Overview of Dynamic Analysis in Abaqus

1. Introduction

Despite static analysis, Abaqus also offers several methods to study dynamic problems. In essence, in a dynamic problem the effect of inertia should be considered in the analysis and the objective is to study the temporal response of the system under specific conditions (for example to simulate the deformation of sheet metals in the metal punching process, to study dynamic loads transmitted to an electronic device when it is dropped on a hard surface, or to study the transient thermal problem).

In dynamic (transient) problems, the solution step often involves direct integration of the system of equations that describe the motion (or other physics) of the system under consideration. Both Abaqus/Standard and Abaqus/Explicit can be used to study these problem. The difference arises from the fact these products adopt different schemes to solve (evolve) the non-linear system of equations that describe the problem. Abaqus/Standard utilizes implicit direct integration schemes for this purpose whereas Abaqus/Explicit employes explicit integration schemes.

In general, explicit schemes are more cost-effective in solving a large non-linear system of equation that describe a dynamic system. Major disadvantage for the explicit methods is the sensitivity of their stability with the choice of the time-step (jump in the time during each solution increment). In the other hand, implicit schemes allow for the selection of larger time-steps, however they impose a larger computational cost for solving a non-linear system. The selection between Explicit and Implicit (Standard) product is sometime limitted by the type of analysis that is needed, and usually it very much depends on the nature of PDE (partial differential equation) describing the physics of the problem (transient analysis is not only limited to the structural problem), type of the boundary conditions describing the system, total amount of analysis time and etc. In the remainder, we will compare these two solution schemes in more details.

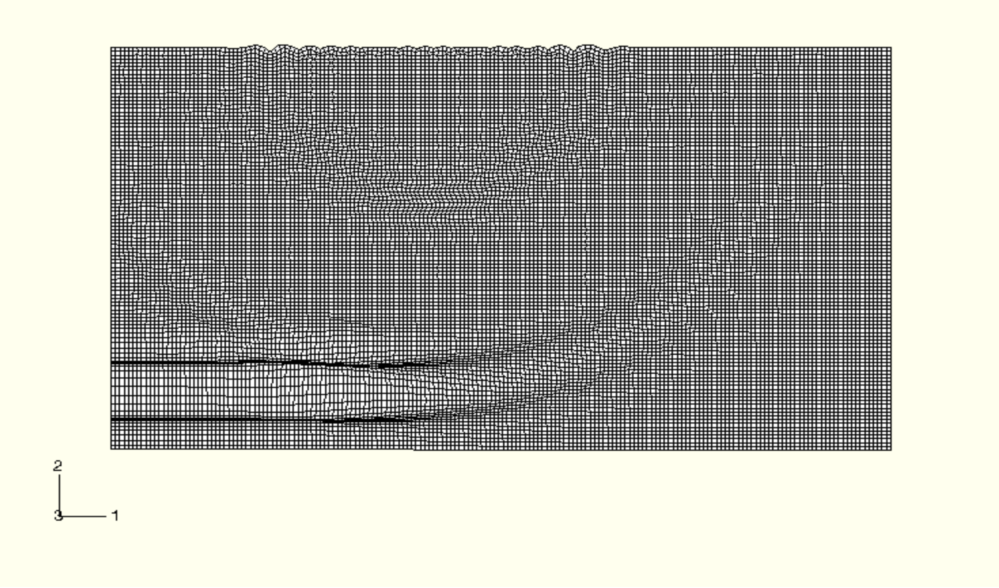

Fig 1- Dynamic analysis of wave propagation in elastic media

2. Implicit versus Explicit Integration Scheme

2.1. Matrix Inversion

Abaqus/Standard provides a choice of implicit operators for the integration purpose, while Abaqus/Explicit uses the central-difference operator. In an implicit dynamic analysis the integration operator matrix must be inverted and a set of nonlinear equilibrium equations must be solved at each time increment. In an explicit dynamic analysis displacements and velocities (applicable to structural problems) are calculated in terms of quantities that are known at the beginning of an increment; therefore, the global mass and stiffness matrices need not be formed and inverted, which means that each increment is relatively inexpensive compared to the increments in an implicit integration scheme.

2.2. Stability

The size of the time increment in an explicit dynamic analysis is limited, because the central-difference operator is only conditionally stable; whereas the implicit operator options available in Abaqus/Standard are unconditionally stable and, thus, there is no such limit on the size of the time increment that can be used for most analyses in Abaqus/Standard (accuracy governs the time increment in Abaqus/Standard).

In Abaqus Explicit, the stability limit for the central-difference method (the largest stable time increment) is closely related to the time required for a stress wave to cross the smallest element dimension in the model; thus, the time increment in an explicit dynamic analysis can be very short if the mesh contains small elements or if the stress wave speed in the material is very high. The method is, therefore, computationally attractive for problems in which the total dynamic response time that must be modeled is only a few orders of magnitude longer than this stability limit; for example, wave propagation studies or some “event and response” applications.

2.3. Element Types

Abaqus/Explicit offers fewer element types than Abaqus/Standard. For example, only first-order, displacement method elements (4-node quadrilaterals, 8-node bricks, etc.) and modified second-order elements are used, and each degree of freedom in the model must have mass or rotary inertia associated with it.

2.4. Advantages of Explicit Method

Main advantages of Abaqus/Explicit are following:

- The analysis cost rises only linearly with problem size, whereas the cost of solving the nonlinear equations associated with implicit integration rises more rapidly than linearly with problem size. Therefore, Abaqus/Explicit is attractive for very large problems (like modeling of vehicle crash)
- The explicit integration method is often more efficient than the implicit integration method for solving extremely discontinuous short-term events or processes (like electronics drop test)
- Problems involving stress wave propagation can be far more efficient computationally in Abaqus/Explicit than in Abaqus/Standard (like modeling of blast waves)

In choosing an approach to a nonlinear dynamic problem you must consider the length of time for which the response is sought compared to the stability limit of the explicit method; the size of the problem; and the restriction of the explicit method to first-order, pure displacement method or modified second-order elements. In some cases the choice is obvious, but in many problems of practical interest the choice depends on details of the specific case. Experience is then the only useful guide.

3. Dynamic Solution Procedures in Abaqus

Generally, dynamic solution techniques in Abaqus can be divided in two types:

- Direct solution: must be used for non-linear problems
- Modal superposition: can be used for linear and mildt non-linear problems

Abaqus provides following direct solution procedures for dynamic analysis:

- Implicit dynamic analysis: can be used problems that involve strongly non-linear transinet dynamic response (this is used in Abaqus/Standard General Step)
- Subspace-based explicit dynamic analysis: the mode shapes (eigen modes) of the system are first extracted using frequency extraction step and then uses as a global basis vector to solve the system using an explicit scheme. This method can be used for mild nonlinear systems that mode shapes are not changed significantly during analysis (therefore can not be used for contact problems)
- Explicit dynamics analysis: this is the method used by Abaqus/Explicit
- Direct-solution steady-state harmonic response analysis: As name implies, it can be used to find steady-state harmonic response of a system. The method of choice when frequency-dependent effects (such as damping, or other model parameters) should be captured

The following modal superposition procedures are provided in Abaqus:

- Mode-based state-state response: can be used to find linearzied system response to harmonic excitations
- Subspace-based steady-state harmonic response analysis: Much cheaper way than direct analysis method to capture frequency-dependent effects
- Mode-based transient response analysis: the modal dynamics analysis procedure that can be used to capture transient response for linear problems using modal superposition
- Response spectrum analysis: can be used to obtain approximate upper bound of the peak significant response of a system to an input spectrum provided by user
- Random response analysis: to capture the linearized response of a system to random excitations
- Complex eigen value extraction: to calculate the complex eigenvalues and the corresponding complex mode shapes of a system

In the sequel more details on some of these solution methods are provides. More detail discussion can be found in the Abaqus Analysis User's Guide Section 6.3.

3.1. Implicit Dynamics Analysis

As mentioned earlier this solution procedure is the most general (and also most expensive) technique available in Abaqus for solving nonlinear problems. This is the method can be used for problems involving nonlinearity (material or geometric), contact and moderate energy dissipation. The procedure can be applied to a broad range of applications that require more sophisticated numerical solution strategies, such as the amount of numerical damping required to obtain convergence and the way in which the automatic time increment algorithm proceeds through the solution. Typical dynamic applications fall into three categories:

- Transient fidelity applications: small time increments are taken to accurately resolve the vibrational response of the structure, and numerical energy dissipation is kept at a minimum for example in the analysis of satellite systems, require minimal energy dissipation. These stringent requirements tend to degrade convergence behavior for simulations involving contact or nonlinearities.
- Moderate dissipation applications: for problem in which moderate amount of energy is dissipated by plasticity, viscous damping, or other effects. Applications include various insertion, impact, and forming analyses. Usually numerical energy dissipation is used in these analyses to reduce solution noise and improve convergence behavior.
- Quasi-static applications: when final static response is only needed and inertia effects are

introduced primarily to regularize unstable behavior. Example of statically unstable behavior include (temporally) unconstrained rigid body modes or “snap-through” phenomena. With adaptive time-stepping strategy, large time increments are taken when possible to obtain the final solution at minimal computational cost. For these problem, considerable numerical dissipation may be required to obtain convergence during certain stages of the loading history.

The classification of the application is determined by the user and Abaqus assigins numerical settings based on this selection. Sometimes accurate results can be obtained with more than one method, therefore the most efficient approach should be used. As a rule of thumb, among these tree classification, high-dissipation quasi-static method results in best convergence behavior.

3.1.1 Time-Integration Method

Abaqus/Standard uses the Hilber-Hughes-Taylor (an extension of Newmark β-method) time integration by default and backward Euler method is used for quasi-static classification. As mentioned earlier for these implicit methods, the tangent matrix should be evaluated and inverted and the non-linear system of equations is solved in each time increment iteratively using Newton's method. The method is unconditionally stable and this is its great advantage compared to conditionally stable explicit methods. Application of a finite time-step may result in numerical damping (energy dissipation, completely different from damping as a material property but same effect) from the system. The amount of energy dissipation can vary among different operator types (for example backward Euler is more dissipative than Hilber-Hughes-Taylor) and can be controlled by tweaking the numerical settings of the method. Table 1 summarizes default numerical values for the parameters used in Abaqus for different applications. User can specify the value for α ($-0.5 <= \alpha < 0$) and the other two parameters are calculated according to this value based on the following relations:

$$\beta = 0.25(1 - \alpha)^2$$
$$\gamma = 0.5 - \alpha$$

Table 1- Defaule parameters for Hilber-Hughes-Taylore time integration scheme

Parameter	Application	
	Transient Fidelity	Moderate Dissipation
α	–0.05	–0.41421
β	0.275625	0.5
γ	0.55	0.91421

The ability of the operator to effectively treat contact conditions is often of considerable importance with respect to their usefulness. For example, some changes in contact conditions can result in “negative damping” (nonphysical energy source) for many time integrators, which can be very undesirable.

3.1.2. Incrementation Scheme

In this analysis, Abaqus uses automatic time incrementation method. The time-step is adjusted depending on the behavior of the Newton iteration and the accuracy of the time integration. For quasi-static analysis, the value of time increment is adjusted based on these criteria:

- The time increment size is reduced if an increment appears to be diverging or if the convergence rate is slow

- The time increment size is reduced if an increment appears to be diverging or if the convergence rate is slow

For moderate dissipation problems same criteria is used plus the maximum allowable time step is set to one-tenth of the step duration. For the transient fidelity analysis the following criteria used to decide about the time step:

- The time increment size is reduced if an increment appears to be diverging or if the convergence rate is slow.
- The time increment size is reduced if changes in contact status are detected during the first attempt of processing an increment. The new increment size is set such that the end of the increment corresponds to the average time of the contact status changes that were detected with the previous increment size. (In such cases an additional very small time increment is used to enforce compatibility of velocities and accelerations across active contact interfaces.)
- The time increment size is reduced if the half-increment residual (out-of-balance force) halfway through a time increment exceeds the half-increment residual tolerance, which is 10,000 times the time average force for a contact analysis or 1000 times the time average force for an analysis without contact.
- The time increment is gradually increased if rapid convergence occurs in previous increments.
- The upper bound for the time increment size is equal to 1/100 of the step duration.

Direct time incrementation can also specified by the user and set to a constant value (generally not recommended). In this case, analysis terminates if convergence tolerances are not satisfied with the maximum number of iterations specified by the user.

3.2. Explicit Dynamic Analysis

As mentioned earlier, explicit analysis is recommended under the following conditions:

- large dynamics problems
- when general/complicated contact conditions are needed (like crash worthiness simulations)
- when large deformations and rotations are happening
- can be used to perform an adiabatic stress analysis if inelastic dissipation generate heat in materials (for example in large strain-rate deformation)
- allows for both automatic and fixed time incrementation

3.2.1. Time-Integration Method

The equation of the motion for the body is integrated using a central-difference integration rule:

$$\dot{u}^N_{(i+\frac{1}{2})} = \dot{u}^N_{(i-\frac{1}{2})} + \frac{\Delta t_{(i+1)} + \Delta t_{(i)}}{2} \ddot{u}^N_{(i)},$$

$$u^N_{(i+1)} = u^N_{(i)} + \Delta t_{(i+1)} \dot{u}^N_{(i+\frac{1}{2})},$$

where u^N refers to the degree of freedom (Displacement/Rotation) numerb N , subscript i refers to the increment number, Δt is the time-step. $\mathbf{u}^{\cdot}$ and $\mathbf{u}^{\cdot\cdot}$ are velocity and accelerations of the previous time step. The efficiency of the explicit time-step originates from the use of diagonal (lumped) element mass matrices since the accelerations at the beginning of the increment are calculated by

$$\ddot{u}^N_{(i)} = (M^{NJ})^{-1}(P^J_{(i)} - I^J_{(i)}),$$

where M^{NJ} is the lumped mass matrix. P^J is the applied load vector, and I^J is the internal force vector. The diagonal mass matrix can be simply inverted. Other reason for added computational efficiency of the method is the fact that the tangent stiffness matrix is not needed and no iteration need to be performed during analysis. The internal force vector is assembled by element contributions.

3.2.2 Stability

When no damping exist, the central difference time integration scheme is stable when

$$\Delta t \leq \frac{2}{\omega_{max}}.$$

where ω_{max} is the highest frequency of the system. With damping when the stability condition is

$$\Delta t \leq \frac{2}{\omega_{max}}(\sqrt{1+\xi_{max}^2} - \xi_{max}),$$

where ξ_{max} is the fraction of critical damping in the mode with highest frequency. As can be noted, introduction of damping reduces the stable time-step size.

3.2.3 Stable Time Increment

Since highest frequency of the system is not known a-priori (or expensive to calculate) an approximation to the stable time increment is often using smallest transit time of a dilatational wave across every element in the mesh

$$\Delta t \approx \frac{L_{min}}{c_d},$$

where L_{min} is the smallest element dimension in the mesh and c_d is the dilatation wave speed. Since this is not a conservative approximate value for the time-step, Abaqus/Explicit uses a smaller factor of this value (between 1 and $1/\sqrt{2}$).

3.2.4 Time Incrementation and Scaling

As mentioned, the time-step used in explicit schemes is limited to the stability bound. A larger values for time-step cause unbounded oscillation in the solution field (and energy) and finally cause divergence. User can scale the stable time-step estimate to reduce the chance of divergence.

3.2.5 Automatic Time Incrementation

Abaqus by default uses and automatic time incrementation scheme. Two types of stable time-step estimation is used for this purpose:

- Element by element estimate : more conservative criteria which uses dillatational wave speed for all elements in the system (compared to a global value for the whole model) to estimate the stable time-step
- Global estimate : less conservative, will be switched to this once algorithm determines it has a reasonable accuracy

3.2.6 Applications

The explicit dynamics procedure is ideally suited for analyzing high-speed dynamic events, but many of the advantages of the explicit procedure also apply to the analysis of slower (quasi-static) processes. A good example is sheet metal forming, where contact dominates the solution and local instabilities may form due to wrinkling of the sheet.

4. Reference

1- Abaqus 6.13 Documentation