Finding the Contact Force

April 2, 2024

[1]:

```python
import sympy as sp

# Defining symbols
p_s, v_s, a_s = sp.symbols(r'\vec{p}_s \vec{v}_s \vec{a}_s')
k, n = sp.symbols(r'k n')
del_t = sp.Symbol(r'\Delta t')
xi, eta = sp.symbols(r'\xi \eta')
p_k, v_k, a_k = sp.symbols(r'\vec{p}_k \vec{v}_k \vec{a}_k')
phi_k = sp.Function(r'\phi_k')(xi, eta)
F_s, fc, R_s = sp.symbols(r'\vec{F}_s f_c \vec{R}_s')
F_k, N, R_k = sp.symbols(r'\vec{F}_k \vec{N} \vec{R}_k')
m_s, m_k = sp.symbols('m_s m_k')
```

Consider a contact pair (a patch and a single node) in which a force resolution needs to be acquired between these entities. The goal is to ensure that applied force moves the node and patch in such a way that node lies on the surface of the patch at the next time step. Such a condition can be achieved by solving the following equation:

$$\vec{p}_s + \vec{v}_s \Delta t + \frac{1}{2}\vec{a}_s \Delta t^2 = \sum_{k=0}^{n-1} \phi_k(\xi, \eta) \left[\vec{p}_k + \vec{v}_k \Delta t + \frac{1}{2}\vec{a}_k \Delta t^2\right]$$

where p, v, and a are the position, velocity, and acceleration of the node and patch at the current time step, and subscript s denotes the slave node while k denotes the nodes that bound the master patch. Alternatively, the above equation can be represented as a matrix multiplication instead of a summation:

$$\vec{p}_s + \vec{v}_s \Delta t + \frac{1}{2}\vec{a}_s \Delta t^2 = \underbrace{\begin{bmatrix} p_{x0} + v_{x0}\Delta t + \frac{1}{2}a_{x0}\Delta t^2 & p_{x1} + v_{x1}\Delta t + \frac{1}{2}a_{x1}\Delta t^2 & \cdots \\ p_{y0} + v_{y0}\Delta t + \frac{1}{2}a_{y0}\Delta t^2 & p_{y1} + v_{y1}\Delta t + \frac{1}{2}a_{y1}\Delta t^2 & \cdots \\ p_{z0} + v_{z0}\Delta t + \frac{1}{2}a_{z0}\Delta t^2 & p_{z1} + v_{z1}\Delta t + \frac{1}{2}a_{z1}\Delta t^2 & \cdots \end{bmatrix}}_{A} \begin{bmatrix} \phi_0(\xi, \eta) \\ \phi_1(\xi, \eta) \\ \vdots \\ \phi_n(\xi, \eta) \end{bmatrix}$$

The acceleration for the slave node and the master patch node can be written as:

$$\vec{a}_s = \frac{\vec{F}_s + f_c\vec{N} + \vec{R}_s}{m_s}$$

$$\vec{a}_k = \frac{\vec{F}_k - f_c\vec{N} \cdot \phi_k(\xi, \eta) + \vec{R}_k}{m_k}$$

where $\vec{F}$ is the internal force known prior to the analysis, f_c is the incremental contact force between the current node and the patch, and $\vec{R}$ is the force due to other contact pairs. $\vec{N}$ is the unit normal at the contact point (ξ, η) and must be in the outward direction of the patch, facing the non-penetrated slave node.

$$N = \frac{\partial p/\partial\xi \times \partial p/\partial\eta}{|\partial p/\partial\xi \times \partial p/\partial\eta|}$$

[2]:

```python
eq1 = sp.Eq(p_s + v_s*del_t + sp.Rational(1, 2)*a_s*del_t**2, sp.Sum(phi_k*(p_k + v_k*del_t + sp.Rational(1, 2)*a_k*del_t**2), (k, 0, n-1)))
eq1
```

[2]:

$$\frac{\Delta t^2 \vec{a}_s}{2} + \Delta t\vec{v}_s + \vec{p}_s = \sum_{k=0}^{n-1} \left(\frac{\Delta t^2 \vec{a}_k}{2} + \Delta t\vec{v}_k + \vec{p}_k\right) \phi_k(\xi, \eta)$$

[3]:

```python
eq2 = eq1.subs([
    (a_s, (F_s + fc*N + R_s)/m_s),
    (a_k, (F_k - fc*N*phi_k + R_k)/m_k)
])
eq2
```

[3]:

$$\frac{\Delta t^2 \left(\vec{F}_s + \vec{N} f_c + \vec{R}_s\right)}{2m_s} + \Delta t\vec{v}_s + \vec{p}_s = \sum_{k=0}^{n-1} \left(\frac{\Delta t^2 \left(\vec{F}_k - \vec{N} f_c \phi_k(\xi, \eta) + \vec{R}_k\right)}{2m_k} + \Delta t\vec{v}_k + \vec{p}_k\right) \phi_k(\xi, \eta)$$

In the matrix form, this results in:

[4]:

```python
A = sp.Matrix([sp.Function('A')(xi, eta, fc)])
eq3 = sp.Eq(eq2.lhs, sp.MatMul(A, sp.Matrix([phi_k])), evaluate=False)
eq3
```

[4]:

$$\frac{\Delta t^2 \left(\vec{F}_s + \vec{N} f_c + \vec{R}_s\right)}{2m_s} + \Delta t\vec{v}_s + \vec{p}_s = \left[A(\xi, \eta, f_c)\right] \left[\phi_k(\xi, \eta)\right]$$

To solve this with the Newton-Raphson method, we have the following

$$\begin{bmatrix} \xi_{i+1} \\ \eta_{i+1} \\ f_{ci+1} \end{bmatrix} = \begin{bmatrix} \xi_i \\ \eta_i \\ f_{ci} \end{bmatrix} - \mathbf{J}^{-1}\mathbf{F}$$

[5]:

```python
F = sp.Matrix([eq3.lhs]) - eq3.rhs
F
```

[5]:

$$\left[\frac{\Delta t^2\left(\vec{F}_s+\vec{N}f_c+\vec{R}_s\right)}{2m_s} + \Delta t\vec{v}_s + \vec{p}_s - A(\xi,\eta,f_c)\phi_k(\xi,\eta)\right]$$

[6]:
```
J = F.jacobian([xi, eta, fc])
J
```

[6]:

$$\left[-A(\xi,\eta,f_c)\frac{\partial}{\partial\xi}\phi_k(\xi,\eta) - \phi_k(\xi,\eta)\frac{\partial}{\partial\xi}A(\xi,\eta,f_c) \quad -A(\xi,\eta,f_c)\frac{\partial}{\partial\eta}\phi_k(\xi,\eta) - \phi_k(\xi,\eta)\frac{\partial}{\partial\eta}A(\xi,\eta,f_c) \quad \frac{\Delta t^2\vec{N}}{2m_s} - \phi_k(\xi,\eta)\frac{\partial}{\partial f_c}\right.$$

Here is a better look of the result:

$$\left[-A(\xi,\eta,f_c)\frac{\partial}{\partial\xi}\phi_k(\xi,\eta)-\frac{\partial}{\partial\xi}A(\xi,\eta,f_c)\phi_k(\xi,\eta) \; -A(\xi,\eta,f_c)\frac{\partial}{\partial\eta}\phi_k(\xi,\eta)-\frac{\partial}{\partial\eta}A(\xi,\eta,f_c)\phi_k(\xi,\eta) \; \frac{\Delta t^2\vec{N}}{2m_s}-\frac{\partial}{\partial f_c}A(\xi,\eta,f_c)\phi_k(\xi,\eta)\right]$$

Recall that for A, we have

$$\begin{bmatrix} p_{x0}+v_{x0}\Delta t+\frac{1}{2}\frac{F_{x0}-N_x f_c\cdot\phi_0(\xi,\eta)+R_{x0}}{m_0}\Delta t^2 & p_{x1}+v_{x1}\Delta t+\frac{1}{2}\frac{F_{x1}-N_x f_c\cdot\phi_1(\xi,\eta)+R_{x1}}{m_1}\Delta t^2 & \cdots \\ p_{y0}+v_{y0}\Delta t+\frac{1}{2}\frac{F_{y0}-N_y f_c\cdot\phi_0(\xi,\eta)+R_{y0}}{m_0}\Delta t^2 & p_{y1}+v_{y1}\Delta t+\frac{1}{2}\frac{F_{y1}-N_y f_c\cdot\phi_1(\xi,\eta)+R_{y1}}{m_1}\Delta t^2 & \cdots \\ p_{z0}+v_{z0}\Delta t+\frac{1}{2}\frac{F_{z0}-N_z f_c\cdot\phi_0(\xi,\eta)+R_{z0}}{m_0}\Delta t^2 & p_{z1}+v_{z1}\Delta t+\frac{1}{2}\frac{F_{z1}-N_z f_c\cdot\phi_1(\xi,\eta)+R_{z1}}{m_1}\Delta t^2 & \cdots \end{bmatrix}$$

The following matrices are constructed using the outer product:

$$\begin{gathered} A' = \vec{p}_k + \vec{v}_k\Delta t + \frac{1}{2}\Delta t^2\left(\vec{F}_k+\vec{R}_k\right) \\ A = A' - \vec{N}\otimes\phi_k\frac{f_c}{2m_k}\Delta t^2 \\ \frac{\partial}{\partial\xi}A = -\vec{N}\otimes\frac{\partial}{\partial\xi}\phi_k\frac{f_c}{2m_k}\Delta t^2 \\ \frac{\partial}{\partial\eta}A = -\vec{N}\otimes\frac{\partial}{\partial\eta}\phi_k\frac{f_c}{2m_k}\Delta t^2 \\ \frac{\partial}{\partial f_c}A = -\vec{N}\otimes\phi_k\frac{1}{2m_k}\Delta t^2 \end{gathered}$$

The matrix A' is constructed as a column stack for each vector.