UCRL- 92218
PREPRINT
CONF-850862--2

DYNA3D, INGRID, and TAURUS – An Integrated, Interactive Software System for Crashworthiness Engineering

David J. Benson
John O. Hallquist
Douglas W. Stillman

This paper was prepared for submittal to the 1985 ASME Computers in Engineering Conference, Boston, MA., August 4-8, 1985.

April, 1985

Lawrence Livermore National Laboratory

DYNA3D, INGRID, and TAURUS - An Integrated, Interactive Software System for Crashworthiness Engineering

David J. Benson, John O. Hallquist, and Douglas W. Stillman
University of California, Lawrence Livermore National Laboratory
P.O. Box 808, Livermore, California 94550 U.S.A.

UCRL--92218

DE85 010928

ABSTRACT

Crashworthiness engineering has always been a high priority at Lawrence Livermore National Laboratory because of its role in the safe transport of radioactive material for the nuclear power industry and military. As a result, the authors have developed an integrated, interactive set of finite element programs for crashworthiness analysis.

The heart of the system is DYNA3D, an explicit, fully vectorized, large deformation structural dynamics code. DYNA3D has the following four capabilities that are critical for the efficient and accurate analysis of crashes: 1) fully nonlinear solid, shell, and beam elements for representing a structure, 2) a broad range of constitutive models for representing the materials, 3) sophisticated contact algorithms for the impact interactions, and 4) a rigid body capability to represent the bodies away from the impact zones at a greatly reduced cost without sacrificing any accuracy in the momentum calculations.

To generate the large and complex data files for DYNA3D, INGRID, a general purpose mesh generator, is used. It runs on everything from IBM PCs to CRAYS, and can generate 1000 nodes/minute on a PC. With its efficient hidden line algorithms and many options for specifying geometry, INGRID also doubles as a geometric modeller.

TAURUS, an interactive post processor is used to display DYNA3D output. In addition to the standard monochrome hidden line display, time history plotting, and contouring, TAURUS generates interactive color displays on 8 color video screens by plotting color bands superimposed on the mesh which indicate the value of the state variables. For higher quality color output, graphic output files may be sent to the DICOMED film recorders. We have found that color is every bit as important as hidden line removal in aiding the analyst in understanding his results.

In this paper, the basic methodologies of the programs are presented along with several crashworthiness calculations.

INTRODUCTION

The greatest difficulty in designing for crashworthiness is evaluating a proposed design. Experiments are the ultimate in accuracy, but they are also the ultimate in cost, and analytical solutions are nonexistent. Computer simulation is a necessity.

Computer hardware has been and continues to be the primary limitation on the size and sophistication of computer simulations. Early software used rigid bodies and plastic hinges, an approach that is worthless for shell structures. Later programs incorporated finite elements but the elements locked, the crash was treated as a quasistatic event, and the crude contact algorithms frequently allowed a structure to pass through itself as it buckled. Only with the advent of supercomputers have programs been developed that are sophisticated enough to accurately model the complicated phenomena involved in a crash.

One of the first finite element programs to take full advantage of vector processes, such as the CRAY-1, is DYNA3D (1), a completely vectorized program that uses explicit integration to solve three-dimensional, inelastic, large deformation structural dynamics problems. Developed by John Hallquist, it was first released in 1976 and has undergone continual development ever since. On a fully configured CRAY XMP-48, DYNA3D can handle nearly a quarter of a million elements in core at a rate of two million element cycles per minute, a figure that will jump to almost eight million once CRAY delivers its multi-tasking software.

For a complicated crash simulation, the amount of data required describe the problem to DYNA3D is enormous, and the amount of output is much, much greater. Interactive pre- and post-processors with color and hidden line graphics are a necessity. INGRID (2), developed by Doug Stillman, generates our DYNA3D meshes, and it is sophisticated enough to be used as a geometric modeller. For plotting time histories and displaying the deformed structures, we use TAURUS (3), originally developed by Bruce Brown.

These three integrated programs - DYNA3D, INGRID, and TAURUS - are the primary analysis tools used at Lawrence Livermore National Laboratory for crashworthiness design in both the weapons and nuclear power programs. In the following sections, an overview of the methods, capabilities and applications of these programs is given.

DYNA3D

A complete development of the theoretical and computational methods of DYNA3D is too long for a single paper; the interested reader can get the basics from the now dated Theoretical Manual (4), and the fine points from the appropriate references (5-10). Our emphasis, after a few paragraphs to establish notation, is on the features in the program that are of particular interest in crashworthiness design.

Our viewpoint is strictly Lagrangian. Material points are identified by their initial location, $\underline{X}$, in the undeformed body. The current position of a point, $\underline{x}(\underline{X},t)$, is a function of time and its initial location.

The principal of virtual work is the foundation of the displacement finite element method. It is easily derived from, the weak form of Cauchy's first law of motion by applying the divergence theorem.

$$\delta\pi = \int_\Omega \rho\ddot{x}_i\,\delta x_i\,d\Omega + \int_\Omega \sigma_{ij}\,\delta x_{i,j}\,d\Omega - \int_\Omega \rho f_i\,\delta x_i\,d\Omega - \int_{\Gamma_t} t_i\,\delta x_i\,d\Gamma = 0 \qquad (1)$$

where

ρ is the density.

$\underline{x}$ is the displacement.

$\underline{\sigma}$ is the Cauchy stress.

$\underline{f}$ is the body force.

$\underline{t}$ is the surface traction.

Ω is the domain of the body.

Γ_t is the section of the boundary of Ω that has tractions applied.

$\delta\pi$ is the virtual work.

The finite element method interpolates $\underline{x}$ throughout the body from its nodal values.

$$x_i = \phi(\xi,\eta,\zeta)_\alpha\, x_{\alpha i} \qquad (2)$$

where

ϕ_α is the isoparametric shape function at node α.

$\underline{x}_\alpha$ is the displacement at node α.

ξ,η,ζ are the isoparametric coordinates of a material point $\underline{X}$.

DYNA3D has isoparametric eight node brick elements, four node shell elements and two node beam elements. The interpolation methods for the shell and beam elements (11) are more complicated than Eq. (2) because of their rotational degrees of freedom. All of the elements are formulated to handle large, nonlinear deformations. We use only linear interpolation because our experience indicates that higher order elements are simply not computationally cost-effective for finite deformation analyses.

Substitution of Eq. (2) into Eq. (1) gives a set of simultaneous equations for the accelerations. The superscript n refers to the n-th integration time step.

$$M_{\alpha i\beta j}\ddot{x}^n_{\beta j} = F^n_{\alpha i} \qquad (3)$$

where

$M_{\alpha i\beta j}$ is the mass matrix.

$F_{\alpha i}$ is the sum of the internal and external forces.

We use a diagonal, lumped mass matrix in DYNA3D, which makes solving for the accelerations in Eq.(3) trivial. A lumped mass matrix also leads to more accurate answers in many situations, especially those involving shock waves.

The velocities and displacements are calculated using central difference integration, an explicit method which is second order accurate.

$$\dot{x}^{n+1/2}_{\alpha i} = \dot{x}^{n-1/2}_{\alpha i} + h^n\,\ddot{x}^n_{\alpha i} \qquad (4)$$

$$x^{n+1}_{\alpha i} = x^n_{\alpha i} + h^{n+1/2}\,\dot{x}^{n+1/2}_{\alpha i} \qquad (5)$$

$$h^{n+1/2} = \frac{1}{2}\left(h^{n+1} + h^n\right) \qquad (6)$$

where

h is the integration stepsize.

As with all explicit integration methods, the central difference method has a restriction on the size of the integration step. The Courant criterion (4) limits the time step to one small enough that a sound wave cannot cross the thinnest element in a single step. Consequently, steps on the order of a microsecond are typical. The extreme simplicity of the equations and the high cost of the reads and writes to disk necessary for solving the banded equations generated by unconditionally stable integration methods (which allow large time steps) make explicit codes like DYNA3D the only practical approach for large problems. When hundreds of millions of words of real memory become available, however, implicit codes such as NIKE3D (12) may be faster than explicit codes.

In addition to the usual features found in completely nonlinear finite element programs, our experience indicates that there are three additional capabilities that are critical to crashworthiness analysis: 1) a broad range of efficiently implemented material models, 2) sophisticated contact algorithms for the impact interactions, and 3) a rigid body capability to represent the bodies away from the impact zones at a greatly reduced cost without sacrificing any accuracy in the momentum calculations.

An explicit finite element program spends most of its time calculating the stresses in a body in order to calculate the contribution of the stress divergence to the right hand side of Eq. (2). The stresses are calculated from constitutive equations that are usually expressed in a rate form.

$$\overset{\nabla}{\sigma}_{ij} = \sigma_{ij}(\dot{\underline{\varepsilon}}) \qquad (7)$$

$$\dot{\sigma}_{ij} = \overset{\nabla}{\sigma}_{ij} + \sigma_{ik}\,\Omega_{kj} + \sigma_{jk}\,\Omega_{ki} \qquad (8)$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \dot{x}_j}{\partial x_i} - \frac{\partial \dot{x}_i}{\partial x_j}\right) \qquad (9)$$

where

$\overset{\nabla}{\underline{\sigma}}$ is the Jaumann stress rate.

$\dot{\underline{\varepsilon}}$ is the strain rate.

$\underline{\Omega}$ is the spin tensor.

The Jaumann stress rate, as opposed to the simple time derivative of stress, is used in Eq. (7) to account for the changes in the stress caused by rigid rotations of the body. For kinematic hardening we use the computationally more expensive Green-Naghdi stress rate in order to avoid the well known oscillatory behavior in pure shear (13).

The implementation of the constitutive relations is every bit as important as the choice of the constitutive relations. Equations (7) through (9) are evaluated literally millions of times during each analysis. The canonical Prandtl-Reuss elastoplasticity model with isotropic and kinematic hardening is a good example of the difficulties associated with creating an efficient implementation of a constitutive relation. In this model, a material behaves elastically if its stress is interior to the cylinderical region described by its yield surface, ϕ, and plastically if the stress is on the yield surface and trying to get outside of it.

$$\phi = \frac{1}{2}(S_{ij} - \alpha_{ij})(S_{ij} - \alpha_{ij}) - \frac{\sigma_y^2}{3} = 0 \qquad (10)$$

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\sigma_{kk} \qquad (11)$$

where

$\underline{S}$ is the deviatoric stress tensor.

$\underline{\alpha}$ is the back stress.

σ_y is the yield stress.

The values of σ, α, and σ_y are determined by integrating over time a set of highly nonlinear differential equations involving the strain rate subject to the constraint that ϕ must always be less than or equal to zero. Many methods have been proposed over the years, including complicated and expensive subincremental methods, but only the radial return method (8), which we use exclusively in DYNA3D, has proven to be efficient enough for large scale calculations. Furthermore, it is usually more accurate than other more expensive methods (14).

Crashes, by definition, involve contact between surfaces; any program used for crashworthiness analysis must have a variety of sophisticated contact/impact algorithms. The first involves two arbitrary surfaces, such as a bumper hitting a barrier, where the surfaces deform and undergo large relative displacements. The second type is single surface contact which occurs when a surface folds over on to itself, e.g., the controlled crush structural members of a car. In DYNA3D, both types of contact are easily modelled, including, friction and automatic closure and separation, by using the penalty method (5). Although Lagrange multipliers are theoretically attractive, their performance is poor. When a node penetrates a surface, a force proportional to the penetration is applied normal to surface on the penetrating node and the reaction force is distributed to the four nodes defining the appropriate surface segment. A friction force based on the normal force is also applied if it is requested.

The most expensive part of contact algorithms is the search algorithm for determining the contact points, and not, as might first be expected, the actual calculation of the contact force. Efficient and reliable search algorithms (5) have been developed for DYNA3D over the years; virtually every production analysis performed with DYNA3D uses the contact/impact algorithms, therefore any flaws in new versions of the algorithms are quickly exposed and corrected.

Momentum is usually the driving force in crashes. Any moving structure must be modelled in its entirety, no matter how small the impact area, in order to assure the accuracy of the momentum calculations. Away from the impact zones, we desire the cheapest possible representation of the body that accurately models the body's inertial properties so that the cost of the analysis is minimized. To that end, David Benson and John Hallquist implemented a rigid body material type in DYNA3D (6) based on the earlier theoretical work by Benson for his thesis (15). Each finite element is assigned a material number in the data. All of the elements that share a common material number, where that material is specified to be rigid, define a rigid body. Separate rigid bodies with different material numbers can also be merged to form a single rigid body. In addition to all of the standard boundary conditions, contact surfaces, and body forces acting on the bodies, joint constraints, such as universal joints, are used to tie bodies together.

Efficiency is emphasized repeatedly in the previous paragraphs because dynamic crashworthiness calculations are inherently the most expensive kind of structural calculations. Dynamics requires thousands to hundreds of thousands of timesteps, and the mesh must be very fine throughout the regions involving impact and buckling, all of which leads to extremely large computational costs. The supercomputers of today are marginal at best for sophisticated crashworthiness calculations. This implies that not only must the algorithms be inherently efficient, but that their actual FORTRAN implementation must be vectorized so that all of a super computer's potential is realized. John Hallquist found in 1979 that vectorizing DYNA3D achieved speed increases ranging from four to fifty over the speed of the original code. Vectorization makes the difference between being able to run large problems and not getting the job done.

INGRID

Mesh generation is an area that has been largely ignored in the finite element literature. Pick up any of the popular textbooks on finite element theory and try to find much more than a small section on mesh generation methods. Most mesh generators are, consequently, much less sophisticated than the analysis software. This lack of sophistication lead Doug Stillman to develop INGRID (2), an interactive, mesh generator.

INGRID describes parts of a finite element model using several methods. The most useful method is a variation on logically regular mesh generation which has been designated index progressions. The index progression allows the analyst to describe most geometries including nodes and elements with roughly the same amount of input as a solids modeller. Boundary conditions, loads, and slide surfaces are located by the same index progressions. Using these methods most parts can be completely generated with fewer than twenty commands.

When all of the parts of a model are generated they must be copied, moved, scaled, rotated, distorted, mapped, or reflected to put them in their proper location relative to the system. Frequently the same copy operation must be performed on several parts so INGRID permits parts to be grouped to form subsystems which can then be copied or moved.

One of the most important advantages of INGRID is virtually any shape can be meshed without having to resort to wedge elements, tetrahedrons, triangular elements, or highly distorted quadrilateral and hexahedral elements. INGRID can, however, generate meshes with such elements if they are desired. These elements are known to be overly stiff or, at best, to be simply inaccurate. Even a few wedge-shaped elements can ruin a crashworthiness analysis by being overly stiff and therefore changing the buckling modes.

INGRID efficiently generates large and complicated meshes because of extensive optimization of algorithms for three-dimensional graphics, surface intersections, mapping, data manipulating, and coincident node removal. Fine meshes are required for accurate analyses; we routinely generate meshes with thirty thousand elements, and meshes with more than one hundred thousand have been generated interactively.

TAURUS

Nonlinear finite element solutions generate such an overwhelming amount of output that tabular output is worthless. We examine the results of a calculation by using TAURUS (3), an interactive graphics postprocessor. The program has two basic modes of displaying output: 1) plotting the deformed geometries, or 2) time history plots.

Deformed geometries are plotted using the hidden line software written by Michael Archuleta in 1973, which is based on Watkin's algorithm. MOVIE.BYU (16), released in 1975, incorporated, with improvements by Bruce Brown, Archuleta's software, making TAURUS and MOVIE.BYU distant relatives. Depending on the graphics device, either contour lines or color fringes are used to display any of the over one hundred element variables (given in Table 1) on the deformed geometry. In our experience, color fringes are as much an improvement over contour lines as hidden line removal is over wire frame graphics.

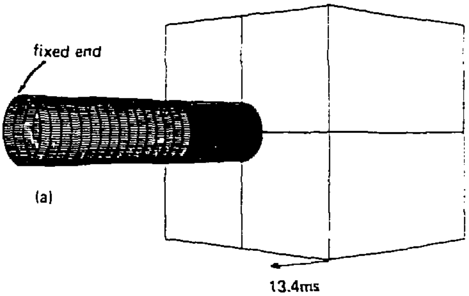
The time history plots of the element, material, nodal and global variables (Tables 1 through 4) are useful to analysts studying the dynamic behavior of structures. While color fringing gives an overall view of the behavior of a structure, at a given time, a time history plot gives a detailed, continuous picture of the important variables over the entire analysis, therefore the two approaches to displaying information are complementary rather than equivalent.

Efficiency is as important for TAURUS as it is for DYNA3D and INGRID. All of the variables for hundreds of output steps, with tens of thousands of elements, must be handled efficiently to prevent the cost of postprocessing from being a significant part of the overall analysis cost. Through the early contributions of Archuleta and Brown, and the more recent work by Hallquist, analysts use TAURUS to evaluate their output with the same speed and lack of concern about cost as they use text editors on their data.

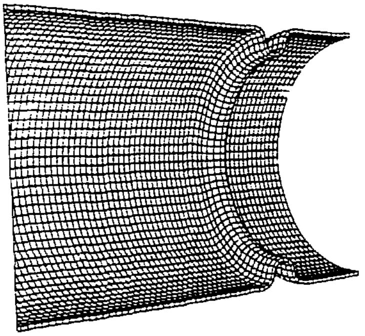
EXAMPLES

Collapse of a Steel Cylinder

One method for limiting the forces transmitted to occupants of a vehicle involved in a crash is by the use of collapsible structural members. One such structural member, shown in Fig. 1, is impacted by a large mass at 13.4 m/sec (30 miles/hour).



(a)



(b)

Fig. 1 (a) Initial configuration of cylinder.
(b) Closeup view showing cross section: One quarter of the cylinder and mass were modeled with symmetric boundary conditions.

The length is 440mm, the diameter of the circular cross section is 100mm, and the wall thickness is 1.5mm. An indentation about the circumference initiates the buckling response. Our mesh consist of 1980 shell elements with five integration points through the thickness. Though more costly per element cycle than solid elements by a factor of 3 to 4, fewer elements are required and a larger time step is permitted thereby cutting overall cost by at least a factor of five over the cost of using solid elements with four elements through the thickness.

The deformation at various times is shown in Fig. 2. Our last computed state is at 10ms and required nearly 12.7 CPU hours on the CRAY-1. We have used our contact capability to keep the surfaces from penetrating. Three interfaces were defined. One was defined between the large mass and the member, and the other two consisted of the inner and outer surfaces of the pipe. Our single-sided surface contact algorithm keeps all nodes on a surface from penetrating through the same surface. Figure 3 compares the initial and final shapes.

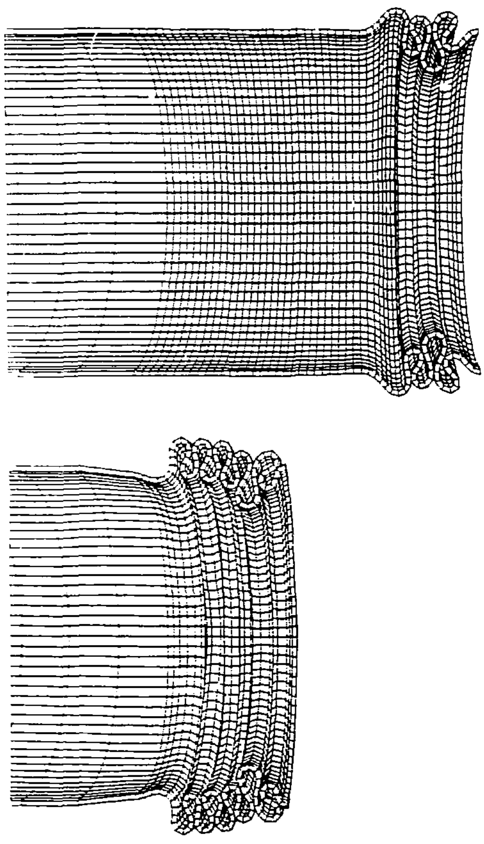


Fig. 2 Deformed cross sections at 5 and 10ms.

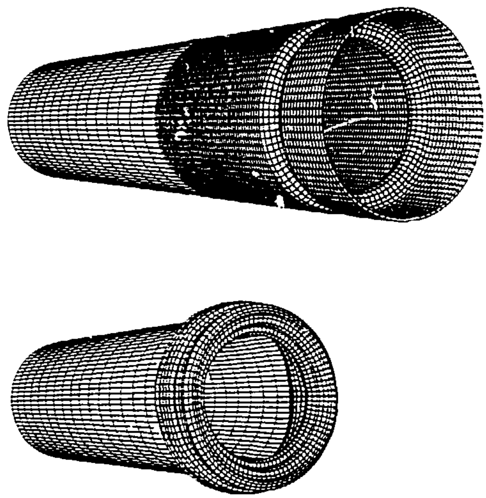


Fig.3 Comparison of initial and final shapes.

Impact of a Earth Penetrator into a Tree

In Fig. 4, an earth penetrator 30cm in diameter is shown just prior to impact with a tree 46cm in diameter. The axis of the penetrator is tangent to the outer diameter of the tree. An initial velocity of 750m/s is assumed for the earth penetrator.

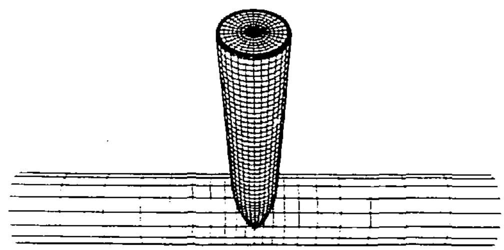


Fig. 4 Earth penetrator at time of impact with tree.

The purpose of the calculation is to determine the angular rotation of projectile prior to ground impact. We therefore modeled the projectile as a rigid body with just 6 degrees of freedom. We modeled the tree as an elastic-plastic material under the assumption that the results would primarily depend on the mass density and not necessarily the constitutive model.

Results are shown in Fig. 5 at 800 μs intervals.

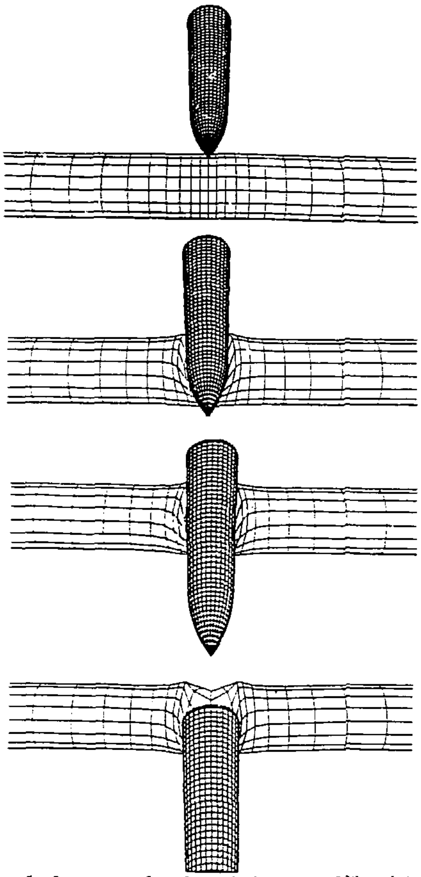


Fig. 5 Sequence of deformed shapes at 800μs intervals.

Calculation time with an elastic projectile was 4 hours, but with the rigid body projectile, calculation time was approximately 6 minutes of CPU on the CRAY-1. Most of the CPU was required for the interface treatment between the projectile and tree. We found that a sufficiently large rotation occurred as to make subsequent penetration into the ground improbable.

Collapse of a Frame Member

This model demonstrates the use of the single sided slide surface for nonaxisymmetric collapse. The frame member is a hollow square box with members attached to the front and back. The system impacts a stone wall. Three single side slide surfaces are required for this problem. The first is inside the beam to keep it from collapsing through itself. Two more are used on either side of the member. Two more slide surfaces are required to connect the front and back masses to the beam. Figure 6 shows the initial configuration and the final deformed shape.

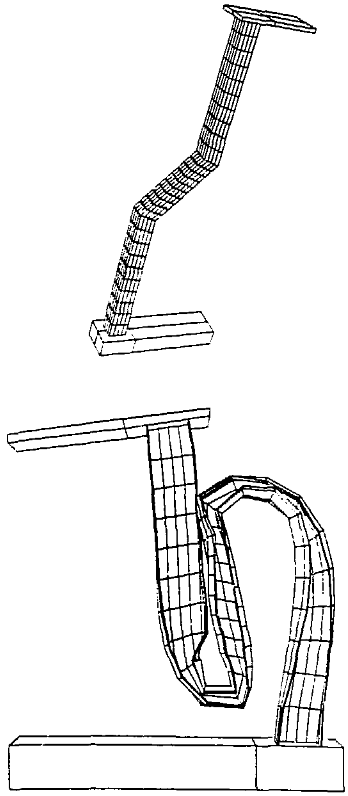


Fig. 6 Collapse of a frame member.

SUMMARY

An overview, with examples, of INGRID, DYNA3D, and TAURUS was presented. They form the integrated software system currently used at Lawrence Livermore National Laboratory for crashworthiness analysis and, nonlinear structural dynamics in general. Mesh generation and postprocessing are no longer the significant hurdles they were ten years ago. The analysis phase, which was impossible, is now feasible only on supercomputers. We are optimistic that the cost will be reduced substantially in the next five years by faster computers, and to a lesser extent, more efficient algorithms.

Element Type	No.	Component
solid elements (2D&3D) stress components	1	x
	2	y
	3	z
	4	xy
	5	yz
	6	zx
	7	effective plastic strain
	8	pressure or average strain
	9	von Mises
	10	1st principal deviator max.
	11	2nd principal deviator
	12	3rd principal deviator min.
	13	maximum shear
	14	1st principal maximum
	15	2nd principal
	16	3rd principal minimum
hexahedrons membranes plates shells	17	x-displacement
	18	y-displacement
	19	z-displacement
	20	maximum displacement
	21	x-velocity
	22	y-velocity
	23	z-velocity
	24	maximum velocity
	25	temperature (TACO3D)
membranes plates shells	26	M_{xx} bending resultant
	27	M_{yy} bending resultant
	28	M_{xy} bending resultant
	29	Q_{xx} shear resultant
	30	Q_{yy} shear resultant
	31	N_{xx} normal resultant
	32	N_{yy} normal resultant
	33	N_{xy} normal resultant
	34	surface stress $N_{xx}/t+6M_{xx}/t^2$
	35	surface stress $N_{xx}/t-6M_{xx}/t^2$
	36	surface stress $N_{yy}/t-6M_{yy}/t^2$
	37	surface stress $N_{yy}/t+6M_{yy}/t^2$
	38	surface stress $N_{xy}/t-6M_{xy}/t^2$
	39	surface stress $N_{xy}/t+6M_{xy}/t^2$
	40	effective upper surface stress
	41	effective lower surface stress
	42	maximum effective surface stress

TABLE 1. Component numbers for element variables. By adding 100, 200, 300, and 400 to component numbers 1 through 16 component numbers for infinitesimal strains, Green-St. Venant strains, Almansi strains and strain rates are obtained, respectively.

No.	Component
1	x-displacement
2	y-displacement
3	z-displacement
4	x-velocity
5	y-velocity
6	z-velocity
7	x-acceleration
8	y-acceleration
9	z-acceleration
10	temperature (TACO3D)

TABLE 2. Component numbers for nodal time history plots.

No.	Component
1	x-rigid body displacement
2	y-rigid body displacement
3	z-rigid body displacement
4	x-rigid body velocity
5	y-rigid body velocity
6	z-rigid body velocity
7	x-rigid body acceleration
8	y-rigid body acceleration
9	z-rigid body acceleration

TABLE 3. Component numbers for material time history plots.

No.	Component
1	x-rigid body displacement
2	y-rigid body displacement
3	z-rigid body displacement
4	x-rigid body velocity
5	y-rigid body velocity
6	z-rigid body velocity
7	x-rigid body acceleration
8	y-rigid body acceleration
9	z-rigid body acceleration
10	kinetic energy

TABLE 4. Component numbers for global variable time history plots.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-Eng-48.

REFERENCES

1. Hallquist, J. O., "Preliminary User's Manuals for DYNA3D and DYNAP (Nonlinear Dynamic Analysis of Solids in Three Dimensions), University of California, Lawrence Livermore National Laboratory, Rept. UCID-17268, 1976 and Rev. 1, 1979.

2. Stillman, D. W., "INGRID: A Three-Dimensional Mesh Generator Based on an Expanded Index Notation," Thesis, The University of Tennessee, 1981.

3. Brown, B. E., Hallquist, J. O., "TAURUS: An Interactive Post-Processor for the Analysis Codes NIKE3D, DYNA3D, TACO3D, and GEMINI," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19392, 1982, and Rev. 1, 1984).

4. Hallquist, J. O., "Theoretical Manual for DYNA3D," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19401, 1982.

5. Hallquist, J. O., Goudreau, G. L., and Benson, D. L., "Sliding Interfaces with Contact-Impact in Large Scale Lagrangian Computations," Proceedings of the Third International Conference on Finite Elements in Nonlinear Mechanics (FENOMECH'84), 1984, Stuttgart, W. Germany.

6. Benson, D. J. and Hallquist, J. O., "A Simple Rigid Body Algorithm for Structural Dynamics Programs - Part I," Proceedings of the Conference on Numerical Methods in Engineering: Theory and Applications, 1985, University College Swansea, United Kingdom.

7. Goudreau, G. L. and Hallquist, J. O., "Recent Developments in Large Scale Finite Element Lagrangian Hydrocode Technology," Journal of Computer Methods in Applied Mechanics Engineering, Vol. 30, 1982. Presented at FENOMECH'81, Stuttgart, W. Germany.

8. Krieg, R. D. and Key, S. W., "Implementation of a Time Dependent Plasticity Theory into Structural Computer Programs," Vol. 20 of Constitutive Equations in Viscoplasticity: Computational and Engineering Aspects, American Society of Mechanical Engineers, New York, N.Y., 1976, pp. 125-137.

9. Hughes, T.J.R., "Stress-Point Algorithm for a Pressure-Sensitive Multiple-Yield-Surface Plasticity Theory," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19339, 1982).

10. Steinberg, D. J. and Guinan, M. W., "A High-Strain-Rate Constitutive Model for Metals," University of California, Lawrence Livermore National Laboratory, Rept. UCRL-80465, 1978.

11. Hughes, T.J.R. and Liu, W. K., "Nonlinear Finite Element Analysis of Shells: Part I. Three-Dimensional Shells," Journal of Computer Methods in Applied Mechanics and Engineering, Vol. 26, 1981, pp. 331-362.

12. Hallquist, J. O., "NIKE3D: An Implicit Finite-Deformation, Finite Element Code for Analyzing the Static and Dynamic Response of Three-Dimensional Solids," University of California, Lawrence Livermore National Laboratory, Rept. UCID-18822, 1981 and Rev. 1, 1984.

13. Johnson, G. C. and Bammann, D. J., "A Discussion of Stress Rates in Finite Deformation Problems," Rept. SAND-82-8821, 1982, Sandia National Laboratories.

14. Krieg, R. D. and Krieg, D. B., "Accuracies of Numerical Solution Methods for the Elastic-Perfectly Plastic Model," Journal of Pressure Vessel Technology, Vol. 99, ASME, No. 4, 1977, p. 510.

15. Benson, D. J., "The Simulation of Deformable Mechanical Systems Using Vector Processors," Dissertation The University of Michigan, 1983.

16. Christiansen, H. N. and Stephenson, M. B., "MOVIE.BYU - A General Purpose Computer Graphics Display System," Proceedings of the Symposium on Applications of Computer Methods in Engineering, University of Southern California, Los Angeles, Vol. 2, Aug. 1977, pp. 759-769.