ME451
Kinematics and Dynamics of Machine Systems

Introduction to Dynamics
Newmark Integration Formula
[not in the textbook]
December 9, 2014

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:

  \[ M\ddot{q} + C\dot{q} + Kq = F(t) \]

- Not quite our problem, but good stepping stone
  - Square matrices \( M \), \( C \), and \( K \) are constant
  - \( F(t) \) 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 $t_0, t_1, t_2, ...$

  ![Time Grid](image)

  - **Stage 1/3** – Newmark’s formulas relate position to acceleration and velocity to acceleration:
    
    \[
    q_{n+1} = q_n + h \dot{q}_n + \frac{h^2}{2} [(1 - 2\beta) \ddot{q}_n + 2\beta \ddot{q}_{n+1}] \equiv p(\ddot{q}_{n+1}) \\
    \dot{q}_{n+1} = \dot{q}_n + h [(1 - \gamma) \ddot{q}_n + \gamma \ddot{q}_{n+1}] \equiv v(\ddot{q}_{n+1})
    \]

  - **Stage 2/3** – Newmark’s method (1957) discretizes the second order EOM:
    
    \[
    M\ddot{q} + C\dot{q} + Kq = F(t) \iff M\ddot{q}_{n+1} + C\dot{q}_{n+1} + Kq_{n+1} = F(t_{n+1})
    \]
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 $t_{n+1}$
  \[ M\ddot{q}_{n+1} + C\dot{q}_{n+1} + Kq_{n+1} = F(t_{n+1}) \]

- Use the discretization formulas to replace $q_{n+1}$ and $\dot{q}_{n+1}$ in terms of the accelerations $\ddot{q}_{n+1}$ using formulas on previous slide:
  \[ q_{n+1} = p(\ddot{q}_{n+1}) \quad \text{and} \quad \dot{q}_{n+1} = v(\ddot{q}_{n+1}) \]

- Obtain algebraic problem in which the unknown is the acceleration (denoted here by $x$):
  \[ M \cdot x + C \cdot v(x) + K \cdot p(x) = F(t_{n+1}) \]
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.

\[
M\ddot{q} + C\dot{q} + Kq = F(t)
\]

\[
\begin{align*}
M\ddot{q} + \Phi_q^T\lambda - Q^A(\dot{q}, q, t) &= 0 \\
\Phi(q, t) &= 0
\end{align*}
\]

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

Linear Finite Element Dynamics Problem

Nonlinear Multibody Dynamics Problem
Stage 3/3:
Discretization of the Constrained EOM (1/3)

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

\[
\begin{align*}
\mathbf{M} \ddot{\mathbf{q}}_{n+1} + \Phi_q^T(q_{n+1}) \lambda_{n+1} - \mathbf{Q}^A(\dot{q}_{n+1}, q_{n+1}, t_{n+1}) &= 0 \\
\frac{1}{\beta h^2} \Phi(q_{n+1}, t_{n+1}) &= 0
\end{align*}
\]

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

\[
\begin{align*}
q_{n+1} &= q_n + h \dot{q}_n + \frac{h^2}{2} [(1 - 2\beta) \ddot{q}_n + 2\beta \ddot{q}_{n+1}] \equiv p(\ddot{q}_{n+1}) \\
\dot{q}_{n+1} &= \dot{q}_n + h [(1 - \gamma) \ddot{q}_n + \gamma \ddot{q}_{n+1}] \equiv v(\ddot{q}_{n+1})
\end{align*}
\]

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

\[
\begin{bmatrix}
M \ddot{q}_{n+1} + \Phi_q^T(q_{n+1}) \lambda_{n+1} - Q^A(\dot{q}_{n+1}, q_{n+1}, t_{n+1}) \\
\frac{1}{\beta h^2} \Phi(q_{n+1}, t_{n+1})
\end{bmatrix} = 0
\]

- 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 \( \Phi(q, t) = 0 \) to get the positions \( q \))
Stage 3/3: Discretization of the Constrained EOM (3/3)

- Define the following two functions:

\[
\bar{\Psi}(\ddot{q}_{n+1}, \dot{q}_{n+1}, q_{n+1}, \lambda_{n+1}) \triangleq M\ddot{q}_{n+1} + \Phi_q^T(q_{n+1})\lambda_{n+1} - Q^A(\dot{q}_{n+1}, q_{n+1}, t_{n+1})
\]

\[
\bar{\Omega}(q_{n+1}) \triangleq \frac{1}{\beta h^2} \Phi(q_{n+1}, t_{n+1})
\]

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

- To make this clear, define the new functions:

\[
\Psi(\ddot{q}_{n+1}, \lambda_{n+1}) \equiv \bar{\Psi}(\ddot{q}_{n+1}, \dot{q}_{n+1}(\ddot{q}_{n+1}), q_{n+1}(\ddot{q}_{n+1}), \lambda_{n+1})
\]

\[
\Omega(\ddot{q}_{n+1}) \equiv \bar{\Omega}(q_{n+1}(\ddot{q}_{n+1}))
\]

- Therefore, we must solve for \(\ddot{q}_{n+1}\) and \(\lambda_{n+1}\) the following system

\[
\begin{bmatrix}
\Psi(\ddot{q}_{n+1}, \lambda_{n+1}) \\
\Omega(\ddot{q}_{n+1})
\end{bmatrix} = 0
\]
Chain Rule for Computing the Jacobian (1/3)

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

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

relies on the Jacobian

\[
\begin{bmatrix}
\frac{\partial \Psi}{\partial \ddot{q}_{n+1}} & \frac{\partial \Psi}{\partial \lambda_{n+1}} \\
\frac{\partial \Omega}{\partial \ddot{q}_{n+1}} & \frac{\partial \Omega}{\partial \lambda_{n+1}}
\end{bmatrix}
\]

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

- Note that, from the Newmark formulas we get

\[
\frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} = \frac{\partial p(\ddot{q}_{n+1})}{\partial \ddot{q}_{n+1}} = \beta h^2 I_{nc \times nc} \quad \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} = \frac{\partial v(\ddot{q}_{n+1})}{\partial \ddot{q}_{n+1}} = \gamma h I_{nc \times nc}
\]
Consider

\[ \Psi(\ddot{q}_{n+1}, \lambda_{n+1}) = \ddot{\Psi}(\ddot{q}_{n+1}, \dot{q}_{n+1}(\ddot{q}_{n+1}), q_{n+1}(\ddot{q}_{n+1}), \lambda_{n+1}) \]

\[ = M\ddot{q}_{n+1} + \Phi_q^T(q_{n+1})\lambda_{n+1} - Q^A(\dot{q}_{n+1}, q_{n+1}, t_{n+1}) \]

Apply the chain rule of differentiation to obtain

\[ \frac{\partial \Psi}{\partial \ddot{q}_{n+1}} = \frac{\partial \ddot{\Psi}}{\partial \ddot{q}_{n+1}} + \frac{\partial \ddot{\Psi}}{\partial \dot{q}_{n+1}} \frac{\partial \dot{q}_{n+1}}{\partial \ddot{q}_{n+1}} + \frac{\partial \ddot{\Psi}}{\partial q_{n+1}} \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} = \frac{\partial \ddot{\Psi}}{\partial \ddot{q}_{n+1}} + \gamma h \frac{\partial \ddot{\Psi}}{\partial \dot{q}_{n+1}} + \beta h^2 \frac{\partial \ddot{\Psi}}{\partial q_{n+1}} \]

\[ \frac{\partial \Psi}{\partial \ddot{q}_{n+1}} = M + \gamma h \left(- \frac{\partial Q^A}{\partial \ddot{q}_{n+1}}\right) + \beta h^2 \left( \frac{\partial (\Phi_q^T \lambda)}{\partial q_{n+1}} - \frac{\partial Q^A}{\partial q_{n+1}} \right) \]

and

\[ \frac{\partial \Psi}{\partial \lambda_{n+1}} = \Phi_q^T \]
**Chain Rule for Computing the Jacobian (3/3)**

- Consider

\[
\Omega(\ddot{q}_{n+1}) = \tilde{\Omega}(\ddot{q}_{n+1}(\ddot{q}_{n+1})) = \frac{1}{\beta h^2} \Phi(q_{n+1}, t_{n+1})
\]

- Apply the chain rule of differentiation to obtain

\[
\frac{\partial \Omega}{\partial \ddot{q}_{n+1}} = \frac{\partial \tilde{\Omega}}{\partial q_{n+1}} \frac{\partial q_{n+1}}{\partial \ddot{q}_{n+1}} = \beta h^2 \frac{\partial \tilde{\Omega}}{\partial q_{n+1}} = \beta h^2 \left( \frac{1}{\beta h^2} \Phi_q \right)
\]

\[
\frac{\partial \Omega}{\partial \ddot{q}_{n+1}} = \Phi_q
\]

and

\[
\frac{\partial \Omega}{\partial \lambda_{n+1}} = 0
\]
Solving the Nonlinear System

- Newton’s method applied to the system

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

- Jacobian obtained as

\[
J = \begin{bmatrix}
\frac{\partial \Psi}{\partial \dot{q}} & \frac{\partial \Psi}{\partial \lambda} \\
\frac{\partial \Omega}{\partial \dot{q}} & \frac{\partial \Omega}{\partial \lambda}
\end{bmatrix} = \begin{bmatrix}
\mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{q}} + \beta h^2 \left( \frac{\partial (\Phi_{\dot{q}}^T \lambda)}{\partial \dot{q}} - \frac{\partial \mathbf{Q}^A}{\partial \dot{q}} \right) \\
\Phi_q \Phi_q^T
\end{bmatrix}
\]

- Corrections computed as

\[
\begin{bmatrix}
\Delta \dot{q} \\
\Delta \lambda
\end{bmatrix} = \begin{bmatrix}
\mathbf{M} - \gamma h \frac{\partial \mathbf{Q}^A}{\partial \dot{q}} + \beta h^2 \left( \frac{\partial (\Phi_{\dot{q}}^T \lambda)}{\partial \dot{q}} - \frac{\partial \mathbf{Q}^A}{\partial \dot{q}} \right) \\
\Phi_q
\end{bmatrix}^{-1} \begin{bmatrix}
\Psi(\ddot{q}^{(old)}, \lambda^{(old)}) \\
\Omega(\ddot{q}^{(old)})
\end{bmatrix}.
\]

\[
\begin{bmatrix}
\ddot{q}^{(new)} \\
\lambda
\end{bmatrix} = \begin{bmatrix}
\ddot{q}^{(old)} \\
\lambda
\end{bmatrix} - \begin{bmatrix}
\Delta \dot{q} \\
\Delta \lambda
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
M & \Phi_q^T \\
\Phi_q & 0
\end{bmatrix}
\begin{bmatrix}
\dot{q} \\
\lambda
\end{bmatrix} = \begin{bmatrix}
Q^A
\end{bmatrix}
\]

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

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

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

\[
J = \begin{bmatrix}
M - \gamma h^2 Q^A + \beta h^2 \left( \frac{\partial (\Phi_q^T \lambda)}{\partial q} - \frac{\partial Q^A}{\partial q} \right) & \Phi_q^T \\
\Phi_q & 0
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
\Psi^{(old)} \\
\Omega
\end{bmatrix} = \begin{bmatrix}
M \ddot{q} + \Phi_q^T (q) \lambda - Q^A(q, t) \\
\frac{\partial \Phi_q^T (q)}{\partial q} \ddot{q} + \Phi_q \lambda(t)
\end{bmatrix}
\]

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

\[
J \cdot \begin{bmatrix}
\Delta \dot{q} \\
\Delta \lambda
\end{bmatrix} = \begin{bmatrix}
\Psi^{(old)} \\
\Omega
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
\ddot{q}^{(new)} \\
\lambda^{(new)}
\end{bmatrix} = \begin{bmatrix}
\ddot{q}^{(old)} \\
\lambda^{(old)}
\end{bmatrix} - \begin{bmatrix}
\Delta \dddot{q} \\
\Delta \lambda
\end{bmatrix}
\]

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

\[
\text{err} = \left\| \begin{bmatrix}
\Delta \dddot{q} \\
\Delta \lambda
\end{bmatrix} \right\|_\infty
\]

- Is error less than tolerance?

If NO, need to further improve accelerations and Lagrange multipliers.

If YES, store $q$ and $\dot{q}$ at $t_{n+1}$. Use the final acceleration values to calculate positions and velocities $q$ and $\dot{q}$ 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

---

![Graphs showing the iteration process for Newton, Modified Newton, and Quasi Newton methods.](image-url)
Quasi Newton Method
for the Dynamics Problem (1/3)

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

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

- Jacobian obtained as

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

- Terms that we have not computed previously:
  - Partial derivative of reaction forces with respect to positions $\frac{\partial (\Phi_q^T \lambda)}{\partial q}$
  - Partial derivative of applied forces with respect to positions $\frac{\partial \Phi_q^A}{\partial q}$
  - Partial derivative of applied forces with respect to velocities $\frac{\partial \Phi_q^A}{\partial \dot{q}}$
Approximate the Jacobian by ignoring these terms

Nonlinear equations:

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

Exact Jacobian:

\[
J = 
\begin{bmatrix}
\frac{\partial \Psi}{\partial q_{n+1}} & \frac{\partial \Psi}{\partial \lambda_{n+1}} \\
\frac{\partial \Omega}{\partial q_{n+1}} & \frac{\partial \Omega}{\partial \lambda_{n+1}}
\end{bmatrix}
= 
\begin{bmatrix}
M - \gamma h \frac{\partial Q^A}{\partial q_{n+1}} + \beta h^2 \left( \frac{\partial (\Phi_q^T \lambda_{n+1})}{\partial q_{n+1}} - \frac{\partial Q^A}{\partial q_{n+1}} \right) \\
\Phi_q & 0
\end{bmatrix}
\]

Approximate Jacobian:

\[
\tilde{J} = 
\begin{bmatrix}
M & \Phi_q^T \\
\Phi_q & 0
\end{bmatrix}
\]

Therefore, we modify the solution procedure to use a Quasi Newton method
The actual terms dropped from the expression of the exact Jacobian

$$
\beta h^2 \frac{\partial (\Phi_q')}{\partial \dot{q}} \quad \beta h^2 \frac{\partial Q^A}{\partial q} \quad \gamma h \frac{\partial Q^A}{\partial \dot{q}}
$$

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

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

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

Find consistent initial conditions for generalized positions and velocities.

At the initial time \( t_0 \)

Calculate the generalized accelerations and Lagrange multipliers.

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

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

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

\[
J =
\begin{bmatrix}
M & \Phi_q^T \\
\Phi_q & 0
\end{bmatrix}
\]

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

\[
\begin{bmatrix}
\Psi^{(old)} \\
\Omega
\end{bmatrix} = \begin{bmatrix}
M\ddot{q} + \Phi_q^T(q)\lambda - Q^A(q, q, t) \\
\frac{\partial}{\partial \dot{q}} \Phi(q, t)
\end{bmatrix}
\]

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

\[
J \begin{bmatrix}
\Delta\ddot{q} \\
\Delta\lambda
\end{bmatrix} = \begin{bmatrix}
\Psi^{(old)} \\
\Omega
\end{bmatrix}
\]

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

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

\[
err = \left\| \begin{bmatrix}
\Delta\ddot{q} \\
\Delta\lambda
\end{bmatrix} \right\|_{\infty}
\]

Is error less than tolerance?  

NO  
Need to further improve accelerations and Lagrange multipliers

YES

Store \( \ddot{q} \) and \( \lambda \) at \( t_{n+1} \). Use the final acceleration values to calculate positions and velocities \( q \) and \( \dot{q} \) 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.edú