A comparison of DYNA3D, NIKE3D and LS-DYNA

…Seshu Nimmala
Nov23, 2004

1.0 DYNA3D: DYNA3D (ref. (i)) is a nonlinear, explicit, finite element code for analyzing the transient dynamic response of three-dimensional solids and structures.

1.1 BACKGROUND: DYNA3D, a public code, has seen wide application to a variety of problems over the past twenty years (please note the year of the reference: 1999). The applications include transient dynamic problems ranging from crash dynamics to human artery simulations by analysts both at Lawrence Livermore National Laboratory (LLNL) and elsewhere. DYNA3D was originated by Dr. John O. Hallquist of the Methods Development Group at LLNL. During the period 1984–1987, he was joined by Dr. David J. Benson, who is now on the Engineering faculty at the University of California, San Diego. Dr. Hallquist continued as Lead Developer on DYNA3D until 1988, when he left LLNL to pursue a career in private business. Subsequent to Dr. Hallquist, Dr. Bruce E. Engelmann and Dr. Robert G. Whirley served as Lead Developers of DYNA3D, and are responsible for adding many new features, options, and improvements including the YASE shell element. In 1993, both Dr. Engelmann and Dr. Whirley departed for the private sector. Dr. Jerry I. Lin assumed the Lead Developer role for DYNA3D in 1995, and continues to serve this function along with other members of the Methods Development Group.

1.2 CAPABILITIES: The code is fully vectorized (this results in faster runs on vector computers such as Cray) and is available on several computer platforms. DYNA3D includes solid, shell, beam, and truss elements to allow maximum flexibility in modeling physical problems. Many material models are available to represent a wide range of material behavior, including elasticity, plasticity, composites, thermal effects, and rate dependence. In addition, DYNA3D has a sophisticated contact interface capability, including frictional sliding and single surface contact. Rigid materials provide added modeling flexibility. A material model driver with interactive graphics display is incorporated into DYNA3D to permit accurate modeling of complex material response based on experimental data.

As an explicit code,DYNA3D is appropriate for problems where high rate dynamics or stress wave propagation effects are important. For static and low rate dynamic problems, the implicit NIKE3D code (ref. (ii)) may be more suitable. DYNA3D may be applied to quasi-static problems by either using the dynamic relaxation option or by simply applying the external loads slowly and integrating the dynamics equations until all significant transients have died out. In contrast to NIKE3D, DYNA3D uses a large number of relatively small time steps, with the solution being explicit (and inexpensive) at each step. Thus, DYNA3D does not form and solve the large matrix equation typical of implicit codes such as NIKE3D, and does not require iteration at each time step. This often leads DYNA3D to be compute-bound, with modest memory requirements, whereas NIKE3D is often memory or I/O bound due to the assembly of a large stiffness matrix at each time step. Analysis using DYNA3D could be combined with NIKE3D.

There are at least 57 material types, 13 equations of state and several elements (including fluid). The fluid model is most useful for analyzing structures with contained fluids. Large distortions in the fluid make some free-surface and fluid flow problems more amenable to analysis using other analysis codes employing an Eulerian formulation. Caution should therefore be used when applying this material model in situations where large distortions are expected.

Element formulations that include:
• 2-node truss elements
• 2-node integrated beam elements
• 2-node resultant beam elements
• 3-node triangular shell elements
• 4-node quadrilateral shell elements
• 8-node quadrilateral thick shell elements
• 8-node continuum elements.
These element formulations all handle geometric nonlinearities and do not lock for incompressible materials.

DYNA3D has material models that include:
• elasticity and plasticity (isotropic and anisotropic)
• finite elasticity
• volumetric compaction
• rate dependence
• thermal effects
• damage and failure of ductile and brittle solids.

To model nonlinear pressure-volume behavior, DYNA3D has equations of state that include:
• polynomial functions
• high explosive models
• tabulated functions.

A variety of boundary conditions are available, including:
• prescribed velocities
• non-reflecting (transmitting) boundaries
• sliding boundaries along arbitrary planes
• symmetry planes with failure.

Methods of prescribing loads include:
• nodal forces and moments
• follower forces
• surface pressure loads
• body force loads

• loads due to thermal expansion
• loads arising from momentum deposition
• Brode function airblast loads.

DYNA3D has a general interface contact capability which includes:
• frictional sliding
• single surface contact
• nodes impacting on a surface
• tied interfaces
• one-dimensional slidelines
• rigid walls
• material failure along interfaces
• penalty and Lagrangian projection options for constraint enforcement
• fully automatic contact.

The constraint modeling capabilities include:
• single point constraints
• arbitrary nodal constraints.

DYNA3D has the ability to model rigid body dynamics using features such as:
• rigid materials for trusses, beams, shells, and solids
• general rigid body joint definitions between rigid bodies
• inertial property specification by rigid body
• automatic calculation of rigid body inertial properties
• material switching between deformable and rigid idealizations
• merging multiple rigid materials to form one rigid body
• nonlinear springs and dampers to connect rigid or deformable bodies.

The analysis capabilities of DYNA3D include:
• transient dynamic analysis
• static analysis using dynamic relaxation
• dynamic analysis with static initialization from a NIKE3D implicit analysis.

The code may be restarted with a variety of modifications to the analysis, including:
• changes in termination time
• deletion of portions of the model by element or by material
• deletion of sliding interfaces
• modification of boundary conditions on deformable or rigid materials.

There are no inherent limits on the size of a DYNA3D analysis model, and storage allocation is dynamic within the code. Problem size is constrained only by the memory available on the computer. Current generation supercomputers have solved DYNA3D problems with more than 500,000 elements, and computing capabilities continue to expand as new generations of hardware become available. DYNA3D has been parallelized with shared memory parallelization (SMP) and with message passing parallelization (MPI). Massively parallel processing (MPP) versions of DYNA3D have surpassed these capacities by an order of magnitude on select hardware.

DYNA3D is based on a finite element discretization of the three spatial dimensions and a finite difference discretization of time. The explicit central difference method is used to integrate the equations of motion in time. The central difference method is conditionally stable, and stability is governed by the Courant limit on the time step. For solid elements, this limit is essentially the time required for an elastic stress wave to propagate across the shortest dimension of the smallest element in the mesh. Equivalently, this maximum time step may be related to the period of the highest free vibration mode of the finite element mesh. DYNA3D automatically calculates the maximum time step size at each step of the solution, and adjusts the time step accordingly to minimize the number of time steps used in a solution. This feature minimizes the cost of the analysis while assuring that stability is maintained. Time step considerations suggest the use of structural elements (beams and shells) rather than solid elements for modeling structures that are "thin" in some dimension, since this thin direction has been analytically incorporated into the element formulation. The time step for the structural element model is therefore many times larger than it would be for a corresponding solid element model of the same structure.

DYNA3D uses a lumped mass formulation for efficiency. This produces a diagonal mass matrix, which renders the solution of the momentum equation trivial at each step in that no simultaneous system of equations must be solved. In the above equation, are the applied external forces, and are the element internal forces. The new accelerations are easily found, from which the updated velocity and coordinates are calculated using the central difference integration formulas.

Interactive graphics preprocessors and postprocessors include MAZE (Sanford, 1996) and ORION (Hallquist and Levatin, 1985) for the two-dimensional codes and INGRID (Christon, Dovey, and Hallquist, 1992), TAURUS (Spelce and Hallquist, 1991), and GRIZ (Dovey and Spelce, 1996) for the three-dimensional codes. All plotting (except in GRIZ) is accomplished using the DIGLIB public domain graphics library developed by Hal Brand at LLNL. INGRID is the most widely used preprocessor at LLNL.

There are two LLNL codes which can be used to visualize results from DYNA3D: TAURUS and GRIZ. In addition, there are several commercial pre/post-processors which can read and display the DYNA3D database.

2.0 NIKE3D: NIKE3D (ref. (ii)) is a fully implicit three-dimensional finite element code for analyzing the finite strain static and dynamic response of inelastic solids, shells, and beams.

2.1 BACKGROUND: NIKE3D, also a public code, has been used at the LLNL and elsewhere over the past fourteen years (please note the year of the reference: 1995) to study the static, quasi-static, and dynamic response of structures undergoing finite deformations. In 1989 the originator of NIKE3D, Dr. John O. Hallquist, left the Laboratory to pursue a career with his private company. Since its introduction in 1980 numerous improvements have been made to NIKE3D. The most significant development has been the addition of structural elements including trusses, beams, membranes, and shells to augment the 8-node solid element. The first structural element was added by Slater [1982], who implemented a shell element that was a product of his Ph.D. research. At this time NIKE3D was extended and reorganized to include rotational degrees of freedom and "ports" for new element classes. Although Slater's shell was used successfully in many calculations, it has been dropped from NIKE3D in favor of a shell element based on the work of Hughes and coworkers [1981] which, unlike the previous implementation, is valid for finite strains. In a straightforward exercise this shell has been degenerated into a rectangular beam element. Truss and membrane elements are obtained from the beam and shell elements, respectively, by one point integration in the thickness direction while ignoring transverse shear. Many organizational changes have been made to reduce the number of I/O operations that often dominate the solution cost.

The element overlays have been recoded for vectorization. The Green-Naghdi stress rate has been adopted in favor of the Jaumann rate for the solid elements to obtain improved behavior for kinematic hardening in the plasticity model. However, the Jaumann rate has been retained for the structural elements. The implementation of radial return plasticity has eliminated the very costly and less accurate subincrementation algorithm. Plane stress structural elements also use radial return with a simple additional iteration to uniquely solve for the normal strain. A stiffness matrix formulation of Hughes has been adopted that is consistent with the constant pressure assumption in the solid elements. The sliding interface algorithm has also been extended to shell elements and modified to increase its reliability and accuracy.

2.2 CAPABILITIES: Spatial discretization is achieved by the use of 8-node solid elements, 2-node truss and beam elements, and 4-node membrane and shell elements. Over twenty constitutive models are available for representing a wide range of elastic, plastic, viscous, and thermally dependent material behavior. Contact-impact algorithms permit gaps, frictional sliding, and mesh discontinuities along material interfaces. Several nonlinear solution strategies are available, including Full-, Modified-, and Quasi-Newton methods. The resulting system of simultaneous linear equations is either solved iteratively by an element-by-element method, or directly by a factorization method, for which case bandwidth minimization is optional. Data may be stored either in or out of core memory to allow for large analyses.

It utilizes implicit time integration, making it most efficient for static or low rate dynamic problems. For high rate dynamic problems such as highspeed crash and impact, explicitly integrated DYNA3D code is more appropriate. NIKE3D utilizes a relatively small set of elements. All elements use low order interpolation, requiring no mid-side node definitions. This approach chooses highly efficient elements over more costly higher order elements (which have been shown to be more accurate in *linear* analyses.) Using many efficient elements, rather than few higher order elements, the analyst may more accurately describe the complex geometries typical of large finite element models.

Analyses with NIKE3D typically concern loadings applied in several increments, or steps, to accurately resolve geometric or material nonlinearities. Within each step, a nonlinear solver drives the iterative process used to satisfy equilibrium. Within each equilibrium iteration, a system of linear equations must be solved. Several nonlinear and linear solvers are available. For dynamic problems, load steps represent increments in time. In *quasi-static* analysis, these steps are increments in *quasi-time*, a monotonically increasing time-like parameter which characterizes the evolution of the loading. The unconditional stability of the implicit time integrator allows the user to specify the time step size, its choice being governed by convergence and accuracy considerations. Proper choice of time step size, where convergence is maintained, accuracy is maximized, and computational cost is minimized, represents one of the most difficult challenges to the analyst using implicit finite element methods. A variety of diagnostic and automatic time step control tools are available to simplify this task.

The greatest challenge in analysis with NIKE3D is still selection of the time step size. To aid in monitoring the progress of the nonlinear solver, the screen output has been enhanced. Convergence may be controlled using a new residual norm, and/or by a new displacement tolerance whose value does not change as the total deformation in the problem increases. Most importantly, automatic time step control has matured, trapping and recovering from errors that previously generated floating point exceptions.

NIKE3D is based on an updated Lagrangian formulation. During each load step, NIKE3D computes the nodal displacement increments which produce a geometry that satisfies equilibrium at the end of the step. This involves the solution of a set of (potentially) nonlinear equations. Several nonlinear solvers based on the Newton method are available. The nonlinear solution process in each step is iterative. Equilibrium is obtained when one or more user defined convergence tolerances are met. During each equilibrium iteration of the nonlinear solver, NIKE3D recomputes internal and external forces and the global stiffness matrix (if necessary) using the current estimate of geometry.

NIKE3D has at least 23 material models.

NIKE3D contains virtually no mesh generation capability. At LLNL, the three-dimensional mesh generators INGRID and SLIC (Gerhard [1979]) provide preprocessing capability, and support most of NIKE3D's options. Occasionally, users must edit the preprocessor-generated input deck directly, especially to invoke new features. The interactive three dimensional color graphics postprocessors TAURUS and GRIZ may be used to display results from the plot database.

3.0 LS-DYNA: LS-DYNA (ref. (iii)) is a general purpose finite element code for analyzing large deformation dynamic response of structures including structures coupled to fluids.

3.1 BACKGROUND: The origin of LS-DYNA dates back to the public domain software, DYNA3D, which was developed in the mid seventies at the LLNL. Dr. Hallquist, the originator of DYNA3D, resigned from LLNL at the start of 1989 to continue development of LS-DYNA at Livermore Software Technology Corporation (LSTC), a private business.

3.2 CAPABILITIES: The main solution methodology is based on explicit time integration. An implicit solver is available with limited capabilities including structural analysis and heat transfer. Spatial discretization is achieved by the use of four node tetrahedron and eight node solid elements, two node beam elements, three and four node shell elements, eight node solid shell elements, truss elements, membrane elements, discrete elements, and rigid bodies. A variety of element formulations are available for each element type.

A more detailed chronology of the several developments (including bug fixes, efficiency improvements, new capabilities) made could be found in the ref. (iii). More importantly (from fluid-structure interaction point of view), Multi-material Eulerian/ALE fluids(with 2^{nd} order accurate formulations, automatic coupling to shell, brick or beam elements, element with fluid + void, element with multimaterial) have been added in v940 in 1996-97.

LS-DYNA can be used to solve for the steady state or transient fluid flow about a body using Boundary element method.

This code has over 200 materials, over 10 equations of state and several element formulations.

This has been vectorized and parallelized for faster execution.

LS-PREPOST and PATRAN could be used for pre- and post processing operations. Either one is not complete by itself, but could complement each other. ANSYS could also be used as a pre- and post-processor since more recent version comes with LS-DYNA to complement ANSYS's implicit solution capabilities.

4.0 CONCLUSION: It is possible that the references (i) and (ii), available to the public online, have not kept up (more so in case of (ii)) with the recent developments in the codes. So latest information on DYNA3D & NIKE3D could be obtained by contacting LLNL. The codes are available to the public with the approval of LLNL. I searched for ALE formulation in ref. (i), and could not locate any relevant information in there.

LS-DYNA is the flagship product of Livermore Software Technology Corporation and as such it is in the commercial domain. The capabilities are widely tested by the industrial users in a variety of domains. Very good technical support is available at some fee. Code is available for research under a contract agreement (which is somewhat restrictive) with LSTC, however any research developments would proceed under the ownership of LSTC.

In case of DYNA3D and NIKE3D, technical support is somewhat limited (the code could be complex to understand). But there are bright sides as well. As a public code, the use of DYNA3D by outside firms has been widespread, and this has played an important role in its development. Many code shortcomings have been discovered and remedied as a direct result of dialog with outside users in industry. In addition, many capabilities have been suggested or inspired by feedback obtained from Collaborators outside LLNL. This active participation provides important information for future development directions of DYNA3D.

Parallelized version of Dyna3D is named as ParaDyn, and it is also being developed at LLNL (ref. (vii)). This software has been demonstrated to be able to handle 1million to 10 million elements. However, it needs to be examined whether this version is also made available for research.

It seems like these codes are the best for research purposes as the explorative and learning opportunities (research really needs these two components) are unlimited in their use and enhancement to solve specific problems. Also, one could contribute to the spirit of open source development (after exploring with LLNL).

There are other references (iv-vi) pertaining to DYNA3D that could be of interest for further study.

5.0 REFERENCES:

(i) DYNA3D : "A Nonlinear, Explicit, Three-dimensional Finite Element Code for Solid and Structural Mechanics", User Manual, Lawrence Livermore National Laboratory, Oct 1999

(ii) NIKE3D : "A Nonlinear, Implicit, Three-dimensional Finite Element Code for Solid and Structural Mechanics", User Manual, Lawrence Livermore National Laboratory, Apr 1995.

(iii) LS-DYNA : " Theoretical Manual", Livermore Software Technology Corporation, May 1998.

(iv) Hallquist, J. O.,"DYNA3D Course Notes," University of California, Lawrence Livermore National Laboratory, Rept. UCID-19899, Rev. 2, 1987.

(v) Lovejoy, S., Whirley, R., "DYNA3D Example Problem Manual."

(vi) Maker, B., Whirley, R., Engelmann B., "Numerical Integration of Structural Elements in NIKE3D and DYNA3D

(vii) http://www.llnl.gov/str/Raboin.html