ME451
Kinematics and Dynamics of Machine Systems

Introduction to Dynamics

Newmark Integration Formula

[not in the textbook]

December 9, 2014

Dan Negrut
ME451, Fall 2014
University of Wisconsin-Madison

- Quote of the day: “They have computers, and they may have other weapons of mass destruction.”

-- Janet Reno, while US Attorney General

Before we get started…

- Last time[s]
  - Numerical Integration
  - Exam
- Today
  - Solving the constrained equations of motion using the Newmark integration formulas
  - Critical for implementation of simEngine2D
- Project 2 due on 12/16 at 11:59 PM
- Dropped HW policies
  - Lowest 6 scores amongst the MATLAB, pen-and-paper, and ADAMS assignments will be dropped
- Exams graded, scores in Learn@UW
  - Please come to see me this week if you think score doesn't reflect the quality of your work

Before we get started…

- Final Exam: content
  - Part 1: Pen and paper
    - You'll have to generate a pair of acf/adm files but you don't have to use these files unless you go for the bonus
  - Part 2: Bonus (extra credit)
    - You'll have to use simEngine2D and the pair of acf/adm files
  - Score cannot exceed 100%

- Final Exam: logistics
  - Tuesday, December 16, 2014
  - 2:45 PM - 4:45 PM
  - Room: 2109ME (computer lab)
  - MATLAB access – one of two choices:
    - Bring your own laptop
    - Use CAE machine

- Final Project
  - Due on Friday, December 19 at 11:59 PM

Solution Strategy

[Step 3 of the "Three Steps for Dynamics Analysis", see Slide 25]

The numerical solution; i.e., an approximation of the actual solution of the dynamics problem, is produced in the following three stages:

- **Stage 1**: the Newmark numerical integration (discretization) formulas are used to **express the positions and velocities as functions of accelerations**

- **Stage 2**: everywhere in the constrained EOM, the positions and velocities are **replaced using the Newmark numerical integration formulas** and expressed in terms of the acceleration
  - This is the most important step, since through this "discretization" the **differential problem is transformed into an algebraic problem**

- **Stage 3**: the unknowns; i.e., the acceleration and Lagrange multipliers are obtained by **solving a nonlinear system**

Solution Strategy

Ease Into It – Solve Simpler Problem First

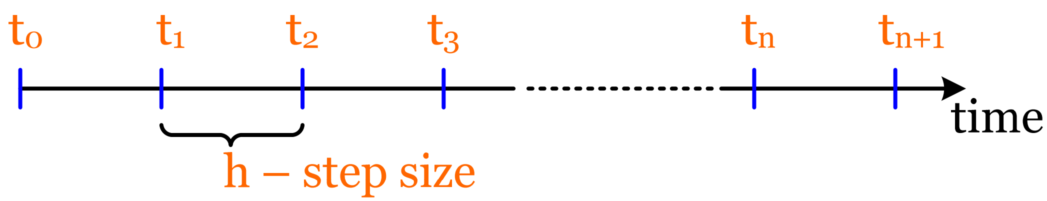
- Solve a Finite Element Analysis (FEA) problem first, then move to DAE
- Linear FEA leads to the following second order differential equation:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}(t)$$

- Not quite our problem, but good stepping stone
  - Square matrices $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are constant
  - $\mathbf{F}(t)$ is the forcing term, time dependent

Newmark Integration Formulas (1/2)

- Goal: find the positions, velocities, accelerations and Lagrange multipliers on a grid of time points; i.e., at $t_0, t_1, t_2, \ldots$



- <u>Stage 1/3</u> – Newmark's formulas relate *position to acceleration* and *velocity to acceleration*:

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right] \equiv \mathbf{p}(\ddot{\mathbf{q}}_{n+1})$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right] \equiv \mathbf{v}(\ddot{\mathbf{q}}_{n+1})$$

- <u>Stage 2/3</u> – Newmark's method (1957) discretizes the second order EOM:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}(t) \quad \Leftrightarrow \quad \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{C}\dot{\mathbf{q}}_{n+1} + \mathbf{K}\mathbf{q}_{n+1} = \mathbf{F}(t_{n+1})$$

Newmark Integration Formulas (2/2)

- Newmark Method
  - Initially introduced to deal with linear transient Finite Element Analysis
  - Accuracy: 1st Order
  - Stability: Very good stability properties
  - Choose values for the two parameters controlling the behavior of the method: $\beta = 0.3025$ and $\gamma = 0.6$

- Write the EOM at each time t_{n+1}

$$\mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{C}\dot{\mathbf{q}}_{n+1} + \mathbf{K}\mathbf{q}_{n+1} = \mathbf{F}(t_{n+1})$$

- Use the discretization formulas to replace $\mathbf{q}_{n+1}$ and $\dot{\mathbf{q}}_{n+1}$ in terms of the accelerations $\ddot{\mathbf{q}}_{n+1}$ using formulas on previous slide:

$$\mathbf{q}_{n+1} = \mathbf{p}(\ddot{\mathbf{q}}_{n+1}) \quad \text{and} \quad \dot{\mathbf{q}}_{n+1} = \mathbf{v}(\ddot{\mathbf{q}}_{n+1})$$

- Obtain algebraic problem in which the unknown is the acceleration (denoted here by $\mathbf{x}$):

$$\mathbf{M} \cdot \mathbf{x} + \mathbf{C} \cdot \mathbf{v}(\mathbf{x}) + \mathbf{K} \cdot \mathbf{p}(\mathbf{x}) = \mathbf{F}(t_{n+1})$$

DAEs of Constrained Multibody Dynamics

- The rigid multibody dynamics problem is more complicated than the Linear Finite Element problem used to introduce Newmark's formulas
  - Additional algebraic equations: kinematic constraints that solution must satisfy
  - Additional algebraic variables: the Lagrange multipliers that come along with these constraints

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{F}(t)$$

Linear Finite Element Dynamics Problem

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^T \lambda - \mathbf{Q}^A(\dot{\mathbf{q}}, \mathbf{q}, t) = \mathbf{0} \\ \mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \end{cases}$$

Nonlinear Multibody Dynamics Problem

- Newmark's method can be applied for the DAE problem, with slightly more complexity in the resulting algebraic problem.

Stage 3/3: Discretization of the Constrained EOM (1/3)

- The discretized equations solved at each time t_{n+1} are:

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{\Phi}_{\mathbf{q}}^{T}(\mathbf{q}_{n+1})\lambda_{n+1} - \mathbf{Q}^{A}(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1}) = \mathbf{0} \\ \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) = \mathbf{0} \end{cases}$$

- Recall that $\mathbf{q}_{n+1}$ and $\dot{\mathbf{q}}_{n+1}$ in the above expressions are functions of the accelerations $\ddot{\mathbf{q}}_{n+1}$:

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right] \equiv \mathbf{p}(\ddot{\mathbf{q}}_{n+1})$$

$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right] \equiv \mathbf{v}(\ddot{\mathbf{q}}_{n+1})$$

Recall, these are Newmark's formulas that express the generalized positions and velocities as functions of the generalized accelerations

Stage 3/3:
Discretization of the Constrained EOM (2/3)

- The unknowns are the accelerations and the Lagrange multipliers
  - The number of unknowns is equal to the number of equations

- The equations that must be solved now are algebraic and nonlinear
  - Differential problem has been transformed into an algebraic one
  - The new problem: find acceleration and Lagrange multipliers that satisfy

$$\begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{\Phi}_{\mathbf{q}}^T(\mathbf{q}_{n+1})\boldsymbol{\lambda}_{n+1} - \mathbf{Q}^A(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1}) \\ \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) \end{bmatrix} = \mathbf{0}$$

- We have to use Newton's method
  - We need the Jacobian of the nonlinear system of equations (chain rule will be used to simplify calculations)
  - This looks exactly like what we had to do when for Kinematics analysis of a mechanism (there we solved $\Phi(\mathbf{q},t)=0$ to get the positions $\mathbf{q}$)

Stage 3/3: Discretization of the Constrained EOM (3/3)

- Define the following two functions:

$$\bar{\mathbf{\Psi}}(\ddot{\mathbf{q}}_{n+1}, \dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, \lambda_{n+1}) \triangleq \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{\Phi}_{\mathbf{q}}^{T}(\mathbf{q}_{n+1})\lambda_{n+1} - \mathbf{Q}^{A}(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1})$$

$$\bar{\mathbf{\Omega}}(\mathbf{q}_{n+1}) \triangleq \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}_{n+1}, t_{n+1})$$

- Once we use the Newmark discretization formulas, these functions depend in fact only on the accelerations $\ddot{\mathbf{q}}_{n+1}$ and Lagrange multipliers $\boldsymbol{\lambda}_{n+1}$

- To make this clear, define the new functions:

$$\mathbf{\Psi}(\ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}) \equiv \bar{\mathbf{\Psi}}(\ddot{\mathbf{q}}_{n+1}, \dot{\mathbf{q}}_{n+1}(\ddot{\mathbf{q}}_{n+1}), \mathbf{q}_{n+1}(\ddot{\mathbf{q}}_{n+1}), \lambda_{n+1})$$

$$\mathbf{\Omega}(\ddot{\mathbf{q}}_{n+1}) \equiv \bar{\mathbf{\Omega}}(\mathbf{q}_{n+1}(\ddot{\mathbf{q}}_{n+1}))$$

- Therefore, we must solve for $\ddot{\mathbf{q}}_{n+1}$ and $\boldsymbol{\lambda}_{n+1}$ the following system

$$\begin{bmatrix} \mathbf{\Psi}(\ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}) \\ \mathbf{\Omega}(\ddot{\mathbf{q}}_{n+1}) \end{bmatrix} = \mathbf{0}$$

Chain Rule for Computing the Jacobian (1/3)

- Newton's method for the solution of the nonlinear system

$$\begin{bmatrix} \mathbf{\Psi}(\ddot{\mathbf{q}}_{n+1}, \boldsymbol{\lambda}_{n+1}) \\ \mathbf{\Omega}(\ddot{\mathbf{q}}_{n+1}) \end{bmatrix} = \mathbf{0}$$

relies on the Jacobian

$$\begin{bmatrix} \frac{\partial \mathbf{\Psi}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \mathbf{\Psi}}{\partial \boldsymbol{\lambda}_{n+1}} \\ \frac{\partial \mathbf{\Omega}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \mathbf{\Omega}}{\partial \boldsymbol{\lambda}_{n+1}} \end{bmatrix}$$

- Use the chain rule to calculate the above partial derivatives.

- Note that, from the Newmark formulas we get

$$\frac{\partial \mathbf{q}_{n+1}}{\partial \ddot{\mathbf{q}}_{n+1}} = \frac{\partial \mathbf{p}(\ddot{\mathbf{q}}_{n+1})}{\partial \ddot{\mathbf{q}}_{n+1}} = \beta h^2 \mathbf{I}_{nc \times nc} \qquad \frac{\partial \dot{\mathbf{q}}_{n+1}}{\partial \ddot{\mathbf{q}}_{n+1}} = \frac{\partial \mathbf{v}(\ddot{\mathbf{q}}_{n+1})}{\partial \ddot{\mathbf{q}}_{n+1}} = \gamma h \mathbf{I}_{nc \times nc}$$

Chain Rule for Computing the Jacobian (2/3)

- Consider

$$\begin{aligned}\mathbf{\Psi}(\ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}) &= \bar{\mathbf{\Psi}}(\ddot{\mathbf{q}}_{n+1}, \dot{\mathbf{q}}_{n+1}(\ddot{\mathbf{q}}_{n+1}), \mathbf{q}_{n+1}(\ddot{\mathbf{q}}_{n+1}), \lambda_{n+1}) \\ &= \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{\Phi}_{\mathbf{q}}^T(\mathbf{q}_{n+1})\lambda_{n+1} - \mathbf{Q}^A(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1})\end{aligned}$$

- Apply the chain rule of differentiation to obtain

$$\frac{\partial \mathbf{\Psi}}{\partial \ddot{\mathbf{q}}_{n+1}} = \frac{\partial \bar{\mathbf{\Psi}}}{\partial \ddot{\mathbf{q}}_{n+1}} + \frac{\partial \bar{\mathbf{\Psi}}}{\partial \dot{\mathbf{q}}_{n+1}}\frac{\partial \dot{\mathbf{q}}_{n+1}}{\partial \ddot{\mathbf{q}}_{n+1}} + \frac{\partial \bar{\mathbf{\Psi}}}{\partial \mathbf{q}_{n+1}}\frac{\partial \mathbf{q}_{n+1}}{\partial \ddot{\mathbf{q}}_{n+1}} = \frac{\partial \bar{\mathbf{\Psi}}}{\partial \ddot{\mathbf{q}}_{n+1}} + \gamma h \frac{\partial \bar{\mathbf{\Psi}}}{\partial \dot{\mathbf{q}}_{n+1}} + \beta h^2 \frac{\partial \bar{\mathbf{\Psi}}}{\partial \mathbf{q}_{n+1}}$$

$$\frac{\partial \mathbf{\Psi}}{\partial \ddot{\mathbf{q}}_{n+1}} = \mathbf{M} + \gamma h \left(-\frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}_{n+1}} \right) + \beta h^2 \left(\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}_{n+1}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}_{n+1}} \right)$$

and

$$\frac{\partial \mathbf{\Psi}}{\partial \lambda_{n+1}} = \mathbf{\Phi}_{\mathbf{q}}^T$$

Chain Rule for Computing the Jacobian (3/3)

- Consider

$$\mathbf{\Omega}(\ddot{\mathbf{q}}_{n+1}) = \bar{\mathbf{\Omega}}(\mathbf{q}_{n+1}(\ddot{\mathbf{q}}_{n+1})) = \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}_{n+1}, t_{n+1})$$

- Apply the chain rule of differentiation to obtain

$$\frac{\partial \mathbf{\Omega}}{\partial \ddot{\mathbf{q}}_{n+1}} = \frac{\partial \bar{\mathbf{\Omega}}}{\partial \mathbf{q}_{n+1}} \frac{\partial \mathbf{q}_{n+1}}{\partial \ddot{\mathbf{q}}_{n+1}} = \beta h^2 \frac{\partial \bar{\mathbf{\Omega}}}{\partial \mathbf{q}_{n+1}} = \beta h^2 \left(\frac{1}{\beta h^2} \mathbf{\Phi}_{\mathbf{q}} \right)$$

$$\boxed{\frac{\partial \mathbf{\Omega}}{\partial \ddot{\mathbf{q}}_{n+1}} = \mathbf{\Phi}_{\mathbf{q}}}$$

and

$$\boxed{\frac{\partial \mathbf{\Omega}}{\partial \lambda_{n+1}} = \mathbf{0}}$$

Solving the Nonlinear System

- Newton's method applied to the system

$$\begin{bmatrix} \mathbf{\Psi}(\ddot{\mathbf{q}}, \lambda) \\ \mathbf{\Omega}(\ddot{\mathbf{q}}) \end{bmatrix} = \mathbf{0}$$

- Jacobian obtained as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{\Psi}}{\partial \ddot{\mathbf{q}}} & \frac{\partial \mathbf{\Psi}}{\partial \lambda} \\ \frac{\partial \mathbf{\Omega}}{\partial \ddot{\mathbf{q}}} & \frac{\partial \mathbf{\Omega}}{\partial \lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}} + \beta h^2 \left(\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}} \right) & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

- Corrections computed as

$$\begin{bmatrix} \Delta \ddot{\mathbf{q}} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}} + \beta h^2 \left(\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}} \right) & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{\Psi}(\ddot{\mathbf{q}}^{(old)}, \lambda^{(old)}) \\ \mathbf{\Omega}(\ddot{\mathbf{q}}^{(old)}) \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(new)} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(old)} - \begin{bmatrix} \Delta \ddot{\mathbf{q}} \\ \Delta \lambda \end{bmatrix}$$

Note: to keep notation simple, all subscripts were dropped. Recall that all quantities are evaluated at time t_{n+1}

Newton Method for Dynamics

At the initial time t_0

Find consistent initial conditions for generalized positions and velocities

Calculate the generalized accelerations and Lagrange multipliers

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

At each integration time step

Increment time: $t_{n+1} = t_n + h$
Define the initial guess for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ to be the values from the previous time step

Update positions and velocities at t_{n+1} using the Newmark formulas using the current accelerations and Lagrange multipliers

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right]$$
$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right]$$

Calculate the Jacobian matrix, using the current values of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\boldsymbol{\lambda}$ at t_{n+1}

$$\mathbf{J} = \begin{bmatrix} \mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}} + \beta h^2 \left(\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}} \right) & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

Evaluate the EOM and scaled constraints, using the current values of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\boldsymbol{\lambda}$ at t_{n+1}. The resulting vector is called the residual vector.

$$\begin{bmatrix} \mathbf{\Psi} \\ \mathbf{\Omega} \end{bmatrix}^{(old)} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^T(\mathbf{q})\lambda - \mathbf{Q}^A(\dot{\mathbf{q}}, \mathbf{q}, t) \\ \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}, t) \end{bmatrix}$$

Compute the correction vector by solving a linear system with the Jacobian as the system coefficient matrix and the residual as the RHS vector.

$$\mathbf{J} \cdot \begin{bmatrix} \Delta\ddot{\mathbf{q}} \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi} \\ \mathbf{\Omega} \end{bmatrix}^{(old)}$$

Correct the accelerations and Lagrange multipliers to obtain a better approximation for their values at time t_{n+1}

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(new)} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(old)} - \begin{bmatrix} \Delta\ddot{\mathbf{q}} \\ \Delta\lambda \end{bmatrix}$$

Compute the infinity norm of the correction vector (the largest entry in absolute value) which will be used in the convergence test

$$\mathrm{err} = \left\| \begin{matrix} \Delta\ddot{\mathbf{q}} \\ \Delta\lambda \end{matrix} \right\|_\infty$$

Is error less than tolerance?

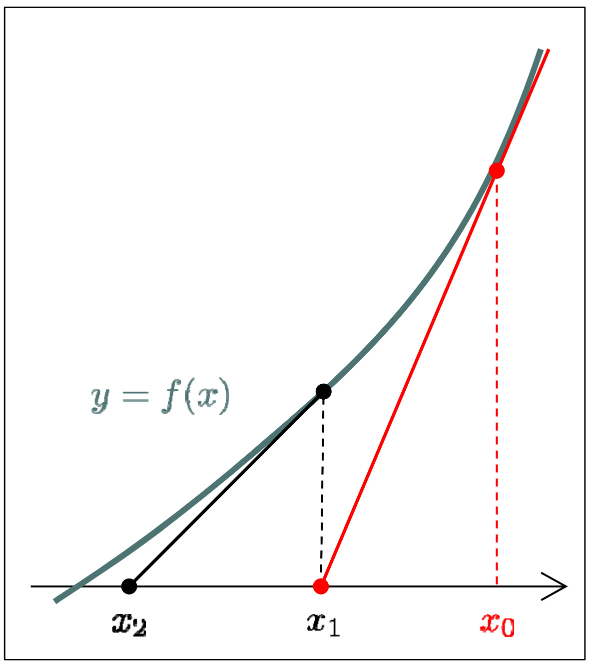
NO
Need to further improve accelerations and Lagrange multipliers

YES

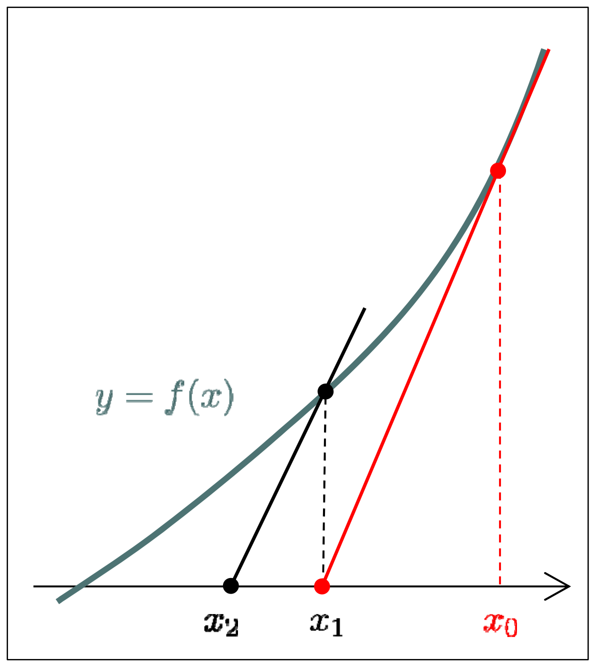
Store $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ at t_{n+1}. Use the final acceleration values to calculate positions and velocities $\mathbf{q}$ and $\dot{\mathbf{q}}$ at t_{n+1}. Use the final Lagrange multiplier values to calculate reaction forces. Store all this information.

Newton-Type Methods
Geometric Interpretation

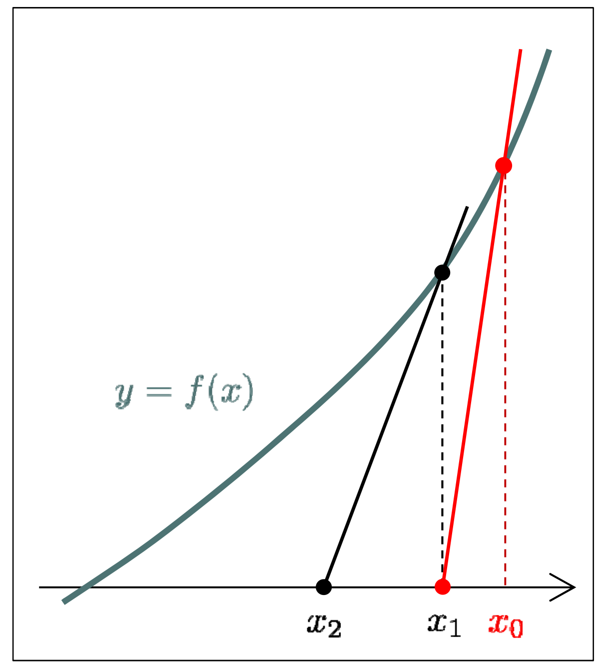
Newton method
At each iterate, use the direction given by the current derivative



Modified Newton method
At all iterates, use the direction given by the derivative at the initial guess



Quasi Newton method
At each iterate, use a direction that only approximates the derivative



Quasi Newton Method for the Dynamics Problem (1/3)

- Nonlinear problem: find $\ddot{q}_{n+1}$ and λ_{n+1} by solving

$$\begin{bmatrix} \mathbf{\Psi}(\ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}) \\ \mathbf{\Omega}(\ddot{\mathbf{q}}_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \mathbf{\Phi}_{\mathbf{q}}^T(\mathbf{q}_{n+1})\lambda_{n+1} - \mathbf{Q}^A(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1}) \\ \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) \end{bmatrix}$$

- Jacobian obtained as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{\Psi}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \mathbf{\Psi}}{\partial \lambda_{n+1}} \\ \frac{\partial \mathbf{\Omega}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \mathbf{\Omega}}{\partial \lambda_{n+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}_{n+1}} + \beta h^2 \left(\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda_{n+1})}{\partial \mathbf{q}_{n+1}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}_{n+1}} \right) & \mathbf{\Phi}_{\mathbf{q}}^T \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

- Terms that we have not computed previously:
  - Partial derivative of reaction forces with respect to positions $\frac{\partial (\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}}$
  - Partial derivative of applied forces with respect to positions $\frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}}$
  - Partial derivative of applied forces with respect to velocities $\frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}}$

Quasi Newton Method for the Dynamics Problem (2/3)

- Approximate the Jacobian by ignoring these terms
- Nonlinear equations:

$$\begin{bmatrix} \boldsymbol{\Psi}(\ddot{\mathbf{q}}_{n+1}, \lambda_{n+1}) \\ \boldsymbol{\Omega}(\ddot{\mathbf{q}}_{n+1}) \end{bmatrix} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}}_{n+1} + \boldsymbol{\Phi}_{\mathbf{q}}^T(\mathbf{q}_{n+1})\lambda_{n+1} - \mathbf{Q}^A(\dot{\mathbf{q}}_{n+1}, \mathbf{q}_{n+1}, t_{n+1}) \\ \frac{1}{\beta h^2}\boldsymbol{\Phi}(\mathbf{q}_{n+1}, t_{n+1}) \end{bmatrix}$$

- Exact Jacobian:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \boldsymbol{\Psi}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \boldsymbol{\Psi}}{\partial \lambda_{n+1}} \\ \frac{\partial \boldsymbol{\Omega}}{\partial \ddot{\mathbf{q}}_{n+1}} & \frac{\partial \boldsymbol{\Omega}}{\partial \lambda_{n+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}_{n+1}} + \beta h^2 \left(\frac{\partial (\boldsymbol{\Phi}_{\mathbf{q}}^T \lambda_{n+1})}{\partial \mathbf{q}_{n+1}} - \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}_{n+1}} \right) & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

- Approximate Jacobian:

$$\tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^T \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}$$

- Therefore, we modify the solution procedure to use a Quasi Newton method

Quasi Newton Method for the Dynamics Problem (3/3)

- The actual terms dropped from the expression of the exact Jacobian

$$\beta h^2 \frac{\partial(\mathbf{\Phi}_{\mathbf{q}}^T \lambda)}{\partial \mathbf{q}} \qquad \beta h^2 \frac{\partial \mathbf{Q}^A}{\partial \mathbf{q}} \qquad \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{\mathbf{q}}}$$

- Is it acceptable to neglect these terms? Under what conditions?
  - As a rule of thumb, this is fine for small values of the step-size; e.g. $h \approx 0.001$
  - But there is no guarantee and smaller values of h may be required

- Note that the terms that we are neglecting are in fact straight-forward to compute

- A production-level multibody package (such as ADAMS) would evaluate these quantities

Quasi Newton Method for Dynamics

At each integration time step

At the initial time t_0

Find consistent initial conditions for generalized positions and velocities.

Calculate the generalized accelerations and Lagrange multipliers.

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

Increment time: $t_{n+1} = t_n + h$.
Define the initial guess for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ to be the values from the previous time step.

Update positions and velocities at t_{n+1} using the Newmark formulas using the current accelerations and Lagrange multipliers.

$$\mathbf{q}_{n+1} = \mathbf{q}_n + h\dot{\mathbf{q}}_n + \frac{h^2}{2}\left[(1-2\beta)\ddot{\mathbf{q}}_n + 2\beta\ddot{\mathbf{q}}_{n+1}\right]$$
$$\dot{\mathbf{q}}_{n+1} = \dot{\mathbf{q}}_n + h\left[(1-\gamma)\ddot{\mathbf{q}}_n + \gamma\ddot{\mathbf{q}}_{n+1}\right]$$

Calculate the approximate Jacobian matrix. Only evaluate this matrix at the first iteration and reuse it at subsequent iterations.

$$\tilde{\mathbf{J}} = \begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_\mathbf{q}^T \\ \mathbf{\Phi}_\mathbf{q} & \mathbf{0} \end{bmatrix}$$

Evaluate the EOM and scaled constraints, using the current values of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, and $\boldsymbol{\lambda}$ at t_{n+1}. The resulting vector is called the residual vector.

$$\begin{bmatrix} \mathbf{\Psi} \\ \mathbf{\Omega} \end{bmatrix}^{(old)} = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_\mathbf{q}^T(\mathbf{q})\boldsymbol{\lambda} - \mathbf{Q}^A(\dot{\mathbf{q}}, \mathbf{q}, t) \\ \frac{1}{\beta h^2}\mathbf{\Phi}(\mathbf{q}, t) \end{bmatrix}$$

Compute the correction vector by solving a linear system. Note that the system matrix is constant during the iterative process.

$$\tilde{\mathbf{J}} \cdot \begin{bmatrix} \Delta\ddot{\mathbf{q}} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{\Psi} \\ \mathbf{\Omega} \end{bmatrix}^{(old)}$$

Correct the accelerations and Lagrange multipliers to obtain a better approximation for their values at time t_{n+1}.

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(new)} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix}^{(old)} - \begin{bmatrix} \Delta\ddot{\mathbf{q}} \\ \Delta\lambda \end{bmatrix}$$

Compute the infinity norm of the correction vector (the largest entry in absolute value) which will be used in the convergence test.

$$\text{err} = \left\| \begin{matrix} \Delta\ddot{\mathbf{q}} \\ \Delta\lambda \end{matrix} \right\|_\infty$$

Is error less than tolerance?

NO
Need to further improve accelerations and Lagrange multipliers

YES

Store $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$ at t_{n+1}. Use the final acceleration values to calculate positions and velocities $\mathbf{q}$ and $\dot{\mathbf{q}}$ at t_{n+1}. Use the final Lagrange multiplier values to calculate reaction forces. Store all this information.

ME451 End of Semester Evaluation

- Please let me know what you didn’t like
- Please let me know what you liked
- Your input is extremely valuable

- Course Evaluation: https://aefis.engr.wisc.edu