Implementation of Dynamic Time-Integration Notes

The Hilber-Hughes-Talyor (HHT) scheme is used in the following form:

**M** **a** = (1+α ) **b** – α **b**0

**d** = **d**pr + β Δt2 **a** , **v** = **v**pr + γ Δt **a**

**d**pr = **d**0 + Δt **v**0 + ( 1/2 - β ) Δt2 **a**0 , **v**pr = **v**0 + (1-γ ) Δt **a**0

γ = 1/2 – α , β = (1-α )2/4

where

α = free parameter controlling algorithmic damping in range -1/3 to 0.

**M** = mass matrix

**a**0, **a** = vector of accelrations at time t and t+Δt, respectively.

**v**0, **v** = vector of velocities at time t and t+Δt, respectively.

**d**0, **d** = vector of displacements or rotations at time t and t+Δt, respectively.

**b**0, **b** = net out of balance forces at time t and t+Δt, respectively; includes all out-of-balance forces other than those captured by the mass matrix. It is envisioned that the exact[[1]](#footnote-1) total out of balance forces at any time are b(d,v) – M a. However the HHT approximations require these two terms to be treated differently, and therefore they are treated separately. This would not be necessary for α=0.

Δt = time increment

HHT recommend to initiate the scheme is initiated using **M** **a** = **b**.

Notes:

1. a0 cannot be calculated from information at time t0 & t alone. Therefore accelerations must be stored, along with the velocity and displacement.
2. Whereas HHT recommend to initiate the scheme using **M** **a** = **b**, this would require a non-singular mass matrix. Therefore it is assumed that initially both **a** and **b** are zero, unless an IEXFLAG option is activated to read the initial values of **a** and **b**.

Implementation procedure:

1. For a given increment, **d** is the unknown displacement at the end of the increment to be determined by Newton iteration. The corresponding velocities and accelerations are calculated from:  
     
   **a** = (**d** - **d**pr)/(β Δt2)  
   **v** = **v**pr + γ (**d** - **d**pr)/(β Δt)  
     
   This calculation is performed outside the element routines, and the resulting velocity v is passed to the element routines. In addition DVDD=γ /(β Δt) is passed to element routines.
2. The element routines calculate the following:   
   **b**=**b**(**d**,**v**)  
   **k**t = ∂(**b**,**d**) + DVDD ∂(**b**,**v**) = effective tangent stiffness  
   **M** = mass matrix (assumed not to depend on **d** and **v**)  
     
   where ∂(**y**,**x**) denotes a matrix in which the jth column is the partial derivative of **y** with respect to the jth element of **x**.  
     
   The algorithms are designed for a consistent mass matrix, but the element routine may of course return a diagonal lumped mass matrix.
3. Outside the element routines, the residual for the time-discretized system, is formed as  
   (1+α) **b** – α **b**0 – **Ma**  
   where the first and last terms are assembled and the middle **b**0 is stored from the previous step. Further the Jacobian is formed as,  
   **J** = (1+α) **k**t + **M**/(β Δt2)  
   This Jacobian and residual is used in the Newton Iterations.
4. The routines to check the consistency of the tangents need to check the consistency of **k**t.
5. Dynamic analysis is a separate IEXFLAG option that does not combine with the Riks arclength method, the user-specified arclength increment will be used as the time increment.
6. The velocity and acceleration is used only when the dynamic analysis option is used. Otherwise velocity and acceleration are assumed to be zero for the purpose of calculating the out of balance forces.

1. Here “exact” refers to time integration only, and not spatial discretization and other approximations made. For “exact” time integration the “exact” total out-of-balance forces are zero at all times. [↑](#footnote-ref-1)