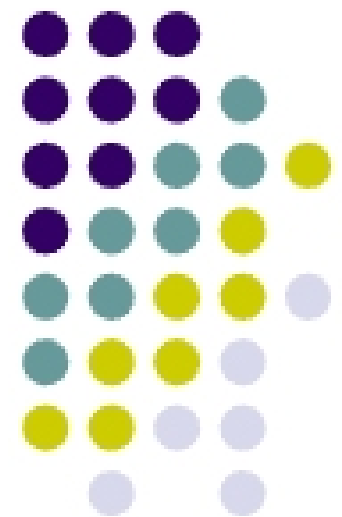


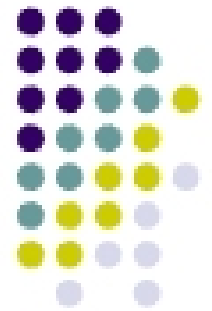
ME451

Kinematics and Dynamics of Machine Systems

Dynamics of Planar Systems
December 9, 2010
Solving Index 3 DAEs using Newmark Method



Before we get started...



- Last Time:
 - Finish the discussion started last time
 - Marked the end of the Dynamics Analysis of Mechanical Systems chapter
- Today
 - Discuss the final exam
 - Inverse Dynamics Analysis
 - Equilibrium Analysis
 - We hug each other and stuff
- Final Exam
 - Tu, Dec. 21 at 5:05 PM, Room: **ME1245**

Done once, at the very beginning of the dynamics simulation

Find initial conditions for general coordinate positions and velocities that satisfy position and velocity constraint equations

Use Newton's equations of motion along with the acceleration constraint equations to find general coordinate accelerations and lambdas at time = 0.

$$\begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q^A \\ \gamma \end{bmatrix}$$

At each time step (solving for the state of the system at t_{n+1} ...)

Increment $t_{n+1} = t_n + h$
Define initial guess for accelerations \ddot{q} and lambdas (take values from previous time step t_n)

Iterate to solve nonlinear system obtained after Newmark discretization of DAEs.

Update t_{n+1} position and velocity using Newmark's formulas and the most recent values for acceleration and lambdas.

$$\begin{cases} q_{n+1} = q_n + h\dot{q}_n + \frac{h^2}{2} [(1-2\beta)\ddot{q}_n + 2\beta\ddot{q}_{n+1}] \\ \dot{q}_{n+1} = \dot{q}_n + h[(1-\gamma)\ddot{q}_n + \gamma\ddot{q}_{n+1}] \end{cases}$$

Get Jacobian of discretized equation. Use most recent q , \dot{q} , \ddot{q} , and lambda at $n+1$

$$\hat{J} = \begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix}$$

Use most recent values at $n+1$ to compute what each equation equals. Each equation will probably not equal zero. These values are called the residuals.

$$\begin{cases} M\ddot{q} + \Phi_q^T(q)\lambda - Q^A(q, \dot{q}, t) = 0 \\ \frac{1}{2} \dot{\Phi}_q(q, \dot{q}, t) = 0 \end{cases}$$

Yes. Need to refine current acc/Lagr. Multiplier values

Compute the correction vector.

$$\begin{bmatrix} \delta\ddot{q} \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix}^{-1} \cdot \Psi(q^{(old)}, \lambda^{(old)})$$

Correct the most recent acceleration and lambda to get better approximations for accelerations and lambdas.

$$\begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix}^{(new)} = \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix}^{(old)} - \begin{bmatrix} \delta\ddot{q} \\ \delta\lambda \end{bmatrix} \quad \text{residual}$$

Compute the largest value present in the correction factor matrix and compare to a specified limit. This is the convergence check (largest correction is easily computed as infinity norm in MATLAB)

Is $\|\delta\ddot{q}\| > 10^{-3}$?

(You might have to play with the value 10^{-3} a bit. 10^{-3} is just a good rule of thumb.)

No.

Store acceleration and lambda for time t_{n+1} . Use this accelerations to compute the position and velocity at t_{n+1} and store these quantities as well.

