GraSMech course 2009-2010
Computer-aided analysis of rigid
and flexible multibody systems

Numerical Integration
of Equations of Motion

Prof. Olivier Verlinden (FPMs)
Olivier.Verlinden@fpms.ac.be

Prof. Olivier Brüls
o.bruls@ulg.ac.be

Modelling steps

- Choose the configuration parameters (q)
- Set up the kinematics: express position, velocity and acceleration (rotational and translational) in terms of q and its first and second time derivatives
- Express the forces in terms of q, its time derivatives and time t
- Build the differential equations of motion
- Numerical treatment of the equations (lesson 5)
**Layout of the presentation**

- General principle of an integration scheme (O. Verlinden)
  - ODEs (without constraints)
    - principle of integration
    - principal methods: multistep (Newmark), Runge-Kutta
    - practical realization of a time step (explicit and implicit)
  - DAEs (with constraints)

- Properties of integration methods (O. Brüls)
  - ODEs
    - Accuracy, stability
    - Stiff ODEs
  - DAEs
    - The index
    - Index reduction methods

---

**Concerned equations**

The equations of motion consist of

- \( n_{cp} \) differential equations (dynamic equilibrium)

\[
M(q) \cdot \ddot{q} + h(q, \dot{q}, t) + B^T \cdot \lambda = g(q, \dot{q}, t)
\]

- \( n_c \) constraint equations
  - at position level \( \phi(q) = 0 \)
  - or at velocity level \( Bq = 0 \)
  - or at acceleration level \( B\ddot{q} + B\dot{q} = 0 \)

\( \Rightarrow n_c + n_{cp} \) differential-algebraic equations (DAE)
The equations of motion are considered in residual form

\[ f(q, \dot{q}, \ddot{q}, \lambda, t) = \begin{cases} \mathbf{M}(q) \cdot \ddot{q} + \mathbf{h}(q, \dot{q}, t) - \mathbf{g}(q, \dot{q}, t) + \mathbf{B}^T \cdot \lambda = 0 \\ \phi(q) = 0 \end{cases} = 0 \]

From the formulation, we can only estimate the residuals \( f(\neq 0) \) for given values of \( q \) and its time derivatives, \( \lambda \) and \( t \).

=> It is the job of the numerical integration to draw them to zero by finding the right values of \( q(t) \) and \( \lambda \)!

---

**Principle of numerical integration**

- We want to integrate the equations of motion without constraints

\[ f(q, \dot{q}, \ddot{q}, t) = \mathbf{M}(q) \cdot \ddot{q} + \mathbf{H}(q, \dot{q}, t) = 0 \]

from initial conditions \( q_0 \) and \( \dot{q}_0 \).

- Step by step procedure

One or several states at and before \( t \)

3xn\(_{cp}\) unknowns

\( h \)= time step
Basic equations

- The $3x_n_{cp}$ corresponding equations are
  - the $n_{cp}$ residuals
    \[ f(q_i^{t+h}, q_i^{t+h}, \dot{q}_i^{t+h}, t + h) = 0 \]
  - and the $2x_n_{cp}$ integrals
    \[ q_i^{t+h} = q_i^t + \int_t^{t+h} \dot{q}_i dt \]
    \[ \dot{q}_i^{t+h} = \dot{q}_i^t + \int_t^{t+h} \ddot{q}_i dt \]

- The integrals are replaced by integration formulas
  \[ \dot{q}_i^{t+h} = \dot{q}_i^t + \beta \dot{q}_i^t + (0.5 - \beta) h^2 \ddot{q}_i + \beta h^2 \dddot{q}_i^{t+h} \]
  \[ \dddot{q}_i^{t+h} = \dddot{q}_i^t + (1 - \gamma) h\dddot{q}_i + \gamma h^2 \dddot{q}_i^{t+h} \]

First-order form

- Integration methods generally deal with first-order differential equations
  \[ \dddot{y} = f(y, t) \text{ or } f(y, \dot{y}, t) = 0 \]
  with integration formulas of the form
  \[ y_i^{t+h} = N(y_i^t, \dot{y}_i^t, \ddot{y}_i^{t+h}) \]

- $N$ second order differential equations
  \[ f(q, \dot{q}, t) = 0 \]
  can be transformed into $2N$ equivalent first-order differential equations
  \[
  \begin{align*}
  \dddot{q} - \dddot{y} &= 0 \\
  f(q, y, \dot{y}, t) &= 0
  \end{align*}
  \]
  with $y = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$
Implicit - explicit

- The integration formula is **implicit** if it involves the term in \( \dot{y}^{t+h} \)
  \[
  \dot{y}_i^{t+h} = \Lambda'(y_i^{t}, \dot{y}_i^{t}, y_i^{t+h})
  \]
  and **explicit** otherwise
  \[
  y_i^{t+h} = \Lambda'(y_i^{t}, \dot{y}_i^{t})
  \]
- Number of steps: number of known configurations at and before time \( t \) involved in the integration formula

Adams-Moulton integration formulas

Implicit multistep methods
Principle: polynomial of order \( m \) on \( \dot{y} \)

- \( m=0 \) (Euler implicit)
  \[
  y_i^{t+h} = y_i^t + hy_i^{t+h}
  \]
- \( m=1 \) (one step, =trapezoidal rule)
  \[
  y_i^{t+h} = y_i^t + \frac{h}{2}y_i^{t} + \frac{h}{2}y_i^{t+h}
  \]
- \( m=2 \) (two steps)
  \[
  y_i^{t+h} = y_i^t - \frac{h}{12}y_i^{t-h} + \frac{8h}{12}y_i^{t} + \frac{5h}{12}y_i^{t+h}
  \]
BDF integration formulas

Implicit multistep methods
Principle: polynomial of order $m$ on $y$

- $m=1$ ($\text{ADAMS-0}$)
  \[ y_{t+h} = y_t + h y_{t+h} \]

- $m=2$ (two steps)
  \[ y_{t+h} = \frac{1}{3} y_{t-h} + \frac{4}{3} y_t + \frac{2h}{3} y_{t+h} \]

- $m=3$ (three steps)
  \[ y_{t+h} = \frac{2}{11} y_{t-2h} - \frac{9}{11} y_{t-h} + \frac{18}{11} y_t + \frac{6h}{11} y_{t+h} \]

General form of multistep formulas

- General form of multistep integration formulas
  \[ \sum_{i=0}^{p+1} \alpha_i y_{t+(1-i)h} = h \sum_{j=0}^{k} \beta_j y_{t+jh} \]
  or
  \[ \alpha_0 y_{t+h} = -\alpha_1 y_t - \alpha_2 y_{t-h} \ldots - \alpha_p y_{t-\phi h} + h \left( \beta_0 y_{t+h} + \beta_1 y_t \ldots + \beta_k y_{t+ph} \right) \]
  method explicit if $\beta_0=0$

- $\text{ADAMS-m: } p=0$ and $k=m$
  $\text{ADAMS-2: } \alpha_0=12, \alpha_1=-12, \beta_0=5, \beta_1=8, \beta_2=-1$

- $\text{BDF-m: } p=m$ and $k=0$
  $\text{BDF-3: } \alpha_0=11, \alpha_1=-18, \alpha_2=9, \alpha_3=-2, \beta_0=6$
Generating second-order formulas

- By applying recursively implicit first-order formulas
  \[ y_{i}^{t+h} = \Lambda'(y_{i}^{\leq t}, y_{i}^{t}, y_{i}^{t+h}) \]
  you can generate second-order formulas

  \[
  \begin{align*}
  q_{i}^{t+h} &= \Lambda'(q_{i}^{\leq t}, q_{i}^{\leq t}, q_{i}^{t+h}) \\
  q_{i}^{t+h} &= \Lambda'(q_{i}^{\leq t}, q_{i}^{\leq t}, q_{i}^{t+h}) \\
  &= \Lambda'(q_{i}^{\leq t}, q_{i}^{\leq t}, \Lambda'(q_{i}^{\leq t}, q_{i}^{\leq t}, q_{i}^{t+h})) \\
  &= \Lambda(q_{i}^{\leq t}, q_{i}^{\leq t}, q_{i}^{t+h})
  \end{align*}
  \]

  => more efficient than working with 2N equations

Equivalent 2\textsuperscript{nd} order ADAMS-Moulton

- \( m=0 \)
  \[
  \begin{align*}
  q_{i}^{t+h} &= \dot{q}_{i}^{t} + h\ddot{q}_{i}^{t+h} \\
  q_{i}^{t+h} &= \dot{q}_{i}^{t} + h\ddot{q}_{i}^{t+h} = \dot{q}_{i}^{t} + h\dot{q}_{i}^{t} + h^{2}\ddot{q}_{i}^{t+h}
  \end{align*}
  \]

- \( m=1 \)
  \[
  \begin{align*}
  q_{i}^{t+h} &= \dot{q}_{i}^{t} + \frac{h}{2}\ddot{q}_{i}^{t} + \frac{h}{2}\ddot{q}_{i}^{t+h} \\
  q_{i}^{t+h} &= \dot{q}_{i}^{t} + h\ddot{q}_{i}^{t} + \frac{h^{2}}{4}\dddot{q}_{i}^{t} + \frac{h^{2}}{4}\dddot{q}_{i}^{t+h}
  \end{align*}
  \]

- \( m=2 \)
  \[
  \begin{align*}
  q_{i}^{t+h} &= \dot{q}_{i}^{t} - \frac{h}{12}\dddot{q}_{i}^{t-h} + \frac{8h}{12}\dddot{q}_{i}^{t} + \frac{5h}{12}\dddot{q}_{i}^{t+h} \\
  q_{i}^{t+h} &= \dot{q}_{i}^{t} - \frac{h}{12}\dddot{q}_{i}^{t-h} + \frac{13h}{12}\dddot{q}_{i}^{t} - \frac{5h^{2}}{144}\dddot{q}_{i}^{t-h} + \frac{40h^{2}}{144}\dddot{q}_{i}^{t} + \frac{25h^{2}}{144}\dddot{q}_{i}^{t+h}
  \end{align*}
  \]
Equivalent 2\textsuperscript{nd} order BDF formulas

- **m=1**
  \[
  q^{t+h} = q^t + h\dot{q}^{t+h} \\
  q^{t+h} = q^t + h\dot{q}^{t+h} = q^t + h\dot{q}^t + h^2\ddot{q}^{t+h}
  \]

- **m=2**
  \[
  \begin{align*}
  q^{t+h} &= -\frac{1}{3}q^{t-h} + \frac{4}{3}q^t + \frac{2h}{3}\dot{q}^{t+h} \\
  \dot{q}^{t+h} &= -\frac{1}{3}q^{t-h} + \frac{4}{3}q^t - \frac{2h}{9}q^{t-h} + \frac{8h}{9}q^t + \frac{4h^2}{9}\dddot{q}^{t+h}
  \end{align*}
  \]

Consistent initial conditions

- The initial conditions are \( q^0 \) and \( \dot{q}^0 \) and must be given by the user.

- The initial accelerations must verify the equations of motion
  \[
  M(q^0) \cdot \ddot{q}^0 + H(q^0, \dot{q}^0, t^0) = 0
  \]
  and are then given by
  \[
  \ddot{q}^0 = -M^{-1}(q^0) \cdot H(q^0, \dot{q}^0, t^0)
  \]
Resolution with explicit formulas

No term in $\ddot{q}^{t+h}$ in the formula $\rightarrow q^{t+h}$ and $\ddot{q}^{t+h}$ can be determined directly from the integration formula:

\[
\begin{align*}
q^{t+h} &= \Lambda(q^{\leq t}, \dot{q}^{\leq t}, \ddot{q}^{\leq t}) \\
\dot{q}^{t+h} &= \dot{\Lambda}(q^{\leq t}, \ddot{q}^{\leq t})
\end{align*}
\]

The accelerations are calculated as for the initial accelerations:

\[
\ddot{q}^{t+h} = -M^{-1} \cdot H(q^{t+h}, \dot{q}^{t+h}, t + h) = -M^{-1} \cdot H(\Lambda(q^{\leq t}, \dot{q}^{\leq t}, \ddot{q}^{\leq t}), \dot{\Lambda}(q^{\leq t}, \ddot{q}^{\leq t}), t + h)
\]

Resolution with implicit formulas

- Once positions and velocities have been replaced by the integration formulas, the only unknowns are the accelerations at $t+h$:

  \[
  f(q^{t+h}, \dot{q}^{t+h}, \ddot{q}^{t+h}, t + h) = 0 \\
  f(\Lambda(q^{\leq t}, \dot{q}^{\leq t}, \ddot{q}^{\leq t}), \dot{\Lambda}(q^{\leq t}, \ddot{q}^{\leq t}, \ddot{q}^{t+h}, t + h) = 0 \\
  F(\ddot{q}^{t+h}) = 0
  \]

- solved by iterative procedure of Newton-Raphson:

  \[
  \ddot{q}^{t+h,n} = \ddot{q}^{t+h,n-1} - J^{-1} \cdot F(\ddot{q}^{t+h,n-1})
  \]

  with $J$ the iteration matrix (Jacobian matrix)
Computation of the iteration matrix

- The accelerations at t+h intervenes at three levels
  \[ \ddot{q}_{t+h} \]
  \[ f(q_{t+h}, \dot{q}_{t+h}, \ddot{q}_{t+h}, t+h) \]

- The iteration matrix is given by
  \[ J_{ij} = \frac{\partial F_i}{\partial \dot{q}_j} + \frac{\partial f_i}{\partial q_j} \cdot \frac{\partial \lambda}{\partial \dot{q}_j} + \frac{\partial f_i}{\partial q_j} \cdot \frac{\partial \lambda}{\partial \ddot{q}_j} \]
  \[ J_{ij} = M_{ij} + C T_{ij} \cdot \frac{\partial \lambda}{\partial \dot{q}_j} + K T_{ij} \cdot \frac{\partial \lambda}{\partial \ddot{q}_j} \]
  with \( K T_{ij} = \frac{\partial f_i}{\partial q_j} \) \( C T_{ij} = \frac{\partial f_i}{\partial q_j} \)

\( K T, C T = \) tangent stiffness and damping matrices

Computation of iteration matrix

- Formulas of Newmark
  \[ q_{t+h} = q_t + h \dot{q}_t + (0.5 - \beta)h^2 \ddot{q}_t + \beta h^2 q_{t+h} \]
  \[ \dot{q}_{t+h} = \dot{q}_t + (1 - \gamma)h \ddot{q}_t + \gamma h \ddot{q}_{t+h} \]
  \[ \rightarrow \frac{\partial \lambda}{\partial \dot{q}_t} = \beta h^2 \]
  \[ \frac{\partial \lambda}{\partial \ddot{q}_t} = \gamma h \]
  \[ \rightarrow J = M + C T \cdot \gamma h + K T \cdot \beta h^2 \]

- an approximate iteration matrix is sufficient to get convergence

- \( J \) tends to \( M \) when \( h \) decreases (always possible to get convergence with \( M \) by decreasing \( h \) -> not efficient with stiff systems)
Integration with constraints

- The system of equations of motion is:
  \[ f^*(q, \dot{q}, \ddot{q}, \lambda, t) = \left( f(q, \dot{q}, \ddot{q}, t) + B^T \cdot \lambda \right) = 0 \]
- or
  \[ f^*(q, \dot{q}, \ddot{q}, \lambda, t) = \left( f(q, \dot{q}, \ddot{q}, t) + B^T \cdot \lambda \right) = 0 \]
- or
  \[ f^*(q, \dot{q}, \ddot{q}, \lambda, t) = \left( f(q, \dot{q}, \ddot{q}, t) + B^T \cdot \lambda \right) = 0 \]

Constraints and initial conditions

- The equations of motion with constraints at acceleration level must be used:
  \[ f^*(q, \dot{q}, \ddot{q}, \lambda, t) = \left( M(q) \cdot \dddot{q} + H(q, \dot{q}, \ddot{q}, t) + B^T \cdot \lambda \right) = 0 \]
- The constraint equations at acceleration level must be used to get consistent initial accelerations:
  \[ \left( \begin{array}{c} \dddot{q}^0 \\ \lambda^0 \end{array} \right) = - \left( \begin{array}{cc} M & B^T \\ B & 0 \end{array} \right)^{-1} \left( \begin{array}{c} H(q^0, \dot{q}^0, t^0) \\ B \cdot \dddot{q}^0 + \sum_i \frac{\partial^2 b}{\partial q_i^2} \dot{q}_i^0 + \frac{\partial^2 b}{\partial t^2} t^0 \end{array} \right) \]
- The same relationship can be used with explicit integration formulas but unavoidable drift of the constraints at velocity and position levels!
Constraints and implicit formulas

- The constraints simply consist of some more nonlinear equations in the accelerations (and Lagrange multipliers) at \( t+h \)

\[
f^*(\dot{q}^{t+h}, \ddot{q}^{t+h}, \dddot{q}^{t+h}, \lambda^{t+h}, t + h) = 0
\]

\[
f^*(\dot{A}, \ddot{A}, \dddot{q}^{t+h}, \lambda^{t+h}, t + h) = F^*(\dot{q}^{t+h}, \lambda^{t+h}, t + h) = 0
\]

whatever the constraint level

- which can be solved by Newton-Raphson

\[
\begin{pmatrix}
\dot{q}^{t+h,n} \\
\lambda^{t+h,n}
\end{pmatrix} = \begin{pmatrix}
\dot{q}^{t+h,n-1} \\
\lambda^{t+h,n-1}
\end{pmatrix} - J^* \cdot \begin{pmatrix}
F(\dot{q}^{t+h,n-1}) + B^T \lambda^{t+h,n-1} \\
b(\Delta(\dot{q}^{t+h,n-1}))
\end{pmatrix}
\]

with \( J^* = \begin{pmatrix} J & B^T \\ B & \frac{\partial \lambda}{\partial \dot{q}^{t+h}} \end{pmatrix} \) (for position level)

Runge-Kutta methods

- So-called « stage » methods: simultaneous resolution of the equations of motion at \( s \) instants \( t_i = t_n + c_i h \) between \( t \) and \( t+h \)

\[
f(\ddot{q}_i, \dddot{q}_i, \kappa_i, t + c_i h) = 0 \quad (i = 1, s) \quad (c_i = \sum_{j=1}^{s} a_{ij})
\]

with formulas

\[
\ddot{q}_i(t_n + c_i h) = q_n + h \ddot{q}_n + h^2 \sum_{j=1}^{s} a_{ij} \left( \sum_{i=1}^{s} a_{ji} \kappa_i \right)
\]

\[
\dddot{q}_i(t_n + c_i h) = \dot{q}_n + h \sum_{j=1}^{s} a_{ij} \kappa_j
\]

- \( \Rightarrow \) \( s \times N \) set of equations to solve simultaneously
Runge-Kutta methods

- The state at time $t_{n+1}$ is given by other formulas

\[
\begin{align*}
\hat{q}_{n+1} &= q_n + h\dot{q}_n + h^2 \sum_{i=1}^{s} b_i \left( \sum_{j=1}^{s} a_{ij} \kappa_j \right) \\
\tilde{q}_{n+1} &= \dot{q}_n + h \sum_{j=1}^{s} b_j \kappa_j \\
\tilde{q}_{n+1} &= \kappa_s \quad (s_i c_s = 1) \\
\text{ou} \quad f(q_{n+1}, \dot{q}_{n+1}, \ddot{q}_{n+1}, t_{n+1}) &= 0
\end{align*}
\]

Example: RKI 5/12 (Radau)

- 2 stages implicit scheme ($\gamma=5/12$ for RKI 5/12)

\[
\begin{align*}
\tilde{q}_1 &= \dot{q}_n + h(\gamma \kappa_1 + (\frac{1}{3} - \gamma) \kappa_2) \\
\tilde{q}_1 &= q_n + \frac{7}{3} h \dot{q}_n + h^2 (4 \gamma^2 - 3 \gamma + \frac{2}{3}) \kappa_1 \\
&\quad + h^2 (-4 \gamma^2 + \frac{7}{3} \gamma - \frac{1}{3}) \kappa_2 \\
\tilde{q}_2 &= \dot{q}_n + h((2 - 3 \gamma) \kappa_1 + (3 \gamma - 1) \kappa_2) \\
\tilde{q}_2 &= q_n + \gamma h \dot{q}_n + h^2 (-12 \gamma^2 + 11 \gamma - 2) \kappa_1 \\
&\quad + h^2 (12 \gamma^2 - 9 \gamma + \frac{5}{3}) \kappa_2
\end{align*}
\]
Resolution for RKI 5/12

- The set of $2N$ nonlinear equations is solved by the usual iterative procedure of Newton-Raphson:

\[
\begin{pmatrix}
\kappa_1 \\
\kappa_2
\end{pmatrix} = 
\begin{pmatrix}
\kappa_1 \\
\kappa_2
\end{pmatrix} - J^{-1} \begin{pmatrix}
\vec{f}(\vec{q}_1, \vec{q}_1, \kappa_1, t_1) \\
\vec{f}(\vec{q}_2, \vec{q}_2, \kappa_2, t_2)
\end{pmatrix}
\]

\[
J =
\begin{pmatrix}
M_1 + \frac{5h}{12} C_1 + \frac{h^2}{3} K_1 & \frac{h}{12} C_1 - \frac{h^2}{18} K_1 \\
\frac{3h}{4} C_2 + \frac{h^2}{2} K_2 & M_2 + \frac{3h}{4} C_2
\end{pmatrix}
\]

with $C = \frac{\partial f}{\partial \vec{q}}$, $K = \frac{\partial f}{\partial \vec{q}}$

- and state at $t_{n+1}$ is given by:

\[
q_{n+1} = q_n + h\dot{q}_n + \frac{h^2}{2}\kappa_1
\]

\[
\ddot{q}_{n+1} = \ddot{q}_n + h\left(\frac{3}{4}\kappa_1 + \frac{1}{4}\kappa_2\right)
\]

\[
\dot{q}_{n+1} = \kappa_2 \text{ or } f(q_{n+1}, \dot{q}_{n+1}, \ddot{q}_{n+1}, t_{n+1}) = 0
\]

Integration – second part

- Properties of integration methods

=> Olivier Brüls
Time integration

I. General principle of an integration scheme (O. Verlinden)

II. Properties and specific methods (O. Brüls)

- **ODEs**
  - Multistep (Newmark), Runge-Kutta
  - Accuracy, stability
  - Stiff ODEs

- **DAEs**
  - The index
  - Index reduction methods

---

**ODEs: Multistep methods**

\[
\begin{align*}
\text{Differential equation:} & \quad \dot{y} = g(y, t) \quad y^T = [q^T \dot{q}^T] \\
+ \text{Adams 2:} & \quad y_{n+1} = y_n + h \left( \frac{5}{12} \dot{y}_{n+1} + \frac{1}{12} \dot{y}_n + \frac{1}{12} \dot{y}_{n-1} \right) \\
\end{align*}
\]

\[
\text{Algebraic equation:} \quad y_{n+1} = y_n + h \left( \frac{5}{12} g_{n+1} + \frac{1}{12} g_n - \frac{1}{12} g_{n-1} \right)
\]

**General form of a linear multistep method (k steps):**

\[
\alpha_0 y_{n+1} + \ldots + \alpha_k y_{n+1-k} = h \left( \beta_0 g_{n+1} + \ldots + \beta_k g_{n+1-k} \right)
\]

- Implicit: \( \beta_0 \neq 0 \) (\( y_{n+1} \) in the RHS \( \Rightarrow \) nonlinear probl)
- BDF: \( \alpha_0 y_{n+1} + \ldots + \alpha_k y_{n+1-k} = h \beta_0 g_{n+1} \)
- Adams: \( y_{n+1} - y_n = h \left( \beta_0 g_{n+1} + \ldots + \beta_k g_{n+1-k} \right) \)
ODEs: Multistep methods

- Newmark:
  \[ M \ddot{q} + h(q, \dot{q}, t) = 0 \]  
  \[ q_{n+1} = q_n + h \dot{q}_n + h^2 (1/2 - \beta) \ddot{q}_n + h^2 \beta \ddot{q}_{n+1} \]  
  \[ \dot{q}_{n+1} = \dot{q}_n + h (1 - \gamma) \ddot{q}_n + h \gamma \ddot{q}_{n+1} \]

  - Implementation: Newmark formulae into (*)
  - Theoretical analysis: \( \ddot{q} = -M^{-1}h = g \)

\[ q_{n+1} - q_n = h (1 - \gamma) g_n + h \gamma g_{n+1} \]

\[ q_{n+1} - q_n = h \dot{q}_n + h^2 (1/2 - \beta) g_n + h^2 \beta g_{n+1} \]

\[ q_n - q_{n-1} = h \dot{q}_{n-1} + h^2 (1/2 - \beta) g_{n-1} + h^2 \beta g_n \]

\[ q_{n+1} - 2q_n + q_{n-1} = h^2 (\beta_0 g_{n+1} + \beta_1 g_n + \beta_2 g_{n-1}) \]

- Two-step method with a one-step implementation

ODEs: Runge-Kutta methods

- Euler explicit:
  \[ y_{n+1} = y_n + h g(y_n, t_n) \]

- Mid-point rule?
  \[ y_{n+1} = y_n + h g(\frac{y_{n+1/2} + y_{n-1/2}}{2}, t_{n+1/2}) \]

  Intermediate Euler step:
  \[ y_{n+1/2} = y_n + \frac{h}{2} g(y_n, t_n) \]

- \( s \) stages:
  \[ t(i) = t_n + c_i h \]
  \[ t_n \quad t(i) \quad t_{n+1} \]

  1-step formula:
  \[ y_{n+1} = y_n + h \left(b_1 g(1) + \ldots + b_s g(s)\right) \]

  \[ g(i) = g(y(i), t(i)) \]

  ERK: \[ y(i) = y_0 + h \left(a_{i1} g(1) + \ldots + a_{i-1} g(i-1)\right) \]

  IRK: \[ y(i) = y_0 + h \left(a_{i1} g(1) + \ldots + a_{iS} g(S)\right) \]
**ODEs: Accuracy**

- Global error: \( e_f = y(t_f) - y_n \) \( n = (t_f - t_0)/h \)
- Order \( p \) if \( e_f = \mathcal{O}(h^p) \) when \( h \to 0 \)

<table>
<thead>
<tr>
<th>Method</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams with ( k ) steps</td>
<td>( k + 1 )</td>
</tr>
<tr>
<td>BDF with ( k ) steps</td>
<td>( k )</td>
</tr>
<tr>
<td>Newmark</td>
<td>1 or 2</td>
</tr>
<tr>
<td>RADAU5 (3 stages)</td>
<td>5</td>
</tr>
</tbody>
</table>

For intermediate \( h \): ! warning constant of errors

**ODEs: Accuracy**

Undamped oscillator: \( \ddot{q} + \omega^2 q = 0 \)

Exact period: \( T = \frac{2\pi}{\omega} \)

Numerical result: \( \bar{T} \)

Periodicity error: \( \frac{\bar{T} - T}{T} \)

![Graph showing periodicity error comparison for different methods](image-url)
ODEs: Stability

- **Undamped oscillator**
  \[ \ddot{q} + \omega^2 q = 0 \]
  \( \omega = 1 \text{ rad/s}, \ q(0) = 1, \ \dot{q}(0) = 0 \)

- **Linear** scalar test equation
  \[ \dot{y} = \sigma y \]

  Stability region

**Euler explicit** \( h = 0.1 \text{ s} \)

---

ODEs: Stability

- **Undamped oscillator**
  \[ \ddot{q} + \omega^2 q = 0 \]
  \( \omega = 1 \text{ rad/s}, \ q(0) = 1, \ \dot{q}(0) = 0 \)

- **Linear** test equation \[ \dot{y} = \sigma y \]

  The stability region includes the left half plane

**Euler implicit** \( h = 0.1 \text{ s} \)
ODEs: Stability

- **Unconditional stability (A-stability):**
  « stable solution for any stable linear system »
  (whatever $h$)

- **Stability regions for BDFs**

  BDF1 = Euler implicit

- **Dahlquist’s barrier for linear multistep methods:**
  Unconditional stability is only possible for $p \leq 2$

---

ODEs: Stability

- **Linear multistep method for** $\dot{y} = g(y, t)$
  
  $\alpha_0 y_{n+1} + \ldots + \alpha_k y_{n+1-k} = h (\beta_0 g_{n+1} + \ldots + \beta_k g_{n+1-k})$

- **Linear test equation:** $\dot{y} = \sigma y$
  
  $(\alpha_0 - \sigma h \beta_0) y_{n+1} + \ldots + (\alpha_k - \sigma h \beta_k) y_{n+1-k} = 0$

- **Eigenvalues of the difference equation**
  
  $(\alpha_0 - \sigma h \beta_0) \zeta^k + \ldots + (\alpha_k - \sigma h \beta_k) \zeta^0 = 0$

- **Spectral radius** $\rho(\sigma h) = \max_j |\zeta_j|$
  « amplification factor » at each time step

- **Stability region:** $\rho(\sigma h) \leq 1$ ($< 1$)
ODEs: Stability

Spectral radius: \( \sigma_h \) on the imaginary axis

\[ \ddot{q} + \omega^2 q = 0 \]

\( \rho_{\infty} \) = measure of the numerical damping

Available methods

Matlab
- \texttt{ode45} = Dormand/Prince method (explicit Runge-Kutta)
- \texttt{ode15s} = BDF of order 1 to 5 (implicit multistep)

On the web
- \texttt{dopri5.f}, \texttt{dopri5.c} (Dormand/Prince method)
- \texttt{lsode.f}, \texttt{dassl.f}, \texttt{cvode.c} (BDF)
- \texttt{radau5.f} (implicit Runge-Kutta)
Example – 2 dof system

Properties of the system

- $k_1 = 79.15 \text{ N/m}$ and $k_2 = 1961 \text{ N/m}$
- $m = 1 \text{ kg}$
- Eigen frequencies: 1 and 10 Hz

Simulation from initial conditions $q_1 = 0$ and $q_2 = 1$

2 dof system – 20 steps per period

- 2 DOF system - Integration with Newmark ($h=0.005 \text{ s}$)
- 2 DOF system - Integration with RKI ($h=0.005 \text{ s}$)
2 dof system – 5 steps per period

Stiff ODEs

- Stiff differential equations: slow & fast modes
  - Example: FE model with thousands of dofs
  - **Accuracy** is mostly required for the slow modes
  - **Stability** is also required for the fast modes

- Explicit methods
  - Restricted stability region ⇒ very small time steps

- Implicit methods
  - Large stability domain ⇒ larger time steps
  - Price to pay: nonlinear equations to solve
Stiff ODEs

Limit of explicit methods

\[ m = 1 \text{ kg}, \quad k_1 = 79.0 \text{ N/m}, \quad k_2 = 2368.7 \text{ N/m} \]

\[ q_1(0) = 1 \text{ m}, \quad \dot{q}_1(0) = 0 \text{ m/s}, \quad q_2(0) = 1 \text{ m}, \quad \dot{q}_2(0) = 0 \text{ m/s} \]

Higher order explicit methods result in similar stability problems

Newmark formulae:

\[ q_{n+1} = q_n + h\dot{q}_n + \frac{h^2}{2} (0.5 - \beta) \ddot{q}_n + \frac{h^2}{2} \beta \dddot{q}_{n+1} \]

\[ \ddot{q}_{n+1} = \ddot{q}_n + h(1 - \gamma) \dddot{q}_n + h \gamma \dddot{q}_{n+1} \]

To be solved with:

\[ M(q_{n+1}) \dddot{q}_{n+1} + h(q_{n+1}, \dot{q}_{n+1}, \dddot{q}_{n+1}) = 0 \]

- From \( q_n, \dot{q}_n, \ddot{q}_n \), we seek \( q_{n+1}, \dot{q}_{n+1}, \dddot{q}_{n+1} \) satisfying
  - one non-linear equation
  - two linear equations

- Choice of parameters
  - Undamped \( \gamma = \frac{1}{2}, \quad \beta = \frac{1}{4} \)
  - Damped \( \gamma = \frac{1}{2} + \alpha, \quad \beta = \frac{1}{4} (\gamma + \frac{1}{2})^2 \)
Newmark formulae:
\[
\begin{align*}
q_{n+1} &= q_n + h\dot{q}_n + h^2(0.5 - \beta)a_n + h^2\beta a_{n+1} \\
\dot{q}_{n+1} &= \dot{q}_n + h(1 - \gamma)a_n + h\gamma a_{n+1}
\end{align*}
\]

**Generalized-\(\alpha\) method** [Chung & Hulbert 1993]
\[
(1 - \alpha_m)a_{n+1} + \alpha_m a_n = (1 - \alpha_f)\ddot{q}_{n+1} + \alpha_f \ddot{q}_n
\]

To be solved with: \(M(q_{n+1})\ddot{q}_{n+1} + h(q_{n+1}, \dot{q}_{n+1}, t_{n+1}) = 0\)

- Two types of acceleration variables: \(a_n \neq \ddot{q}_n\)
- Second-order accuracy + numerical damping

\[
\begin{align*}
\alpha_m &= \frac{2\rho\alpha - 1}{\rho\alpha + 1}, \quad \alpha_f = \frac{\rho\alpha}{\rho\alpha + 1}, \quad \gamma = \frac{1}{2} + \alpha_f - \alpha_m, \quad \beta = \frac{1}{2} \left( \gamma + \frac{1}{\beta} \right)^2
\end{align*}
\]
ODEs: summary

- Multistep
  - Dahlquist’s barrier: stability vs. accuracy
  - Variable step-size: implementation is not trivial

- Runge-Kutta
  - Higher orders of precision are possible
  - IRK: all stages are coupled \( n \times s \) unknowns!

- Generalized-\( \alpha \)
  - One-step method
  - Unconditional stability (numerical damping)
  - Second-order accuracy

DAEs

- Consider the scalar singular perturbation problem:
  \[
  \frac{\dot{y}}{\epsilon} = g_1(y, z, t) \\
  \epsilon \frac{\dot{z}}{z} = \sigma z + g_2(y)
  \]
  if \( \epsilon \to 0 \) \((\sigma / \epsilon \to \infty)\), the system becomes very stiff

- Limit case = DAE:
  \[
  \dot{y} = g_1(y, z, t) \\
  0 = \sigma z + g_2(y) \Rightarrow z = -\frac{1}{\sigma} g_2(y)
  \]

- Underlying ODE?
  \[
  \dot{y} = g_1(y, z, t) \\
  \dot{z} = -\frac{1}{\sigma} \frac{\partial g_2}{\partial y}(y) \cdot g_1(y, z, t)
  \]
  Differentiation index of the original DAE = 1

- Integration procedure:
  \[
  \dot{y} = g^*(y, t) = g_1 \left( y, -\frac{1}{\sigma} g_2(y), t \right)
  \]
DAEs

- Multibody system
  \[ M \ddot{q} + h(q, \dot{q}, t) + B^T \lambda = 0 \]
  \[ \Phi(q) = 0 \]

  - Constraints at velocity level: \[ B \dot{q} = 0 \]
  - Constraints at acceleration level: \[ B \ddot{q} + \dot{B} \dot{q} = 0 \]
  - One more differentiation to obtain \[ \dot{\lambda} \]
  - Differentiation index = 3 !

- Index reduction
  - constraints at velocity level: index-2
  - constraints at acceleration level: index-1 ⇒ drift !!

General forms of DAEs (singular matrix)

- Perturbation index:
  \[ f(\dot{y}, \ddot{y}, t) = \dot{\delta}(t) \]
  \[ \|\ddot{y}(t) - \dot{y}(t)\| = O(\|\delta\|) + O(\|\dot{\delta}\|) + ... + O(\|\delta^{(i-1)}\|) \]

High-index problems = highly sensitive !

<table>
<thead>
<tr>
<th>Multibody simulation</th>
<th>Numerical problems</th>
<th>Precision (constraints)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index-1</td>
<td>~ ODE</td>
<td>Large drift</td>
</tr>
<tr>
<td>Index-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Index-3</td>
<td>Highly sensitive</td>
<td>No error at position level</td>
</tr>
</tbody>
</table>
DAEs: Direct methods

- Generalized-α method [Arnold & B. ’07]

\[ M(q_{n+1})\ddot{q}_{n+1} + h(q_{n+1}, \dot{q}_{n+1}, \lambda_{n+1}) + \dot{B}^T_{n+1}\lambda_{n+1} = 0 \]

\[ \Phi(q_{n+1}) = 0 \]

\[ q_{n+1} = q_n + h\dot{q}_n + h^2(0.5 - \beta)a_n + h^2\beta a_{n+1} \]

\[ \dot{q}_{n+1} = \dot{q}_n + h(1 - \gamma)a_n + h\gamma a_{n+1} \]

\[ (1 - \alpha_m)a_{n+1} + \alpha_m a_n = (1 - \alpha_f)\dot{q}_{n+1} + \alpha_f \dot{q}_n \]

\[ \lambda_{n+1}, \dot{q}_{n+1}, a_{n+1}, \lambda_{n+1} \]

Good prediction required for Newton process

\[ \Rightarrow \] if no convergence, \( h \) may be reduced

Algorithm 1 \([q_{n+1}, \dot{q}_{n+1}, \lambda_{n+1}, 0] = \text{AlphaStep}(q_n, \dot{q}_n, a)\):

- Compute the residuals \( r^q \) (Equation (1)), \( r^\lambda \) (Equation (2))
- if \( \sqrt{\|r^q\|^2 + \|r^\lambda\|^2} < \epsilon \) then break
- end if

\[
\begin{bmatrix}
\Delta q \\
\Delta \lambda
\end{bmatrix} = \begin{bmatrix}
S^{-1} & 0 \\
0 & S^{-1}
\end{bmatrix}
\begin{bmatrix}
r^q \\
r^\lambda
\end{bmatrix}
\]

- Newton iterations

Stopping criterion

Linearized problem
DAEs: Direct methods

- Generalized-\( \alpha \) method [Arnold & B. '07]

  - Tolerance: \( tol = ATOL + RTOL \left( \|M\dot{q}\| + \|h\| + \|B^T \lambda\| + \|\Phi\| \right) \)

  - Iteration matrix: \( S_t = \begin{bmatrix} (M \beta' + C_t \gamma' + K_t) & \Phi_t^T \\ \Phi_t & 0 \end{bmatrix} \)

    \[ \beta' = \frac{1 - \alpha_m}{h^2 \beta (1 - \alpha_t)} \quad \gamma' = \frac{\gamma}{h \beta} \]

    \[ K_t = \partial(M(q)\ddot{q} + h(q, \dot{q}, t)) + B^T(q)\lambda / \partial q \]

    \[ C_t = \partial h(q, \dot{q}, t) / \partial \dot{q} \]

    Bad numerical conditioning for small \( h \)

    \Rightarrow scaling strategy required

DAEs: Direct methods

- Generalized-\( \alpha \) for a double pendulum [Géradin & Cardona '01]

  Undamped scheme

  Damped scheme

  Weak instability for index-3 DAEs!

  stability + accuracy
DAEs: Direct methods

- DAEs direct solvers
  - Generalized-\(\alpha\): index-3 (numerical damping)
  - IRK / RADAU5 (Hairer): index \(\leq 3\)
  - BDF / DASSL (Petzold): index \(\leq 1\)

- Index reduction?
  - Projection
  - Stabilization (Baumgarte, GGL, overdetermined DAEs)
  - Coordinate partitioning / splitting

DAEs: Index reduction

- Projection
  - Index-1 solution: \(\tilde{q}, \dot{q}\)
  - After each time-step: project on the constraints
    \[
    B(q) \dot{q} = 0 \quad \text{with the condition} \quad \min_{\tilde{q}} \|\dot{q} - \dot{\tilde{q}}\|
    \]
    \[
    \Phi(q) = 0 \quad \text{with the condition} \quad \min_{q} \|q - \tilde{q}\|
    \]
    \[
    \tilde{q} \quad \Phi(q) = 0
    \]

\[
\text{UMONS} \quad \text{UCL} \quad \text{GraSMech – Multibody 57}
\]

\[
\text{UMONS} \quad \text{UCL} \quad \text{GraSMech – Multibody 58}
\]
DAEs: Index reduction

- Baumgarte stabilization: $\dot{\Phi} + 2a\Phi + c^2\Phi = 0$
  - Any drift converges dynamically to zero
  - Choice of the parameters?

- GGL formulation:
  $v - \dot{q} = B^T\mu$
  $M(q)\dot{v} + h(q, v, t) + B^T\lambda = 0$
  $B(q)\dot{q} = 0$

  [Gear, Leimkuhler, Gupta '85]

- Overdetermined DAE:
  $v - \dot{q} = 0$
  $M(q)\dot{v} + h(q, v, t) + B^T\lambda = 0$
  $B(q)\dot{q} + \dot{B}(q)\dot{q} = 0$
  $B(q)\dot{q} = 0$
  $\Phi(q) = 0$

  [Führer, Leimkuhler '91]

Generalized coordinate partitioning

- System with $n$ coordinates and $m$ constraints
- Locally, select $n - m$ independent coord. $q^I$, $q = \begin{bmatrix} q^I \\ q^D \end{bmatrix}$
- Solve $\Phi(q) = 0$ for $q^D$: $q^D = \Phi^*(q^I)$
- Eliminate $q^D$ (and derivatives) from the eq. of motion
- Underlying ODE: $\ddot{M}(q^I)\ddot{q}^I + \dot{h}(q^I, \dot{q}^I, t) = 0$
- Efficient numerical implementation

  [Wehage & Haug '82]
DAEs: Index reduction

- Coordinate splitting: \( q = X q^I + Y q^D \)
  \[ \begin{align*}
  q &= X q^I + Y q^D \\
  n \times (n - m) &\quad \text{null space matrix } P:\text{ ok by construction}
  \end{align*} \]

\[ \text{Objectives} \]

\[ \begin{align*}
  B(q) P(q) &= 0 \\
  P(q) &= (I - Y (B(q)^T B(q))^{-1} B(q)) X \\
  P &= \text{basis of the tangent space…}
  \end{align*} \]

- Index-1 DAE:
  \[ \begin{align*}
  P^T(q) (\dot{q} - v) &= 0 \\
  P^T(q) (M(q) \ddot{v} + h(q, v, t)) &= 0 \\
  B(q) \dot{q} &= 0 \\
  \Phi(q) &= 0
  \end{align*} \]

[Yen '93]

DAEs: Index reduction

Null space matrix \((n = 3, m = 1)\)

\[ \begin{align*}
  \phi_1(q) &= 0 \\
  \phi_2(q) &= 0 \\
  \phi_3(q) &= 0 \\
  b_1 &= (\phi_1(q))^T \\
  b_2 &= (\phi_2(q))^T \\
  b_3 &= (\phi_3(q))^T \\
  b_4 &= \ldots \\
  b_{n-m} &= \ldots \\
  B(q) P(q) &= \begin{bmatrix} b_1^T & \ldots & b_{n-m}^T \end{bmatrix} \begin{bmatrix} p_1 \ldots p_{n-m} \end{bmatrix} = 0
  \end{align*} \]

Span the tangent space

\[ \begin{align*}
  q_1, q_2 = \text{independent coord.}
  \end{align*} \]
DAEs: etc...

- Partitioned methods (e.g. half-explicit methods)
  - Implicit method for the « algebraic part »
  - Explicit method for the « differential part »

- Energy conserving schemes, variational integrators…
  - 1st-integrals (energy, momenta) & symplecticity
  - Nonlinear stability analysis
  - Longer integration intervals are possible

Time integration: Conclusion

- Multibody system = index-3 DAE
  - Need sophisticated implicit algorithms
- Combine DAE solvers (RADAU5, DASSL) and
  - Projection (velocity level)
  - Stabilization (GGL, ODAE)
  - Coordinate partitioning / splitting
- Flexible multibody dynamics
  - Large & stiff systems
  - Generalized-α scheme (MECANO: HHT)
  - Energy conserving schemes, variational integrators
Time integration: References

Technical books

Numerical methods in multibody dynamics

A few papers…