

MODAL METHODS FOR TRANSIENT DYNAMIC ANALYSIS IN LS-DYNA

Bradley N. Maker
David J. Benson

Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, CA 94550
(925) 449-2500
maker@lstc.com

Keywords: modal analysis, transient dynamics

ABSTRACT

Modal analysis methods offer an opportunity for tremendous cost savings compared to traditional explicit approaches to transient dynamic analysis. Modal methods approximate the structural response of a body by a linear combination of pre-computed mode shapes, eliminating the need for explicit element processing. LS-DYNA's new modal analysis capability is combined with existing rigid body features to offer the additional advantage of large rigid body motion with superimposed linear modal response. This allows a portion of an LS-DYNA model to be represented by modes, while other parts of the model are treated with standard nonlinear explicit methods. LS-DYNA has also been enhanced to compute several types of modes, including eigen modes, constraint modes, and attachment modes. These modes are written to binary databases which can be viewed using LS-POST, and used as input to subsequent modal analyses. This paper introduces these new features for modal analysis, focusing on modeling procedures, input parameters, and examples of potential cost savings.

INTRODUCTION

For some applications, transient dynamic analysis can be performed very efficiently using modal methods. In contrast to standard direct methods, where the response of each node is computed independently, modal methods represent the response of the structure using a weighted combination of a set of structural mode shapes. The unknowns in modal analysis become the weights (amplitudes) of each mode. If a complete set of modes is used, i.e. the number of modes is equal to the number of degrees of freedom in the model, the method is exact and can be considered as a transformation into a set of generalized coordinates. Typically however, only a small number of modes are used, so the method becomes approximate and a substantial computational savings can be realized.

APPROACH

The underlying assumption in traditional modal analysis is linear superposition. This restricts the application of modal methods to problems of small strain and displacement. *Flexible rigid bodies* were implemented in LS-DYNA to extend modal analysis to accommodate finite rigid body motions.

Flexible Rigid Bodies

Using flexible rigid bodies, the small deformation modal response is computed in a local coordinate system at the center of mass within the rigid body. The rigid body may then undergo finite translation and rotation, together with superimposed (small) modal deformation. An LS-DYNA model may contain one or more flexible rigid bodies, as well as parts which are treated with traditional nonlinear explicit methods. For example, a flexible rigid body may be used to represent the body of a vehicle, while the suspension system may be fully nonlinear.

INPUT PARAMETERS FOR MODAL ANALYSIS

This section describes how to modify a standard transient explicit LS-DYNA input deck to use modal analysis. Recommended input parameters are identified in **boldface** type and included in boxed keyword input syntax for quick reference. A boldface zero value is entered for required input data which is model specific, such as the termination time **term**.

Modal Analysis Procedure

An LS-DYNA modal analysis is performed in two phases. In the first phase, modes are computed and output to a binary database. Then in the second phase, this database is used as input to supply modes for the transient dynamic analysis. In this way, modes may be computed only once and re-used for transient analyses of several loading cases.

The parts in the original model must be subdivided into those which will be flexible rigid bodies, and those which will be treated using nonlinear explicit FEA. Several flexible rigid bodies may be used, and each may include more than one part.

Phase I: Extracting Modes

Modes are computed separately for each flexible rigid body in the model. This means that sub-models must be created from the original input deck for each flexible rigid body. Sub-models can be created with the help of LS-POST by deleting unwanted parts (SelPar button) and then writing the remaining nodes and elements into a new keyword file (Output button). Elastic materials should be used, all applied loads must be removed, and enough boundary constraints must be defined to eliminate rigid body motion. (Note that if “free-free” eigenmodes are left in the model, then LS-DYNA will attempt to identify and remove them during initialization for the transient modal analysis.)

Modes are computed using implicit analyses. An eigenvalue analysis may be used to compute *eigen modes*, which are written to a binary database named “d3eigv”. *Constraint modes* and *attachment modes* can both be computed in a single analysis, and output to a binary database named “d3mode”. These databases can be viewed using LS-POST. One or both of these databases may be supplied as input to the transient analyses of the second phase. Constraint and attachment modes are described in more detail later in this report.

The *CONTROL_IMPLICIT_GENERAL keyword is used to activate the implicit method (**imflag**), and to provide an initial implicit time step size (**dt0**). The step size must be non-zero, but is otherwise ignored for mode extraction.

```
*CONTROL_IMPLICIT_GENERAL
$  imflag      dt0
      1      0.001
```

The *CONTROL_IMPLICIT_EIGENVALUE keyword is used to activate an eigenvalue analysis, and to select the number of eigenmodes **neigv** to be extracted. Several additional parameters are available in this keyword (not shown in box below) for controlling which eigenvalues are computed. By default, the lowest **neigv** eigenvalues are determined.

```
*CONTROL_IMPLICIT_EIGENVALUE
$  neigv
      0
```

The *CONTROL_IMPLICIT_MODES keyword is used to compute constraint and/or attachment modes. Parameter **ncmode** gives the node set ID for nodes where constraint modes are computed, and **namode** gives the node set ID for attachment nodes:

```
*CONTROL_IMPLICIT_MODES
$  ncmode      namode
      93      94
```

One mode can be computed for each degree of freedom (DOF) at each node. The first two node set attributes can be used to restrict mode computation to specific degrees of freedom. These attributes may be defined globally for all nodes in the set using parameters **da1** and **da2**, and later re-defined locally for some nodes using parameters **a1** and **a2**. The first parameters **da1** and **a1** control translational DOFs, and the second parameters **da2** and **a2** control rotational DOFs. If **both** the first and second attribute parameters are zero, modes are computed for all unconstrained degrees of freedom at each node in the set. Otherwise, modes are computed according to the following definitions:

| attribute value | DOF for which modes are computed |
|---------------------|----------------------------------|
| 1 | X |
| 2 | Y |
| 3 | Z |
| 4 | X, Y |
| 5 | Y, Z |
| 6 | X, Z |
| 7 | X, Y, Z |
| 8 | (none) |
| da1 = 0 and da2 = 0 | (all) |

For example, to compute constraint modes for all DOF at nodes 13-16, the following node set definition could be used:

```
*SET_NODE_LIST
$   sid      da1      da2
   93         0         0
$   nid1     nid2     nid3     nid4
   13        14        15        16
```

and to compute attachment modes for only the translational DOF at nodes 41-45, the following node set definition could be used:

```
*SET_NODE_LIST
$   sid      da1      da2
   94         7         8
$   nid1     nid2     nid3     nid4     nid5
   41        42        43        44        45
```

As an example using local parameters **a1** and **a2**, the following node set definition would compute attachment modes for only the translational DOF at nodes 41-43, and for all DOF except the rotation about the Z-axis at nodes 44 and 45:

```
*SET_NODE_COLUMN
$   sid      da1      da2
   94         7         8
$   nid      a1       a2
   41         0         0
   42         7         8
   43         0         0
   44         7         4
   45         7         4
```

Phase II: Transient Dynamic Modal / Explicit Analysis

In preparation for the transient dynamic analysis, the original explicit model must be modified to define the flexible rigid bodies and their associated mode databases:

- All parts belonging to flexible rigid bodies must use the rigid material *MAT_RIGID (type 20).
- Multiple parts belonging to the same flexible rigid body must be merged using *CONSTRAINED_RIGID_BODIES keyword. The first part ID in this list becomes the master part ID for the flexible rigid body.
- Constraints between parts belonging to the same flexible rigid body must be removed. This includes, for example, spotwelds connecting inner and outer body panels which will be represented as a single flexible rigid body. Since these constraints were active when the mode databases were computed, they will be implicitly enforced when these modes are superimposed to compute motion during the transient analysis.
- Interactions between flexible rigid bodies may not be imposed by constraints. Penalty-type joints and contact interfaces between flexible rigid bodies are allowed.

The *PART_MODES keyword must be defined for each flexible rigid body. Parameter **pid** gives the part ID, or master part ID if multiple parts have been merged together. The total number of modes used is given by **nmfb**.

```
*PART_MODES
$   pid      nmfb      form      ansid      format      kmflag      nupdf
   0         0         1         0         0         0         0
$ filename (maximum 80 characters)
<database_filename>
$   mstart    mstop    dampf
   0         0         0.0
```

Parameter **form** selects the type of database(s) to read. Values for LS-DYNA's databases are:

| form | binary database type |
|------|--|
| 3 | LS-DYNA eigenvalue |
| 4 | LS-DYNA constraint/attachment mode |
| 5 | both LS-DYNA databases (filenames must be "d3eigv" and "d3mode") |

Parameter **filename** is used to enter the name of the modes database (ignored when **form**=5). Parameters **mstart**, **mstop**, and **dampf** can be used to add damping to some or all of the modes. A nominal value of one percent of critical damping has been used with good results on some models (**dampf** = 0.01).

By default, the motions of all nodes in the flexible body are updated every time step. Some computational savings can be realized if it is possible to skip these updates for a large group of nodes. Nodes which are skipped will appear to remain motionless in d3plot and other output databases. To use this feature, create a node set containing only those nodes which will be updated. Enter this node set ID using parameter **ansid**, and define parameter **nupdf**=1. Then use the same node set ID with the *PART_ATTACHMENT_NODES keyword in the definition of the master flexible rigid body part. WARNING: All nodes which receive applied loading must be included in the update node set.

EXAMPLE: CANTILEVERED SHELL

A simple example demonstrating the accuracy of the modal analysis method is given by the cantilevered strip of shell elements shown in figure 1. The shells are fully supported at their left end, and a load is applied in the vertical direction at their right end. The load is increased in magnitude from time=0 to time=0.002, then abruptly released, allowing the structure to vibrate freely. All deflections are small enough for linear theory to apply.

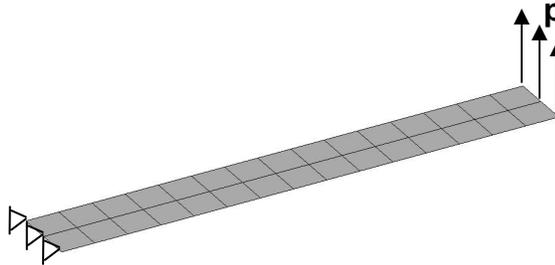


Figure 1 – Simple cantilevered strip of shell elements, supported on left end and loaded on right end.

In the first phase of the analysis, an eigenvalue simulation was used to extract the lowest twenty modes of the model. These modes were output to the binary database “d3eigv”. (Note that this database can be viewed and the modes animated using LS-POST.)

In the second phase of the analysis, the material type of the shell part was redefined as rigid. The *PART_MODES keyword was added to indicate the name of the “d3eigv” database, and the number of modes within that database to use in the simulation. Two transient analyses were performed, the first using all twenty modes, and the second using only the lowest three modes.

The vertical displacement response of a node point at the tip of the cantilevered strip is plotted as a function of time in figure 2. Results are shown for three solution methods: a standard explicit finite element simulation, an explicit modal simulation using the lowest twenty modes of the structure, and an explicit modal simulation using the lowest three modes. All results show similar behavior as the dynamic response causes oscillations in the tip deflection during loading, then complex dynamic response during free vibration including several simultaneously excited frequencies.

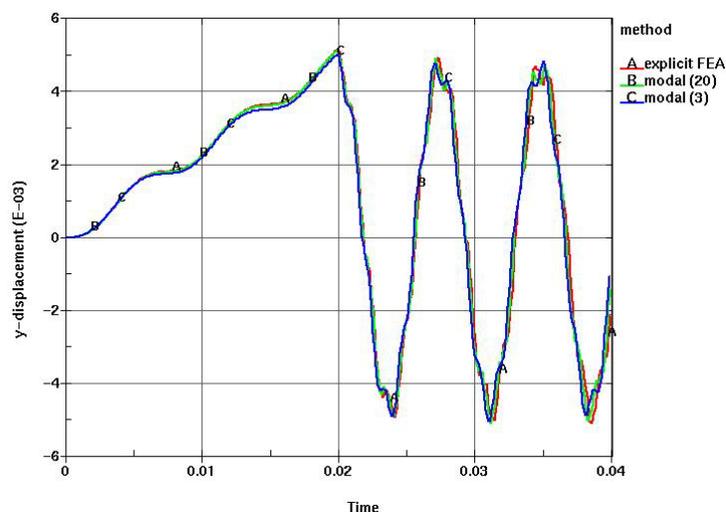


Figure 2 – Dynamic response of tip node during loading, followed by free vibration. Solutions using explicit FEA, explicit modal (20 modes), and explicit modal (3 modes) are shown. Surprising accuracy is obtained in this simple example using only three modes to represent the structure.

EXAMPLE: SHIP UNDER MINE ATTACK

An unfitted, dry ship was subjected to mine attack type loading in a model consisting of approximately 10,000 shell elements (figure 3). Loading was applied as a bi-linear pressure spike on the forward hull, spanning two compartments. The loading was chosen to excite the primary low frequency modes of the ship. The loading magnitude and duration, shown in figure 4, were selected to give realistic response magnitudes. The simulated time interval was 0.5 seconds.

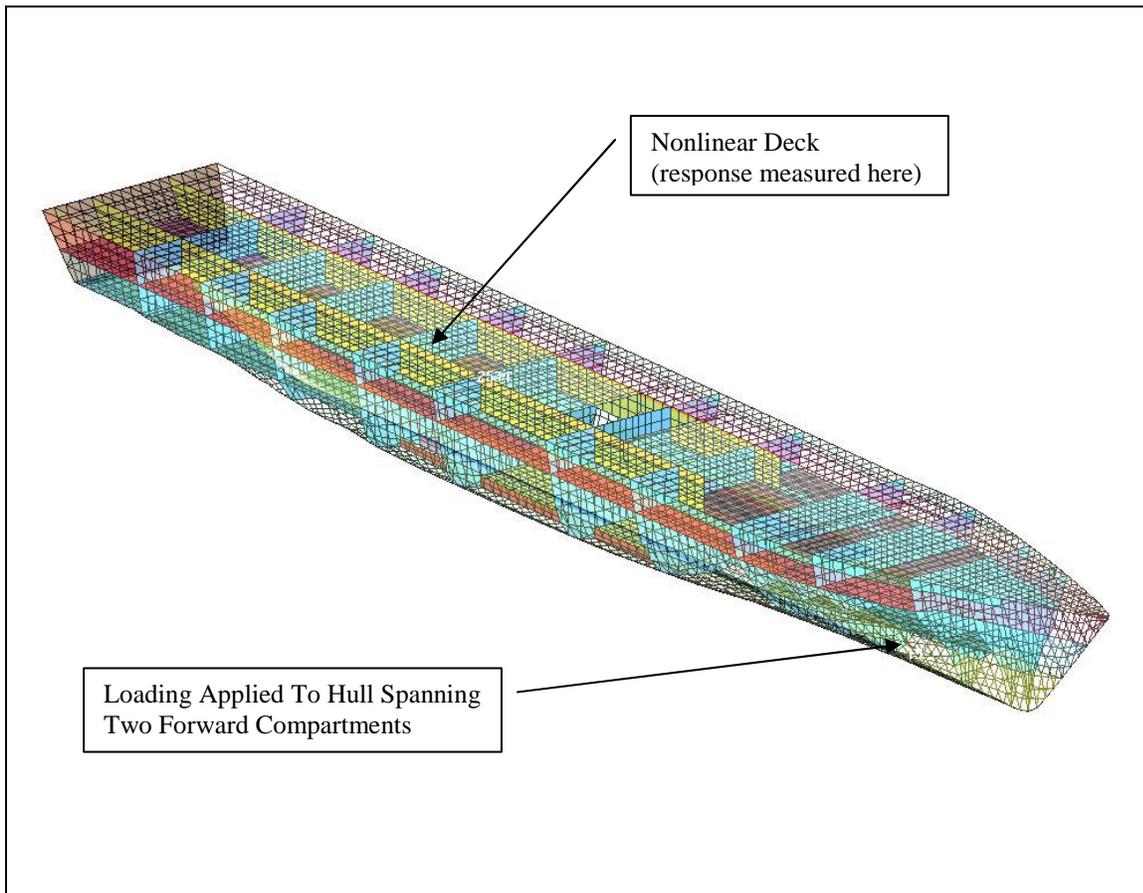


Figure 3: Finite element model of a ship, with transparent hull and top deck used to reveal inner structure. No superstructure nor internal fittings are included, and no effects of water were represented.

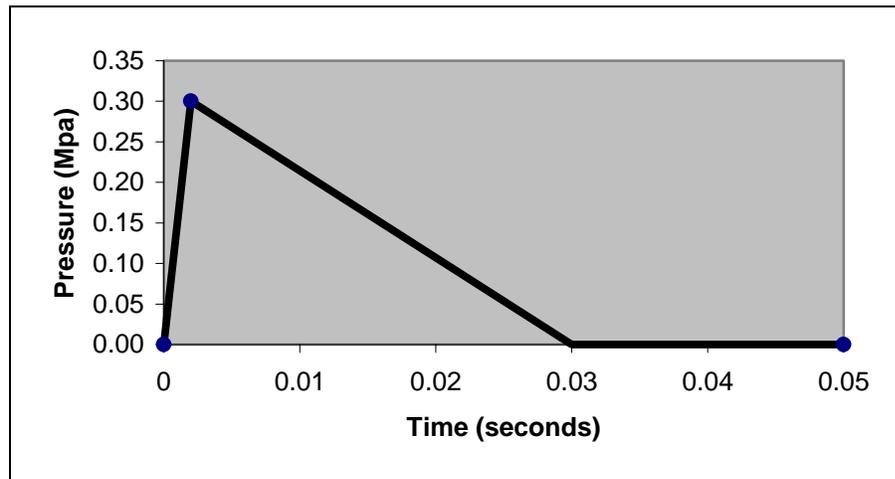


Figure 4: Pressure time history for "mine attack" loading applied to forward hull section of the ship model. Applied pressure was zero after time shown.

Response was measured at the center of a deck selected near the middle of the ship. The entire deck within this compartment was modeled using nonlinear explicit elements. The remainder of the ship was treated with three methods: nonlinear explicit using element type 2, nonlinear explicit using element type 16, and by modal representation. Element type 2, the (default) reduced integration Belytschko-Tsay shell, is the fastest element and introduces the most damping. Element type 16 is a fully integrated enhanced strain shell. It is typically about 2.5 times slower than type 2, but provides more accurate high frequency response, introduces less damping, and typically compares more favorably with linear elements.

For the modal method, constraint modes were generated for all three translational degrees of freedom at each node around the perimeter of the isolation deck, producing a total of 120 modes. Timing data presented in table 1 shows that this modal representation was 85 times faster than the nonlinear explicit method using element type 2, and 300 times faster than the nonlinear explicit method using element type 16.

| method | CPU seconds |
|-----------------------------|-------------|
| explicit, shell type 16 | 129,049 |
| explicit, shell type 2 | 36,586 |
| modal, 120 constraint modes | 430 |

Table 1: CPU times for the full, dry ship simulation on an IBM-B80 workstation using one CPU. The modal method was up to 300 times faster than the explicit nonlinear method.

Figure 5 shows the response predictions obtained from the nonlinear explicit and modal simulations. Resultant velocity is plotted for a node point located in the center of the isolation deck. The modal solution, using one percent critical damping, shows good accuracy in peak velocity prediction, and reasonable resolution of low frequency response, although high frequency content in the modal solution is lacking. Significant differences are apparent between the nonlinear explicit solutions, especially in the later half of the simulation. Shell formulation 2 is much less expensive, but also shows reduced high frequency content compared with shell formulation 16. In view of these differences, the accuracy of the modal approximation is reasonably good.

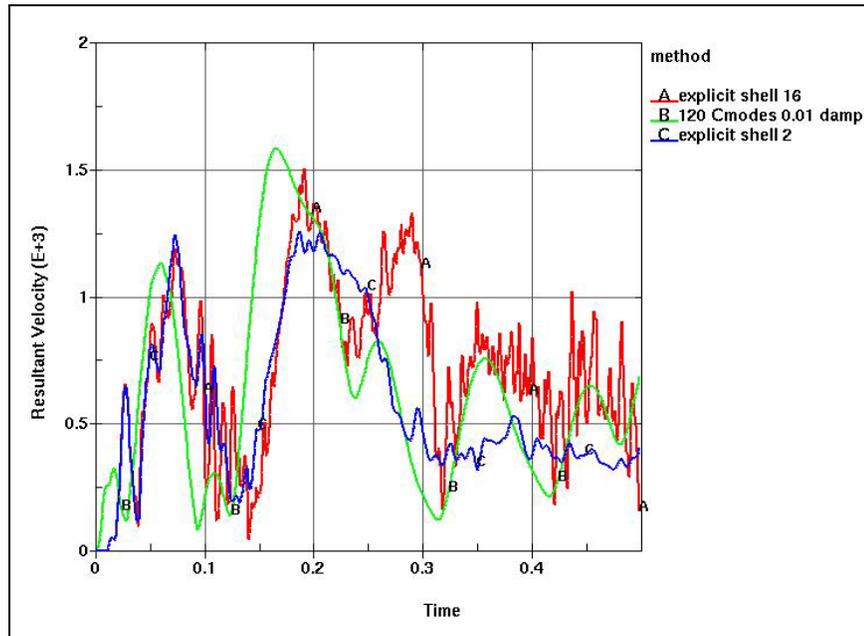


Figure 5: Resultant velocity (mm/sec) vs. time (sec), from explicit and modal simulations. A damping factor of one percent of critical damping was used in the modal simulation. The modal solution lacks high frequency content.

APPENDIX: BASIC FORMULATION FOR MODAL ANALYSIS

The first step in a modal analysis is to extract a set of response modes. These can be obtained by performing an implicit eigenvalue analysis in LS-DYNA. After computing the stiffness and mass matrices $[\mathbf{K}]$ and $[\mathbf{M}]$, the following system is constructed

$$([\mathbf{K}] - \omega^2 [\mathbf{M}])\Phi = 0$$

and solved for the eigenvalues ω^2 and corresponding eigenvectors (mode shapes) Φ_i . For an n degree-of-freedom (DOF) system, a total of n eigenvalue/eigenvector pairs could be found. Typically, only the lowest m pairs are determined, since these will participate most significantly in the transient response of the model. These mode shapes are orthogonal to each other,

$$\Phi_i \Phi_j = 0,$$

and are normalized so that the assembled set of mode shapes form an $n \times m$ modal matrix Φ which is orthogonal to both the mass and stiffness matrices in the following way:

$$\Phi^T [\mathbf{M}] \Phi = [\mathbf{I}], \quad \Phi^T [\mathbf{K}] \Phi = [\omega^2]$$

where $[\mathbf{I}]$ is the $m \times m$ identity matrix, and $[\omega^2]$ is a diagonal matrix of eigenvalues ω^2 .

Modal Decomposition

Using the modal matrix, the discrete equations of motion for the full $n \times n$ system

$$[\mathbf{M}]\ddot{\mathbf{u}} + [\mathbf{K}]\mathbf{u} = \mathbf{f}(\mathbf{t})$$

can be re-written into a smaller mxm system

$$[\mathbf{I}]\ddot{\mathbf{z}} + [\omega^2]\mathbf{z} = \mathbf{p}(\mathbf{t}) \quad (1)$$

where the n-dimensional unknown displacement \mathbf{u} , acceleration $\ddot{\mathbf{u}}$, and loading function $\mathbf{f}(\mathbf{t})$ have been decomposed into their smaller, m-dimensional counterparts \mathbf{z} , $\ddot{\mathbf{z}}$, and $\mathbf{p}(\mathbf{t})$ as:

$$\mathbf{u} = \Phi\mathbf{z}, \quad \ddot{\mathbf{u}} = \Phi\ddot{\mathbf{z}}, \quad \mathbf{p}(\mathbf{t}) = \Phi^T\mathbf{f}(\mathbf{t}). \quad (2)$$

A tremendous advantage of the modal method is that since the modes are orthogonal, the resulting mxm system of equations (1) is uncoupled (matrices $[\mathbf{I}]$ and $[\omega^2]$ are diagonal). This means that solutions can be obtained for the unknowns associated with each mode independently, without solving a simultaneous set of linear algebraic equations.

The modal equations of motion (1) are integrated in time, and once solutions are known for the m-dimensional system, solutions for the complete n-dimensional system are recovered using equations (2).

Modal time integration by implicit method

The modal equations of motion (1) can be integrated in time using either implicit or explicit methods. Implicit “direct” integration is often performed using the Newmark method, in a similar manner to that used for an implicit transient dynamic analysis. The modal displacement, velocity and acceleration vectors at time steps (n) and ($n+1$) are related by the Newmark equations:

$$\begin{aligned} \dot{\mathbf{z}}_{n+1} &= \dot{\mathbf{z}}_n + [(1-\gamma)\ddot{\mathbf{z}}_n + \gamma\ddot{\mathbf{z}}_{n+1}]\Delta t \\ \mathbf{z}_{n+1} &= \mathbf{z}_n + \dot{\mathbf{z}}_n\Delta t + \left[\left(\frac{1}{2} - \beta \right) \ddot{\mathbf{z}}_n + \beta \ddot{\mathbf{z}}_{n+1} \right] \Delta t^2 \end{aligned}$$

where Δt is the time step size, and parameters $\gamma = 0.5$ and $\beta = 0.25$ are typically chosen for unconditional stability. Using these equations, the numerical algorithm becomes:

- 1) Compute inertial contributions to modal stiffness and load vectors

$$\begin{aligned} [\hat{\omega}^2] &= [\omega^2] + \frac{1}{\beta\Delta t^2} [\mathbf{I}] \\ \hat{\mathbf{p}}(\mathbf{t}) &= \mathbf{p}(\mathbf{t}) + [\mathbf{I}] \left(\frac{1}{\beta\Delta t^2} \mathbf{z}_n + \frac{1}{\beta\Delta t} \dot{\mathbf{z}}_n + \frac{1-2\beta}{2\beta} \ddot{\mathbf{z}}_n \right) \end{aligned}$$

- 2) Solve uncoupled system for modal amplitudes at step n+1:

$$\mathbf{z}_{n+1} = \frac{\hat{\mathbf{p}}_i(t)}{[\hat{\omega}^2]}$$

- 3) Update modal velocity and acceleration

$$\begin{aligned} \ddot{\mathbf{z}}_{n+1} &= \frac{1}{\beta\Delta t^2} (\mathbf{z}_{n+1} - \mathbf{z}_n) - \frac{1}{\beta\Delta t} \dot{\mathbf{z}}_n - \frac{1-2\beta}{2\beta} \ddot{\mathbf{z}}_n \\ \dot{\mathbf{z}}_{n+1} &= \dot{\mathbf{z}}_n + [(1-\gamma)\ddot{\mathbf{z}}_n + \gamma\ddot{\mathbf{z}}_{n+1}]\Delta t \end{aligned}$$

- 4) Proceed to the next time step, repeating steps 1-3 above.

The advantage of the implicit method is its unconditional stability, which allows an arbitrarily large time step size Δt to be selected. In practice, the implicit step size is selected to accurately resolve the highest frequency of interest. Note that since $[\hat{\omega}^2]$ is a diagonal matrix, each modal DOF can be updated independently, and all of the above equations are scalar. This makes the implicit method very computationally efficient, an seemingly superior to the explicit method described below. However, when more complex features such as constraint and attachment modes are introduced to the modal analysis, the diagonal nature of $[\hat{\omega}^2]$ is lost, and a system of simultaneous equations must be solved in each implicit time step.

Modal time integration by explicit method

A simple alternative time integration scheme is the explicit method. The numerical algorithm is:

- 1) Solve for acceleration at time n using the modal equations (1):

$$\ddot{\mathbf{z}}_n = \mathbf{p}(t) - [\omega^2]\mathbf{z}_n$$

- 2) Integrate modal acceleration to obtain modal velocity at the mid-step:

$$\dot{\mathbf{z}}_{n+1/2} = \dot{\mathbf{z}}_{n-1/2} + \ddot{\mathbf{z}}_{n+1}\Delta t$$

- 3) Integrate mid-step modal velocity to obtain modal amplitude at n+1:

$$\mathbf{z}_{n+1} = \mathbf{z}_n + \dot{\mathbf{z}}_{n+1/2}\Delta t$$

- 4) Proceed to the next time step, repeating steps 1-3 above.

The explicit algorithm does not require solution of a system of linear equations, but is only conditionally stable. The stable time step size is limited by the largest modal frequency ω_{\max} included in the simulation:

$$\Delta t < \frac{2}{\omega_{\max}}$$

Fortunately, this step size is usually much larger than the critical step size for the finite element mesh, and the explicit modal method becomes much more efficient than the explicit finite element method. Standard modal analysis in LS-DYNA currently employs the explicit time integration method.

Constraint Modes

When the eigenmodes of the unsupported model are used for modal analysis, the solution to the modal equations of motion (1) gives amplitudes \mathbf{z} for each mode shape. These amplitudes often have little engineering significance, and meaningful results are obtained only after solving (2) to recover the motions of each node. In some cases, the motions of a relatively small set of nodes are of particular interest. These degrees of freedom can be added directly to the modal system (1) using constraint modes.

Constraint modes directly introduce response at vital constraint points in the model, which is not necessarily included in the lowest m eigenmodes. Constraint modes introduce a full spectrum of frequency response. As a consequence, the stable time step size for explicit integration of the modal equations of motion (1) may be substantially reduced. When constraint modes are used, the eigenvalue extraction model should be modified by constraining all degrees of freedom associated with constraint modes. In this way, constraint modes contribute 100% of the motion to their respective degrees of freedom.

Constraint modes are computed by assembling the stiffness matrix for the unsupported structure, and extracting to storage columns associated with each constraint DOF. Then all constraint DOF are eliminated and the resulting

linear system is factorized. Using the extracted columns as force vectors, backsolves are performed to compute the constraint modes. Amplitudes for the constraint DOF associated with each column are unit, while the others are zero.

If the stiffness matrix is partitioned into constraint DOFs “C” and unconstrained DOFs “U”,

$$\begin{bmatrix} \mathbf{K}_{UU} & \mathbf{K}_{UC} \\ \mathbf{K}_{CU} & \mathbf{K}_{CC} \end{bmatrix} \begin{bmatrix} \mathbf{u}_U \\ \mathbf{u}_C \end{bmatrix} = \begin{bmatrix} \mathbf{f}_U \\ \mathbf{f}_C \end{bmatrix},$$

then the constraint modes can be expressed simply as:

$$\Phi_C = \begin{bmatrix} \mathbf{K}_{UU}^{-1} \mathbf{K}_{UC} \\ \mathbf{I}_{CC} \end{bmatrix}$$

Note also that the factorized linear system obtained by this procedure is precisely the one used to produce eigenmodes with zero amplitude at the constraint DOFs.

Attachment Modes

Similar to constraint modes, attachment modes introduce response at vital locations in the model where sensors or other models are to be attached. Attachment modes also introduce a full spectrum of frequency response, limiting the stable stepsize for explicit methods.

Attachment modes are computed in LS-DYNA at a user-defined set of degrees of freedom. One mode is produced for each such degree of freedom. For each mode, a force vector is applied to the model which has unit value at the corresponding attachment DOF, and zero elsewhere. The nodal displacements produced by this force vector are the attachment mode.

Complications arise when the linear system is singular due to lack of constraints and the presence of rigid body modes. This can be circumvented by applying constraints to eliminate rigid body motion, or by using inertia-relief modes (yet to be implemented in LS-DYNA).

Complications With Constraint and Attachment Modes

The use of constraint and attachment modes complicates the modal solution. Since the frequencies associated with these modes are not necessarily low, they can substantially decrease the stable time step size required for explicit time integration methods. These frequencies can be estimated using the Rayleigh quotient:

$$\omega^2 = \frac{\Phi_i K_{ij} \Phi_j}{\Phi_i M_{ij} \Phi_j}$$

In addition, constraint and attachment modes do not generally produce an uncoupled system of equilibrium equations (1), since the reduced mass and stiffness matrices $\Phi_C^T [\mathbf{M}] \Phi_C$ and $\Phi_C^T [\mathbf{K}] \Phi_C$ are not necessarily diagonal, as they would be if eigenmodes were used. This requires a non-trivial system of simultaneous linear equations to be solved each time step, involving matrix-vector multiplication, and in the case of implicit time integration, matrix inversion. These additional operations appear to be worthwhile, however, since constraint and attachment modes proved in this study to produce much more accurate modal response than did a similar number of eigenmodes.

REFERENCES

Kardestuncer, H. (1987). Finite Element Handbook. McGraw-Hill, New York, ISBN 0-07-033305-X

Wilson, E.L. (2000). Three Dimensional Static and Dynamic Analysis of Structures, Third Edition. Computers and Structures, Inc., ISBN 0-923907-00-9

