Newton Solver Reference

François Faure and Matthieu Nesme

August 27, 2014

Abstract

Detailing the non-linear time-stepping scheme implemented in the CompliantNLImplicitSolver component.

1 Notations

- x: positions
- v: velocities
- $f(x, v, t)$: forces for given positions and velocities at time t
- h: time step
- $(.)^-$, $(.)^+$: a state at, respectively, the beginning and the end of the time step
- $\Delta x = x^+ - x^-$: variation of position during the time step
- $\Delta v = v^+ - v^-$: variation of velocity during the time step
- α, β: blending parameters such as $f^* = \alpha f^+ + (1 - \alpha)f^-$ and $v^* = \beta v^+ + (1 - \beta)v^-$. Corresponding Data are called implicitVelocity and implicitPosition.
- M: mass

2 Euler Integration

$$\left\{ \begin{array}{rcl} \Delta x & = & hv^* \\ M\Delta v & = & hf^* \end{array} \right.$$

from explicit $\alpha = \beta = 0$ to implicit $\alpha = \beta = 1$.

3 Non-linear Solver

The method computes the next velocity v^+, such that $e \equiv M\Delta v - hf^*$ is satisfied.

Based on the Newton's method, an approximate solution is iteratively improved by solving a linear equation system based on the Jacobian of the residual of the equation to satisfy. A first guess is computed with the regular, linearized system (cf the linear time-stepping scheme in the Compliant plugin doc).

Stating that

$$e \equiv M(v^+ - v^-) - h(\alpha f^+ + (1-\alpha)f^-)$$

we obtain the jacobian

$$\frac{\partial e}{\partial v^+} = M - h\alpha \frac{\partial f^+}{\partial v^+}$$

A first order approximation of the Taylor serie of $f^+ = f(x^+, v^+, t{+}h)$ gives

$$\begin{array}{rcl} f^+ & = & f(x^-, v^-, t) + \frac{\partial f}{\partial x}\Delta x + \frac{\partial f}{\partial v}\Delta v \\ & = & f^- + K\Delta x + B\Delta v \\ & = & f^- + K(h(\beta v^+ + (1-\beta)v^-)) + B(v^+ - v^-) \end{array}$$

with $K = \frac{\partial f}{\partial x}$ the stiffness matrix and $B = \frac{\partial f}{\partial v}$ the damping matrix.

So

$$\frac{\partial f^+}{\partial v^+} = h\beta K + B$$

and

$$\frac{\partial e}{\partial v^+} = M - h\alpha B - h^2\alpha\beta K$$

4 Newton Step Length

Two different strategies are implemented:

- naïve sub-step approach: a predefinied portion (Data 0<newtonStepLength<1) of the correction is applied successively while the error is decreasing. This is simple and quite efficient. For 10% (default) it seems to always converge.
- Backtracking algorithm (Data newtonStepLength=1): try to find the maximum amount of correction to apply that decreased "sufficiently" the error. The line search described in *Numerical Recipies* (chapter Globally Convergent Methods for Nonlinear Systems of Equations) is employed. Even if this method should be better, it sometimes diverges, maybe because of its magical thresholds?!