100.
END
```

VARIABLES:

NODE	Node at which the force is applied
IDIR	Loading direction: 1=F_x, 2=F_y, 3=F_z, 4=M_x, 5=M_y, 6=M_z
IFUNC	Number of time history function defining load magnitude versus time
SCALE	Scale factor to be applied to time function value

NOTES:

1. Positive forces act in the positive global coordinate directions. Positive moments act in the direction of a positive right-handed rotation about the axis in question.

2. Time history function values are obtained at a given time by linear interpolation of curve number IFUNC, and then multiplied by SCALE to obtain the actual load value at that time.

HISTORY FUNCTIONS
INPUT DATA BLOCK

Defines the time variation of element pressure forces and nodal loads.

STATUS: Optional (Required if a **PRESS**URE or **FORC**E input block is present)

FORMAT:

HIST

ICURV, NPT, t_1, F_1, t_2, F_2, ..., t_{NPT}, F_{NPT}

<repeat as needed to define all time functions>

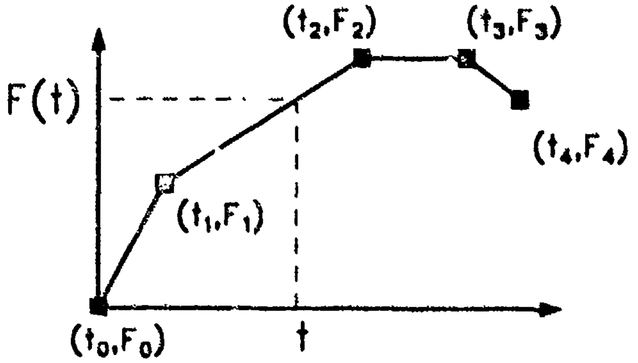
END

EXAMPLE:

```
HIST
1, 2, 0, 50, 1000, 50          ! Constant function F(t) = 50
3, 4                           ! Function no. 3, defined by 4 points:
0, 0, 0.0001, 1000  &          ! Linear rise up to F(.0001) = 1000,
0.0002, 0           &          ! linear decrease down to 0 at time t = 0.0002,
1, 0                           ! then F = 0 for all subsequent times
END
```

VARIABLES:

ICURV	ID number of time history function to be defined (≤ 20)
NPT	Number of data points [t,F(t)] for this time function (≤ 50)
t_i	Time value for the i^{th} data point
F_i	Function value for the i^{th} data point



HISTORY FUNCTIONS
INPUT DATA BLOCK (CONTINUED)

NOTES:

1. Time values must increase monotonically for each function.

2. Function values F(t) are used in computing the magnitude of element pressure loads and nodal forces. Values at times between the data points are obtained by linear interpolation.

3. To define a time function which goes to zero at a given time, and remains zero thereafter, make sure that the last *two* data points define F=0 (see the example above). Function values for times $t > t_{NPT}$ are computed by extrapolation using the last two data points.

INITIAL VELOCITY
INPUT DATA BLOCK

Defines initial conditions on nodal velocity components at selected nodes.

STATUS: Optional (ignored when restarting)

FORMAT:

INIT

V_x, V_y, V_z, NFIRST, NLAST, NINCR

<repeat as needed to define all initial conditions>

END

EXAMPLE:

```
INIT
0, 0, -5000, 1, 7, 2        ! Z velocity of -5000 at nodes 1, 3, 5, 7
0, -2500, 0, 20             ! Y velocity of -2500 at node 20
END
```

VARIABLES:

V_x	Initial velocity in the global X direction
V_y	Initial velocity in the global Y direction
V_z	Initial velocity in the global Z direction
NFIRST	First node to which initial velocities V_x, V_y, V_z apply
NLAST	Last node to which initial velocities V_x, V_y, V_z apply
NINCR	Node number increment

NOTES:

1. Initial velocities specified here are applied only to nodes which are otherwise unconstrained. That is, velocity constraints defined in the **BOUNDARY CONDITIONS** input block take precedence over the initial conditions.

LAMINATES
INPUT DATA BLOCK

Defines cross-sectional properties for plate/shell elements.

STATUS: Optional (ignored when restarting)

FORMAT:

<u>LAMI</u>
LAMNO, NLAYER, $MATL_1$, $MATL_2$, ..., $MATL_{NLAYER}$, &
Z_1, Z_2, ..., $Z_{NLAYER+1}$, ANG_1, ANG_2, ..., ANG_{NLAYER}
<repeat as needed to define all laminates>
<u>END</u>

EXAMPLE:

```
LAMI
1, 1, 3, -0.125, 0.125                 ! One layer, material #3, t=0.25
3, 1, 4, 0, 0.050                      ! One layer, midsurface offset from nodes
2, 2, 1, 3, -0.1, 0, 0.1               ! Two layers with different materials
8, 5, 1, 2, 4, 2, 3,           &       ! Five-layer laminate constructed from four
 -0.25, -0.0004, 0.,           &       !       different materials
 0.246, 0.250, 0.255
END
```

VARIABLES:

LAMNO — Laminate ID number

NLAYER — Number of layers

$MATL_i$ — Material ID for the i^{th} layer (bottom to top)

Z_i — Thickness coordinate at the lower surface of the i^{th} layer. The last value is the top of the last (uppermost) layer.

ANG_i — Material orientation angle θ for the i^{th} layer (see figure), in degrees.

NOTES:

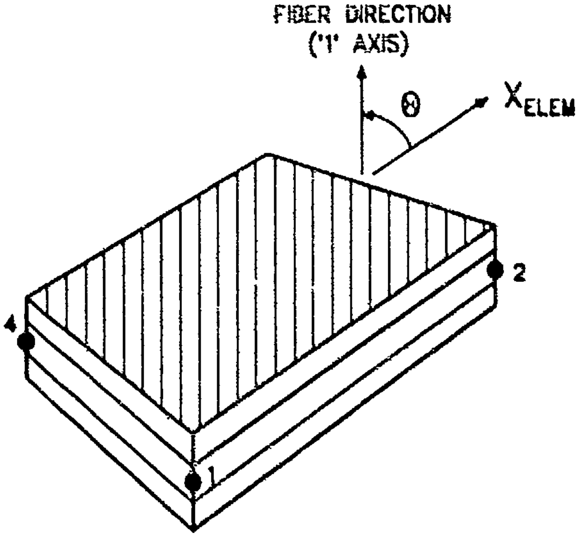
1. Laminate ID's should be numbered from 1 to the total number of laminates. Laminates may be defined in any order.

2. Material ID's refer to materials defined in the **MAT2D** input block.

3. Note that the thickness of layer 'i' is defined by the difference Z_{i+1}-Z_i.

NC⌃ES (Continued):

4. ANG$_i$ is important only if the material is orthotropic (material model '2' in the **MAT2D** input block). The figure shows the positive sense of ANG$_i$ as seen from the top of an element.



LINKED ELEMENTS
INPUT DATA BLOCK

Defines pairs of elements whose failure properties are linked

STATUS: Optional (ignored when restarting)

FORMAT:

LINK
$M_1, NS_1, M_2, NS_2, \ldots, M_n, NS_n$
<repeat as needed to define all failure links>
END

EXAMPLE:

```
LINK
1, 2                 ! Element 2 fails when element 1 fails
7, 13, 18, 100       ! Element 13 linked to 7, 100 linked to 18
2120, 3159           ! Use additional lines as needed
END
```

VARIABLES:

M_i	i^{th} master element
NS_i	i^{th} slave element

NOTES:

1. Only 3-D solid elements (8-node HEX or 4-node TET) may be linked.

2. Failure of a master element is determined by the material model for that element. When a master element fails, the corresponding slave element is marked as failed also.

3. The usual use of this option is to eliminate very soft, elastic layers (e.g., silicone or PVB interlayers) when adjacent structural layers have failed. If such elements are retained in the calculation, they normally contribute little to the internal forces, but may cause numerical problems due to excessive deformation.

4. When an element is deleted, its mass is retained at all connected nodes, but the element no longer develops internal forces.

MASS
INPUT DATA BLOCK

Defines concentrated masses at selected points of the model

STATUS: Optional (ignored when restarting)

FORMAT:

```
MASS
NODE, TRMASS, XRI, YRI, ZRI
<repeat as needed to define all point masses>
END
```

EXAMPLE:

```
MASS
1022, 0.001, 0, 0, 0          ! Point mass of 0.001 units at node 1022
END
```

VARIABLES:

NODE	Node at which the concentrated mass is attached
TRMASS	Translational mass (the same for all three directions)
XRI	Rotational inertia about an axis parallel to global X
YRI	Rotational inertia about an axis parallel to global Y
ZRI	Rotational inertia about an axis parallel to global Z

NOTES:

1. Translational mass values should have units of FT^2/L (weight divided by the appropriate gravitational constant). For instance, in pound-inch-second units, the mass of a one-pound weight is:

$$m = \frac{1\ lb.}{386\ in./\mathrm{sec}^2} = 0.00259\ lb\text{-}\mathrm{sec}^2/in.$$

2. Concentrated masses are added to the mass of all finite elements connected to a node to obtain the total mass of the node.

3. Body forces (see the **BODY** input block) are *not* applied automatically to concentrated masses. If a force per unit mass acts on a point mass, the force must be defined in terms of nodal forces (see the **FORC** and **HIST** input blocks).

MAT2D
INPUT DATA BLOCK

Defines material properties to be used in 2-D (plate/shell) elements.

STATUS: Required when 2-D elements are used (ignored when restarting)

FORMAT:

<u>MAT2</u>
MATL, ITYPE, p_1, p_2, ..., p_n
<repeat as needed to define all materials>
<u>END</u>

FORMATS FOR INDIVIDUAL MATERIAL MODELS:

<u>Material Model 1: Elastic-Plastic, Rate-Sensitive Isotropic Material</u>
MATL, *1*, RHO, E, XNU, ALPHA, YIELD, H, RSCALE, REXP, SULT

<u>Material Model 2: Orthotropic Material with Brittle Failure</u>
MATL, *2*, RHO, E1, E2, XNU12, G12, G13, G23, SF, SD

EXAMPLE:

```
MAT2
1, 1, 0.000254, 10.E6, 0.3, 8.5E-6, &          ! Aluminum 6061-T6
40000, 416000, 0.0001538, 0.25, 45000.
2, 2, 0.000373, 33.22E6, 23.35E6, 0.268, &   ! Titanium MMC
9.071E6, 9.071E6, 8.922E6, 160000., 65000.
END
```

VARIABLES:

MATL	Material property ID
ITYPE	Material model type (1 = isotropic, 2 = orthotropic)
RHO	Reference state density
E	Young's modulus in tension, E
XNU	Poisson's ratio, ν
ALPHA	Coefficient of thermal expansion, α
YIELD	Static yield stress, σ_y
H	Work-hardening modulus, $H = EE'/(E-E')$
RSCALE	Factor for strain rate sensitivity of yield stress, 1/D
REXP	Exponent for rate sensitivity of yield stress, 1/p
SULT	Ultimate stress, σ_{ult}

MAT2D
INPUT DATA BLOCK (CONTINUED)

VARIABLES (Continued):

E1	Extensional modulus in material '1' direction, E_1
E2	Extensional modulus in material '2' direction, E_2
XNU12	Inplane Poisson's ratio, ν_{12}
G12	Inplane shear modulus, G_{12}
G13	Transverse shear modulus, G_{13}
G23	Transverse shear modulus, G_{23}
SF	Failure stress for fiber direction, σ_f
SD	Failure stress for delamination, σ_d

NOTES:

1. The bulk (pressure-volume) behavior of a 2-D isotropic material is assumed to be linear, with the bulk modulus defined by

$$K = \frac{E}{3(1-2\nu)}$$

The 2-D material model assumes a zero transverse normal stress for use in plate elements (i.e., σ_{zz}=0). The bulk response and the elastic-plastic deviatoric response are determined simultaneously, in such a way that the zero normal stress constraint is satisfied.

2. H defines the work-hardening characteristics of the material, and is assumed to be constant. This implies a *bilinear* representation of the stress-strain curve. If E is the elastic modulus, and E' is the slope of the post-yield stress-strain curve, then

$$H = \frac{EE'}{E-E'}$$

3. The rate sensitivity of the yield stress is defined by RSCALE (1/D) and REXP (1/p). If d is the effective deviatoric strain rate, the instantaneous yield stress is computed from YIELD*(1 + (d*RSCALE)**REXP). That is:

$$\sigma_y = \sigma_y^0 \left[1 + \left(\frac{d}{D} \right)^{\frac{1}{p}} \right]$$

MAT2D
INPUT DATA BLOCK (CONTINUED)

NOTES (Continued):

4. When the von Mises effective stress at an integration point reaches SULT, the point in question is marked as failed and is dropped from the calculation.

5. Orthotropic properties E_1, E_2, ν_{12}, G_{12}, G_{13}, and G_{23} are referred to *material axis directions* 1, 2, 3. Normally the 1 axis corresponds to the fiber, or stiff, direction. The orientation of the material '1' axis with respect to the local coordinate axes for a plate element is part of the laminate definition data (see the **LAMI**NATE input block). The material '3' axis must correspond to the thickness direction.

6. The orthotropic 2-D material model is *linear to failure*. The material behaves elastically until the fiber-direction normal stress reaches σ_f, at which time the integration point is considered to be failed.

7. When the resultant transverse shear stress at a point reaches σ_d, failure by delamination is predicted. After delamination occurs, the point in question cannot transmit the transverse shear forces which normally accompany local bending stress. Stresses at delaminated points are computed on the basis of the membrane (reference surface) strains only. In other words, a plate/shell element in which all points are delaminated is incapable of developing bending stresses, and behaves like a membrane element.

MAT3D
INPUT DATA BLOCK

Defines material properties to be used in 3-D (solid) elements.

STATUS: Required when 3-D elements are used (ignored when restarting)

FORMAT:

```
MAT3
MATL, ITYPE, p1, p2, ..., pn
<repeat as needed to define all materials>
END
```

FORMATS FOR INDIVIDUAL MATERIAL MODELS:

Material Model 1: Elastic-Plastic, Rate-Sensitive Isotropic Material

```
MATL, 1, RHO, BULK1, BULK2, BULK3, SHEAR,  &
YIELD, RSCALE, REXP, H, SULT, Q2, Q1
```

Material Model 2: Newtonian Viscous Fluid

```
MATL, 2, RHO, BULK1, BULK2, BULK3, VISC, Q2, Q1
```

Material Model 3: Elastic-Plastic Material with Discontinuous P-V Behavior

```
MATL, 3, RHO, BULK1, BULK2, BULK3, BULKT, SHEAR,  &
YIELD, RSCALE, REXP, H, SULT, Q2, Q1
```

EXAMPLE:

```
MAT3
1, 1, 0.000254, 8.333E6, 0, 0, 3.846E6,      &     ! Aluminum 6061-T6
40000, .0001538, 0.25, 416000, 4500
2, 1, 5.E-5, 2000, 1200, 800, 500            &     ! Nonlinear elastic
3, 3, 0.0000888, 337000, 729000, 2020000,    &     ! "Bird properties" for
     1000, 30000, 3000, 0, 1, 300, 1500            ! birdstrike simulation
END
```

VARIABLES:

MATL	Material property ID
ITYPE	Material model type (1 = isotropic solid, 2 = viscous fluid)
RHO	Reference state density

MAT3D
INPUT DATA BLOCK (CONTINUED)

VARIABLES (Continued):

BULK1	Linear bulk (pressure-volume) modulus, K_1
BULK2	Quadratic bulk modulus, K_2
BULK3	Cubic bulk modulus, K_3
BULKT	Bulk modulus in tension, K_t
SHEAR	Elastic shear modulus, G
YIELD	Static yield stress, σ_y
RSCALE	Factor for strain rate sensitivity of yield stress, 1/D
REXP	Exponent for rate sensitivity of yield stress, 1/p
H	Work-hardening modulus, H = EE'/(E-E')
SULT	Ultimate stress, σ_{ult}
Q2	Quadratic artificial bulk viscosity coefficient, Q_2
Q1	Linear artificial bulk viscosity coefficient, Q_1
VISC	Viscosity coefficient 2μ for viscous fluid

NOTES:

1. The bulk (pressure-volume) behavior of a 3-D isotropic material is assumed to be linear in tension, with the rates of change of pressure and compression ratio $\eta = \frac{\rho}{\rho_0} - 1$ being related by the linear bulk modulus K_1. In compression, the remaining moduli K_2 and K_3 define the nonlinear pressure-volume behavior. In particular:

$$p = \begin{cases} K_1 \rho_0 \eta & \eta < 0 \\ K_1 \rho_0 \eta + K_2 \rho_0 \eta^2 + K_3 \rho_0 \eta^3 & \eta > 0 \end{cases}$$

For material model 3, a different bulk modulus (K_t) is used when $\eta < 0$.

2. The rate sensitivity of the yield stress is defined by RSCALE (1/D) and REXP (1/p). If d is the effective deviatoric strain rate, the instantaneous yield stress is computed from YIELD*(1 + (d*RSCALE)**REXP). That is:

$$\sigma_y = \sigma_y^0 \left[1 + \left(\frac{d}{D} \right)^{\frac{1}{p}} \right]$$

MAT3D
INPUT DATA BLOCK (CONTINUED)

NOTES (Continued):

3. H defines the work-hardening characteristics of the material, and is assumed to be constant. This implies a *bilinear* representation of the stress-strain curve. If E is the elastic modulus, and E' is the slope of the post-yield stress-strain curve, then

$$H = \frac{EE'}{E-E'}$$

4. When the von Mises effective stress in an element reaches SULT, the point in question is marked as failed and is dropped from the calculation.

5. For the viscous fluid model, the current stress is defined by:

$$\sigma_{ij} = 2\mu d_{ij} - p\delta_{ij}$$

in which 2μ = VISC, and p is computed from the equation of state (see note 1 above).

6. Q2 and Q1 define the *artificial bulk viscosity*, a numerical device used to suppress "noise" associated with shock wave propagation. Normally the default values (Q2 = 1.5, Q1 = 0.06) are adequate. The artificial pressure in a compressed element is

$$q = \rho Le(Q_2 Le - Q_1 c)$$

Here $L = V^{1/3}$ is a characteristic element length, $e = d_{kk}$ is the volumetric strain rate, and c is the local sound speed.

MPOST
INPUT DATA BLOCK

Requests an MPOST-formatted results file.

STATUS: Optional

FORMAT:

MPOST
END

NOTES:

1. Output is written to the MPOST file whenever restart output is written.
2. The output file is assigned the name 'MPOST'; if the computer system uses file extensions, the default extension is used (e.g., MPOST.DAT under VAX/VMS).

NODE COORDINATES
INPUT DATA BLOCK

Defines initial positions for all nodes in the model.

STATUS: Required (ignored when restarting)

FORMAT:

```
NODE
IPT, X, Y, Z, IGEN
<repeat as needed to define all node points>
END
```

EXAMPLE:

```
NODE
1, 5.2, 3.6, -4.2             ! Single node
9, 9.2, 3.6, -4.2, 2          ! Generate nodes 3, 5, 7, 9
END
```

VARIABLES:

IPT	Current node number
X, Y, Z	Cartesian coordinates of the node
IGEN	Node number increment for use in node generation

NOTES:

1. To generate a series of nodes equally spaced along a line, enter the first node in the series (with IGEN=0), followed by the last node with an appropriate value of IGEN. For example, the data:

```
7, 34., -20.3, 0.
13, 40., -20.0, 0., 2
```

is equivalent to:

```
7, 34., -20.3, 0.
9, 36., -20.2, 0.
11, 38., -20.1, 0.
13, 40., -20.0, 0.
```

PARAMETERS
INPUT DATA BLOCK

Defines parameters and options which control the dynamic solution and results output.

STATUS: Required

FORMAT:

```
PARA
TIME, TMAX, TREST, DTMIN, DTMAX, INCR, INCMAX, IREST, &
NINTPL, ISTAT, ITRACE
END
```

EXAMPLE:

```
PARA
0, 0.0025, 0.00125, 0.00025,, &   ! New analysis; solution up to 2.5 ms but
0, 1000, 1, 5, 0, 40              ! not exceeding 1000 time steps;
                                  ! restart/results output every 1.25 ms;
                                  ! trace output every 40 time steps;
                                  ! 5 int. points/layer in shells.
END
```

VARIABLES:

TIME	Time at start of solution (ignored when restarting)
TMAX	Final time value for which a solution is requested
TREST	Time interval for restart file output
DTMIN	Lower limit on solution time step
DTMAX	Upper limit on solution time step
INCR	Time step number at start of solution (ignored when restarting)
INCMAX	Maximum allowable time step number
IREST	Switch for restart/results file output (0 = no output, 1 = output)
NINTPL	Number of integration points per layer for plate elements
ISTAT	0 = dynamic solution, 1 = static solution by dynamic relaxation
ITRACE	Number of time steps between output dumps to TRACE file

NOTES:

1. TMAX and INCMAX provide *termination conditions* for the analysis. The solution stops whenever the time value reaches TMAX or the time step number reaches INCMAX, whichever comes first. If either TMAX or INCMAX is omitted (or set to zero), the other parameter provides the termination condition. If both TMAX and INCMAX are zero or missing. no analysis will be performed.

2. TREST is a *time interval* (real valued) between restart output dumps. When the solution time value approaches a multiple of TREST, the time step is modified so that output can be saved at the requested time.

3. DTMIN is useful for ensuring that the solution does not become "bogged down" with an extremely small time step when severe mesh distortions occur. If the critical time step falls below DTMIN, restart output is written and the solution stops.

4. DTMAX can be used to force a small time step in situations where stability is not a limiting factor. An example is an impact problem in which the initial spacing and relative velocity of the impacting bodies is such that extreme penetration might occur on the first time step, when contact has not yet been established.

5. NINTPL *must be odd,* and is limited to the range 3-11.

6. Normally ISTAT = 0 (transient dynamic analysis). When ISTAT = 1, the program attempts to obtain a static solution for the specified external forces, by computing the damped dynamic response of the finite element model. This process is called *dynamic relaxation.* If this option is selected, it usually is appropriate to:

   - set TMAX very large (e.g., TMAX = 10^{20});
   - make INCMAX rather large, to allow sufficient time steps for convergence;
   - request restart/results output at the end of the analysis (IREST = 1, TREST = TMAX), since intermediate states are rarely of interest; and
   - supply the tolerances FRATIO and VRATIO in the **TOLERANCES** input block.

PRESSURE
INPUT DATA BLOCK

Defines time-dependent pressure loads on element faces

STATUS: Optional

FORMAT:

```
PRES
IBEG, IEND, IFACE, IFUNC, SCALE
<repeat as needed to define all pressure loads>
END
```

EXAMPLE:

```
PRES
1, 10, 4, 2, -20.75          ! Load face 4 of HEX elements 1-10
70, 90, 6, 1, -1.0           ! Load plate elements 70-90
END
```

VARIABLES:

IBEG First element in a series of elements to be loaded
IEND Last element in the series
IFACE Face (surface) number on which load is applied
IFUNC Time history function defining pressure versus time
SCALE Scale factor to be applied to time history function

NOTES:

1. All elements with ID's between IBEG and IEND are loaded. If IEND is omitted, a pressure load is applied only to element IBEG.

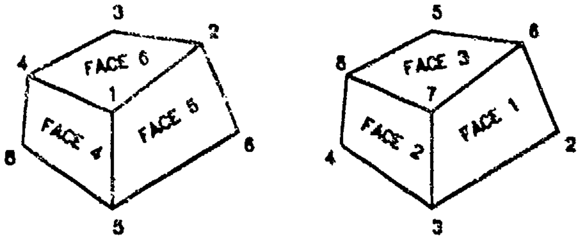
2. Valid face numbers are between 1 and 6 for solid elements. *For plate elements, set IFACE = 6.* In terms of the element connectivity pattern (see **ELEMENTS** input block), the six surfaces are defined as follows:

Face Number	Nodes Defining Face
1	2 6 7 3
2	3 7 8 4
3	5 8 7 6
4	1 4 8 5
5	1 5 6 2
6	1 2 3 4

NOTES (Continued):

3. Positive values of the pressure define a force acting *outward* from the element face.

4. At any particular time in the solution, the actual pressure to be applied to an element is determined by interpolating time history function IFUNC linearly, and then multiplying by SCALE.



REFERENCE
INPUT DATA BLOCK

Defines geometric limits for the mesh for use in plotting

STATUS: Optional (ignored when restarting)

FORMAT:

```
REFE
XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX
END
```

EXAMPLE:

```
REFE
0, 10, 0, 20, -1, 1
END
```

VARIABLES:

XMIN, XMAX	Plotting limits on the X axis
YMIN, YMAX	Plotting limits on the Y axis
ZMIN, ZMAX	Plotting limits on the Z axis

NOTES:

1. Results output from X3D is not affected by this option. It is provided for use in situations where a program to be used for output needs "help" in finding appropriate geometric limits for plotting.

RESTART
INPUT DATA BLOCK

Requests a restart of a previous analysis, and initiates restart file input.

STATUS: Optional (required when restarting)

FORMAT:

```
REST
INCOLD
END
```

EXAMPLE:

```
REST
3500          ! Read restart dump from time step 3500 and continue
END
```

VARIABLES:

INCOLD   Time step number from which the analysis is to be resumed

NOTES:

1. This block is required whenever an analysis is to be resumed from a point of interruption.

2. The old restart file is assumed to reside on file "OREST". Default values are used for file extensions on all systems (e.g., OREST.DAT under VAX/VMS).

3. If no restart checkpoint for time step INCOLD exists on the restart file, a fatal error occurs and execution stops.

4. Input data in many of the input blocks may be changed when restarting. Only data which are being replaced or modified need to be present in the input file for the restart job. Note that the **PARAMETERS** input block must be changed when restarting, to define new termination conditions (e.g., a later TMAX or higher INCMAX).

RIGID WALL
INPUT DATA BLOCK

Defines arbitrary planar surfaces through which nodes may not move.

STATUS: Optional (ignored when restarting)

FORMAT:

```
RIGI
A, B, C, D
<repeat as needed to define all rigid walls>
END
```

EXAMPLE:

```
RIGI
1, 0, 0, -5           ! Barrier at X = -5 (nodes to right)
-1, 0, 0, 5           ! Barrier ar X = -5 (nodes to left)
0.5, 0.8660254, &     ! 30° incline in (X,Y) plane, passing through
   0, 2               !   the points (0, 2.3095) and (4, 0)
END
```

VARIABLES:

A, B, C	X, Y, and Z coefficients in equation of constraint plane
D	Constant on right-hand side of constraint equation

NOTES:

1. Each rigid wall constraint defines a surface in space through which nodes of the model may not pass. Each constraint has the general form:

$$AX + BY + CZ \geq D$$

where X, Y, and Z refer to the current position of *any node* in the finite element model. For example, to define a barrier at X = -2, with all nodes confined to the side toward the origin, we impose the constraint X ≥ -2. Therefore, let A = 1, B = 0, C = 0, and D = -2.

SOFT BODY
INPUT DATA BLOCK

Defines properties and initial conditions for soft-body impact modeling.

CAUTION

This input block defines data for an "unstructured mesh" technique for soft-body impact problems. The method is experimental, and is not recommended for production analyses. Presently the most reliable approach for this class of problems is to use tetrahedral elements for the projectile, and define CONTact surfaces through which the projectile and target may interact.

STATUS: Optional (ignored when restarting)

FORMAT:

```
SOFT
NODLIM, CHDIST, VOLNOD, RHOZRO, BK1, BK2, BK3, BKT, &
VX0, VY0, VZ0
END
```

EXAMPLE:

```
SOFT
2000            &     ! Soft body mesh starts with node 2001
0.25            &     ! Characteristic mesh dimension is 0.25 units
0.025           &     ! Each mass element has initial volume 0.025
0.888E-4        &     ! Specific gravity 0.95 (0.03431 lb/in3)
337322, 729253  &     ! P-V properties for water (3 terms) with
2020809, 0      &     !   no tensile stiffness
0, 0, -8000           ! Initial velocity -8000 along Z axis
END
```

VARIABLES:

NODLIM	Number of last node point in structure mesh
CHDIST	Characteristic dimension (interaction zone) for nodal forces
VOLNOD	Initial volume of each mass element in soft body mesh
RHOZRO	Initial density of the soft body mesh
BK1	Linear bulk modulus in compression
BK2	Quadratic bulk modulus in compression
BK3	Cubic bulk modulus in compression
BKT	Linear bulk modulus in tension
V_i0	Initial velocity components for all nodes in soft body mesh

NOTES:

1. The soft body mesh consists of nodes NODLIM+1 through the highest-numbered node in the model. All nodes in this range are controlled by a self-contained internal force model, and should not be connected to other finite elements.

2. CHDIST defines the maximum separation of neighboring mass elements which may interact through internal forces. Normally CHDIST should be set to about twice the mesh spacing in the most coarsely divided direction.

3. VOLNOD and RHOZRO define the size and density of each mass element in the soft body mesh. Note that if the highest-numbered node in the model is N_{max}, the the total mass of the soft body mesh is:

   $$M = [\ N_{max} - \text{NODLIM}\ ] \times \text{VOLNOD} \times \text{RHOZRO}$$

   Nonuniform masses may be added to nodes in the soft body mesh using the **MASS** input block.

4. The equation of state used for the soft-body mesh is identical to that used for 3-D solids (see the **MAT3D** block), except that the tensile bulk modulus (BKT) may be different from the compressive value.

5. VX0, VY0, and VZ0 define the initial components of velocity for all nodes in the soft body mesh. Nonuniform initial velocity values may be input using the **INITIAL** input block. All initial velocity values are subject to the boundary conditions prescribed in the **BOUN** input block.

TANGENTIAL ROTATION
INPUT DATA BLOCK

Defines constraints on rotational DOFs along a shell edge.

STATUS: Optional (replaced if present when restarting)

FORMAT:

TANG
N_1, N_2, N_3, ..., N_{MAX}
<repeat as needed to define all edge lines>
END

EXAMPLE:

```
TANG
12, 32, 65, 84, 0    &      ! A list of four nodes (0 marks the end),
92, 105, 128                ! followed by a list of three nodes
END
```

VARIABLES:

N_i List of nodes whose tangential rotation is to be constrained

NOTES:

1. The usual use of this data block is to define symmetry boundary conditions on rotational degrees of freedom. A shell with a symmetry edge in the (x,y) plane, for instance, should have constraints applied to the Z translation DOF, and also to the rotational DOF *about a line tangent to the edge.* The particular DOF involved generally is some combination of θ_x and θ_y, and changes as the shell deforms.

2. The list of nodes N_1, N_2, ... defines a series of nodes along a shell edge, each of which is subjected to a tangential rotation constraint. The tangent direction to the edge at node N_k in the list is determined by passing a curve through the current positions of nodes N_{k-1}, N_k, and N_{k+1}, and finding the tangent to this curve at node N_k. For the first and last nodes in a list, only two nodes are used to find the tangent direction. The direction about which the rotation is constrained is updated at each time step of the solution.

3. A zero value within this data block ends one list of nodes and begins a new list (i.e., a new shell edge). Be sure to use only *one* zero to separate lists of nodes; two successive zeros will cause subsequent data to be ignored.

TITLE
INPUT DATA BLOCK

Defines a descriptive problem title for printer and file output.

STATUS: Required (optional when restarting)

FORMAT:

```
TITL
ITITLE
END
```

EXAMPLE:

```
TITL
Spherical shell with initial velocity at equator
END
```

VARIABLES:

ITITLE	Alphanumeric problem title, 80 characters or less in length

TOLERANCES
INPUT DATA BLOCK

Defines parameters and tolerances which control the time integration procedure.

STATUS: Optional

FORMAT:

```
TOLE
DTFRAC, HGDAMP, HGCONS(1), HGCONS(2), HGCONS(3), FRATIO, &
VRATIO, CONTOL
END
```

EXAMPLE:

```
TOLE
0.95, 0.10, 0.02, 0.02, 0.02, 0.01, 0.01, 0.2
END
```

VARIABLES:

DTFRAC Fraction of the critical time step to be used for integration

HGDAMP Anti-hourglass damping fraction for eight node solid elements

HGCONS(1) Anti-hourglass stiffness fraction for membrane modes in plate finite elements

HGCONS(2) Anti-hourglass stiffness fraction for transverse deflection modes in plate finite elements

HGCONS(3) Anti-hourglass stiffness fraction for rotational modes in plate finite elements

FRATIO Force convergence tolerance for static solution by dynamic relaxation

VRATIO Velocity convergence tolerance for static solution by dynamic relaxation.

CONTOL Relative motion tolerance for contact solution.

NOTES:

1. DTFRAC is used at each solution time step to modify the allowable time step computed by the program, and must be between 0 and 1. The default value is 0.75.

NOTES (Continued):

2. HGDAMP controls the magnitude of the restoring forces used to suppress the unstable deformation modes which are possible in 8-node solid elements with one-point integration. Too small a value may permit instabilities to grow and propagate throughout the model, while too large a value may affect accuracy. HGDAMP values between 0.1 and 0.3 are typical. Note that large values of HGDAMP reduce the allowable time step: for HGDAMP = 0.1, the critical time step is reduced by about 11%; when HGDAMP = 0.3, the time step is 35% smaller than the undamped critical value.

3. HGCONS(i) are anti-hourglass coefficients for plate elements. The plate element uses a stiffness method of hourglass control (rather than damping). The three values define independent anti-hourglass controls for membrane, transverse displacement, and rotational hourglassing modes, respectively. While the values of HGCONS(i) do not affect the allowable time step, larger values may reduce the accuracy of the solution. HGCONS(i) values in the range 0.01-0.05 are recommended.

4. FRATIO and VRATIO control the criteria for convergence of a static solution by dynamic relaxation. The unbalanced forces at all nodes are required to be less than FRATIO times the maximum applied nodal force. The velocity components at all nodes are required to be less than VRATIO times the largest velocity component occurring in the solution.

5. CONTOL controls the amount of motion which may occur during a time step due to the enforcement of contact constraints. The absolute displacement of any node, divided by the length of the shortest element edge to which the node is connected, will be made less than CONTOL. The magnitude of the nodal motions is controlled by reducing the solution time step. Specifying a DTMIN value (see **PARAMETERS** input block) is highly recommended when CONTOL is small, to prevent the time step from approaching zero.

TRACE NODES
INPUT DATA BLOCK

Defines a list of nodes for which output will be written to the TRACE output file.

STATUS: Optional

FORMAT:

TRAC
$N_1, N_2, N_3, ..., N_n$
END

EXAMPLE:

```
TRAC
1, 12, 13, 16, 17, 22, 35, 7, 9 &
302, 485, 2217, 666
END
```

VARIABLES:

N_i — i^{th} node for which intermediate results are to be recorded during the solution

NOTES:

1. Results recorded for each trace node consist of the initial conditions, the current position, and the current velocity.
2. Nodes are listed on the TRACE file in the order defined in this input block.
3. TRACE file results are recorded after a prescribed number of time steps (see parameter ITRACE in the **PARAMETERS** input block).

@DEBUG
INPUT DATA BLOCK

Requests diagnostic output or other undocumented operations.

STATUS: Optional

FORMAT:

```
@DEB
 KEYWORD = value
 <repeat as needed to select all options>
END
```

EXAMPLE:

```
@DEB
 SBPRES = 10
END
```

VARIABLES:

KEYWORD	Name of a diagnostic output option or internal function
value	Value to be assigned (typically an output frequency)

NOTES:

1. This input block normally is used for debugging and quality control purposes. Refer to the X3D source code for further information.

APPENDIX B
PRE- AND POST-PROCESSING

INTRODUCTION

This Appendix describes several utility programs which support pre- and post-processing of X3D finite element models. They are:

- PATX3D: convert PATRAN finite element model to X3D format
- X3POST: produce tabular output listings and PATRAN results files

PATX3D: PATRAN INPUT DATA INTERFACE

PATX3D is a data conversion utility which translates finite element model data from PATRAN to X3D. Some of the more specialized data types which exist in each of the programs are not supported by PATX3D; however, the basic geometric data, which represent the bulk of most models, are converted automatically. The remainder of this section describes the conventions used by the PATX3D interface, and defines modeling guidelines and restrictions which will minimize problems with data conversion.

Overview

PATX3D accepts as input a PATRAN formatted Neutral File, and writes an input data file for X3D. The most common data types which are common to PATRAN and X3D (node coordinates, elements, and constraints) are read from the Neutral File and written to the X3D data file. The X3D data must be edited to add materials data, loads, and control information, which typically represent a very small fraction of the final input.

Basic documentation of the PATRAN Neutral File is contained in Chapter 29 of the PATRAN User's Manual. The X3D input formats are documented in this report (see Appendix A).

Data Types

The relationship between data types in X3D and PATRAN is summarized in the Table on the next page. Data conversion is limited to node coordinates, elements, constraints, and contact surface data. Since the elements in X3D are low-order (bilinear or trilinear), they correspond to the simplest element configurations available in PATRAN. Note that the normal method of creating contact surface data for X3D is with PATRAN TRI elements, while QUAD (or HEX) shapes are preferred for the "real" finite elements.

RELATIONSHIP BETWEEN X3D AND PATRAN DATA TYPES

X3D DATA BLOCK	PATRAN DATA TYPE	CREATED BY
NODE coordinates	NODE	GFEG
ELEMents	HEX, TET, WEDGE, QUAD	CFEG, PFEG
BOUNdary conditions	NODE	GFEG, DFEG
CONTact surfaces	TRI/3/n1, TRI/3/n2	CFEG

Finite Elements

The standard X3D finite elements are a 4-node plate, corresponding to PATRAN QUAD elements, and 4 or 8 node solids, corresponding to the PATRAN TET and HEX elements respectively. The steps required to create nodes and elements in PATRAN are:

1. use GFEG to create a regular array of nodes on one or more patches or hyperpatches;
2. generate finite elements on the same region(s) with CFEG; and
3. assign property ID's to the elements using PFEG (if not done in the CFEG step).

For both QUAD (plate) and HEX or TET (solid) elements, the element configuration is defined in PATRAN using the notation "QUAD/n/e" or "HEX/n/e", where 'n' is the number of nodes per element and 'e' is a configuration code. X3D uses the simplest form of each element, with the configuration code omitted. For example:

CFEG, P100, QUAD

creates 4-node plate elements compatible with X3D on patch number 100, and

CFEG, HP79, HEX

creates 8-node solids compatible with X3D on patch 79.

The use of WEDGE and TRI elements is not recommended, even though these elements will be converted properly by the translator. These shapes are treated as degenerate cases of the standard X3D elements and are inferior to QUAD and HEX elements. When a model is constructed from degenerate patches, a few WEDGE or TRI elements may be created when the region is paved with QUAD or HEX elements. It is for this reason (only) that PATX3D supports the degenerate element forms. We recommend that their use be limited to situations in which PATRAN is forced to create them. The TET element normally is less effective than the HEX element, but may be useful for some situations in which extremely large mesh distortions occur.

Property Data

Even though PATX3D does not translate material properties data from PATRAN, it is a good idea to assign material ID's to elements as they are created in PATRAN. This can be accomplished with either the CFEG or PFEG commands in PATRAN. Suppose, for instance, we wish to generate solid elements for X3D on hyperpatch number 22, and assign the material ID 3 to each of these elements. We can use either:

```
CFEG, HP22, HEX/8
PFEG, HP22, HEX/8,, 3
```

or

```
CFEG, HP22, HEX/8,, 3
```

The X3D data created by PATX3D will contain the material code (3) for all of the solid elements generated, although the properties for material 3 must be entered "by hand" in the X3D data file.

The use of negative numbers to distinguish between material ID's and property ID's, as used in PATRAN, is not necessary with PATX3D. In X3D, only a single property ID is needed for each element, corresponding to the material number for solids and the laminate number for plates. PATX3D uses the absolute value of the property ID specified in the PATRAN PFEG (or CFEG) command.

Boundary Conditions

Nodal constraints may be created during the GFEG (node generation) operation, or later using the PATRAN DFEG command. Both types of nodal constraints will be translated properly by PATX3D, but the enforced value of the displacement must be zero. A single enforced displacement value, as supported in PATRAN, is not appropriate in a transient dynamic solution. Note that all constraints listed in the neutral file will be applied in the basic coordinate system (X,Y,Z) in X3D, since PATX3D does not translate coordinate system definitions.

Contact Surface Data

PATRAN may be used to create contact surface data for X3D, provided certain conventions are followed. All surfaces to be analyzed for contact in X3D must be classified as "master" or "slave" surfaces, which then are arranged in pairs, or "sets". During the dynamic solution, all points on the slave surface for set '1' are tested for contact with all master elements in set '1'; slave points in set '2' are compared with all master elements in set '2'; and so on. Usually the master surface is the more rigid (or more coarsely meshed) of the two potential contact surfaces.

In PATRAN, we create contact surface sets as if they were finite elements, using a different element type and configuration code. All contact elements created in PATRAN for X3D must be TRI elements (3-node triangles). If a contact surface actually consists of plate finite elements, it must be CFEG'ed twice: once to create the plate elements (QUADs), and once again to create the contact elements (TRIs). For solid regions, one or more of the bounding surfaces must be CFEG'ed with TRI elements to define the contact surface. For solids in which failure may occur over a significant region of the body, it may be appropriate to mesh the entire solid with TRI contact elements.

The contact surface set is defined by the element configuration code (e.g., TRI/3/e) in PATRAN. Master surfaces must be assigned configuration codes ending in '1': 1, 11, 21, 31, and so on. The corresponding slave surfaces are numbered 2, 12, 22, etc. If, for example, there is a single pair of potential contact surfaces, use:

```
CFEG, Pmmm, TRI/3/1
```

to create contact elements on the master surface, and:

```
CFEG, Pnnn, TRI/3/2
```

for the slave surface. PATX3D will translate up to 10 pairs of contact surfaces.

One further restriction exists in creating contact data, since the elements so created are not treated as finite elements in X3D. All "real" finite elements (QUADs, HEXes, and TETs) must be numbered before the contact surface elements. The simplest ways of ensuring that this is the case are to generate the contact elements (TRI's) last; or specify a high starting value when CFEGing the contact surfaces (so the contact element numbers are the largest), and then compress the element numbers before writing the neutral file.

While this limitation is likely to disappear in later versions of X3D, it may cause problems with the present release of the code.

X3POST: POSTPROCESSING OUTPUT FROM X3D RESTART FILE

X3POST is an interactive program which reads the X3D standard restart file, and produces detailed output at specific points in time (as opposed to time histories). The program is menu-driven, and offers the following output options:

Tabular Output Listings:

- Solid element state variables
- Plate element integration point stresses

PATRAN Results File Output:

- Displacement results file
- Nodal results file
- Element results file

PATRAN Session/PCL File Output:

- Single-increment failed-element list file
- Multi-increment displacement PCL file

X3POST works with data for one time step at a time, providing "snapshots" of deformed geometry and element state variables as output.

X3POST reads the X3D restart file, OREST, immediately on startup, and loads model data for the first time step found on the file. The file OREST (or OREST.DAT on VAX/VMS systems) must be present in the current ID or file directory, or linked to file OREST through logical name assignments. Data for following time steps may be loaded by requesting the next available time step, or by requesting a specific time step number.

The tabular output options in X3POST allow for listing of either solid element or plate element results at integration points. For solid elements, the results listed consist of centroid values of:

- mean stress (hydrostatic pressure)
- von Mises effective stress
- average density
- effective plastic strain

For plate elements, results are listed at each integration point through the element thickness; quantities output are:

- inplane stresses σ_{xx}, σ_{yy}, σ_{xy}
- von Mises effective stress
- effective plastic strain
- flag for integration point failure

In either case, a range of elements can be specified for which output will be listed.

X3POST produces PATRAN results files, session files, and PCL files which can be used to produce plots of deformed geometry, nodal velocities, and element stresses and state variables. The format of the results files is as described in Chapter 27 of the PATRAN User's Manual, *Results Evaluation*. All PATRAN-related output files are formatted files, so they can be created on one machine and later transferred to another machine of a different architecture and still be useful.

X3POST-related files are summarized in the following Table. File contents and options are described more fully in the subsequent section.

FILE NAME	FORM	DESCRIPTION
(logical unit 5)	Formatted	Terminal input.
(logical unit 6)	Formatted	Terminal output.
OREST	Binary	Input restart/postprocessing data. This was created by X3D with the file name NREST, and can also be used as a restart input data file by X3D.
Dxxx	Formatted	Output PATRAN displacements results file for time step xxx.
Nxxx	Formatted	Output PATRAN nodal results file for time step xxx.
Exxx	Formatted	Output PATRAN element results file for time step xxx.
Fxxx	Formatted	Output failed-element list PATRAN session file for time step xxx.
DISPPCL	Formatted	Output displacements processing PATRAN PCL file for all increments found in file OREST.

All files use the system default naming conventions. For instance, the DISPPCL file would be named DISPPCL on a UNIX system; on a VAX/VMS system, where the default file type for FORTRAN-generated files is .DAT, the same file would be called DISPPCL.DAT. File names which are generated using time step numbers do not include leading zeros; for example, the PATRAN displacements results files for time steps 72 and 258 would be called D72 and D258, respectively, on a UNIX system.

The *Displacement Results File* contains the model displacements at a particular time step by nodes. This file is needed for plotting deformed model geometry. The data are stored in three columns, with x, y, and z displacements in columns 1, 2, and 3 respectively. Nodal rotational displacements are not calculated within X3D and are not recorded here.

The *Nodal Results File* contains additional nodal variables for plotting in PATRAN. At present these nodal variables consist of the three velocity components at the node, followed by the magnitude of the velocity vector. The data are stored in four columns, with x, y, and z velocity components stored in columns 1, 2, and 3 respectively; column 4 contains the magnitude of the nodal velocity vector.

The *Element Results File* contains results for a single time step, by elements. Results are arranged in columns, which are listed in the following Table:

COLUMN DEFINITIONS FOR PATRAN ELEMENT RESULTS OUTPUT

Column	3-D Solid Element Result	Plate Element Result
1	Stress σ_{xx}	Force resultant N_x
2	Stress σ_{yy}	Force resultant N_y
3	Stress σ_{zz}	Force resultant N_{xy}
4	Stress σ_{yz}	Moment resultant M_x
5	Stress σ_{xz}	Moment resultant M_y
6	Stress σ_{xy}	Moment resultant M_{xy}
7	Density ρ	Max. principal stress
8	Pressure p	Min. principal stress
9	von Mises stress σ_{eq}	Max. von Mises stress
10	Effective plastic strain	Max. eff. plastic strain
11	Failure index (1 if element is failed, otherwise 0)	Failure index (fraction of integration points failed)
12	Internal energy	Internal energy

The stress components listed in the Table for solids refer to the global coordinate directions, and occur at the element center. The solid element results may be interpreted as mean values for the element.

Element results for plates include the effect of several integration points through the element thickness. The force and moment resultants are integrals through the plate thickness:

$$N_{ij} = \int_{z_l}^{z_u} \sigma_{ij}\, dz \qquad M_{ij} = \int_{z_l}^{z_u} z\sigma_{ij}\, dz$$

The results columns labeled "max" or "min" are the largest or smallest values of all the integration points in a particular element. The failure index computed for plates is a fraction between 0 and 1, since the plate element may be failed through part of its thickness only.

X3POST also can produce PATRAN session and PCL files. These can help automate the processing of deformed geometry plots. Because large deformations are possible during analysis using X3D, including failed elements in a deformed geometry model plot can obscure the behavior of the model. The X3POST session files help a PATRAN user eliminate failed elements from the PATRAN active set. This can provide a clearer visualization of the model's response. As an example, Figure B.1 shows a transparency impact hidden-surface plot with and without failed elements included in the display. When the failed elements have been removed, the example clearly shows transparency failure; in the other view, transparency failure is not apparent.

The single-increment *Failed-Element List Session File* contains a list of all failed elements in the model during a particular time step. The file contains PATRAN directives that store the list of failed elements as a *named component,* specific for that time step, in the user's model database. The named component can later be used to erase those elements from the PATRAN active set in order to eliminate those elements from the deformed geometry plot. The user can process the file directives by invoking the unprompted PATRAN READ command, then entering the failed-element list file name for a particular increment. Howvere, most users prefer instead to invoke the multi-increment displacement processing PCL file, described below, which automatically reads in the directives stored in the failed-element list files.

The multi-increment *Displacement Processing PCL File* contains PATRAN Command Language directives which automate the processing of a large number of displacement results files for a model. The file is generated by X3POST only when the user requests the option to create PATRAN results files for *all* increments. In that case, X3POST generates a single displacement processing PCL file, and displacement results files, element results files, and failed-element list session files for all time steps found on the

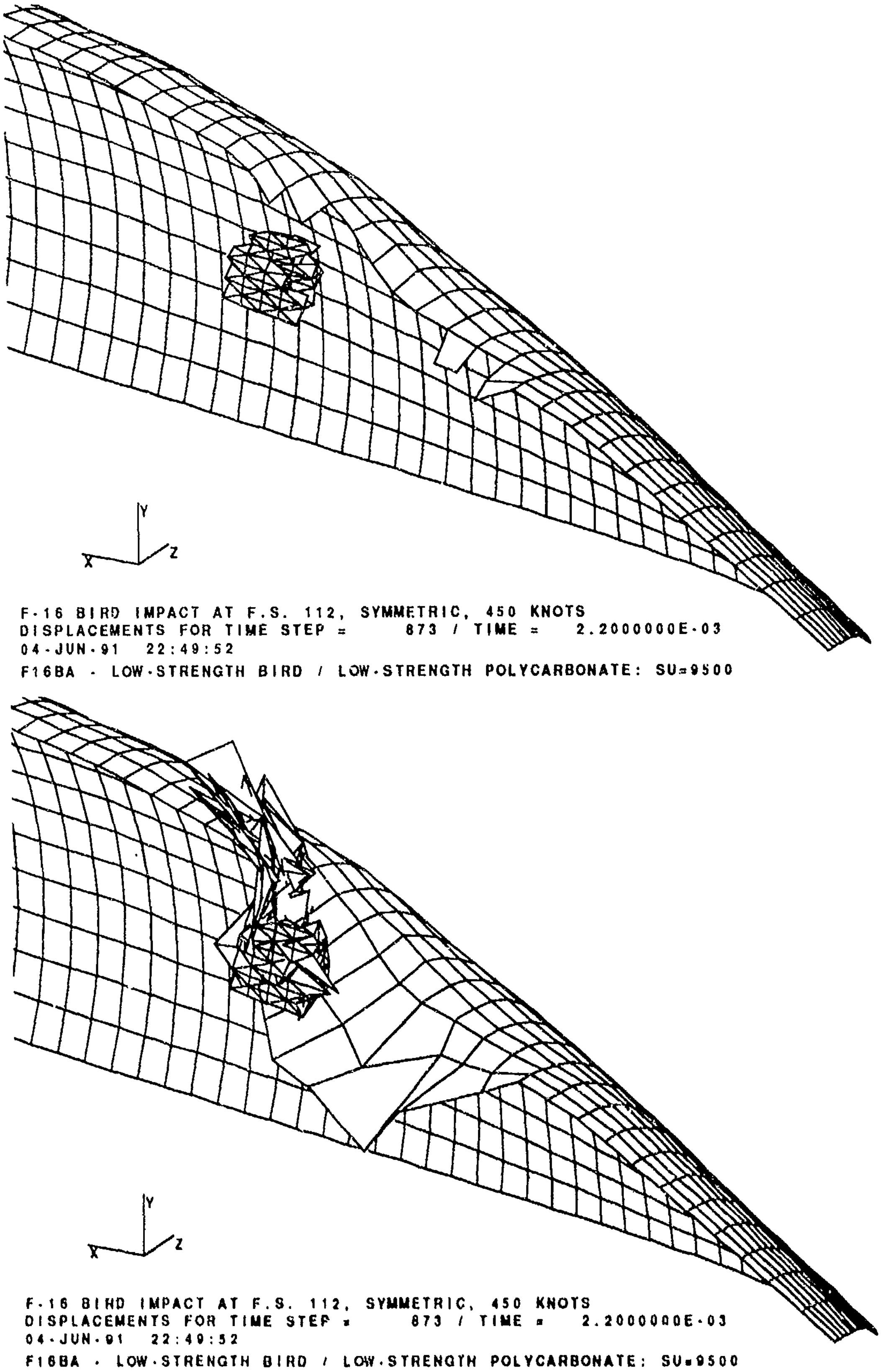

Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 450 KNOTS
DISPLACEMENTS FOR TIME STEP = 873 / TIME = 2.2000000E-03
04-JUN-91 22:49:52
F16BA - LOW-STRENGTH BIRD / LOW-STRENGTH POLYCARBONATE: SU=9500
Y
X
Z
F-16 BIRD IMPACT AT F.S. 112, SYMMETRIC, 450 KNOTS
DISPLACEMENTS FOR TIME STEP = 873 / TIME = 2.2000000E-03
04-JUN-91 22:49:52
F16BA - LOW-STRENGTH BIRD / LOW-STRENGTH POLYCARBONATE: SU=9500

Figure B.1. Transparency Impact Deformed-Shape Plot With and Without Failed Elements Removed from the Display.

X3D restart file. Using the displacement processing PCL file requires version 2.4 or later of PATRAN. The PCL file's routines have been tested with versions 2.4 and 2.5 of PATRAN.

The remainder of this section discusses using PATRAN and the displacement processing PCL file DISPPCL to display multiple displacement results.

A displacement processing PCL file generated by X3POST is a standalone PCL program, specific to the analysis from which it was generated. When used within PATRAN, the file requires presence of the PATRAN displacement results files, failed-element list session files, and optionally, element results files which X3POST generated at the same time as the PCL file. Files within the group should not be substituted for files generated by another analysis. If a user should do so inadvertently, the PCL file may process displacements incorrectly with no warnings to the user.

Below are some steps a user might follow in processing displacements with PATRAN. This example describes an X3D analysis performed on a CRAY under UNICOS. It shows procedures used in displaying PATRAN results from this analysis on a Sun which is connected to the CRAY via a network link.

1. **Run X3POST on the CRAY.**
   Generate results for all increments found in the X3D restart file. Use the "All Increments" option from the X3POST main menu: this also generates the PCL file *DISPPCL.*

2. **Transfer appropriate analysis results files to the Sun.**
   Put these in their own directory to avoid confusion with results files from other analyses. Required files are:

   - All *Dxxx* files (displacements results files).
   - All *Fxxx* files (failed-element list session files).
   - All *Exxx* files (element results files); these are required only if you want to plot element results.
   - The displacement processing PCL file *DISPPCL.*

   File names should not be changed when the files are copied: the PCL program uses file names which are explicitly stored within its code for reference to the other files.

3. **Use a copy of the PATRAN database file for this model.**
   The PCL program makes many permanent changes to the PATRAN database file. No model parts are created or deleted, but the active set is modified substantially. It's probably most convenient for the user to work with a copy of the model database, then discard the database copy when the plotting session is completed.

4. **Run PATRAN from the directory where the results files are located.**
Get the PATRAN file ready to display displacement results:

   - Define the NORMAL named component if it doesn't yet exist.
   - Set the displacements scale factor to 1.0 if desired:

```
SET,SINTERP,ON
SET,SFACT,1.0
```

   - Set ZNULL off if desired:

```
SET,ZNULL,OFF
```

   - Define an appropriate view angle if your terminal does not let you change the viewing angle locally.

5. **Read in the displacements processing PCL file.**
Use the standard PATRAN PCL syntax:

```
!!INPUT DISPPCL
```

You should need to read in the PCL file only once per PATRAN session: once the PCL functions have been stored, they can be invoked repeatedly.

6. **Run the displacements processing PCL function EXEC_HIDE.**
Use the standard PATRAN PCL syntax:

```
!EXEC_HIDE()
```

This brings up the main menu for :he PCL program. Note that the program temporarily turns off the on-screen menu mode ("SET,MENU,OFF") if you have that turned on. The menu mode is restored, if necessary, when the program is completed.

Note that when responding to EXEC_HIDE's prompts, you can usually enter either the number of a menu choice or its first letter. If using the first letter would be considered an ambiguous response for a particular menu, then only the number is valid.

7. **Modify file name convention if needed.**
When you invoke it, EXEC_HIDE internally sets a default file name ending according to an initial file search. The search patterns, in order of preference, are ".DAT", ".", and " " (blank). You should rarely need to change the default. If the file ending is determined incorrectly by the program, then EXEC_HIDE has an option that lets you manually define a string to be added to the end of the file name it uses. For example, if your PATRAN displacement file names on a VAX

looked like "D123.;1", then the PCL program's trailing string might have to be changed to a period (".").

In particular, you probably need this option if you see the PATRAN error message:

```
*RESULTS FILE HAS A WIDTH THAT IS NOT...
```

Invoke the option by selecting menu item number 5 ("File names") and enter the required string; don't include double quotes.

8. **Process the failed-element list session files.**
   Select menu item number 4 ("Process failed-element lists"), then item number 1 ("Read in lists"). The process may take some time to complete. This step needs to be run only once for a PATRAN session.

9. **Plot the displacements.**
   From the main menu, select menu item number 1 ("Do plots"). This brings up a menu with several options, with some options that are not displayed on the menu bar:

   - To display the time value and other information associated with the current time step, press the space bar and enter <Return>.
   - You can step through the stored increment list without plotting, if you like. Enter "2" or "+" and <Return> to change to the next stored time step; enter "-" and <Return> to move a step backwards.
   - The menu has two lines of choices; to save display space only one line is shown at a time. To view the second line of choices, select item number 3 ("More commands"). Note that for the "Do plots" menu, you can select an item even if it is not displayed on the current menu line.
   - If you want to specify an increment, select menu item number 5 ("Specify incrt"). The PCL program will jump to the increment number which you specify; or, if the requested increment does not exist in the stored increment list, the program will select the first increment on the list which is greater than the value specified.
   - You can invoke a single-line PATRAN command by selecting item number 6 ("Enter PATRAN cmd") if needed. This is meant to be used only for quick SET/SHOW commands.
   - To plot the current time step dipslacements, select menu item number 1 ("Plot"). When the plot is complete, the PCL program will pause to allow for a hardcopy or screen capture.

10. **Enter "End" when done.**
You can quit running the PCL procedure at any time. If you want to rerun the PCL program during the same PATRAN session, you can skip the !! INPUT DISPPCL step; enter ! EXEC_HIDE() to restart the process. Again, if the failed-element lists have been already read during this PATRAN session, they do not need to be read in again (although it won't hurt anything to do so).

11. **Options to set PATRAN WINDOW or VIEW parameters.**
By default, the PCL program uses the current PATRAN view when displaying deformed-geometry results. Users may wish to display several views of a model's behavior at a particular time step, and users with older-technology graphics terminals have to change views by using PATRAN commands.

The PCL program can read and store PATRAN commands that request several views of a model when the PCL program is processing a time step; this permits a user to see the model from several angles without having to run the PCL program several times. The commands are stored internally by sets, and the PCL program processes each set of commands before issuing a "RUN,HIDE". The PCL program pauses after each "RUN,HIDE" for a possible hardcopy or screen capture operation.

There is a maximum of 5 sets permitted; each set can have a maximum of 10 commands; each command may have a maximum of 80 characters.

Currently, the only way to invoke this option is to have command sets stored in a file and to read the file from within EXEC_HIDE. The user should select the main menu item number 3 ("Set WINDOW/VIEW params"), then item number 1 from the next menu ("Read sets from file").

The command set file is basically a PATRAN session file fragment. No command checking is performed on these commands before they are executed within the PCL program, so the user should be sure that the commands will work as desired before reading the file. The data format for the file is simple: command sets are terminated by a line with a "%" character in it at any position. Any other contents of that line are ignored. An example data file is shown below; this file contains two command sets:

```
VIEW
1
10.18,-63.2,0.
WIN
1
.85
7
  %
VIEW
6
-1000,0,0
  %
```

Note that EXEC_HIDE communicates with PATRAN by issuing its commands from the PATRAN "MODE" menu; commands that you use within these optional sets should not leave PATRAN at a menu level with an unsatisfied request for input.

You can clear these command sets from the PCL program memory by getting back to the WINDOW/VIEW Parameters menu and selecting item 2 ("Clear all sets").

12. **Entering a PATRAN command from within the PCL program.**
You can enter a PATRAN command by selecting menu item number 2 from the main menu ("Enter a PATRAN command"). This option is meant to provide quick access to SET/SHOW commands; do not enter a command that will require terminal input, because PATRAN will attempt to read those values from the PCL program. To execute more elaborate commands, it's best to exit EXEC_HIDE, execute the PATRAN command, and then run EXEC_HIDE again.

This function is also available from the "Do plots" menu.

13. **Changing the plot type.**
EXEC_HIDE works by invoking the PATRAN command RUN,HIDE with several options. By default, EXEC_HIDE plots deformed-shape plots by invoking RUN,HIDE,DEFORM. You can change the options from the main menu by selecting item 6 ("Modify plot type"); this function is also available from the "Do plots" menu.

Most of the items on the Options menu are on/off switches: if an item is currently marked "ON" and you select it, then it will be set off.

Some items require presence of the element results files: the CONTOUR, FILL, and FRINGE options all use element results files. If you use one of these options, then define the appropriate file column by selecting item number 7 ("Results col"). Note that you must set the appropriate assignment or contour levels outside of EXEC_HIDE; the PCL procedure always uses the current contour level or assignment level settings. Item 5 ("PARIN") is an option, used to request parametric interpolation, that can be invoked if either FILL or CONT is active.

You can also request a different base name from this menu. By default, EXEC_HIDE works from the database component named NORMAL; you can request a different name if you like. The named component must exist in your PATRAN database before it is accepted here. This option would be useful, for example, if you wanted to define one named component that showed nodes and one that didn't. In that case, you could define one as NORMAL and the other as BASE2 before you ran EXEC_HIDE, and could select either one of those from this menu.

You can request that the plot images be stored in a PATRAN hardcopy file by switching on the HARDCOPY option.

Finally, you can define up to two title strings to be placed on resulting plots. These are displayed using PATRAN's RUN,TITLE,X=x,Y=y command. If you need only one title line, enter blanks in the other when you define it. When placing the title strings, note that if you have HARDCOPY switched on and the title lines are plotted in the terminal alpha region, the titles may not end up on the hardcopy file. In that case, try to make the Y values larger than 20 per cent. This is a PATRAN limitation.

A log file of an interactive terminal session which illustrates the preceding steps is listed below. The session shows activities on a Sun machine, and includes TCP/IP file transfers and PATRAN commands which use the PCL program to plot displacements. The results have been edited to remove some redundant information.

The example uses an X3D model of a bird impact against an aluminum plate. In this example, the bird is modeled with tetrahedral elements, and the aluminum plate is modeled with quadrilateral elements.

As shown in the log file, X3POST has already been run on a CRAY/UNICOS machine. A PATRAN database file of the model already exists on the Sun (patran.dat.1). In the example session, the user defines the NORMAL named component so as to include:

- All aluminum plate quad elements.
- All bird model tetrahedral elements.
- All bird model nodes (these begin with node number 106).

The user excludes from the NORMAL set the following:

- All contact elements (any triangle element is assumed to be a contact element here).
- Any node not used in the bird model. Nodes not included in the bird model are in the range 1 through 105.

The example log file follows below:

```
ss2:/scratch/held/b45wa01/ ls -l
total 0

ss2:/scratch/held/b45wa01/ ftp cray

Connected to sn418.wpafb.af.mil.
220 sn418 FTP server (Version 5.5 Sun Sep 23 14:27:26 EDT 1990) ready.
Name (cray:mccartre): heldtw

230 User heldtw logged in.

ftp> cd /tmp/held/b45wa01

250 CWD command successful.

ftp> dir

200 PORT command successful.
150 Opening ASCII mode data connection for /usr/ucb/ls.
total 12048
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D114
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D153
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D192
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D231
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D270
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D310
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D350
-rw-------  1 heldtw   D840200     19485 Jun  5 22:59 D38
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D391
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D432
-rw-------  1 heldtw   D840200     19485 Jun  5 23:00 D468
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D491
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D512
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D530
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D545
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D560
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D575
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D590
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D605
-rw-------  1 heldtw   D840200     19485 Jun  5 23:01 D606
-rw-------  1 heldtw   D840200     19485 Jun  5 22:59 D76
-rw-------  1 heldtw   D840200     51547 Jun  5 23:01 DISPPCL
-rw-------  1 heldtw   D840200      5510 Jun  5 23:01 DISPSES
-rw-------  1 heldtw   D840200    182846 Jun  5 22:59 E114
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E153
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E192
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E231
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E270
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E310
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E350
-rw-------  1 heldtw   D840200    182846 Jun  5 22:59 E38
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E391
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E432
-rw-------  1 heldtw   D840200    182846 Jun  5 23:00 E468
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E491
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E512
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E530
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E545
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E560
-rw-------  1 heldtw   D840200    182846 Jun  5 23:01 E575
```

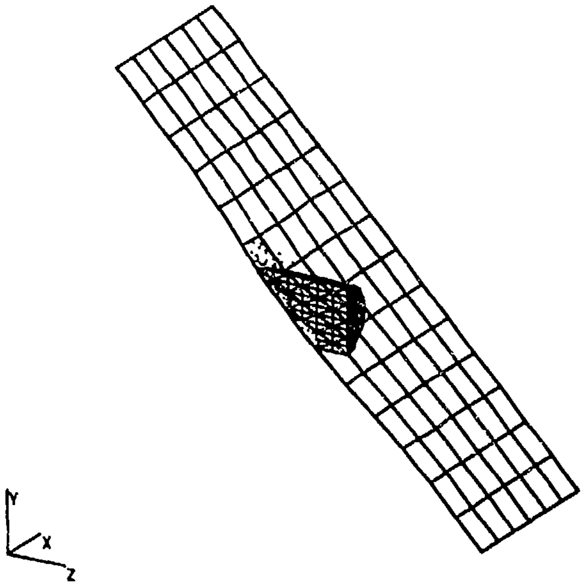
```
-rw-------  1 heldtw   D840200     182846 Jun  5 23:01 E590
-rw-------  1 heldtw   D840200     182846 Jun  5 23:01 E605
-rw-------  1 heldtw   D840200     182846 Jun  5 23:01 E606
-rw-------  1 heldtw   D840200     182846 Jun  5 22:59 E76
-rw-------  1 heldtw   D840200       1180 Jun  5 23:00 F114
-rw-------  1 heldtw   D840200       1299 Jun  5 23:00 F153
-rw-------  1 heldtw   D840200       1247 Jun  5 23:00 F192
-rw-------  1 heldtw   D840200       1445 Jun  5 23:00 F231
-rw-------  1 heldtw   D840200       1355 Jun  5 23:00 F270
-rw-------  1 heldtw   D840200       1291 Jun  5 23:00 F310
-rw-------  1 heldtw   D840200       1486 Jun  5 23:00 F350
-rw-------  1 heldtw   D840200        846 Jun  5 22:59 F38
-rw-------  1 heldtw   D840200       1463 Jun  5 23:00 F391
-rw-------  1 heldtw   D840200       1342 Jun  5 23:00 F432
-rw-------  1 heldtw   D840200       1141 Jun  5 23:00 F468
-rw-------  1 heldtw   D840200        991 Jun  5 23:01 F491
-rw-------  1 heldtw   D840200        832 Jun  5 23:01 F512
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F530
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F545
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F560
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F575
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F590
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F605
-rw-------  1 heldtw   D840200        785 Jun  5 23:01 F606
-rw-------  1 heldtw   D840200       1026 Jun  5 22:59 F76
-rw-------  1 heldtw   D840200      93089 Jun  5 22:44 b45wa01.dat
-rw-------  1 heldtw   D840200    1393016 Jun  5 22:59 b45wa01.dmp
-rw-------  1 heldtw   D840200        404 Jun  5 22:59 b45wa01.err
-rw-------  1 heldtw   D840200     195967 Jun  5 22:59 b45wa01.out
-rw-------  1 heldtw   D840200      47483 Jun  5 22:59 b45wa01.trc
-rw-------  1 heldtw   D840200       9931 Jun  5 23:01 xpout
226 Transfer complete.
4376 bytes received in 1.2 seconds (3.6 Kbytes/s)

ftp> prompt
Interactive mode off.

ftp> mget D* F* E*

200 PORT command successful.
150 Opening ASCII mode data connection for D114 (19485 bytes).
226 Transfer complete.
local: D114 remote: D114
19891 bytes received in 0.48 seconds (40 Kbytes/s)
200 PORT command successful.
150 Opening ASCII mode data connection for D153 (19485 bytes).
226 Transfer complete.
local: D153 remote: D153
19891 bytes received in 0.49 seconds (40 Kbytes/s)

(Gets all the Dxxx, Fxxx, and Exxx files, and the DISPPCL file.)

226 Transfer complete.

ftp> bye

221 Goodbye.

ss2:/scratch/held/b45wa01/ cp ~/ss2-files/b45/patran.dat.1 .

ss2:/scratch/held/b45wa01/ cp ~/ss2-files/b45/b45-view.txt .
```

```
ss2:/scratch/held/b45wa01/ ls

D114           D545           E310           E606           F512
D153           D560           E350           E76            F530
D192           D575           E38            F114           F545
D231           D590           E391           F153           F560
D270           D605           E432           F192           F575
D310           D606           E468           F231           F590
D350           D76            E491           F270           F605
D38            DISPPCL        E512           F310           F606
D391           DISPSES        E530           F350           F76
D432           E114           E545           F38            b45-view.txt
D468           E153           E560           F391           patran.dat.1
D491           E192           E575           F432
D512           E231           E590           F468
D530           E270           E605           F491

ss2:/scratch/held/b45wa01/ cat b45-view.txt
VIEW
1
10.18,-63.2,0.
WIN
1
.85
7
  %
VIEW
6
-1000,0,0
  %

ss2:/scratch/held/b45wa01/ patran2

=========================================================================

                              P A T R A N

                              RELEASE 2.5

                 COPYRIGHT 1979-1990  PDA ENGINEERING

=========================================================================

 ENTER DEVICE MNEMONIC, "?" OR "STOP": 4107

INPUT "GO", "SES", "HELP", PATRAN EXECUTIVE DIRECTIVE OR "STOP".

>GO

PATRAN DATA FILE?  1.NEW  2.OLD  3.LAST  4.STOP

>3

PREPARING THE DATA BASE SUB-SYSTEM FOR "OLD" FILE:
"patran.dat.1"
RESTART FROM THE FILE LAST CLOSED 27-MAY-91   20:43:25

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>SET,GRAPHICS,OFF
 "GRAPHICS" IS NOW OFF (WAS ON ).
```

```
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>SET,ACTIVE,ALL
 PHASE1 IS OFF.

>NODE,1T105,ERASE

>TRI,1T#,ERASE

>NAME,NORMAL,ADD

>SET,GRAPHICS,ON
 "GRAPHICS" IS NOW ON  (WAS OFF).

>SET,SINTERP,ON
 "SINTERP" IS NOW ON  (WAS OFF).

>SET,SFACT,1.0
"SFACTOR" IS NOW   1.0000     (WAS   0.1000    ).

>SET,ZNULL,OFF
 "ZNULL" IS NOW OFF (WAS ON ).

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>VIEW

VIEW? 1.AXES ABS 2.AXES REL 3.SCREEN ABS 4.SCREEN REL
      5.INCREMENTAL 6.FROM 7.VIEW PLANE 8.END

>1

 INPUT AXES ABSOLUTE THETA X,Y,Z FOR SCREEN  1

>20.2,60.3,0.0

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>WIN

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>4

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>1

 INPUT MAGNIFICATION POWER FOR SCREEN  1

>0.85

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>2

 INPUT DECIMAL OFFSET FACTOR IN X,Y FOR SCREEN  1   (EG: .33,.25)

>0,-0.05
```

```
WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>E

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>!!INPUT DISPPCL

!!INPUT DISPPCL
Compiling: exec_hide
  Compiled: exec_hide
Compiling: check_up
  Compiled: check_up
  ...
Compiling: mod_opts
  Compiled: mod_opts
Compiling: show_cols
  Compiled: show_cols

>!EXEC_HIDE()

!EXEC_HIDE()

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>5

Current file name ending is:
Enter new name ending to use (eg .DAT):
>(Enter a space and <Return> to keep the name ending blank.)

File name ending now is....:

Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>4

Failed-element list processing.
1. Read in lists  2. Delete existing failed-element names  3. End

>1

Warning: Current active set will be modified.
Continue ? Y/N...:

>Y

Number of files to process: 21
Processing file number 1 of 21: F38
Processing file number 2 of 21: F76
  ...
Processing file number 21 of 21: F613
```

```
1. Read in lists  2. Delete existing failed-element names  3. End

>E


Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>1

Begin or resume hidden surface plot processing.
Now: 38 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>(space bar and <Return>:  This displays information about time step 38.)

Incr: 38 (1 of 21) / Time:  1.0000e-04 / Hardcopy: OFF / Title: OFF
Now: 38 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>2

Now: 76 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>+

Now: 114 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>+

Now: 153 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>-

Now: 114 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>3

Now: 114 - Enter: 5. Specify incrt  6. Enter PATRAN cmd  7. Options  8. End

>5

Enter requested increment:

>275

Incr: 309 (8 of 21) / Time:  8.0000e-04 / Hardcopy: OFF / Title: OFF
Now: 309 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>1

Processing increment 309 (Number 8 of 21)...
 Command as processed: RUN,HIDE,DEFO

(Now does a plot.)
```

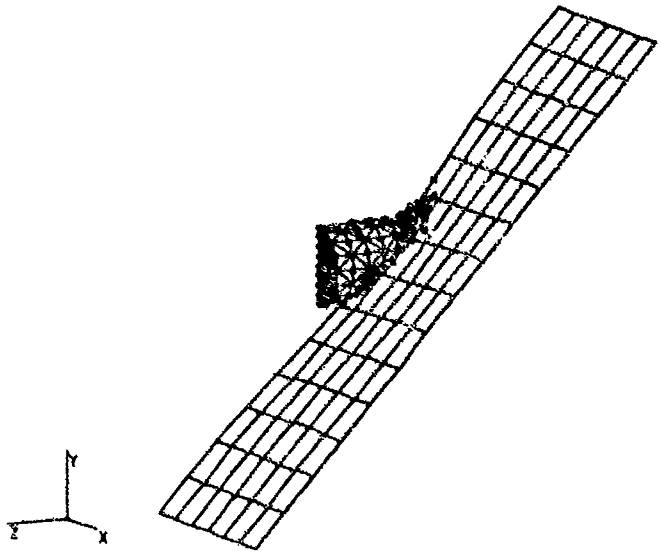


```
45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
DISPLACEMENTS FOR TIME STEP =      309 / TIME =   7.9999998E-04
27-MAY-91  18:08:53
```

```
Done with incrt 309; hardcopy was: OFF...
Now: 349 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>E


Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>E

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>!EXEC_HIDE()

!EXEC_HIDE()

Using some of the values from your last run.
Enter any character to continue:
>

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
```

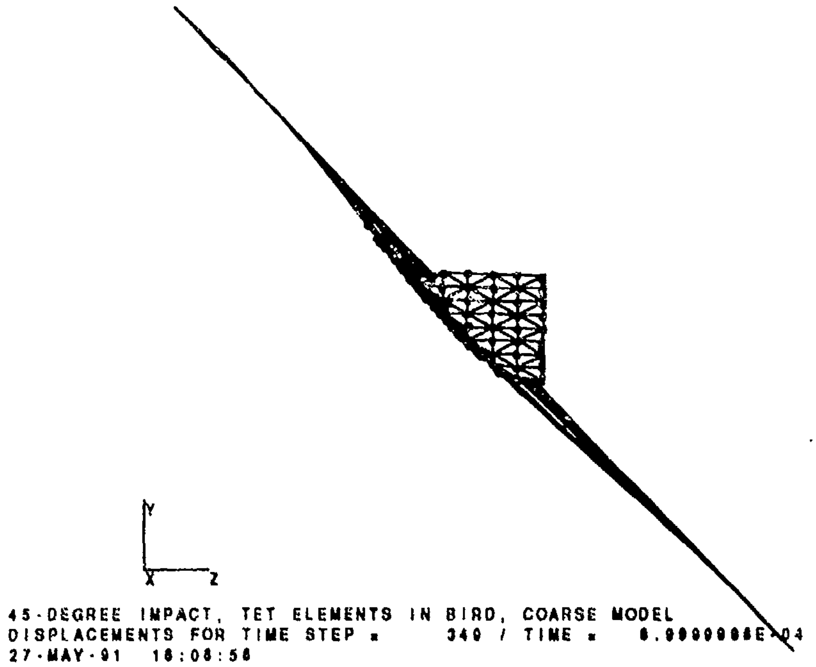
```
Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>3

WINDOW/VIEW Parameter Command Sets.
1. Read sets from file  2. Clear all sets  3. End and return to main menu

>1

Enter the name of the file with command set(s):

>B45-VIEW.TXT

B45-VIEW.TXT
Set 1 - Add: VIEW
Set 1 - Add: 1
Set 1 - Add: 10.18,-63.2,0.
Set 1 - Add: WIN
Set 1 - Add: 1
Set 1 - Add: .85
Set 1 - Add: 7
Set 2 - Add: VIEW
Set 2 - Add: 6
Set 2 - Add: -1000,0,0

WINDOW/VIEW Parameter Command Sets.
1. Read sets from file  2. Clear all sets  3. End and return to main menu

>E


Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>1

Begin or resume hidden surface plot processing.
Now: 349 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>3

Now: 349 - Enter: 5. Specify incrt  6. Enter PATRAN cmd  7. Options  8. End

>6

Enter a single-line PATRAN command:

>SET,RADN,1.8

SET,RADN,1.8
'RADNODE' IS NOW   1.8000     (WAS       0.    ).

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

Now: 349 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>1
```

```
Processing increment 349 (Number 9 of 21)...
 Command as processed: RUN,HIDE,DEFO

(Echoes some commands stored in b45-view.txt -- no keyboard input below.)

VIEW
VIEW? 1.AXES ABS 2.AXES REL 3.SCREEN ABS 4.SCREEN REL
      5.INCREMENTAL 6.FROM 7.VIEW PLANE 8.END
1
 INPUT AXES ABSOLUTE THETA X,Y,Z FOR SCREEN  1
10.18,-63.2,0.
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
WIN
WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT
1
 INPUT MAGNIFICATION POWER FOR SCREEN  1
.85
WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT
7
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

(Now does a plot.)
```



```
45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
DISPLACEMENTS FOR TIME STEP =      349 / TIME =   8.0000998E-04
27-MAY-91   18:08:58

Done with incrt 349, view set 1 of 2; hardcopy was: OFF...
Enter Y to continue, N to end:

>Y
```

```
(Echoes more commands stored in b45-view.txt -- no keyboard input below.)

VIEW
VIEW? 1.AXES ABS 2.AXES REL 3.SCREEN ABS 4.SCREEN REL
      5.INCREMENTAL 6.FROM 7.VIEW PLANE 8.END
6
 INPUT VIEW POINT FOR SCREEN  1
-1000,0,0
 VIEW POINT=   -1000.0           0.            0.
 CENTER=   6.0000           0.      -2.1200     AND DCENTER=   1006.0
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

(Now does a plot.)
```


Y
X
Z
45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
DISPLACEMENTS FOR TIME STEP = 349 / TIME = 8.9999998E-04
27-MAY-91 18:08:56

```
Done with incrt 349, view set 2 of 2; hardcopy was: OFF...
Now: 389 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>E

Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>E

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>STOP
RESTART DATA BEING WRITTEN ON 27-MAY-91   21:10:57 TO:
"patran.dat.1"
PDA/PATRAN COMPLETED
```

The next example continues post-processing of the above bird impact analysis, and shows how one might use some of the optional settings available in EXEC_HIDE. In this case, the user wants to display von Mises stresses in the aluminum target, and to store the resulting image on a PATRAN hardcopy file.

This example illustrates the following options:

- Use of a different base named component. Here, the user wants to view only the target stresses, and defines a named component BASE3 with all nodes, TRI elements, and TET elements removed. BASE3 is actually named component NORMAL, which was defined earlier, minus the remaining nodes and all TET elements (the bird elements are TET elements).
- Definition and placement of title strings.
- Use of the hardcopy switch. Output goes to a PATRAN hardcopy file, and the user is responsible for any actual hardcopy output.
- Definition of spectrum colors and assignment levels appropriate for this analysis. These must be defined before running EXEC_HIDE.

The example log file follows below:

```
ss2:/scratch/held/b45wa01/ patran2

==========================================================================

                              P A T R A N

                              RELEASE 2.5

                 COPYRIGHT 1979-1990  PDA ENGINEERING

==========================================================================

ENTER DEVICE MNEMONIC, "?" OR "STOP": 4107

INPUT "GO", "SES", "HELP", PATRAN EXECUTIVE DIRECTIVE OR "STOP".

>GO

PATRAN DATA FILE?  1.NEW  2.OLD  3.LAST  4.STOP

>3
(Use the PATRAN database file from the prior example.)

PREPARING THE DATA BASE SUB-SYSTEM FOR "OLD" FILE:
"patran.dat.1"
RESTART FROM THE FILE LAST CLOSED 27-MAY-91   21:10:57
```

```
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
```

(Modify NORMAL so only the plate elements remain; name the new component BASE3.)

```
>SET,GRAPHICS,OFF
 "GRAPHICS" IS NOW OFF (WAS ON ).

>NAME,NORMAL,PLOT
 DO YOU WANT THE CURRENT ACTIVE SET CLEARED? (Y/N)
>Y

>NODE,1T#,ERAS

>TET,1T#,ERAS

>NAME,BASE3,ADD
 NAMED COMPONENT "BASE3" ADDED.

>SET,GRAP,ON
 "GRAPHICS" IS NOW ON  (WAS OFF).

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
```

(Change the spectrum to values appropriate for Postscript hardcopy.)

```
>SET,SPEC,5,0/3/2/6/4
```

(Look at von Mises stresses; these are in column 9 of the element results file.)

```
>RUN,ASSIGN,COL.9

INPUT THE NAME OF THE ELEMENT RESULTS FILE:

>E567

 DATA WIDTH =     12
 FILE TITLE =45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
             ELEMENT RESULTS FOR TIME STEP =      567 / TIME =   1.6999999E-03
             27-MAY-91  18:09 35
 DATA VALUES RANGE FROM  0.270E+05 TO  0.743E+05
```

(If you want to process several time steps, pick results values that will be appropriate for them all.)

```
ASSIGNMENT? 1.AUTO 2.MANUAL 3.SEMI-AUTO 4.THRESHOLD
            5.CURRENT LEVELS 6.END
>3

INPUT THE BASE VALUE AND A DELTA VALUE

>0.8E5, -.2E5

 ASSIGNED COLOR CODE RANGES FOLLOW:
  A  0.8000E+05 TO  0.6000E+05        B  0.6000E+05 TO  0.4000E+05
  C  0.4000E+05 TO  0.2000E+05        D  0.2000E+05 TO  0.0000E+00
    ASSIGNING COLORS TO 84 ELEMENTS:^G
*NOTICE***  SETTING "COLCODE" ON.

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
```

```
(Adjust the viewing angle and window for this model.)

>VIEW

VIEW? 1.AXES ABS 2.AXES REL 3.SCREEN ABS 4.SCREEN REL
      5.INCREMENTAL 6.FROM 7.VIEW PLANE 8.END

>1

 INPUT AXES ABSOLUTE THETA X,Y,Z FOR SCREEN  1

>10.18,-63.2,0.

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>WIN

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>1

 INPUT MAGNIFICATION POWER FOR SCREEN  1

>0.85

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>2

 INPUT DECIMAL OFFSET FACTOR IN X,Y FOR SCREEN  1   (EG: .33,.25)

>0, -0.05

WINDOW? 1.ZOOM  2.MOVE  3.CENTER  4.FIND  5.CORNERS  6.RESTORE  7.EXIT

>E

MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU

>!!INPUT DISPPCL

!!INPUT DISPPCL
Compiling: exec_hide
  Compiled: exec_hide
Compiling: check_up
  Compiled: check_up
   ...
Compiling: mod_opts
  Compiled: mod_opts
Compiling: show_cols
  Compiled: show_cols

>!EXEC_HIDE()

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
```

```
Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>6

(Turn on the FILL switch below.)

Options mods -- current command is: RUN,HIDE,DEFO
1. DEFO=ON  2. CONT=OFF  3. FILL=OFF  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=OFF  7. Results col=1  8. Title=OFF  9. Base name=NORMAL  10. END

>3

Current assignment levels will be used.
Enter a character to continue:
>

(Set the RUN,ASSIGN results column number below; von Mises stresses are in
column 9.)

Options mods -- current command is: RUN,HIDE,DEFO,FILL
1. DEFO=ON  2. CONT=OFF  3. FILL=ON  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=OFF  7. Results col=1  8. Title=OFF  9. Base name=NORMAL  10. END

>7

Current results columns is: 1
Enter new results column number or zero to show types:
>9

(Request a different base named component below.)

Options mods -- current command is: RUN,HIDE,DEFO,FILL
1. DEFO=ON  2. CONT=OFF  3. FILL=ON  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=OFF  7. Results col=9  8. Title=OFF  9. Base name=NORMAL  10. END

>9

Current base name is: NORMAL
Enter a new base name or a blank to keep the old one:
>BASE3

Checking the new base name...

(BASE3 is accepted; next, turn on the hardcopy switch below.)

Options mods -- current command is: RUN,HIDE,DEFO,FILL
1. DEFO=ON  2. CONT=OFF  3. FILL=ON  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=OFF  7. Results col=9  8. Title=OFF  9. Base name=BASE3  10. END

>6

(Define some title strings and their placement below.  Note that there are two
title strings available; define them in order.)

Options mods -- current command is: RUN,HIDE,DEFO,FILL,HARDC
1. DEFO=ON  2. CONT=OFF  3. FILL=ON  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=ON  7. Results col=9  8. Title=OFF  9. Base name=BASE3  10. END

>8
```

```
Title option is now: OFF
Title line 1 location command is now: RUN,TITLE,X=1.,Y=10.
Title line 1 string is now :
1. Chng option (ON/OFF)  2. Chng locations  3. Chng strings  4. Test  5. End

>1

Title option is now: ON
Enter a character to continue:
>

Title option is now: ON
Title line 1 location command is now: RUN,TITLE,X=1.,Y=10.
Title line 1 string is now :
1. Chng option (ON/OFF)  2. Chng locations  3. Chng strings  4. Test  5. End
```

(Define locations: put the title strings in the upper left corner of the image.)

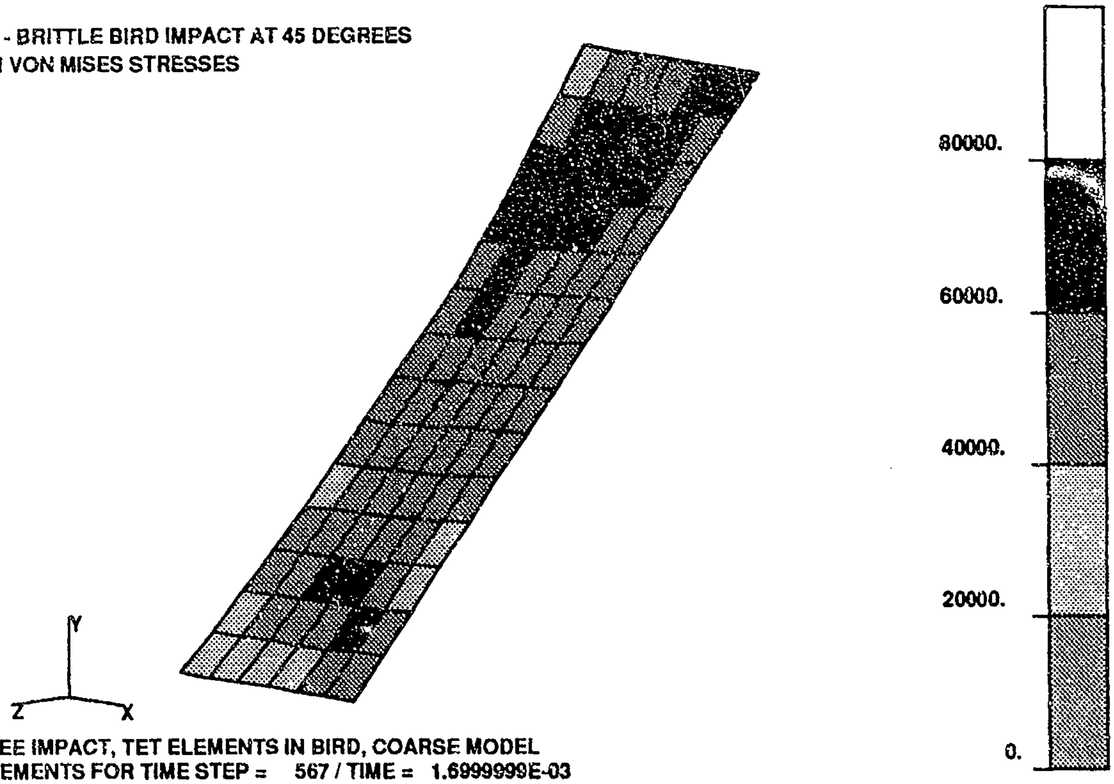
```
>2

Title location command (1) is now: RUN,TITLE,X=1.,Y=10.
Choose: 1. X=1./Y=10.  2. X=55./Y=95.  3. Input X/Y  4. End/no change

>3

Enter screen X percent value (0-100):

>1

Enter screen Y percent value (0-100):

>98

Title location command (2) is now: RUN,TITLE,X=1.,Y=5.
Choose: 1. X=1./Y=5.  2. X=55./Y=90.  3. Input X/Y  4. End/no change

>3

Enter screen X percent value (0-100):

>1

Enter screen Y percent value (0-100):

>95

Title option is now: ON
Title line 1 location command is now: RUN,TITLE,X=1.,Y=98.
Title line 1 string is now :
1. Chng option (ON/OFF)  2. Chng locations  3. Chng strings  4. Test  5. End
```

(Enter the title strings. If a string is currently blank, nothing will show in the prompt. If you want only one title string to be displayed, then enter some blanks for the other string.)

```
>3

Title string (1) is now :
Change it ? Y/N...:

>Y
```

```
Enter a title string (max 75 chars):

>B45W10A - BRITTLE BIRD IMPACT AT 45 DEGREES

Title string (2) is now :
Change it ? Y/N...:

>Y

Enter a title string (max 75 chars):

>MAXIMUM VON MISES STRESSES

Title option is now: ON
Title line 1 location command is now: RUN,TITLE,X=1.,Y=98.
Title line 1 string is now : B45W10A - BRITTLE BIRD IMPACT AT 45 DEGREES
1. Chng option (ON/OFF)  2. Chng locations  3. Chng strings  4. Test  5. End

>E

(Done setting the plot options.  Note the options on the RUN,HIDE command below:
this is how the images will be displayed.)

Options mods -- current command is: RUN,HIDE,DEFO,FILL,HARDC
1. DEFO=ON  2. CONT=OFF  3. FILL=ON  4. FRIN=OFF  5. PARIN=OFF
6. HARDC=ON  7. Results col=9  8. Title=ON  9. Base name=BASE3  10. END

>E


Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>1

Begin or resume hidden surface plot processing.
Now: 38 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>3

Now: 38 - Enter: 5. Specify incrt  6. Enter PATRAN cmd  7. Options  8. End

>5

Enter requested increment:

>567

Incr: 567 (17 of 21) / Time:  1.7000e-03 / Hardcopy: ON / Title: ON
Now: 567 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>P

Processing increment 567 (Number 17 of 21)...
 Command as processed: RUN,ASSIGN,COL,9
 Command as processed: RUN,HIDE,DEFO,FILL,HARDC
 Command as processed: RUN,TITLE,X=1.,Y=98.,HARDC
 Command as processed: RUN,TITLE,X=1.,Y=95.,HARDC

(Now does the plot.)
```

B45W10A - BRITTLE BIRD IMPACT AT 45 DEGREES
MAXIMUM VON MISES STRESSES



45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
DISPLACEMENTS FOR TIME STEP = 567 / TIME = 1.6999999E-03
27-MAY-91 18:09:39

```
Done with incrt 567; hardcopy was: ON...
Now: 582 - Enter: 1. Plot  2. +/- Step incrt  3. More commands  4. End

>E

Hidden Surface Plotter Main Menu for Analysis:
  45-DEGREE IMPACT, TET ELEMENTS IN BIRD, COARSE MODEL
1. Do plots  2. Enter a PATRAN command  3. Set WINDOW/VIEW params
4. Process failed-elemt lists  5. File names  6. Modify plot type  7. End

>E

MENU,MODE
MODE? 1.GEOMETRY 2.ANALYSIS MODEL 3.ANALYZE 4.RESULTS
      5.INTERFACE 6.STOP 7.PDA_PCL 8.USER_MENU
>STOP
RESTART DATA BEING WRITTEN ON 27-MAY-91   22:43:25 TO:
"patran.dat.1"
PDA/PATRAN COMPLETED
```

(End of example.)

APPENDIX C
PROGRAMMING INFORMATION

This Appendix provides brief technical information on installing and using X3D and the related pre- and post-processing programs, PATX3D and X3POST.

The X3D programs have been run on several types of machines. Some of the computer systems and the operating systems which were current at the time of this documentation are as follows:

CRAY X-MP/216:

- Operating system: UNICOS 5.1.1
- FORTRAN compiler: cf77 4.0.3.8

CRAY Y-MP8/864:

- Operating system: UNICOS 6.0.11
- FORTRAN compiler: cf77 4.0.3.10

DEC VAX 8650:

- Operating system: VMS 5.3-1
- FORTRAN compiler: DEC FORTRAN 5.4-79

Sun 4:

- Operating system: SunOS 4.1.1
- FORTRAN compiler: f77 1.4

IBM RS/6000 Model 530:

- Operating system: AIX 3.1
- FORTRAN compiler: xlf 01.01

Machine-specific instructions for compiling and running the X3D programs are described in later paragraphs. General information about installing the programs follows below.

The X3D programs are written entirely in FORTRAN, and comply closely with the ANSI FORTRAN 77 standard. However, each machine version includes a small amount of nonstandard code which is specific to that machine alone. Except for the nonstandard "INCLUDE" statement (discussed below), any nonstandard code is highly localized within the programs, and can be replaced with standard (but less aesthetic) FORTRAN code if required.

The X3D programs are maintained in a nonrunning "master" format, in which the machine-specific code has been commented out with strings corresponding to the target machines. The comment strings must be removed with an editor to create valid FORTRAN source code. The strings which must be removed are "C-VAX" for the VAX versions; "C-CRY" for the CRAY versions; "C-SUN" for the Sun versions; and "C-IBM" for the IBM RS/6000 RISC workstations.

The X3D programs use the nonstandard "INCLUDE" FORTRAN statement to include COMMON block definitions and double precision information in the source code at compilation time. The machines listed above (and many other machines) support use of the INCLUDE statement in their compilers. The include-files required to be present when compiling the X3D programs are:

```
patx3d.ins
x3com.ins
x3d.ins
x3hid.ins
x3post.ins
xpcom.ins
xscom.ins
```

References to these file names within the code are always specified using lower case letters, so UNIX-based machines require that the include-files, as present in the appropriate directory, have file names with lower case letters.

PATX3D is a standalone program, and X3D can be compiled and linked as a standalone program. X3POST, however, requires routines from the X3D source file. Normally, the user should create an object library from the X3D object file, and use the library when linking to create the X3POST executable file.

X3D and X3POST typically are run using single precision floating point variables (32 bit words on VAX, Sun, and IBM machines; 64 bit words on CRAY machines). Users who wish to create double precision versions of the codes must edit the include-file *x3d.ins* and remove the "C-DBL" string there; this is the only change required to convert the codes to double-precision. The **entire** versions of the codes then must be recompiled using the new version of the include-file.

Most users should not need to use double precision versions of the codes. IBM RS/6000 users may find that a double-precision version of X3D actually runs slightly faster for some cases than the single-precision version, but note that disk space requirements for the double-precision restart file are approximately twice that of the single-precision version. Note also that X3POST must be compiled as a double-precision version if it is to read a double-precision restart file, or if it is to be linked with a double-precision version of the X3D object library.

Some transparency analysis runtime statistics are shown below. Note that a critical resource in an analysis is disk space. This can be further aggravated by running X3POST later: the example problem shown stored 60 time steps worth of data in the restart file, and if the user requests PATRAN results files for all 60 time steps, there is an additional 24 Mbyte disk requirement beyond the amount shown in the table.

Example problem specifics (this is a version of the examples described in Section 3.3):

- Problem title . . . . : F16BE.DAT
- Number of plates (transparency elements) . . : 928
- Number of tetrahedral solids (bird elements) . : 960
- Bird material model . . . . : low-strength
- Initial bird velocity . . . . : 350 knots
- Restart file output . . . . : total of 60 time steps written
- Trace file output . . . . : 30 nodes, every 20 time steps
- All X3D program versions were single-precision versions. This gave 32-bit precision results on all the listed machines except the Cray machines, which gave 64-bit precision results.
- The number of time steps calculated varies because of differing architectures and round-off errors.

SAMPLE TRANSPARENCY ANALYSIS RUNTIME STATISTICS

Machine Vendor and Model	Operating System and Compiler Versions	F76BE Run Time Seconds (Hrs:Mins)	Time Steps Calculated	OREST File Size (Mbytes)	TRACE File Size (Mbytes)	Printable Output File (Mbytes)	Total Disk Space (Mbytes)
CRAY Y-MP8/864	UNICOS 6.0.11 cft77 4.0.3.10	2,418 (0:40)	6,327	27.574	0.870	0.409	28.853
CRAY X-MP/216	UNICOS 5.1.10 cft77 4.0.3.8	3,199 (0:53)	6,327	27.574	0.870	0.409	28.853
IBM RS/6000 Model 530	AIX 3.1 xlf 01.01	5,380 (1:30)	6,302	13.767	0.865	0.408	15.040
Silicon Graphics SGI 4D/320 VGX	IRIX Sys. V, Rel. 3.3.2 MIPS f77 2.01	6,448 (1:47)	6,275	13.767	0.861	0.408	15.036
Sun 4 (SPARC)	SunOS 4.1.1 Sun f77 1.4	15,689 (4:21)	6,368	13.767	0.875	0.408	15.050
VAX 8650	ULTRIX 3.1 VAX fort 4.7-271	20,054 (5:34)	6,318	13.767	0.867	0.408	15.042
VAX 8650	VMS 5.3-1 Fortran 5.4-79	20,345 (5:39)	6,316	13.791	0.878	0.416	15.085
Harris HCX-9	CX/UX 4.1-A f77 4.1	37,734 (10:39)	6,380	13.767	0.878	0.408	15.053

CRAY X-MP and Y-MP UNICOS Versions

CRAY machines provide the fastest processing times of the all machines supported. A sample NQS batch job file used to run X3D on a CRAY X-MP is listed below.

```
# QSUB -s /bin/sh
# QSUB -lM 4mw  -lT 9900  -lt 9800
# QSUB -eo
# QSUB -r f16an

set -vxS
ja
trap 'ja -stc' 0 15

#
# -- X3D on Cray X-MP / UNICOS 5.1.10
#
#    Current with X3D 2.5.

# -- Get into TMPDIR and copy input data.

cd $TMPDIR

cp $HOME/f16an/f16an.dat  f16an.dat

# -- Run X3D.

$HOME/x3d25/x3d.exe <f16an.dat >f16an.out 2>errfile

cat errfile

ls -l

# -- Save some files.

cp f16an.out   $HOME/f16an/.
cp NREST       $HOME/f16an/f16an.dmp
cp TRACE       $HOME/f16an/f16an.trc

# -- All done.
```

A sample NQS batch job used to compile X3D, PATX3D, and X3POST on a CRAY X-MP is listed below. The code inlining option available with the cft77 compiler can speed up execution times by several per cent, so its use is illustrated below. Note that the procedure creates FORTRAN source files from the master versions of the codes.

```
# QSUB -s /bin/sh
# QSUB -lM 4mw  -lT 250
# QSUB -eo
# QSUB -r x3dcom5

# -- Compile X3D, X3POST, and PATX3D under UNICOS.
#
#    Current with UNICOS 5.1.1 and cft77 4.0.3.8
#
#    Current with X3D 2.5.
#
```

```
#    Use the cf77 tool to request automatic vectorization.
#
#    Compile x3d.f twice - once to get an executable that has
#    been code inlined, and once to get a non-inlined object
#    file for use with a library.
#
#   Note that the inlined version of x3d.exe did not work correctly
#    for us under an early version of UNICOS 6.0.0 - use the
#   non-inlined compile option if you think you will have problems.
#
# -- File information:
#    -----------------
#    You will need the following files in your working directory
#    to compile the three codes:
#
#    Master source versions:
#      1. x3d.mas
#      2. patx3d.mas
#      3. x3post.mas
#
#    Include-files (not all are used in all programs).
#    File names must be exactly as shown below, with
#    lower case letters:
#      1. patx3d.ins
#      2. x3d.ins
#      3. x3com.ins
#      4. x3hid.ins
#      5. x3post.ins
#      6. xpcom.ins
#      7. xscom.ins
#
# -- End of file information.
#
#    04-Jun-91 -- TH

set -vxS
ja
trap 'ja -stc' 0

# -- Code is in $HOME/x3d25.
#    We need routines in x3d.f to link with x3post,
#    so create object files for both and then a library
#    of the x3d code.

cd $HOME/x3d25

ls -l

# -- Check up on the precision we're compiling with.
#    Should be single on this machine...

cat x3d.ins

# -- Commented out code above means we're compiling single
#    precision...


# -- Edit up some Fortran source from the master files.
#    First x3d, then others...

if [ -f x3d.f ]
  then
  mv x3d.f x3d.$$
fi
```

```
ex x3d.mas <<EOF
%s/C-CRY//g
w! x3d.f
q
EOF

if [ -f x3post.f ]
  then
  mv x3post.f x3p.$$
fi

ex x3post.mas<<EOF
%s/C-CRY//g
w! x3post.f
q
EOF

if [ -f patx3d.f ]
  then
  mv patx3d.f pat.$$
fi

ex patx3d.mas<<EOF
%s/C-CRY//g
w! patx3d.f
q
EOF

# -- Now compile everything, and link x3d and patx3d.

#    Use the statement below instead if you don't want
#    code inlining in x3d.exe:
#    cf77 -Zv -v                -o x3d.exe    x3d.f

cf77 -Zv -v -Wf"-o inline" -o x3d.exe    x3d.f

cf77 -Zv -v                -o patx3d.exe patx3d.f

cf77 -Zv -v -Wf"-b x3d.o"                x3d.f

cf77 -Zv -v -Wf"-b x3post.o"             x3post.f

# -- Make a library.

bld rzv  x3d.a  x3d.o

# -- Now link these to get x3post.

segldr -l ./x3d.a  -o x3post.exe -V x3post.o

# -- Clean up.

rm *.o

# -- Done.
```

A sample UNICOS Bourne shell script used to run X3POST interactively on the CRAY is listed below:

```
#!/bin/sh
# -- Bourne shell script to run X3POST  with a command line
#    option to specify the restart file name.
#    Good only for UNICOS because of the non-standard "assign"
#    command (this uses the UNICOS 5.1 syntax for "assign").
#    04-Jun-91 -- TH
#
#    Usage:
#      1. Store this file in your $HOME/bin subdirectory.
#         Call this file "x3post".
#         Make sure that x3post has user executable privilege:
#            chmod u+x x3post
#      2. Make sure that your $HOME/bin subdirectory is in your
#         path (enter "set" to see the current path value).
#      3. Log out then log in again.
#      4. To run x3post, enter:
#            x3post restartfile.name
#
FILENV=assign.$$
export FILENV
if [ $# -eq 0 ]
then
  echo "Error:"
  echo "You must specify the restart file name as parameter 1."
  echo "Example: x3post cylinder.dmp"
elif [ -f $1 ]
then
  assign -a $1 OREST
#
#   Edit the path below to be the directory where you have
#   x3post.exe stored.
#
  $HOME/x3d25/x3post.exe
else
  echo "Error: Cannot find file $1 to use with x3post."
fi
if [ -f assign.$$ ]
then
  rm assign.$$
fi
```

VAX/VMS Versions

A sample DCL batch command procedure used to run X3D is listed below:

```
$! VAX/VMS Command Procedure to Run X3D in Batch Mode
$! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
$!
$!  This file is called B45C.COM on the VAX.
$!  It runs X3D on the B45C dataset.
$!
$!  Usage: $submit /keep /noprint /log=[] B45C
$!
$!  Results are written to the following files:
$!     B45C.OUT  -  line printer file
$!     B45C.DMP  -  restart file
$!     B45C.TRC  -  trace file
$!
$ set verify
$!
$ set def [scratch.b45c]
$!
$ assign/user b45c.dat for005
$ assign/user b45c.out for006
$ assign/user b45c.dmp nrest
$ assign/user b45c.trc trace
$!
$ run work$fier:[scratch.x3d]x3d
$!
$ sho process /all
```

A sample DCL command procedure used to compile the X3D programs is listed below. Note that the procedure creates the FORTRAN source code from the master versions of the codes.

```
$! -- VAX/VMS command procedure to compile and link
$!    PATX3D, X3D, and X3POST.
$!
$! 1. You must have the following include-files in the
$!    working subdirectory:
$!       1. PATX3D.INS
$!       2. X3COM.INS
$!       3. X3D.INS
$!       4. X3HID.INS
$!       5. X3POST.INS
$!       6. XPCOM.INS
$!       7. XSCOM.INS
$!
$! 3. The procedure builds VAX FORTRAN source code from
$!    master versions of the codes.  By default, programs
$!    use single precision real arithmetic.  You must have
$!    the following master code files in your working
$!    subdirectory:
$!       1. PATX3D.MAS
$!       2. X3D.MAS
$!       3. X3POST.MAS
$!
```

```
$! 4. PATX3D is a standalone program.
$!    X3POST requires some routines from X3D, so the
$!    procedure builds an object library from X3D and
$!    uses it when linking X3POST.
$!
$   on error then goto getout
$!
$! -- Make PATX3D.
$!
$   edit/nocommand patx3d.mas
s/c-vax// w
write patx3d.for
quit
$   for patx3d
$   del patx3d.for;0
$   link patx3d
$!
$! -- Make X3D.
$!
$   edit/nocommand x3d.mas
s/c-vax// w
write x3d.for
quit
$   for x3d
$   del x3d.for;0
$   lib/create x3d x3d
$   del x3d.obj;0
$   lib/extract=(x3d)/output=x3d.obj x3d
$   link x3d,x3d/lib
$!
$! -- Make X3POST.
$!
$   edit/nocommand x3post.mas
s/c-vax// w
write x3post.for
quit
$   for x3post
$   del x3post.for;0
$   link x3post,x3d/lib
$!
$ getout:
$   exit
```

A sample DCL interactive command procedure used to run X3POST is listed below:

```
$! VAX/VMS Command Procedure to Run X3POST Interactively
$! ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
$!
$!  Usage:  @X3POST name
$!          (this command file is called X3POST.COM)
$!  Where:  'name' is the root name of the restart file
$!
$!  Results are read from:
$!     name.DMP  -  restart file
$!
$ ver_set = 'f$verify()'
$ set verify
$ assign/user sys$command sys$input
$ assign/user 'P1'.dmp    orest
$ run work$fier:[scratch.x3d]x3post
$ set noverify
$ if ver_set then set verify
$ exit
```

Sun SunOS Versions

An example UNIX script file used to run X3D in the background on a Sun is listed below:

```
#!/bin/sh

set -vx

#
# X3D script file for SunOS 4.x background run.
#
# This script is file: tbcyl.scr.
# Make sure it has execute permission:
#     chmod +x tbcyl.scr
#
# From the C shell use:
#  /usr/bin/nohup  tbcyl.scr > tbcyl.log &
#
# 04-Jun-91 -- TH

pwd

date

# What host ran this job?

/bin/uname -a

cd /scratch

mkdir tbcyl

cd tbcyl

cp $HOME/x3d-runs/tbcyl/tbcyl.dat .

ls -l

# --- Now run...

/usr/bin/time $HOME/x3d25/x3d.exe \
    < tbcyl.dat  > tbcyl.out  2>tbcyl.err

cat *err

# --- Note - time above is in seconds...

# --- Now run x3post.

mv NREST OREST

$HOME/x3d25/x3post.exe <<EOF >xpout
o
a
s
EOF

ls -l

# --- Done.
```

An example Bourne shell script file used to compile the X3D programs is listed below. Note that we have had problems with code optimization under versions 1.2 and 1.3 of the Sun FORTRAN compiler. This procedure creates FORTRAN source files from the master versions of the codes.

```
#!/bin/sh

set -vx

# --- X3D / X3POST / PATX3D compile script for Sun 4.
#
# Current with X3D 2.5.
#
# This system uses SunOS 4.1.1; Sun FORTRAN 1.4.
#
# Note: Optimization causes problems in prior releases
# of Sun's FORTRAN on SPARC machines.
#
#   For f77 1.2 use:
#     f77 -Nx400 -c x3d.f
#
#   For f77 1.3.1 use:
#     f77 -V -fast -O1 -Nx400 -c x3d.f
#
# This script is file x3dcomp.scr.
# Make sure it has execute permission:
#     chmod +x x3dcomp.scr
# From the C shell (normal shell for the Sun), use:
#    /usr/bin/nohup  x3dcomp.scr >x3dcomp.log &
#
# -- File information:
#    -----------------
#    You will need the following files in your working directory
#    to compile the three codes:
#
#    Master source versions:
#      1. x3d.mas
#      2. patx3d.mas
#      3. x3post.mas
#
#    Include-files (not all are used in all programs).
#    File names must be exactly as shown below, with
#    lower case letters:
#      1. patx3d.ins
#      2. x3d.ins
#      3. x3com.ins
#      4. x3hid.ins
#      5. x3post.ins
#      6. xpcom.ins
#      7. xscom.ins
#
# -- End of file information.
#
#
# 04-Jun-91 -- TH

pwd

date
```

```
# --- What host ran this problem?

/bin/uname -a

# --- Make a working subdirectory and go there.
#     This job will create a bunch of files, so put them
#     in a place that will be easy to clean up.

cd /tmp
mkdir x3dcomp

cd x3dcomp
mkdir $$
cd $$

pwd

cp $HOME/x3d25/*.ins .

cp $HOME/x3d25/*.mas .

# -- Check up on the precision we're compiling with.
#    Should be single on this machine...

cat x3d.ins

# -- Commented out code above means we're compiling single
#    precision...

# --- Edit up x3d first to get FORTRAN files from the master file.
#     We want to make a library file for later use with x3post.

ex x3d.mas <<EOF
%s/C-SUN//g
w! x3dfull.for
q
EOF

# --- Now fsplit into separate files and compile.
#     ar-type library modules are stored by file, not by
#     program section, so we need one program per file.

fsplit x3dfull.for >fsplit.out

rm x3dfull.for

f77 -V -fast  -Nx400 -c *.f

# --- Make a library.

ar crv x3d.a  *.o

ranlib x3d.a

# --- Make an executable.
#     Note that fsplit may have named the main program X3D.f,
#     so change the object file name if necessary...

if [ -f X3D.o ]
  then
  mv  X3D.o  x3d.o
fi

f77 -V -fast -o x3d.exe  x3d.o  x3d.a
```

```
# --- Clean up the mess and work on x3post and/or other
#     codes.

rm  *.o  *.f

# --- Another edit to get Sun-specific source.

ex x3post.mas<<EOF
%s/C-SUN//g
w! x3post.f
q
EOF

f77 -V -fast -Nx400 -c  x3post.f

f77 -V -fast -Nx400 -o x3post.exe  x3post.o  x3d.a


ex patx3d.mas<<EOF
%s/C-SUN//g
w! patx3d.f
q
EOF

f77 -V -fast -Nx400 -o patx3d.exe  patx3d.f

# --- Done with compiles -- copy back the executables.
#     Clean up the rest of the subdirectory later, in case
#     something needs checking.

cp  *.exe  $HOME/x3d25/.

cp  *.a    $HOME/x3d25/.

date

ls -l

# --- All done.
```

IBM RS/6000 AIX Versions

An example AIX script file used to run X3D in the background on an IBM RS/6000 is listed below:

```
#!/bin/ksh

set -vx

# X3D run on an IBM RS/6000 AIX 3.1.
#
# Korn shell script file for background run.
#
# This file is: f16bj.scr
# Make sure it has execute permission:
#     chmod +x f16bj.scr
#
# Do:
#    nohup f16j.scr >f16j.log &
# to run in no-logout background.
#
# X3D 2.5 - single-precision run.
#

date

cd /m115/scratch

mkdir f16bg

cd f16bg
pwd

cp $HOME/f16/f16bg.dat .

timex $HOME/x3d25/x3d.exe  \
  <f16bg.dat >f16bg.out  2>errfile

cat err*

df /m115

ls -l

# -- Run x3post...

mv NREST OREST

timex $HOME/x3d25/x3post.exe <<e-o-f >xpout
o
a
s
e-o-f

df /m115

ls -l

# -- Done.
```

An example Korn shell script file used to compile the X3D programs is listed below. Note that this procedure creates FORTRAN source files from the master versions of the codes.

```
#!/bin/ksh

set -vx

# IBM RS/6000 script file to compile X3D and
# X3POST.  No version of PATX3D yet, but it should be
# easy to port.
#
# Current with X3D 2.5.
#
# This sytem uses AIX 3.1 and XLF FORTRAN 01.01.
#
# This file is: x3dcomp.scr
# Make sure it has execute permission:
#     chmod +x x3dcomp.scr
#
# Do:
#     nohup x3dcomp.scr >x3dcomp.log &
# to run in no-logout background.
#
# Note: ibmtim.c is a user-written C routine
# that gets time/date from the system.  See
# SUBROUTINE TSTAMP in X3D.MAS for the source
# code and instructions (also on what to do if
# you don't want to use ibmtim.c / ibmtim.o).
#
# -- File information:
#    -----------------
#    You will need the following files in your working directory
#    to compile the three codes:
#
#    Master source versions:
#      1. x3d.mas
#      2. patx3d.mas
#      3. x3post.mas
#
#    Include-files (not all are used in all programs).
#    File names must be exactly as shown below, with
#    lower case letters:
#      1. patx3d.ins
#      2. x3d.ins
#      3. x3com.ins
#      4. x3hid.ins
#      5. x3post.ins
#      6. xpcom.ins
#      7. xscom.ins
#
# -- End of file information.
#
# 14-May-91 -- TH

date

cd $HOME/x3d25

pwd
```

```
# --- Verify what precision we're working with here...
cat x3d.ins

# --- Code commented out above means a single-precision compile.

# --- Edit up some master files.
#     Rename existing FORTRAN files to something else
#     first, if necessary.
#     First compile x3d, then x3post...

if [ -f x3d.f ]
  then
    mv x3d.f x3d.$$
fi

if [ -f x3post.f ]
  then
    mv x3post.f x3p.$$
fi

# -- This edit sequence includes activation of the call to IBMTIM:

ex x3d.mas <<EOF
/IBMTIM ( DAT
s/C-IBMC//g
%s/C-IBM//g
w! x3d.f
q
EOF

ex x3post.mas<<EOF
%s/C-IBM//g
w! x3post.f
q
EOF

# --- Now compile and link.
#     Routine ibmtim.c is already in the directory...

xlf -c -O x3d.f

xlf -c -O x3post.f

xlc -c    ibmtim.c

# All compiled now (hopefully), so link them
# together.  The Wl,-bloadmap option generates
# a fairly small load map - good to see what
# happened at link/bind time.

xlf -o x3d.exe -Wl,-bloadmap:x3d.map \
             x3d.o ibmtim.o

xlf -o x3post.exe  -Wl,-bloadmap:x3post.map \
             x3post.o  x3d.o  ibmtim.o

ls -l

# --- Done.
```

APPENDIX D
MATERIAL PROPERTY TABLES

The table of properties in this Appendix presents typical material constant values for several engineering materials. These data are intended as a useful reference and basis for comparison with properties from other sources. However, we wish to emphasize that the values in the table are taken from a variety of sources, and may vary considerably in quality. Some data simply are not available for particular materials, and are omitted from the table; for example, the absence of a value for 'D' or 'p' does not necessarily imply that the material in question is insensitive to strain rate.

When using the listed material properties with X3D, remember that:

- densities should be converted to force-length-time units (i.e., divided by 'g'); and

- values of 1/D and 1/p (not D and p) are required to define the strain rate sensitivity in the X3D input.

Additional details on the material models and data requirements can be found in Section 2 (theory) and Appendix A (input data descriptions).

MATERIAL CONSTANTS FOR SEVERAL COMMON MATERIALS

Material Description	Density lbf/in^3	E ksi	H ksi	ν	α με/°F	σ_y ksi	D sec^{-1}	p	σ_u ksi	Data Source
Acrylic	0.043	450	50.	0.35	41.	10.			20.	
Aluminum 2024-T3	0.098	10,600.		0.33	12.6	50.			70.	2,5
Aluminum 6061-T6	0.098	10,600.	416.	0.33	13.3	40.	6,500.	4	45.	2,5
Beryllium QMV	0.067	40,000.		0.024	6.4	27.			33.	5
Copper, annealed	0.322	15,600		0.355	9.2	5.			32.	5
Glass, common	0.094	10,000.		0.244		10.				5
Magnesium, HK31A-H24	0.065	6,400.		0.35	14.	29.			37.	5
Polycarbonate	0.043	325	36.1	0.36	34.7	7.14	196,560.	12.	11	6,7
Polyurethane	0.037	1.		0.43	110				2.7	6
PVB 38 DBS	0.038	0.2		0.5	130.				3.1	6
PVB 3GH				0.17	50.				4.4	6
Silicone	0.037	0.2		0.33	184.				0.5	6
Steel, mild	0.283	29,500	30.0	0.287	8.4	30.	40.4	5.	60	2,3,4
Steel, 304 stainless	0.283	28,000	395.5	0.30	9.9	25	100.	10	90	1
Titanium 6Al 4V	0.164	16,800		0.30	4.7	120.			130.	5
Titanium B120VCA	0.175	14,800	108.8	0.30	5.2	190			200	5

Notes

1 H = EE'/(E-E') where E' is the post-yield modulus.

2 1 psi = 0.006895 MPa (1 MPa = 145.03 psi)

DATA SOURCES FOR MATERIAL PROPERTY TABLE

1. M. J. Forrestal & M. J. Sagartz, Elastic-plastic response of 304 stainless steel beams to impulse loads, *J. Appl. Mech.* **45**, 685-7, 1978.

2. N. Perrone & P. Bhadra, A simplified method to account for plastic rate sensitivity with large deformations, *J. Appl. Mech.* **46**, 811-6, 1979.

3. S. A. Meguid & J. D. Campbell, Elastic-plastic tension-torsion in a circular bar of rate-sensitive material, *J. Appl. Mech.* **46**, 311-6, 1979.

4. M. J. Manjoine, Influence of rate of strain and temperature on yield stresses of mild steel, *J. Appl. Mech.* **11**, 211-8, 1944.

5. T. Baumeister & L. S. Marks, *Standard Handbook for Mechanical Engineers*, 7ed., McGraw Hill, New York, 1967.

6. J. H. Lawrence, Jr., *Guidelines for the Design of Aircraft Windshield/Canopy Systems*, AFWAL-TR-80-3003, Wright-Patterson AFB, 1980.

7. J. T. Ryan, Impact and yield properties of polycarbonate as a function of strain rate, molecular weight, thermal history, and temperature, *Pol. Engng. & Sci.* **18(4)**, 264-7, 1978.

☆U.S. Government Printing Office 1991—648-127/62088