

Crashworthiness Simulations with DYNA3D

DALE A. SCHAUER, CAROL G. HOOVER, GREGORY J. KAY, ANTHONY S. LEE,
AND ANTHONY J. DE GROOT

Current progress in parallel algorithm research and applications in vehicle crash simulation is described for the explicit, finite-element algorithms in DYNA3D. Two challenging algorithm research problems are addressed: problem partitioning methods and parallel algorithms for contact at material interfaces. Two prototype parallel contact algorithms have been developed for treating the cases of local and arbitrary contact. Demonstration problems for local contact are crashworthiness simulations with 22 locally defined contact surfaces and a vehicle-barrier collision modeled with arbitrary contact. A simulation of crash tests conducted for a vehicle striking a U-channel small signpost embedded in soil has been run on both the serial and parallel versions of DYNA3D. A significant reduction in computational time has been observed when running these problems on the parallel version. However, to achieve maximum efficiency, complex problems must be appropriately partitioned, especially when contact dominates the computation.

DYNA3D (1) and its derivative commercial computer programs have set the world's standard for three-dimensional finite-element structural dynamic codes. DYNA3D is routinely used by the automobile industry to simulate and evaluate the crashworthiness of a vehicle. The Lawrence Livermore National Laboratory (LLNL), in collaboration with FHWA and NHTSA, has undertaken the task of simulating and studying the responses of vehicles and their occupants during impact conditions.

To conduct a vehicle-roadside safety structure computer simulation, accurate finite-element models of the vehicle and the roadside structure are required. The fidelity of the vehicle model is crucial to the structural behavior of the roadside feature as well as to the proper postcrash vehicle trajectory. Therefore, considerable effort is required to develop a vehicle finite-element mesh. The mesh generation process begins with digitizing all vehicle components. The digitized data are then used to create the mathematical representation of the real parts. Many modeling approximations and assumptions such as mesh discretization and boundary conditions between various components are used to bridge the physical and the mathematical models.

The amount of detail and number of complexities that are included in these simulations are bounded by the estimated running times on a Cray YMP. Future crash simulation studies will take advantage of the increased speeds and capacities of the newly developed parallel versions of DYNA3D. The increased speeds will allow parametric studies that are not currently feasible to be made, and the increased capacities will allow for the inclusion of detail that will promote simulation convergence toward physically meaningful solutions. LLNL crash simulation studies are briefly described, as are recent efforts to develop new algorithms for next-generation massively parallel computers.

Lawrence Livermore National Laboratory, Box 808, Livermore, Calif. 94551.

ALGORITHM DEVELOPMENT METHODS

Parallel Finite-Element Method and Problem Partitioning

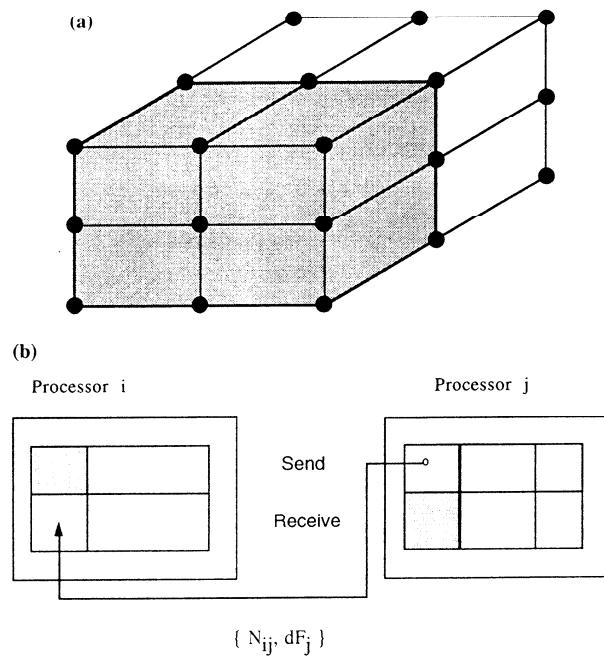
DYNA3D provides a numeric solution of the continuum equations in Lagrangian coordinates using an adiabatic approximation. The continuum equations are solved by using a finite-element treatment for the spatial derivatives and an explicit, centered time integration. The solution method uses an element-by-element technique so that matrix inversion is not necessary.

Current massively parallel computers consist of a set of processors connected with a high-speed network. The DYNA3D parallel algorithms are designed to divide the problem mesh into subdomains and to execute the full DYNA3D program on a subdomain in each processor. Solutions on subdomains are connected by communicating data across processors along the boundaries of the subdomains. Our algorithms assign elements to one processor and nodes to any processor with a connected element. Consequently, nodes on the boundaries, *shared nodes*, are stored in more than one processor, as shown in Figure 1(a). The time integration for the nodal coordinates is duplicated for the shared nodes. The motion for shared nodes is duplicated by adding a communication step when the nodal forces are calculated. For example, the force on a shared node due to internal deformations of the elements is a sum of the partial force calculated from connected elements in the same processor and connected elements in other processors [Figure 1(b)]. In the more general case the total nodal force is a sum of contributions from applied loads, element deformations, contact-restoring forces, and other boundary conditions:

$$\delta F = \delta F_{\text{boundary}} - \delta F_{\text{internal}} + \delta F_{\text{contact}} \quad (1)$$

The additive force contributions allow separate parallel treatment for the element deformation mechanics, contact algorithms, and boundary forces.

One challenge in parallel algorithm development is the subdivision of a mesh to satisfy two requirements: (a) the processors should have roughly equal amounts of computational work and (b) the communication of data between processors should be minimized. Several methods for subdividing meshes have been developed recently by using algorithms evolving from graph tree theory, grid refinement methods, spectral methods, and inertia based methods. Hendrickson and Leland have developed and collected a large collection of mesh partitioning methods, which is provided in the software package CHACO (2,3). We have designed preprocessing software for mesh subdivision that uses the CHACO software



$$dF_i = dF_i + [dF_j + dF_k + \dots] \quad dF_j = dF_j + [dF_i + dF_k + \dots]$$

FIGURE 1 Parallel strategy for DYNA3D. (a) Shaded elements are in processor 1; unshaded elements are in processor 2. Nodes are stored in the processors containing a connected element. Shared nodes are along the boundary and are stored in both processors. (b) Processor j sends the computed force for a shared node to processor i (unshaded). Processor i sends the computed force for a shared node to processor j (shaded).

product. The finite-element mesh, a list of nodes and the element-to-node connectivity, must be transformed into a list of element-to-element connections, the dual mesh, for partitioning algorithms. We have developed $O(N_{\text{element}})$ algorithms (4) to produce the dual mesh for CHACO input. The result from the CHACO algorithms is a list with elements assigned to processors. These data are used in a third preprocessing step to generate nodal assignments to processors and the list of shared nodes in each processor. Figure 2 illustrates the geometric partitioning of an automobile mesh composed of 125 materials and a mixture of solid, shell, and beam elements.

Parallel Algorithms for Boundary Conditions

There are two kinds of boundary conditions in DYNA3D: (a) applied loads and (b) prescribed nodal velocities, accelerations, or constraint conditions. Load curves that may be selected when boundary conditions are applied are read into all processors. The technique for developing a parallel version of any of these boundary conditions is to determine whether the calculations are based on data associated with element face or volume data (material properties) or whether the calculations are based strictly on nodal data (displacements, velocities, and accelerations).

Pressure loads are applied to element faces and are interpolated to evaluate nodal force contributions. Correct updating of the nodal forces requires that the forces be calculated in one processor only

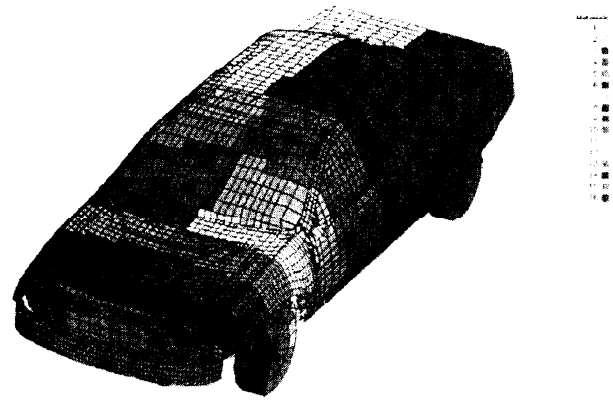


FIGURE 2 Partitioning of an automobile mesh. This is a 16-processor partition with no vertex or edge weighting. The elements are equally divided among the processors. Large elements at the back of the car represent a rigid material. The mesh is composed of a mixture of elements (solids, shells, and beams).

and then communicated along with the internal-element force contributions for shared nodes. Nodal constraints illustrate the other method for applying boundary conditions. Tied nodes, a set of nodes defined in the input, have a common, constrained degree of freedom. The application of the constraint requires only knowledge of the nodal data. To avoid communicating data when the constraint is applied, the preprocessing software assigns all of the nodes in a tied node set to any processor containing one or more of the nodes. Although the constraint condition may be calculated in more than one processor, the cost is quite negligible. We have used strategies similar to those described earlier to implement the boundary conditions needed for our applications, such as prescribed velocities, tied node sets, spot-welded nodes, stonewalls, and single-point constraints.

Parallel Contact Algorithms and Dynamic Partitioning

Contact algorithms are designed to provide an interface pressure to prevent the surface nodes of a material from penetrating through the surface of an adjacent material. There are three steps for detecting and preventing nodal penetration at material interfaces: (a) for each surface node, find the closest node in the remaining set of surface nodes; (b) for pairs of closest nodes, look up the element surface patches for one and determine whether the other node has penetrated; and (c) for a penetrated node, calculate restoring forces to move the node to the surface boundary. A parallel contact method must localize the search step to avoid the communication associated with a global search. In addition, the partitioning for contact calculations is based on the geometry of the surfaces rather than the element volumes. Thus, an optimal partitioning for the contact surfaces is usually different from that for the elements. Finally, in typical engineering applications, contact can be characterized as being (a) highly localized or (b) quite arbitrary and unpredictable because of large deformation or the motion of parts in the mesh. The two parallel contact algorithms discussed later are selected on the basis of whether the contact is local or arbitrary.

For localized contact we use the algorithm described previously for tied node sets. The contact calculations are duplicated in any processor that contains at least one of the surface nodes. This technique is advantageous because it is easy to implement. It is efficient for problems in which the contact surfaces are a small part of the overall calculations. Benchmarks for testing this method include (a) the buckling of a box beam (single surface, folding contact); (b) a cask-drop calculation using a two-surface, symmetric penalty algorithm; and (c) an earth-penetrator calculation using tied slide surfaces.

Arbitrary contact is implemented in DYNA3D in the automatic contact algorithm of Whirley and Engelmann (5) and is based on a bucket sorting algorithm developed by Benson and Hallquist (6). Bucket sorts impose a geometric grid of buckets over the surface node positions. In the parallel algorithm the buckets along the boundaries of processors are duplicated. The partitioning for contact assumes that processors are arranged along a line in one direction on the mesh, usually the longest direction. Surface node coordinates along this direction are sorted and evenly allocated to the processors. At each time step the nodal positions and velocities are sent from the element partition to the contact partition, and after the contact is calculated the nodal contact forces are communicated back to the element partition. An advantage of this approach for arbitrary contact is that the contact surfaces can be repartitioned as needed on the basis of the dynamics of the contact surfaces.

This approach, using two partitioning strategies, is novel. The element partitioning uses the more sophisticated and expensive grid refinement and recursive spectral methods because is applied only once for preprocessing the mesh. By contrast the geometric partitioning used in the arbitrary contact is implemented directly in the DYNA3D program, and it is inexpensive to repartition the surfaces when needed for load balancing.

Benchmark applications for the localized contact algorithm are the buckled cradle in the rigid pole-center bumper impact simulation and the U-channel signpost impact model (Figures 3 and 4). The Taurus sedan simulations planned in future projects will use the parallel implementation for arbitrary contact.

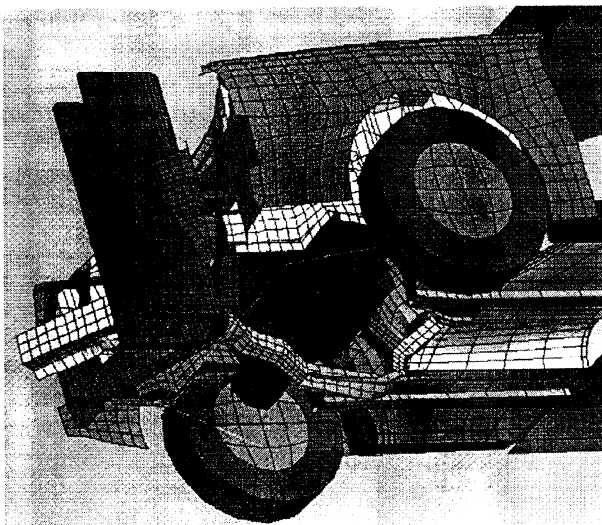


FIGURE 3 Buckled cradle in the rigid pole-center bumper impact simulation.

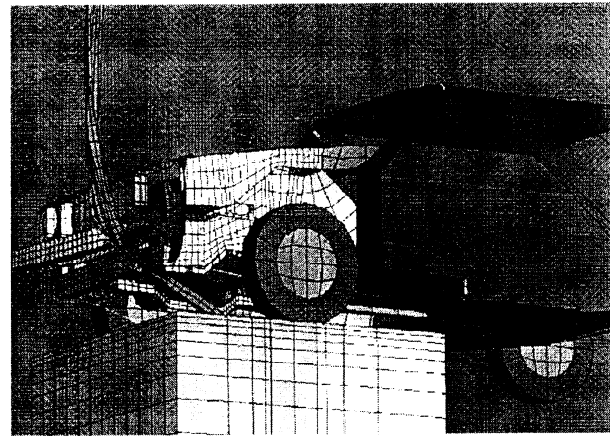


FIGURE 4 U-channel signpost impact model.

Performance

Two factors affect the scalability and efficiency of parallel calculations. First, algorithms must be designed so that the total communication time is limited to less than 10 percent of the total computation time. Second, the algorithms must not include large sections of serial calculations (or equivalently, calculations redundantly duplicated in all processors). The first condition is a *surface-to-volume* effect related to the amount of communication time for the shared nodal data on the surfaces cut by processor boundaries compared with the computational time required for the calculations within the volume in a processor. For good parallel performance this places a lower limit on both the number of elements and the portion of the contact surfaces allocated to each processor.

The algorithms described previously avoid serial calculations for all the expensive portions of the calculations. For example, all calculations for element processing (strain rates, the application of material models and equations of state, element stress, and energy calculations) and contact are performed in parallel. Nodal time integration calculations are duplicated only for the shared nodal data. Serial calculations are performed only at initialization if needed and for inexpensive portions of the time-dependent calculations, such as the application of nodal constraints.

The design of production parallel runs in the future will require characterizing the performance of all of the DYNA3D algorithms to provide guidelines to the analyst for selecting an optimal number of processors for the algorithms in a specific application. Our performance results on current parallel systems (a Cray Research T3D and a Meiko CS-2) show that problems composed primarily of solid elements will perform efficiently with roughly 1,000 elements allocated in a processor. In contrast, we find that applications dominated by shell elements, such as the simulations of automobile crashworthiness, perform efficiently with as few as 250 elements in a processor. With this criterion a 32-processor calculation for the standard DYNA3D box beam test with a factor of 2 refinement in each direction realizes a 90 percent parallel efficiency for the element calculations and 71 percent efficiency for the dynamic-automatic-contact algorithm. The nodes-in-processor contact algorithm can lose scalability if the partitioning for the elements produces a significant number of shared contact nodes in each processor. The flexibility of the preprocessing software can be used to prevent this. For

example, contact surfaces can be weighted to provide bias to the volume decomposition. Another technique is to allocate contact surface pairs fully within a subset of processors.

APPLICATIONS

For the past few years LLNL has been a participant in a pilot program funded by FHWA of the U.S. Department of Transportation to establish the capability of using finite-element analysis techniques in evaluating and designing highway roadside safety structures. Traditionally, highway roadside safety structures are designed and evaluated strictly by extensive testing according to *NCHRP Report 350: Recommended Procedures for the Safety Performance Evaluation of Highway Features* (7).

We have used a series of crashworthiness models with varying degrees of complexity. The first model simulates a vehicle impact with a roadside barrier followed by a two-surface contact interaction with a pole rather than a barrier. This has been executed on both a serial and a parallel version of DYNA3D. Next is a 1990 Ford Taurus frontal impact simulation that uses single-surface contact to include the effects of the bending fender and hood. The most sophisticated model of this series will incorporate engine parts, full body deformation, and parts composed of 125 different materials. Although we have only run this on a serial version of DYNA3D, we have exercised several different partitioning schemes for this complex and detailed vehicle model using the parallel version of DYNA3D. This is followed by the simulation of occupant responses through coupled DYNA3D-crash victim simulation (CVS) code calculations and the development of finite-element representations of vehicle suspensions that can be used in DYNA3D vehicle-roadside barrier impact simulations. Neither of these has been executed on the parallel version of DYNA3D, but both are prime candidates for setup and execution in the near future.

Vehicle-Roadside Barrier Impact Simulation

As part of FHWA's computer simulation pilot program, a DYNA3D finite-element model of a 1982 Honda Civic hatchback was developed so that the model can be used to correlate the results from four full-scale impact tests using Civic hatchbacks. LLNL has modified the original vehicle model to create a central processing unit-efficient and numerically stable Honda Civic model. Many non-structural parts or components that have low frontal impact resistance in the original mesh were eliminated. The load-carrying members at the front end of the Civic that are critical to the vehicle response to frontal impact were modified to enforce the proper boundary conditions and the contact surface definitions between the various components. Failures observed in the crash tests, such as cradle buckling and engine mount breakage, were incorporated in the modified model. The deformed shape of the buckled cradle from the center bumper-rigid pole simulation is depicted in Figure 3. Note that the aft engine mount is separated from the engine, an effect similar to that from crash test observations. The DYNA3D-simulated deceleration history at the rear seat correlates very well with the test data.

One of the crash tests conducted with the Honda Civic was a 32-km/hr (20 mph) impact on a U-channel small signpost embedded in soil. The vehicle response and the signpost deformation were drastically different from those from the rigid pole tests. The interactions

between the soil and the U-channel as well as the uncertainty of the soil properties are the dominant factors influencing the results. The simulation model for the U-channel signpost impact is shown in Figure 4. Here the soil was modeled explicitly to emphasize the interaction between the soil and the signpost. Currently, LLNL is developing soil material models for two typical backfills used in road construction. This will make possible better simulation of the impact interaction between the soil and signpost and provide hardware designers with the tools required to improve roadside hardware.

We have run this simulation on both the serial and parallel versions of DYNA3D. Considerably less computation time is required on the parallel version. We are continuing to adjust partitioning parameters for several variations in boundary and initial conditions to gain insight into the resulting effects on run time. Our goal is to be able to suggest the best partitioning approach, based on the complexity of the problem and the degree of contact (interacting surfaces) during a crashworthiness simulation.

Taurus Frontal Impact Simulation

A version of the Taurus frontal impact mesh was obtained from NHTSA with the intent of running a full frontal impact simulation with DYNA3D that could be compared with available test data. A comparison of measured (NHTSA Test 1600 performed by Calspan on April 25, 1991) and calculated engine acceleration for the full frontal impact of the Taurus sedan into a rigid wall at 56.3 km/hr (35 mph) is provided in Figure 5.

As a result of the Taurus frontal impact calculation many improvements were made in the DYNA3D automatic contact algorithms. Some of these improvements were for constrained nodal pairs, triangular shells, solid-shell interfaces, and reentrant corners in the mesh. New automatic contact options that were added to DYNA3D include the debugging aids for interpenetration and constraints on the region of automatic contact activity by spatial windows and material identifications. Improvements were also made in

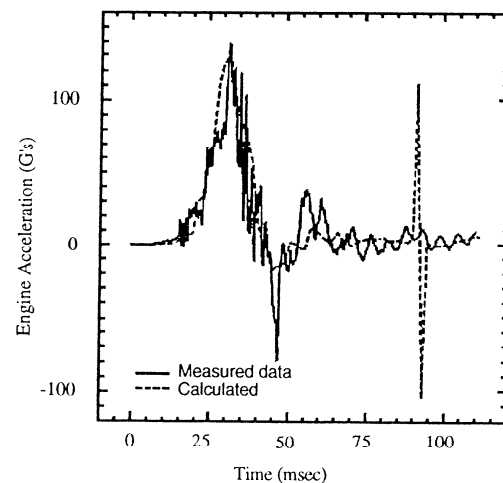


FIGURE 5 Taurus sedan full frontal impact simulation results for a 56.3-km/hr (35 mph) collision with a rigid wall.

the nodal constraint–spot weld options of DYNA3D and in the rotational updates of rigid materials.

We have not yet attempted to run this crash simulation on the parallel version because of internal LLNL priority conflicts. However, we have tried several different partitioning schemes on this large vehicle mesh. One partitioning result is shown in Figure 2. In the near future we plan to execute this on the parallel version. On the basis of our recent experience with other complex problems with extensive internal contact, we expect a considerable improvement in the speed of execution.

Occupant Responses and Suspension Modeling

Neither of the following problems has been executed on the parallel version of DYNA3D. However, these are two excellent examples of problems that may benefit significantly from a parallel computation. These have been very challenging computations for the serial version of DYNA3D.

Recent code developments at LLNL have coupled DYNA3D with its structural response features to CVS codes with their validated occupant response capabilities. Nonlinear explicit finite-element analysis (FEA) vehicle meshes have been available for some time, but validated FEA vehicle–occupant simulations are still in the developmental stages. Vehicle–occupant simulations can currently be achieved through coupled FEA–CVS code calculations. These coupled simulations take advantage of the structural response capabilities of the FEA codes and the validated occupant response capabilities of the CVS codes, can provide detailed information about occupant responses to vehicle safety features, and can be performed over a wide range of crash scenarios.

A mesh that was used in a coupled DYNA3D–CVS calculation is shown in Figure 6. This calculation simulated the occupant and vehicle responses to an impact into a rigid wall at 56.3 km/hr. This

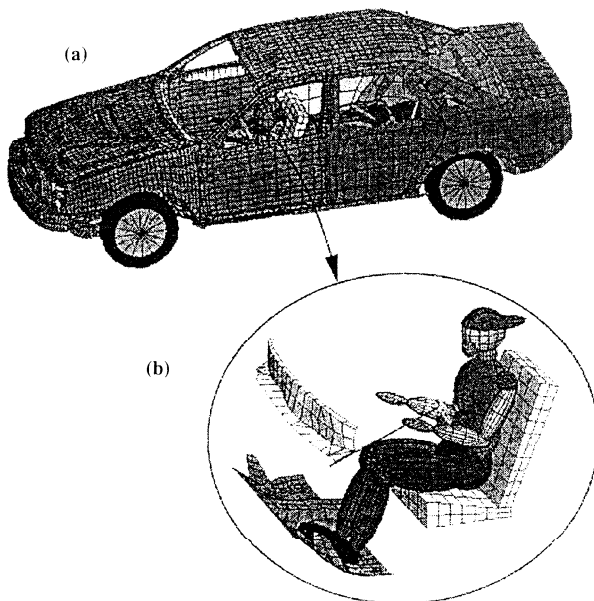


FIGURE 6 Taurus mesh with Hybrid III occupant: (a) complete FEA mesh, (b) Hybrid III crash dummy.

simulation coupled DYNA3D to the rigid-body CVS code MADYMO. (8). It is an excellent candidate for a massively parallel calculation and exhibits a high level of complexity for a crashworthiness simulation.

An example of the increasing complexity of the vehicle meshes is the current collaboration with FHWA to develop two finite-element model representations of vehicle handling systems that can be used in DYNA3D vehicle–roadside barrier impact simulations. Vehicle suspension systems are being modeled for use in a light truck and a light passenger vehicle. These models will be constructed to simulate the primary responses of damaged and undamaged vehicle handling systems during roadside impact conditions. Suspension and tire–pavement attributes will be incorporated into the vehicle handling models. The wheel assemblies will contact the ground and will be capable of rotation and translation. We are developing suspension models that allow for the redistribution of the vehicle weight during impact and that simulate damage or failure in a manner that is consistent with observations. A generic suspension system that meets these criteria is illustrated in Figure 7.

CONCLUSIONS

We have developed a robust set of algorithms that allow efficient massively parallel computation for a variety of dynamic problems containing surface contact. The performance studies on several different types of parallel hardware show that our algorithms are not hardware dependent. Examples such as the buckling of a box beam, a cask drop, an earth penetrator calculation, and a vehicle crash simulation of an impact with a U-channel signpost have been completed on both serial and parallel versions of DYNA3D. We have noted a significant reduction in computational time when running these problems on the parallel version. However, to achieve maximum efficiency, complex problems must be appropriately partitioned, especially when contact dominates the computation. We will continue to investigate and evaluate the performance of other parallel contact algorithms to treat efficiently the full range of problems simulated with DYNA3D. To improve performance we will pursue optimization studies, analyze storage requirements, and evaluate the use of machine-dependent coding related to memory access and cache use. With today's advances in massively parallel computer processing hardware, in impact dynamics, and in FEA code material modeling, it has become possible to analyze many large,

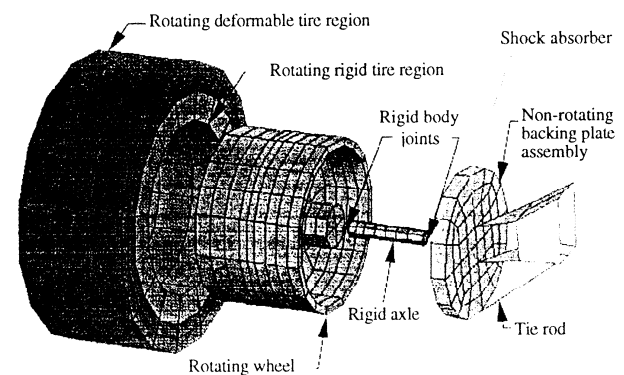


FIGURE 7 Generic finite-element front suspension.

complex vehicle impact problems by numeric methods. By combining these tools roadside safety structures can be designed and improved on the basis of FEA. Computer simulations can also provide hardware designers with the flexibility to analyze a vehicle impact scenario that may be difficult or impossible to implement in full-scale testing.

ACKNOWLEDGMENTS

Support for this project was provided by FHWA, NHTSA, and LLNL under the auspices of a U.S. Department of Energy contract.

REFERENCES

- Whirley, R. G., and B. E. Engelmann. *DYNA3D: A Nonlinear, Explicit, Three Dimensional Finite Element Code for Solid and Structural Mechanics—User Manual*. UCRL-MA-107254 Rev. 1. Lawrence Livermore National Laboratory, Livermore, Calif., 1993.
- Hendrickson, B., and R. Leland. *Technical Report SAND 92-1460*. Sandia National Laboratories, Albuquerque, N.M., 1992.
- Hendrickson, B., and R. Leland. *Technical Report SAND 93-2339*. Sandia National Laboratories, Albuquerque, N.M., 1993.
- Procassini, R. J., A. J. De Groot, and J. D. Maltby. *PARTMESH: Partitioning Unstructured Finite Element Meshes for Solution on a Massively Parallel Processor—User Manual*. UCRL-MA-118774. Lawrence Livermore National Laboratory, Livermore, Calif., 1994.
- Whirley, R. G., and B. E. Engelmann. *Automatic Contact in DYNA3D for Vehicle Crashworthiness*. UCRL-53868-93. Lawrence Livermore National Laboratory, Livermore, Calif., 1994, pp. 2–29.
- Benson, D. J., and J. O. Hallquist. A Single Surface Contact Algorithm for the Post Buckling Analysis of Shell Structures. *Computer Methods and Applied Mechanics*, Vol. 78, 1990, p. 141.
- Ross, H. E., D. L. Sicking, R. A. Zimmer, and J. D. Michie. *NHCHRP Report 350: Recommended Procedures for the Safety Performance Evaluation of Highway Features*. TRB, National Research Council, Washington, D.C., 1993.
- MADYMO USER'S MANUAL 3D*. TNO Road-Safety Research Institute, Delft, The Netherlands, 1992.

This document was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor the University of California 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 U.S. Government or the University of California. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the U.S. Government or the University of California and shall not be used for advertising or product endorsement purposes.