NIKE3D Enhancement and Support

Michael A. Puso
(925) 422-8198
puso1@llnl.gov

The objective of this work is to develop, maintain, and support the implicit structural mechanics finite element code NIKE3D. New features are added to accommodate engineering analysis needs. Maintenance includes bug fixes and code porting to the various platforms available to engineering analysts. User support includes assisting analysts in model debugging and providing general analysis recommendations.

Project Goals

Ongoing code support requires new features to meet our engineering community's user demands. Our goals included the following highlights in NIKE3D development:

1. add bucket sort to mortar contact to speed up contact searching;
2. add new Hierarchical Data Format (HDF) stress initialization restart for coupling to ParaDyn;
3. integrate key NIKE3D contact and solver features to Diablo; and
4. perform an extensive benchmarking study using NIKE3D and Diablo.

Relevance to LLNL Mission

Structural analysis is a core competency of LLNL's Engineering. In-house maintenance, support, and code enhancement are crucial for meeting Engineering's many analysis needs. NIKE3D, in particular, is a premier code for handling difficult non-linear static structural analysis problems.

FY2009 Accomplishments and Results

The mortar contact algorithms previously added to NIKE3D have proven very successful in solving many difficult production analyses and are now

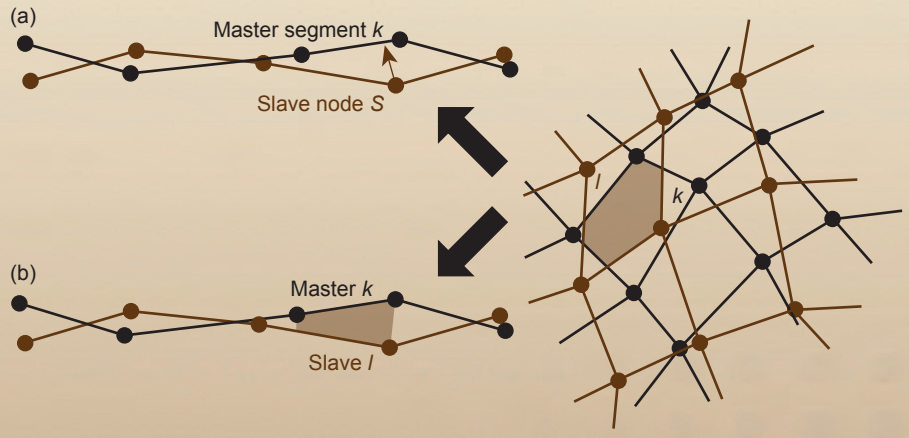

Figure 1. Illustrations of (a) node-on-segment contact algorithm, using a closest point approach to gather node-segment pairs and compute the associated gap between slave node *S* and master segment *k*; and (b) segment-to-segment contact algorithm, gathering segment-segment pairs and computing weighted volume between slave segment *l* and master segment *k*.

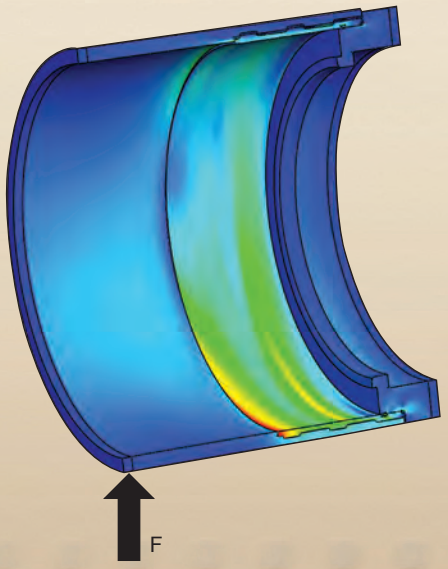

Figure 2. Aft coupler joint analysis. The mesh includes 167,000 elements, (550,000 dof) and over 13,000 contact segments. It required approximately half an hour of wall time to run, using 64 processors for the linear equation solving phases in NIKE3D.

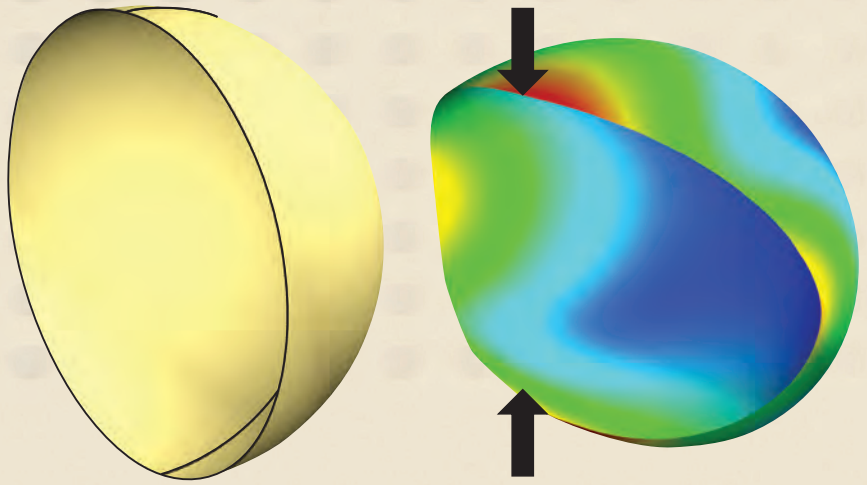

Figure 3. Bending of thin steel shell membrane. The radius is 10 in. and the thickness is 0.04 in.

the preferred contact method for our analysts. The classical contact algorithms are node-on-segment [Fig. 1(a)] whereas mortar contact is a segment-to-segment type contact algorithm [Fig. 1(b)]. The mortar contact method smoothes the contact results and thus provides more robust response. Unfortunately, the segment-to-segment approach is radically different and little of the existing node-on-segment code could be reused for its implementation.

The popularity of the node-on-segment approach was mainly due to its simplicity and it evolved to accommodate a number of different search algorithms and formulations such as local search, bucket sort, and automatic contact. The segment-to-segment approach is more complicated and some of NIKE3D's contact features still need to be implemented in that context.

The latest enhancement is the bucket sort for replacing the simple "brute force" search originally implemented. The old mortar sort algorithm used order N^2 operation (where N is the number of contact facets). The bucket sort, appropriately implemented, reduces to order N operations. The new bucket sort defines a 3-D grid based on a characteristic length. This grid is then populated with master and slave segments. Each cell in that grid is called a bucket, and candidate segment interactions need to be searched only between adjacent buckets.

Bucket sorts achieve optimal speed when all segments are equal size but can approach N^2 operation counts when segment sizes differ. To alleviate this problem, segments can be binned according to sizes and a multiple pass bucket sort approach used. The speed up can be dramatic.

Figure 2 shows an aft coupler joint from a composite material assembly. The contact search time using the old search algorithm required 2500 s whereas the new bucket sort required only 318 s.

The stress initialization feature was upgraded to accommodate the new HDF file format used by ParaDyn. This feature is useful for static initialization of models prior to assessing fast transient response. Whereas explicit dynamic relaxation is often used for such initialization, an implicit code such as NIKE3D can be faster. Figure 3 shows the bending of a thin shell; Fig. 4 displays wall clock times required to do static initialization as the mesh is refined. ParaDyn typically demonstrates near perfect scaling as the number of processors is increased for a fixed mesh size. However, here the problem geometry is fixed and the number of elements increases, requiring a smaller time step size. As the same total time interval must be simulated to dampen to a static response, the number of ParaDyn time steps to solution increases. NIKE3D's parallel direct solver cannot scale perfectly either, but maintains a performance edge across the range of problem size considered.

FY2010 Proposed Work

Our plans for FY2010 include 1) modularizing and upgrading data structures and improving shell routines; and 2) implementing an interior point method within a quasi-Newtonian framework for the solution of unilateral contact problems.

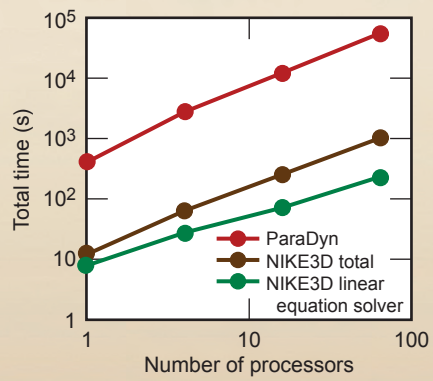

Figure 4. Scaling plot for stress initialization of a thin sphere. Here, the mesh is refined as the number of processors is increased. ParaDyn typically scales very well but does not in this type of benchmark since the time step shrinks with successive refinements. NIKE3D is not parallel but the direct linear solver is. This solver typically scales very well.