Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

A general contact detection algorithm for finite element analysis

M.W. Heinstein, S.W. Attaway, J.W. Swegle, F.J. Mello
Sandia National Laboratories, Albuquerque, New Mexico 87185, USA

A contact detection algorithm has been developed to address difficulties associated with the numerical simulation of contact in nonlinear finite element structural analysis codes. Mechanics problems which have self-contacting surfaces, tearing and eroding surfaces, and multi-body impact are solved using the new algorithm. The proposed algorithm is portable between dynamic and quasi-static codes and can efficiently model contact between a variety of finite element types including shells, bricks, beams and particles. The algorithm is composed of (1) a location strategy that uses a global search to decide which slave nodes are in proximity to a master surface and (2) an accurate detailed contact check that uses the projected motions of both master surface and slave node. The capability of the new algorithm is illustrated with several example problems.

INTRODUCTION

An increasingly important aspect of large-scale finite element structural simulations is the efficient and accurate determination of contact between deformable bodies. At Sandia National Laboratories, the PRONTO [1] transient dynamics codes and the SANTOS [2] and JAC [3] quasistatic codes have been used to solve a wide variety of problems involving contact and other nonlinearities. However in some cases, the range of their applicability could be increased by improving the efficiency and accuracy of contact detection.

Contact detection algorithms of interest here define a set of nodes called slave nodes and a set of surface patches called master surfaces. A slave node is simply a nodal point on the surface of the mesh and a master surface is defined using the side of a finite element on the surface. Contact detection is accomplished by monitoring the displacements of the slave nodes throughout the calculation for possible penetration of a master surface. This can be done by matching a slave node to a set of master surfaces that it potentially could contact (called neighborhood identification), followed by a detailed contact check that determines which of the candidate master surfaces the slave node contacts. A contact constraint which depends on the point of contact, amount of penetration, and the direction in which the slave node should be pushed back is then enforced in the restoration phase of the contact algorithm.

Several algorithms for efficient neighborhood identification have been proposed in the literature. These include surface side-set pairing [4], surface tracking [1][5][6], bucket searching [7], and pinball contact [8]. These current technologies work successfully for a large variety of contact problems. There are some problems, however, where improvements in the current method are required.

In particular, these improvements could address two distinct concerns with current search techniques. One concern is that nearly all neighborhood identification techniques are for nodes. For example, the result of a search is always the closest master *node* for a given slave node (with one exception in pinball overlap). A slave node does not always contact a master surface connected to the closest master node. Figure 1 shows a very simple shell self-contact example demonstrating this fact. The slave node 2 is tracking master node 14, however it is actually contacting master surface 8-9 (defined by surface nodes 8-9). The tracked master node 14 implies that a detailed contact check would consider master surfaces 13-14 and 14-15 only and not master surface 8-9 as it should. Contact is always made between a slave node and a master *surface*, and an improved neighborhood identification should reflect this.

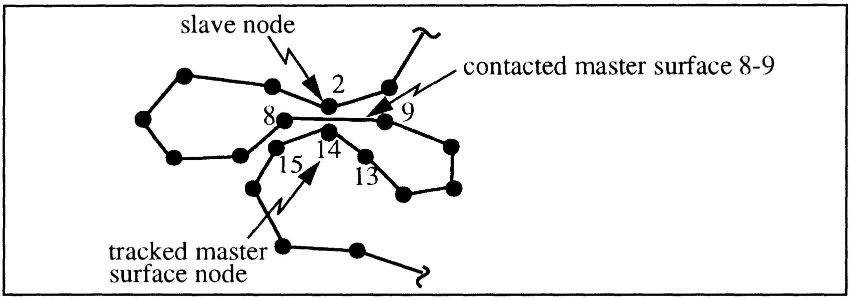


Figure 1. Slave node 2 tracking closest master node 14 results in a missed contact

Another concern is that current global sorting and searching routines depend on the problem domain. The efficiency of bucket searching is adversely affected by problems that grow or expand in spatial dimension. Significant improvements in the speed and efficiency of the search phase could be made if it depended only on the number of master surfaces and slave nodes in the problem, and *not* on the problem geometry.

In addition to these improvements, the detailed contact check could also benefit from more accurate determination of contact. Current algorithms typically use a predicted deformed configuration for contact checking (obtained by extrapolating the previous time step solution forward in time). Contact is detected when a slave node penetrates a master surface in the predicted deformed configuration. This often gives rise to an incorrect contact detection since the added information of contact surface motion is not used.

Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

PROPOSED CONTACT DETECTION ALGORITHM

The proposed contact detection algorithm outlined in the following section offers improvements in the efficiency and accuracy of contact detection. The improvements are the result of a neighborhood identification strategy that uses a global contact search and an accurate detailed contact check that uses the projected motion of both master and slave surfaces. The algorithm does not require contact surface pairing and, therefore, easily handles self-contacting surfaces, eroding and tearing surfaces, and random contact between multiple bodies. The algorithm also resolves the ambiguities that arise because of the surface discretization and from using only the deformed configuration for detailed contact checks.

Proposed Neighborhood Identification Strategy

The proposed neighborhood identification strategy uses a global search to determine what slave nodes are in close proximity to a master surface. It accumulates these potential interactions by constructing a local neighborhood around a master surface and globally searches for all slave nodes that are in the neighborhood. This local neighborhood is defined by bounding the space occupied by a master surface at its known location at time *t* and its predicted location at time *t+dt*. Figure 2 shows a bounding box for a master surface over one time step. Another box, called the capture box is also constructed to collect slave nodes that potentially contact the master surface.

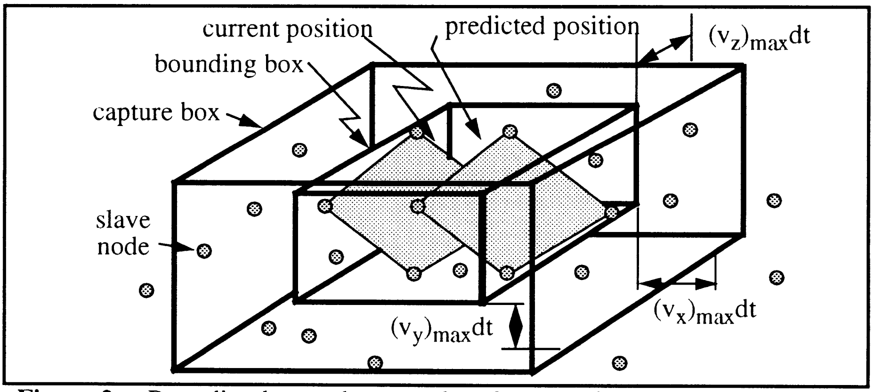


Figure 2. Bounding box and capture box for a moving master surface

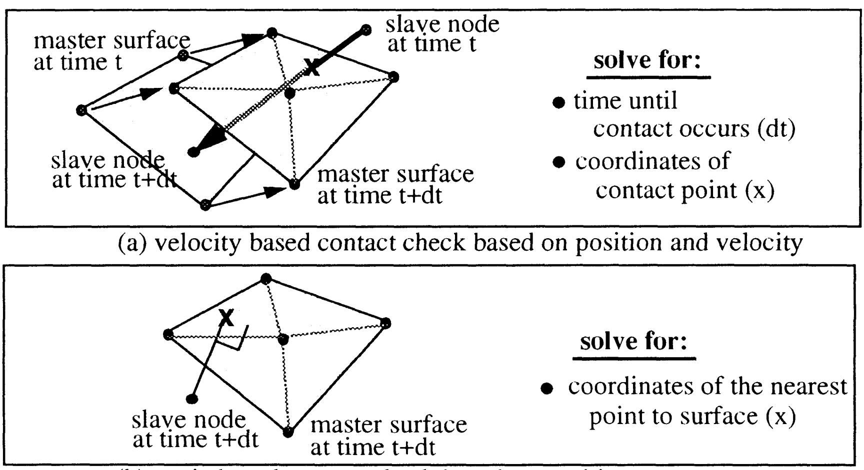
For example, suppose the maximum distance any slave node moves in one time step is $\delta_x=(v_x)_{max}dt$ in the x-direction. A slave node within a distance δ_x from the bounding box could potentially contact the master surface. This holds for the y- and z-directions as well so that a capture box can be constructed, as shown in Figure 2. Any slave node inside the capture box should therefore be considered for potential contact with the master surface. After defining the capture box, a binary search is done to identify the slave nodes inside the capture box.

Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

A binary search algorithm uses $7n$ memory locations and requires on the order of $m_s log_2 n$ comparisons where m_s is the number of master surfaces and n is the number of slave nodes. The algorithm is based on a particle search technique developed by Swegle [19] and depends only on the number of slave nodes n and not on the geometry of the problem. The binary locates two pointers into an ordered list of slave nodes for each coordinate direction. One is a lower pointer corresponding to the first slave node inside the capture box and the other is an upper pointer corresponding to the last slave node inside the capture box. From these two pointers a list is inferred that represents all the slave nodes inside the master surface capture box based on a single coordinate. The final step is to intersect the three lists to find the slave nodes in the capture box of the master surface in all three coordinate directions simultaneously. At this point, all potential contacts have been identified. The next step is to determine if contact has actually occurred using the detailed contact check.

Proposed Detailed Contact Check

The proposed detailed contact check distinguishes between slave nodes that are not in contact and those that are already in contact. For slave nodes just coming into contact, a velocity based contact check is used to identify the physically correct point of contact and time of contact, as shown in Figure 3a. The contact is determined by solving for the point in time when the moving slave node is co-planar with the moving master surface. For slave nodes already in contact with a master surface, a static contact check is used, as shown in Figure 3b.



(a) velocity based contact check based on position and velocity

(b) static based contact check based on position

Figure 3. Master slave tracking using velocity and static contact check

The static based contact check accounts for the discontinuous surface normal, and thereby avoids not finding any contact when there should be one or finding multiple solutions to a single contact.

Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

EXAMPLE PROBLEMS

The examples considered in this section demonstrate the added capabilities that are possible with both the global search algorithm and detailed contact check. Some of the added capabilities include modelling (1) structures that buckle and fold onto themselves, (2) structures that have materials which tear and create new surfaces, and (3) problems where multiple body contact/impact is occurring.

Self-Contacting Impact: Buckling of Shell-Like Structures

The following example demonstrates the self contacting capability of the contact detection algorithm. This feature is important for modelling crash dynamics where buckling, tearing, and self contact is common. The cylindrical shell (can) structure shown in Figure 4 is impacted by plate. The can is 0.25 inches thick, has an inside radius of 5 inches, and is initially 15 inches long. The bottom of the can is constrained in all directions. The 22x11 in. plate is 2.5 inches thick and is initially tilted at a 10 degree angle as it impacts the can at 5000 in/s. Contact between any of the slave nodes and master surfaces on both surface sets (one for the plate and one for the can) are detected during the analysis. The deformed shapes at various times are shown in Figure 4. The self-contact of the buckled can is evident in several locations.

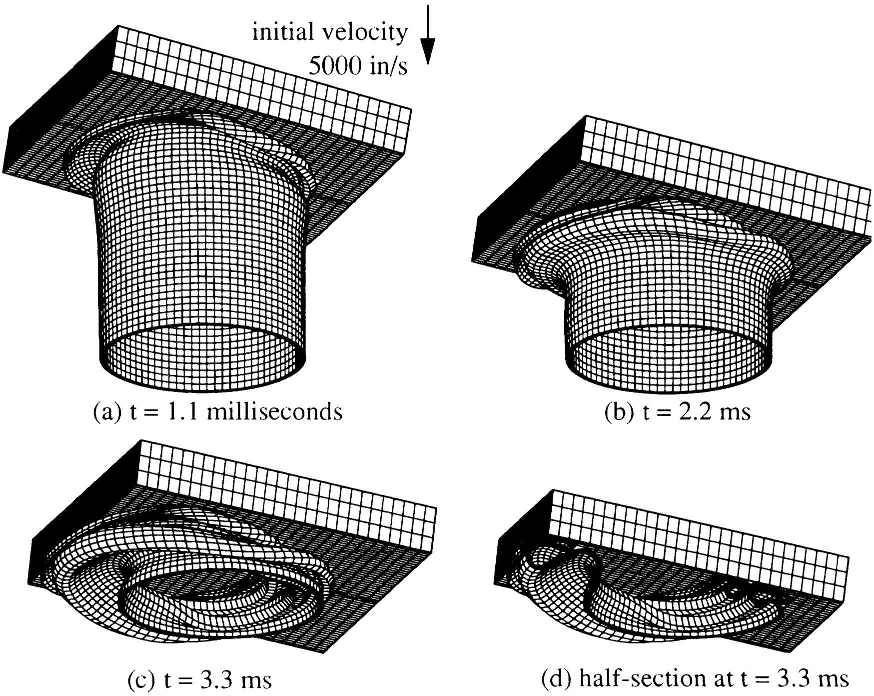
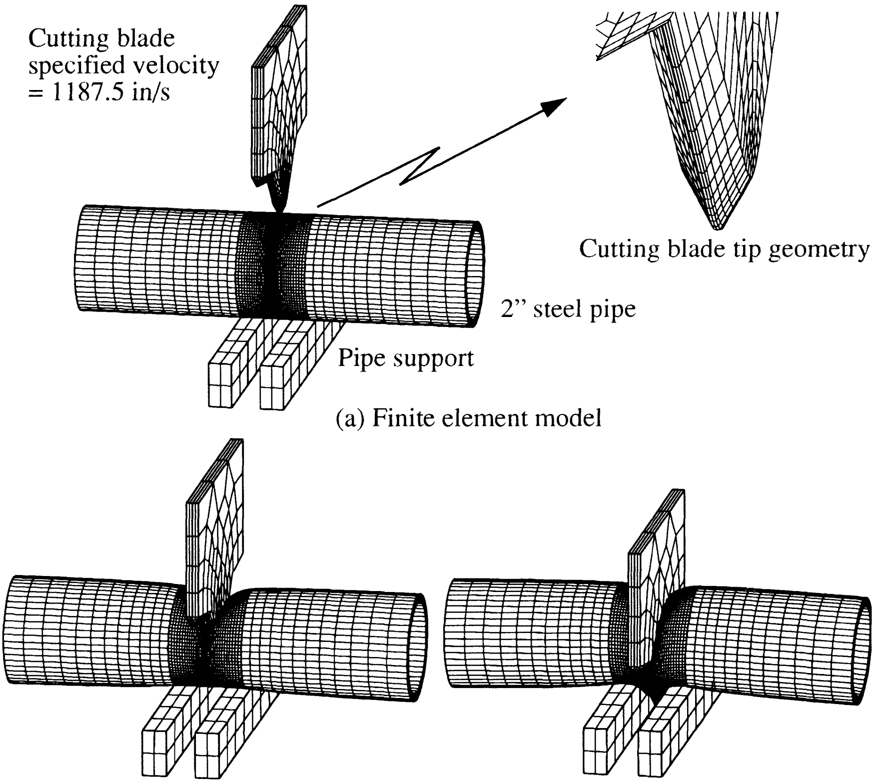


Figure 4. Deformed meshes of the buckled can at various times

Automatic Contact Surface Redefinition: Cutting of a Steel Pipe

In the following example, the capability of automatically redefining the contact surface is demonstrated. For this problem, and others like it, the new surface that is generated as a result of tearing can find itself in contact with other surfaces of the body. The algorithm is illustrated with a simulation of a pipe cutting process. The process involves a hardened, 0.25 inch thick, wedged-shape cutting blade that is forced through a two inch, schedule 40 pipe that is resting on a support, as shown in Figure 5. The blade initially indents and punctures through one side of the pipe and then progressively tears the pipe walls, as shown in Figure 5. The tear is simulated by deleting elements in which the material damage has accumulated to 0.9. The power law hardening model used for the pipe material accumulates damage when it exceeds a failure strain of 1.27 and is loaded with a positive maximum principal stress [9]. The newly created surface is automatically included in the contact algorithm by redefining the surface after any elements are deleted. As the simulation progresses, the edges of the tear are in contact with the moving cutting blade. The simulation continues until the blade cuts through the pipe at approximately 3 milliseconds.



(a) Finite element model

(b) 1.5 milliseconds

(c) 2.5 milliseconds

Figure 5. Pipe cutting simulation results at various times

Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

Multi-Body Impact: Elastic-Plastic Bar Impacting Bricks

One of the added capabilities of the new contact detection algorithm is the efficient modelling of multi-body impact without *a-priori* definition of contact surfaces. For problems where random contact is anticipated, as in this example, each body could potentially impact any other body. Assuming that each block has $n \approx 50$ slave nodes, a contact pairing algorithm would require $19^2(2n\,log(2n)) = 239{,}843$ comparisons, whereas the proposed global contact searching algorithm requires only $19n\,log_2(19n) = 9397$ comparisons. The multi-body impact example, shown in Figure 6, has an elastic-plastic bar impacting a stack of 17 elastic bricks resting against a stationary elastic-plastic wall. All slave nodes and master surfaces on the bodies were automatically defined using the proposed algorithm. The surface definition algorithm is described in [10].

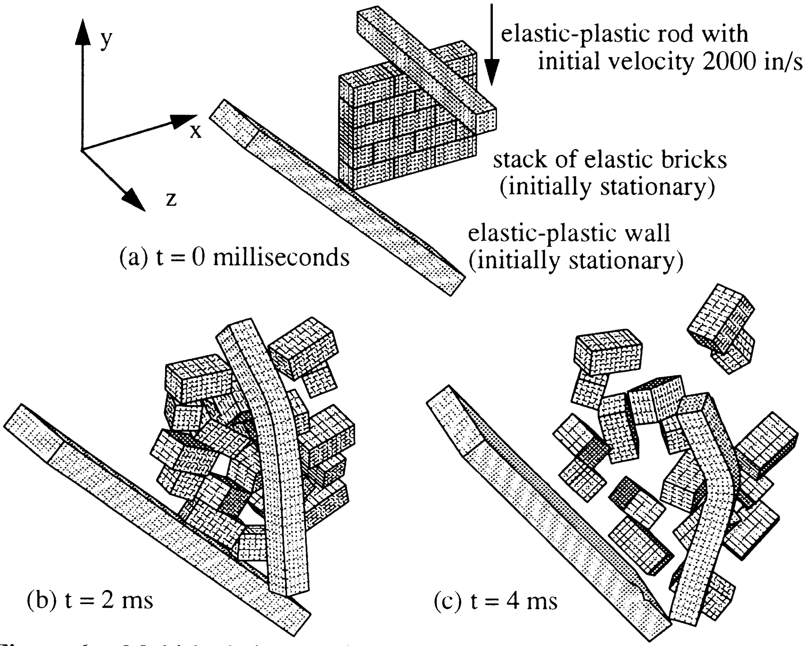


Figure 6. Multi-body impact simulation results at various times

CONCLUSIONS

A proposed algorithm has been presented that offers improvements in contact detection. The improvements are a result of a neighborhood identification strategy that uses a global contact search and a detailed contact check, distinguishing between surfaces that are impacting and ones that are already in contact. The key features of the new contact detection algorithm are:

- The efficient global search allows for global contact. This means that a new capability for modelling self contacting structures and eroding or tearing surfaces is now available.

Transactions on Engineering Sciences vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3533

- The position and velocity of both the slave node and master surface are considered in determining initial contact. This results in a physically correct determination of the contact location.
- The known locations of contacting surfaces and their velocities are used consider only physically meaningful contacts in the detailed contact check.
- An automatic surface definition algorithm allows for a simplified user input in many cases. One can, for example, include all surfaces of a body by specifying that the material be considered for global contact.

The capabilities of the new contact detection algorithm have been demonstrated with several example problems.

ACKNOWLEDGEMENTS

This work, performed at Sandia National Laboratories, was supported by the U. S. Department of Energy under Contract DE-AC04-76DP00789.

REFERENCES

1 Attaway, S.W., *Update of PRONTO2D and PRONTO3D Transient Solid Dynamics Programs*, SAND90-0102, Sandia National Laboratories, Albuquerque, NM 87185, 1990.

2 Stone, C.M., *SANTOS: A Two-Dimensional Finite Element Program for the Quasistatic Large Deformation, Inelastic Response of Solids*, SAND90-0543, Sandia National Laboratories, Albuquerque, NM 87185, in preparation.

3 Biffle, J.H., *JAC3D - A Three-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate Gradient Method*, SAND87-1305, Sandia National Laboratories, Albuquerque, NM 87185, in preparation.

4 Zhong, Z.H. and Nilsson, L., "A Contact Searching Algorithm for General 3D Contact-Impact Problems," *Computers and Structures*, Vol. 34, No. 2, pp. 327-335, 1990.

5 Hallquist, J.O. and Benson, D.J., *User's Manual for DYNA3D: Nonlinear Dynamic Analysis of Structures*, Rev. 3, UCID-19592, Lawrence Livermore National Laboratories, 1987.

6 Hibbitt, Karlsson and Sorensen, Inc., *Contact Calculations with ABAQUS - ABAQUS Explicit Users Manual*, Hibbitt, Karlsson and Sorensen, Inc., 1992.

7 Benson, B.J. and Hallquist, J.O., "A Single Surface Contact Algorithm for the Post-Buckling Analysis of Structures," *Computer Methods in Applied Mechanics and Engineering*, Vol. 78, pp. 141-163, 1990.

8 Belytschko,T. and Neal, M.O., "Contact-Impact by the Pinball Algorithm with Penalty and LaGrangian Methods," *Int. J. Numerical Methods Eng.*, Vol. 31, pp. 547-572, 1991.

9 Stone, C.M., Wellman, G.W., and Krieg, R.D., *A Vectorized Elastic/Plastic Power Law Hardening Material Model Including Luders Strain*, SAND90-0153, Sandia National Laboratories, Albuquerque, NM 87185, March 1990.

10 Heinstein, M.W., Attaway, S.W., Swegle, J.W., and Mello, F.J., *A General Purpose Contact Detection Algorithm for Nonlinear Structural Analysis Codes*, SAND92-2141, Sandia National Laboratories, Albuquerque, NM 87185, October 1992.