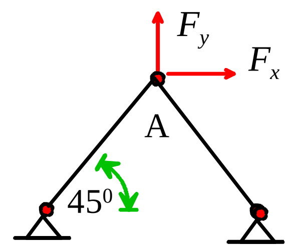
Division of Engineering
Brown University

EN234: Computational methods in Structural and Solid Mechanics

Homework 7: Nonlinear problems
Due Wednesday Nov 20, 2013

1. This problem illustrates the Newton-Raphson method for solving nonlinear equilibrium equations with a very simple example. Consider the truss structure shown in the figure – the two members both have stiffness *k*. When undeformed, the two members have lengths *L*. Loads F_x, F_y act on the joint at A, inducing displacements u_x, u_y .



   a. Write down the total potential energy of the system. Do not assume small deflections.
   b. By minimizing the potential energy derive two nonlinear equilibrium equations for u_x, u_y. Your equations should have the form

$$R_x(u_x, u_y) - F_x = 0$$
$$R_y(u_x, u_y) - F_y = 0$$

   where R_x, R_y are two (pretty messy) functions to be determined
   c. The equilibrium equations can be solved for u_x, u_y by means of the Newton-Raphson method. To do this, we start with some initial guess w_x, w_y for the solution, and then repeatedly correct it by solving

$$\frac{\partial R_x}{\partial u_x} dw_x + \frac{\partial R_x}{\partial u_y} dw_y = -R_x + F_x$$
$$\frac{\partial R_y}{\partial u_x} dw_x + \frac{\partial R_y}{\partial u_y} dw_y = -R_y + F_y$$

   and then correcting the solution so that $w_x + dw_x \quad w_y + dw_y$ (hopefully) approaches the solution. Implement this procedure in a simple MATLAB code.
   d. Test your code by plotting the deformed structure for a few representative values of F_x, F_y (you should be able to make the structure exhibit 'snap through' buckling)

2. Implement an element in FEACHEAP that will solve boundary value problems involving a rate independent, power-law isotropic hardening elastic-plastic solid, with incremental stress-strain relations

$$\Delta\varepsilon_{ij} = \Delta\varepsilon_{ij}^{e} + \Delta\varepsilon_{ij}^{p}$$

$$\Delta\varepsilon_{ij}^{e} = \frac{1+\nu}{E}\left(\Delta\sigma_{ij} - \frac{\nu}{1+\nu}\Delta\sigma_{kk}\delta_{ij}\right) \qquad \Delta\varepsilon_{ij}^{p} = \Delta\varepsilon_{e}\frac{3}{2}\frac{S_{ij}}{\sigma_{e}}$$

$$S_{ij} = \sigma_{ij} - \sigma_{kk}\delta_{ij}/3 \quad \sigma_{e} = \sqrt{\frac{3}{2}S_{ij}S_{ij}} \quad \Delta\varepsilon_{e} = \sqrt{\frac{2}{3}\Delta\varepsilon_{ij}^{p}\Delta\varepsilon_{ij}^{p}}$$

and a yield criterion

$$\sigma_{e} - Y_{0}\left(1+\frac{\varepsilon_{e}}{\varepsilon_{0}}\right)^{1/n} = 0$$

Your solution should include the following steps:

2.1 Devise a method for calculating the stress $\sigma_{ij}^{(n+1)}$ at the end of a load increment. Use a fully implicit computation, in which the yield criterion is exactly satisfied at the end of the load increment. Your derivation should follow closely the procedure discussed in class, except that

a. After computing the elastic predictor for the stress, you should check and see if the stresses are below yield (use the yield criterion). If so, the elastic predictor is the correct stress.

b. If the elastic predictor exceeds yield, the relationship between $\sigma_{e}^{(n+1)}$ and $\Delta\varepsilon_{e}$ must be calculated using the yield criterion, i.e. you should calculate $\Delta\varepsilon_{e}$ such that

$$\sigma_{e}^{n+1} - Y_{0}\left(1+\frac{\varepsilon_{e}+\Delta\varepsilon_{e}}{\varepsilon_{0}}\right)^{1/n} = 0$$

You can use the approach discussed in class (show that $S_{ij}^{n+1} = S_{ij}^{*} - \Delta\varepsilon_{e}\frac{E}{1+\nu}\frac{3}{2}\frac{S_{ij}^{n+1}}{\sigma_{e}^{n+1}}$, set $S_{ij}^{n+1} = \beta S_{ij}^{*}$) to obtain a nonlinear equation that can be solved for $\Delta\varepsilon_{e}$ using Newton-Raphson iteration.

2.2 Calculate the tangent stiffness $\partial\sigma_{ij}^{(n+1)}/\partial\Delta\varepsilon_{kl}$ for the rate independent solid, by differentiating the result of 1. This is pretty horrible, but simpler than the rate dependent case. Make sure your expression is symmetric.

2.3 Implement the results of 1 and 2 in your code. It is simplest to do this in the fully integrated element, but you can use your B-bar element as well if you prefer. If you do this, it is important to use the correct element residual vector – instead of

$$R_{i}^{a} = \int_{V}\sigma_{ij}\frac{\partial N^{a}}{\partial x_{j}}dV$$

in a standard element, you need to use

$$R_{i}^{a} = \int_{V}\sigma_{ij}\frac{\partial N^{a}}{\partial x_{j}} + \sigma_{kk}\left(B_{ia}^{vol} - \frac{1}{3}\frac{\partial N^{a}}{\partial x_{i}}\right)dV \qquad B_{ia}^{vol} = \frac{1}{3V}\int_{V}\frac{\partial N^{a}}{\partial x_{i}}dV$$

(the residual vector is not used for a linear elasticity computation – it is zero – so this was not necessary in the linear elasticity code unless you used the nonlinear solver).

You will also need to modify your input file to activate the nonlinear equation solver. It is a good idea to apply the load in a series of steps – for example, to ramp the load up from zero to 10 in 5 steps, you can use the following key words:

```
% The HISTORY key defines a time history that can be applied to DOFs or
%distributed loads.  The numbers in the table are time,load value, and are
%interpolated linearly by the code
        HISTORY, dload_history
          0.d0, 0.d0
          1.d0,  10.d0

% Syntax here is element set, face #, history name, nx,ny,(nz) (time dependent
%pressure to element face in direction (nx,ny,nz))

%        DISTRIBUTED LOADS
%          end_element, 4, dload_history, 1.d0,0.d0,0.d0
%        END DISTRIBUTED LOADS

%   The STATIC STEP key initializes a static load step

     STATIC STEP

%     The TIME STEP key defines values of parameters controlling time stepping.
%        These parameters are passed to subroutine staticstep.
%        The parameters must be entered in the correct order

          TIME STEP
%        Initial time step value
           0.2d0
%        Max and min time step (making the max 0.2 will ensure at least 5 steps)
           0.2d0, 0.001d0
%        Max no. time steps (should stop after 5 steps when t=1 unless there is
%a cutback in time step caused by poor convergence)
           15
%        Stop time
           1.d0
%        Time interval between state prints and no. steps between state prints
           1000.d0, 1
%        Time interval between user prints and no. steps between user prints
           1000.d0, 1000
% Syntax here is solver type, nonlinear equations, NR tolerance, max iterations.
         SOLVER, FACTOR, NONLINEAR, 0.00001, 20
```

It is also really helpful to run the CHECK STIFFNESS on your code to make sure that the residual and stiffness are consistent – if not, the Newton-Raphson iterations are unlikely to converge. You can test your element by comparing its predictions to the MATLAB version…

2.4 Test your code by using it to calculate the stress-strain relation for the viscoplastic material under uniaxial tension. Model the specimen using a single 8 noded brick, and use material properties $E=10000,\ \ \nu=0.3\ \ Y=18,\ \varepsilon_0=0.5\ \ n=10$.

3. **Optional** Implement a finite-strain F-bar hyperelastic element in FEACHEAP. For simplicity, consider a compressible Neo-Hookean material with stress-strain relation

$$\sigma_{ij}=\frac{\mu_1}{J^{5/3}}\left(B_{ij}-\frac{1}{3}B_{kk}\delta_{ij}\right)+K_1\left(J-1\right)\delta_{ij}$$

The tangent stiffness for this material is

$$C^{e}{}_{ijkl} = \frac{\mu_1}{J^{2/3}}\left(\delta_{ik} B_{jl} + B_{il}\delta_{jk} - \frac{2}{3}\left\{ B_{ij}\delta_{kl} + B_{kl}\delta_{ij} \right\} + \frac{2}{3}\frac{B_{qq}}{3}\delta_{ij}\delta_{kl} \right) + K_1(2J-1)J\delta_{ij}\delta_{kl}$$

To implement the F-bar element, we re-write the virtual work equation as

$$\int_{V_0} \tau_{ij}[\bar{F}_{kl}]\delta\bar{L}_{ij} dV_0 - \int_{V_0} \rho_0 b_i \delta v_i dV_0 - \int_{\partial_2 V_0} t_i^* \delta v_i \eta dA_0 = 0$$

Where $\bar{\mathbf{F}}, \delta\bar{\mathbf{L}}$ are modified deformation and velocity gradients, computed as

$$\bar{F}_{ij} = F_{ij}\left(\eta / J\right)^{1/n}, \ \delta\bar{L}_{ij} = \delta L_{ij} + \delta_{ij}(\delta\dot{\eta}/\eta - \delta L_{kk})/n$$

where $n=2$ for a 2D problem and $n=3$ for a 3D problem, while J=det($\mathbf{F}$), and

$$F_{ij} = \delta_{ij} + \frac{\partial u_i}{\partial x_j} = \delta_{ij} + \sum_{a=1}^{n} \frac{\partial N^a}{\partial x_j} u_i^a$$

$$\delta L_{ij} = \frac{\partial \delta v_i}{\partial y_j} = \frac{\partial \delta v_i}{\partial x_k}\frac{\partial x_k}{\partial y_j} = \frac{\partial \delta v_i}{\partial x_k} F_{kj}^{-1} \qquad = \sum_{a=1}^{n} \frac{\partial N^a}{\partial x_k} F_{kj}^{-1} \delta v_i^a$$

$$\eta = \frac{1}{V_{0el}} \int_{V_{0el}} \det(\mathbf{F}) dV \qquad \dot{\eta} = \frac{1}{V_{0el}} \int_{V_{0el}} J F_{ji}^{-1} \dot{F}_{ij} dV = \frac{1}{V_{0el}} \int_{V_{0el}} J L_{kk} dV$$

The modified virtual work equation must be solved for the unknown nodal displacements by Newton-Raphson iteration. As usual, the Newton-Raphson procedure involves repeatedly solving the following system of linear equations for corrections to the displacement field dw_k^b

$$K_{aibk} dw_k^b = -R_i^a + F_i^a$$

where

$$F_i^a = \int_{V_0} \rho_0 b_i N^a dV_0 + \int_{\partial_2 V_0} t_i^* N^a \hat{\eta} dA_0$$

$$R_i^a = \int_{V_0} \tau_{ij}[\bar{F}_{kl}] \frac{\partial N^a}{\partial y_j} + \frac{\tau_{pp}[\bar{F}_{kl}]}{n}\left(\frac{1}{\eta}\frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) dV_0$$

and $\tau_{ij} = \eta\sigma_{ij}$ is the Kirchhoff stress, calculated from the constitutive equation using $\bar{\mathbf{F}}$ to determine the strain measures. The consistent tangents follow from linearizing the virtual work equation

$$K_{aibk} = \int_{V_0} \frac{\partial \tau_{mj}[\bar{F}_{kl}]}{\partial \bar{F}_{pq}} \frac{\partial \bar{F}_{pq}}{\partial u_k^b} \left(\delta_{im} \frac{\partial N^a}{\partial y_j} + \frac{\delta_{mj}}{n}\left(\frac{1}{\eta}\frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) \right) dV_0$$

$$+ \int_{V_0} \tau_{mj}[\bar{F}_{kl}] \frac{\partial}{\partial u_k^b} \left(\delta_{im} \frac{\partial N^a}{\partial y_j} + \frac{\delta_{mj}}{n}\left(\frac{1}{\eta}\frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) \right) dV_0$$

Where

$$\frac{\partial N^a}{\partial y_j} = \frac{\partial N^a}{\partial x_k} F_{kj}^{-1}$$

Some tedious algebra shows that the integrands can be reduced to

$$\frac{\partial \tau_{mj}[\bar{F}_{kl}]}{\partial \bar{F}_{pq}} \frac{\partial \bar{F}_{pq}}{\partial u_k^b} \left(\delta_{im} \frac{\partial N^a}{\partial y_j} + \frac{\delta_{mj}}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) \right)$$

$$= \bar{C}^e_{mjpr} \left(\frac{\partial N^b}{\partial \bar{y}_r} \delta_{pk} + \frac{\delta_{pr}}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial w_k^b} - \frac{\partial N^b}{\partial y_k} \right) \right) \left(\delta_{im} \frac{\partial N^a}{\partial y_j} + \frac{\delta_{mj}}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) \right)$$

$$= \bar{C}^e_{ijkl} \frac{\partial N^b}{\partial \bar{y}_l} \frac{\partial N^a}{\partial y_j} + \bar{C}^e_{ijpp} \frac{1}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial w_k^b} - \frac{\partial N^b}{\partial y_k} \right) \frac{\partial N^a}{\partial y_j} + \bar{C}^e_{jjkl} \frac{\partial N^b}{\partial \bar{y}_l} \frac{1}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right)$$

$$+ \bar{C}^e_{jjpp} \frac{1}{n^2} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial w_k^b} - \frac{\partial N^b}{\partial y_k} \right) \left(\frac{1}{\eta} \frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right)$$

Where $\bar{\mathbf{C}}^e$ is the tangent stiffness, but again computed using $\bar{\mathbf{F}}$, and

$$\frac{\partial N^a}{\partial \bar{y}_j} == \frac{\partial N^a}{\partial x_k} \bar{F}_{kj}^{-1} \qquad \frac{\partial \eta}{\partial u_i^a} = \frac{1}{V_{0el}} \int_{V_{0el}} \det(\mathbf{F}) \frac{\partial N^a}{\partial y_i} dV$$

In addition

$$\frac{\partial}{\partial u_k^b} \left(\delta_{im} \frac{\partial N^a}{\partial y_j} + \frac{\delta_{mj}}{n} \left(\frac{1}{\eta} \frac{\partial \eta}{\partial u_i^a} - \frac{\partial N^a}{\partial y_i} \right) \right) = -\delta_{im} \frac{\partial N^a}{\partial y_k} \frac{\partial N^b}{\partial y_j} + \frac{\delta_{mj}}{n} \left(\frac{1}{\eta} \frac{\partial^2 \eta}{\partial u_i^a \partial u_k^b} - \frac{1}{\eta^2} \frac{\partial \eta}{\partial u_i^a} \frac{\partial \eta}{\partial u_k^b} + \frac{\partial N^a}{\partial y_k} \frac{\partial N^b}{\partial y_i} \right)$$

with

$$\frac{\partial N^a}{\partial \bar{y}_j} == \frac{\partial N^a}{\partial x_k} \bar{F}_{kj}^{-1} \qquad \frac{\partial^2 \eta}{\partial u_i^a \partial u_k^b} = \frac{1}{V_{0el}} \int_{V_{0el}} \det(\mathbf{F}) \left(\frac{\partial N^b}{\partial y_k} \frac{\partial N^a}{\partial y_i} - \frac{\partial N^b}{\partial y_i} \frac{\partial N^a}{\partial y_k} \right) dV$$

You could check this element by modeling a near incompressible pressurized hyperelastic cylinder, and comparing the numerical solution to the analytical one.