EVALUATION OF NONLINEAR STRUCTURAL DYNAMIC RESPONSES
USING A FAST-RUNNING SPRING-MASS FORMULATION*A. S. Benjamin, B. S. Altman, and J. D. Gruda
Sandia National Laboratories, Albuquerque, New Mexico, U.S.A.

1. INTRODUCTION

It has been a fond desire of structural analysts to be able to calculate the dynamic responses of very large, highly nonlinear systems without expending a year's allocation of computer funds with every run. In today's world, accurate finite-element simulations of large nonlinear systems may require meshes composed of hundreds of thousands of degrees of freedom. Even with today's fast computers and the promise of ever-faster ones in the future, central processing unit (CPU) expenditures for such problems could be measured in days. Many contemporary engineering problems, such as those found in risk assessment, probabilistic structural analysis, and structural design optimization, cannot tolerate the cost or turnaround time for such CPU-intensive analyses, because these applications require a large number of cases to be run with different inputs. For many risk assessment applications, analysts would prefer running times to be measurable in minutes. There is therefore a need for approximation methods which can solve such problems far more efficiently than the very detailed methods and yet maintain an acceptable degree of accuracy. For this purpose, we have been working on two methods of approximation: neural networks and spring-mass models.

Neural network models differ from spring-mass models in that they do not solve the governing physical response equations of the system. Instead, they rely completely upon the calibration, or training, of parameters in the network to achieve agreement with the very detailed finite-element models. Our efforts at present are focused on recurrent neural networks, which simulate the response of the system in a time marching sense, by using the values of the response variables at the current time as inputs to determine the values one time step in the future.

Spring-mass models, on the other hand, solve a set of differential equations that result from a physical representation that is an analogue of the real system. The equations are simpler in form than those employed for finite-element analysis, because they consist of vector rather than tensor variables. Since the models are physically based, they do not require nearly as much training as neural network models. However, spring-mass models have limitations that cause some of the responses of the system to be less accurately calculated than others.

This paper presents our work and results to date for spring-mass modeling and analysis, since we are further along in this area than in the neural network formulation. It describes the physical and numerical models contained in a code we developed called STRESS, which stands for "Spring-mass Transient Response Evaluation for Structural Systems" [1]. The paper also presents results for a demonstration problem, and compares these with results obtained for the same problem using PRONTO3D [2], a state-of-the-art finite element code which was also developed at Sandia.

2. MODELING CONCEPTS

2.1 Springs and Masses

STRESS solves the equations of motion for a set of masses connected by springs that can have nonlinear stiffnesses. The analyst starts from a geometric representation of a structural system, such as the "patch" and "hyperpatch" geometry produced by the code PATRAN [3]. From this geometry, he or she creates a finite element mesh that is typically an order of magnitude coarser than that required for finite-

* This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

element analyses. STRESS converts the mesh to a spring-mass array by placing point masses at the vertices of all the elements, then connecting all the masses within each element by springs. A schematic of this process is shown in Figure 1.

Figure 1. Schematic Illustrating Spring-Mass Modeling Concepts in the STRESS Code.

The originating finite-element mesh can consist of both brick (hexahedral) elements, as in Figure 1a, and shell (quadrilateral) elements, as in Figure 1b. For each shell element, STRESS creates two virtual hexahedral elements to simulate the shell thickness before allocating springs and masses. During the subsequent deformation, the shell is treated as rigid in its transverse direction, so that the thickness remains constant. This treatment relaxes the Courant condition on the time step, which would normally be very restrictive because of the very small mass-to-mass spacing across the shell. It therefore becomes possible to account for bending of the shells utilizing relatively large time steps without incurring instabilities.

2.1 Equations of Motion

The velocities and displacements for each mass are evaluated by the following equations, which relate accelerations to forces according to Newton's law:

$$\frac{d\vec{U}_i}{dt} = -\frac{1}{M_i} \sum_{j=1}^{n_j} \bar{\Lambda}_{i,j} \sum_{m=1}^{(n_m)_{i,j}} \sigma_{i,j,m} A_{i,j,m} \quad (1)$$

and

$$\frac{d\vec{X}_i}{dt} = \vec{U}_i \quad (2)$$

As mentioned earlier, these are vector equations in which the variables $\vec{U}_i$, $\vec{X}_i$, and $\bar{\Lambda}_{i,j}$ have components in the three coordinate directions. They represent, respectively, the velocity and displacement components of mass i and the direction cosines for the spring that connects mass i with mass j . The variables $\sigma_{i,j,m}$ and $A_{i,j,m}$ are respectively the stress (tension positive) in the spring connecting masses i and j for material m and the area over which it acts. Lastly, M_i is the mass.

Because many of the springs are defined to lie on the boundaries of the elements in the original mesh, a single spring can be shared by as many as four neighboring elements. Because each element can be composed of a different material, any spring connecting two masses, i and j , may be subdivided into as many as four springs of different material. The number of materials comprising a composite spring is denoted by the symbol $(n_m)_{i,j}$ in Equation (1).

2.2 Stresses and Areas

The stresses are evaluated from constitutive relationships that are similar to those used in finite-element codes. Strains utilized in the constitutive model are the fractional changes in length of the springs relative to their initial lengths. In STRESS, the following three constitutive models are employed:

- (1) A perfectly elastic model. The only input is the elastic modulus.
- (2) An elastic-plastic model with nonlinear strain hardening. Inputs are the elastic modulus, yield stress, hardening modulus, hardening exponent, and relaxation factor. The latter represents the fraction of isotropic versus kinematic hardening.
- (3) A generalized stress-strain model with nonlinear strain hardening. Inputs are a user-provided table of stresses versus strains, the elastic modulus, and the relaxation factor.

The areas over which the springs act are evaluated from the dimensions of the finite elements in the original mesh. The following equation is used:

$$A_{i,j,m} = \frac{3C}{n_s(n_F)_{i,j}} \sum_{k \in F_{i,j,m}} A_k |\cos(\theta_k)_{i,j}| \quad (3)$$

Referring to Figure 2, k represents any face of an element of the original mesh, and $F_{i,j,m}$ represents the set of such faces that (1) reside in an element that contains the spring and that is composed of material m , and (2) contain one of the end points of the spring, either i or j . As shown in the figure, A_k is the area of face k and $(\theta_k)_{i,j}$ is the angle between the spring (i,j) and the normal to the face, $\vec{n}_k$. The index n_s is the number of springs in an element, namely 28; each element contains 12 springs that define the edges of the element, 12 that lie on the diagonals of the faces of the element, and 4 that lie on the body-centered diagonals. The index $(n_F)_{i,j}$ is the number of faces of an element that attach to the spring, i.e., 2 for edge springs, 4 for face-diagonal springs, and 6 for body-diagonal springs. The constant C is user-provided, and is the same for all springs. It is generally set between 0.5 and 1.0. The higher value gives the correct area for uniaxial loading of an infinitesimally thin rod of square cross-section. Lower values tend to correct for artificial stiffness that can be caused by the coarse meshing.

Figure 2. Schematic Illustrating Determination of Areas A_k and Angles $(\theta_k)_{i,j}$ in the STRESS Code.

The use in Equation (1) of stresses based on constitutive relationships and areas based on cell dimensions differentiates STRESS from earlier spring-mass codes, such as SHOCK [4]. Those codes relied on load-deflection tables provided by the analyst for each spring. Normally, the analyst obtained suitable load-deflection data from the results of static analyses, but the process was laborious and it was necessary to limit the number of masses and springs to a small number. The present method requires only one user-determined constant, namely C , which has the same value across the mesh.

2.3 Targets and Contact Surfaces

Deformation is initiated when the deformable body strikes a target whose shape is specified by the user. Currently, the target is assumed to be rigid. Equations (1) and (2) are solved for each mass, subject to the condition of impenetrability of the target, using an explicit, second-order-correct, time-marching process.

Contact surfaces can be specified between pairs of material blocks that are composed of hexahedral elements, but not shell elements. A current simplification in the algorithm is that when physical contact occurs between two material blocks, they are assumed to move as a single rigid body. If they should subsequently separate, they are again treated as deformable bodies.

Because the target and contact algorithms currently used in STRESS are simplistic compared to a number of existing finite element codes, including PRONTO3D, a more sophisticated approach is being developed to replace them. This effort is part of the continuing development of the code.

2.4 Element Death

In applications where plastic strains are very high, it is important to account for the fact that materials rupture. The effects of local rupturing on subsequent responses in other parts of the system can be major.

Recently, an element death option was added to the PRONTO finite element code to simulate rupturing of materials. (We use the acronym PRONTO, hereafter, to represent the code PRONTO3D.) For each material, the analyst provides an input for the maximum sustainable value of a response variable. For ductile materials, a commonly used response variable is the equivalent plastic strain, or EQPS. In a finite element, the EQPS is roughly proportional to the root mean square of the differences in the directional plastic strains along each of the three principal axes. When the calculated EQPS exceeds the input maximum, PRONTO makes the element "disappear" and apportions its mass to the neighboring elements.

STRESS has an analogous model for material death, in which the analyst provides an input for each material denoting the maximum plastic strain that a spring made of that material can sustain. The user can choose either of two options. In the first, each spring breaks independently when the maximum plastic strain is exceeded. In the second, both the spring itself and all other springs that are attached to either of the masses at the ends of the first spring are assumed to break. The second option is provided to simulate the modeling in PRONTO, where exceedance of the maximum EQPS results in omnidirectional rupturing.

For the death option, we assume that the directional plastic strain in a spring is the same as the equivalent plastic strain in a finite element. Thus, we set the maximum plastic strains for the springs equal to the maximum EQPS values used in PRONTO.

2.5 Postprocessing

The STRESS code produces an EXODUS [5,6] database, which is fully compatible with the utilities that are available in the Sandia Engineering Analysis Code ACCESS System, SEACAS [7]. For example, we can use the code BLOT [8] to display deformations and other responses calculated by STRESS.

In addition, we have written a postprocessor called READEXO, which calculates a set of outputs that are useful for risk assessment. Most significant of these is the calculation of accelerations from displacements using a specified time increment, Δt . The equation for this time-averaged acceleration is as follows:

$$\bar{A}_i(t) = \frac{\bar{X}_i(t + \Delta t) + \bar{X}_i(t - \Delta t) - 2\bar{X}_i(t)}{(\Delta t)^2} \quad (4)$$

where $\bar{X}_i(t)$ is the cumulative vector displacement of mass i at time t . Calculation of accelerations in this manner has the benefit of simulating the accelerations often applied in dynamic impact experiments. A typical loading pattern used to test components, for example, consists of a roughly triangular acceleration pulse applied over a period of 0.3 ms or higher. Component failure thresholds, correspondingly, are often expressed in terms of average accelerations experienced over such a time increment. Use of Equation (4) with $\Delta t = 0.3$ ms, therefore, provides a convenient means for comparing responses to experimentally determined component failure thresholds.

3. COMPARISON OF RESULTS FOR A DEMONSTRATION PROBLEM

3.1 Problem Description

To test the STRESS code relative to PRONTO, we set up a sample problem representative of the types of systems we consider at Sandia. The outer part of the system is an aeroshell which is in the shape of a conical frustum (see Figure 3a). The aeroshell consists of a heat shield material and a metallic substrate. Inside the aeroshell are several blocks of components. One of these (the only one shown in the figure) is composed of a cylindrical metal housing inside of which there are 4 small components that are partially encapsulated in foam. The housing is attached by 8 bolts to the flange of a baseplate and to a supporting structure that is part of the aeroshell substrate.

We initially performed analyses for this configuration using PRONTO with a mesh consisting of about 25,000 finite elements [9]. In this analysis, the vertical midplane was assumed to be a plane of symmetry so that only half of the system had to be modeled. The heat shield and components were meshed with hexahedral brick elements and the substrate, housing, baseplate, and most of the supports and attachments were treated as shells. Each bolt was modeled as a small brick element connecting appropriate shell elements in the housing, flange, and support structure.

In preparation for using STRESS, we created an array of about 3,900 masses by utilizing the PATRAN geometry file that was developed for the PRONTO analysis. To enable consideration of nonsymmetric loading, we developed a full 3-D model and did not assume planar symmetry. We meshed all of the components and shells much more coarsely than in the PRONTO analysis, and to avoid very small elements, we treated the housing as being directly attached to the baseplate flange and support structure rather than connected by bolts. We took this attachment to fail when the plastic strain in the baseplate flange reached 0.06, in order to achieve agreement with PRONTO on the timing of failure for the bolts. In Equation (3), we set $C = 0.5$ as per the discussion in Section 2.2. The total number of springs in the STRESS model was approximately 70,000.

We ran the PRONTO and STRESS codes for a variety of cases involving lateral impacts at various velocities upon a narrow rail and upon an infinitely wide flat plate. For cases involving the rail, the location of impact was taken to be near the location of the 4 components (see Figure 3a). For cases involving the flat plate, the surface of the plate was oriented parallel to the bottom surface of the aeroshell, but the direction of velocity was perpendicular to the body axis. Also, the forward and aft vertical shells were removed, so that the impact occurred simultaneously along the length of the aeroshell.

The principal differences between the PRONTO model and the STRESS model may be summarized as follows:

- (1) The PRONTO model was based on finite elements, whereas STRESS utilized springs and masses;
- (2) Meshing for the PRONTO model was much finer than for the STRESS model;
- (3) The PRONTO model allowed for sliding contacts between bricks and/or shells, whereas STRESS at present can handle only fixed contacts (i.e., contacts with an infinite friction coefficient), and these can occur only between brick elements;
- (4) Much of the aft two-thirds of the body was modeled as rigid in the PRONTO analysis, whereas the whole body was treated as deformable in STRESS;
- (5) Planar symmetry was assumed for the PRONTO model but not for the STRESS model. The assumption of planar symmetry is consistent with the geometries and loadings considered in the demonstration problem.

3.2 Results and Discussion

Comparisons of PRONTO and STRESS results for several cases are shown in Figures 3 and 4 and in Table 1. Figure 3(b) compares the deformations of the aeroshell, the housing, and the 4 components at a particular time during an impact into the rail at 24.4 m/s (80 ft/s). Figure 3(c) depicts the average resultant accelerations of the components as a function of time for the same rail calculation. Figure 4 presents analogous results for a lateral impact on a flat plate at 9.1 m/s (30 ft/s). Table 1 provides the peaks of the accelerations plotted in Figures 3(c) and 4, as well as for two additional cases not shown in the figures. The acceleration peaks are the most relevant measure of whether the components would survive or fail.

The acceleration for a component, c , was evaluated from the following equation:

(a) Undeformed Body

(b) Deformations at 4.0 ms

(c) Accelerations

Figure 3. Comparison of Deformation and Acceleration Results for Lateral Impact on a Stationary Rail at 24.4 m/s (80 ft/s).

Figure 4. Comparison of Accelerations for Lateral Impact on a Semi-Infinite Flat Plate at 9.1 m/s (30 ft/s).

Table 1. Comparison of Peak Average Component Accelerations for Various Impact Scenarios.

Impact Scenario	Component			
	1	2	3	4
Flat Plate @ 12.2 m/s (40 ft/s)				
PRONTO	610	660	620	570
STRESS	480	540	800	670
Difference	- 21%	-18%	+29%	+18%
Flat Plate @ 9.1 m/s (30 ft/s)				
PRONTO	660	740	870	860
STRESS	450	490	790	700
Difference	- 32%	-34%	-9%	-19%
Rail @ 24.4 m/s (80 ft/s)				
PRONTO	8230	5840	3350	3360
STRESS	9580	5240	1280	1360
Difference	+16%	-10%	-62%	-60%
Rail @ 14.3 m/s (47 ft/s) (Aeroshell Removed)				
PRONTO	4770	3440	2950	2910
STRESS	5320	1130	780	970
Difference	+12%	-67%	-74%	-67%

$$\|\bar{A}_c(t)\| = \frac{1}{n_{iec}} \sum_{i \in c} \sqrt{\bar{A}_i(t) \cdot \bar{A}_i(t)} \quad (5)$$

where the acceleration of each mass, $\bar{A}_i(t)$, was calculated using Equation (4) with $\Delta t = 0.3$ ms. The symbol n_{iec} represents the number of masses contained within the component c . It may be noted that some researchers prefer to use a mass-averaged acceleration in place of the nodal averaging performed in Equation (5).

The calculations with STRESS were performed both on a Silicon Graphics workstation and on a Cray-YMP supercomputer. The computational time step was $1.2 \mu\text{s}$ for the calculations in Figure 3 and $2.0 \mu\text{s}$ for those in Figure 4. These values were found by experimentation to produce no instabilities and to provide adequate time resolution. The time steps in PRONTO, by comparison, varied during the course of each run, ranging between $1.0 \mu\text{s}$ and $0.01 \mu\text{s}$. CPU expenditures for the STRESS runs were about 3 to 5 hours per case on the workstation and about 40 minutes to an hour per case on the Cray. Because of the model size, the calculations with PRONTO had to be performed on the Cray, and a corresponding run required about 30 hours of CPU time (a factor of about 30 to 50 greater than the STRESS calculations).

As illustrated in Figure 3(b), the STRESS model was able to simulate PRONTO results quite well for the deformation and rupturing of the aeroshell, the deformation of the housing, and the relative displacements of the inside components. The figure shows, however, that there was a tendency for relatively more rotation of components to occur with the STRESS model. This difference is at least partly attributable to the following two factors. First, the STRESS model used a fairly coarse meshing of the shells that support the components, leading to sharper angularities in the shell deformations under compressive loads. Second, the contact algorithm currently in STRESS did not allow the contacting surfaces to slide over one another. In addition to these factors, which could lead to overrotation in the STRESS computation, there could be factors leading to underrotation in the PRONTO computation (such as the use of a rigid body approximation). We have not completed our investigation of this question.

Table 1 illustrates that for the flat plate impacts, peak accelerations of the various components were predicted fairly well by the STRESS code. Differences between PRONTO and STRESS for the peak values of the various components varied from -34% to +29% for the time period examined. This result is considered to be favorable because accelerations, being the second derivative of displacements, tend to be highly sensitive to modeling subtleties. This sensitivity is evidenced in Figure 4, where it is apparent that there were significant differences between the models in the overall temporal response and timing of the peaks. For the intended applications of STRESS, the temporal differences are not as important as the magnitudes of the peaks.

The results for the rail impacts were more mixed, as illustrated in Table 1 and Figure 3(c). On the positive side, the STRESS model predicted the magnitudes and timing of the acceleration peaks for Component 1 very well (less than 20% difference in the peak magnitudes). Since this component experienced the highest acceleration of all the components, that result is a very favorable one. (The results that are considered favorable are highlighted by a box in Table 1.) On the negative side, some of the peak accelerations for the other components tended to be significantly lower with STRESS than with PRONTO. The differences between the PRONTO and STRESS models in the predictions for the rail impacts are believed to result primarily from the greater rotational energy imparted to the components in the STRESS model after the first acceleration peak. As mentioned above, we are in the process of investigating why this occurs. We believe, however, that finer meshing of some of the internal shells would resolve some of the differences.

4. CONCLUSIONS

The STRESS code was developed to enable structural analysts to estimate the nonlinear responses of complex systems to dynamic loading at a fraction of the cost of a detailed finite-element analysis. It differs from other spring-mass codes in that the relationships between forces and displacements are calculated internally from the material constitutive models and the cell geometries, rather than being determined

externally by the analyst. This allows the analyst to employ models that are much more detailed than usual spring-mass models but much less detailed than most finite element models.

For a complicated system like the one analyzed in the demonstration, one can expect reasonably accurate results from STRESS for about 1/50th to 1/30th of the CPU expenditure required by a detailed finite-element model. The increase in speed of computation is principally the result of the fewer degrees of freedom and much higher time step that is afforded by using meshes that are coarser and less uniform than a finite-element mesh.

Accuracy levels from a STRESS analysis are consistent with the needs of risk assessments and other applications where precision can be traded for the sake of computational speed. Errors of less than 35% in most of the key response variables can be anticipated for relatively coarse meshes as long as the mesh resolution is adequate in regions of high, nonuniform loading (e.g., near the point of impact for a sharp object). Improvements to the contact algorithm, which are the principal focus of ongoing development activities, may reduce the errors incurred in problems where there are multiple contacts between different material blocks.

While simplified models can provide very useful approximations for the responses of complex systems, they do not replace the need for more precise analyses with the very detailed codes. The value of the simplified models lies in their ability to scan the effects of a wide variety of parameter choices on the responses of the system. A select number of detailed finite-element analyses are essential both to benchmark the simplified models and to corroborate any major conclusions that result from the analyses.

REFERENCES

- [1] A. S. Benjamin and B. S. Altman, Spring-mass Transient Response Evaluation for Structural Systems (STRESS), Users Manual, Sandia National Laboratories, Albuquerque, NM, report to be published.
- [2] L. M. Taylor and D. P. Flanagan, PRONTO3D - A Three-Dimensional Transient Solid Dynamics Program, Report SAND86-0594, Sandia National Laboratories, Albuquerque, NM, 1987.
- [3] P3/PATRAN User Manual, Publ. No. 903000, PDA Engineering PATRAN Division, Costa Mesa, CA, 1992.
- [4] V. K. Gabrielson and R. T. Reese, SHOCK Code User's Manual, A Computer Code to Solve the Dynamic Response of Lumped-Mass Systems, Report SCL-DR-69-98, Sandia National Laboratories, Albuquerque, NM, 1969.
- [5] W. C. Mills-Curran, A. P. Gilkey, and D. P. Flanagan, EXODUS: A Finite Element File Format for Pre- and Postprocessing, Report SAND87-2997, Sandia National Laboratories, Albuquerque, NM, 1992.
- [6] L. A. Schoof and V. R. Yarberr, EXODUS II: A Finite Element Data Model, Report SAND92-2137, Sandia National Laboratories, Albuquerque, NM, 1994.
- [7] G. D. Sjaardema, Overview of the Sandia National Laboratories Engineering Analysis Code Access System, Report SAND92-2292, Sandia National Laboratories, Albuquerque, NM, 1993.
- [8] A. P. Gilkey and J. H. Glick, BLOT - A Mesh and Curve Plot Program for the Output of a Finite Element Analysis, Report SAND88-1432, Sandia National Laboratories, Albuquerque, NM, 1989.
- [9] J. D. Gruda, The Dynamic Response of a Weapon's Internal Components to a High Speed Impact, ASME/JSME Pressure Vessel and Piping Conference, Honolulu, HI, July 1995.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.