EN234: Computational methods in Structural and Solid Mechanics

Homework 6: Nonlinear materials
Due Wed Oct 21, 2015

School of Engineering
Brown University

In this homework you will extend EN234FEA to solve problems for nonlinear materials. You can choose whether to implement a hypoelastic material (simple) or a cyclic plasticity model (much harder). Solve either problem 1 or problem 2 (you are welcome to solve both, if you wish)….

1. Extend EN234FEA to solve problems with the simple hypoelastic material model discussed in class. The stress-strain relation is:

$$\sigma_{ij} = \frac{2}{3}\sigma_e \frac{e_{ij}}{\varepsilon_e} + K\varepsilon_{kk}\delta_{ij}$$

where

$$e_{ij} = \varepsilon_{ij} - \frac{1}{3}\varepsilon_{kk}\delta_{ij} \qquad \varepsilon_e = \sqrt{\frac{2}{3}e_{ij}e_{ij}}$$

$$\frac{\sigma_e}{\sigma_0} = \begin{cases} \sqrt{\dfrac{1+n^2}{(n-1)^2} - \left(\dfrac{n}{n-1} - \dfrac{\varepsilon_e}{\varepsilon_0}\right)^2} - \dfrac{1}{n-1} & \varepsilon_e \le \varepsilon_0 \\ \left(\dfrac{\varepsilon_e}{\varepsilon_0}\right)^{1/n} & \varepsilon_e \ge \varepsilon_0 \end{cases}$$

and K is the bulk modulus (usually taken to be large, to model a near-incompressible material).

The tangent modulus is

$$\frac{d\sigma_{ij}}{d\varepsilon_{kl}} = \frac{4}{9}\left(E_t - E_s\right)\frac{e_{ij}e_{kl}}{\varepsilon_e^2} + \frac{2}{3}E_s\left(\frac{\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}}{2} - \frac{\delta_{ij}\delta_{kl}}{3}\right) + K\delta_{kl}\delta_{ij}$$

where $E_s = \sigma_e / \varepsilon_e$, and $E_t = d\sigma_e / d\varepsilon_e$ (for index notation practice, see if you can verify the derivation given in class)

To implement this material model you will need to do complete the following steps (shown for a 3D element; the vectors/matrices will change for 2D):

1. You can either (1) create a new element subroutine (eg copy your B-bar element from last week into a new file) or else add a conditional statement in your existing B-bar element from last week that will allow the user to choose between a standard linear elastic material or a hypoelastic material (if you like, you can use the value of the first 'Element Property' to identify the material model, or you can simply count the number of material properties)
2. Write a subroutine to calculate the stress vector $[\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]$ and the material tangent stiffness matrix $[D]$ given a value for the *total* strain vector (i.e. strain+dstrain in the code) $[\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, 2\varepsilon_{12}, 2\varepsilon_{13}, 2\varepsilon_{23}] + [\Delta\varepsilon_{11}, \Delta\varepsilon_{22}, \Delta\varepsilon_{33}, 2\Delta\varepsilon_{12}, 2\Delta\varepsilon_{13}, 2\Delta\varepsilon_{23}]$. The D matrix can be calculated as follows

$$[D] = \frac{4}{9\varepsilon_e^2}(E_t - E_s)\underline{e}\otimes\underline{e} + \frac{E_s}{3}\begin{bmatrix} 2 & & & & & 0 \\ & 2 & & & & \\ & & 2 & & & \\ & & & 1 & & \\ & & & & 1 & \\ 0 & & & & & 1 \end{bmatrix} + \left(K - \frac{2E_s}{9}\right)\begin{bmatrix} 1 & 1 & 1 & & & 0 \\ 1 & 1 & 1 & & & \\ 1 & 1 & 1 & & & \\ & & & 0 & & \\ & & & & 0 & \\ 0 & & & & & 0 \end{bmatrix}$$

where $\underline{e} = [e_{11}, e_{22}, e_{33}, e_{12}, e_{13}, e_{23}]$ is the total deviatoric strain vector (calculated using the total strain at the end of the step – note there is no factor of 2 in the shear strains). You can compute the dyadic product of two vectors in FORTRAN using

```
    e_dyadic_e =  spread(e,dim=2,ncopies=6)*spread(e,dim=1,ncopies=6)
```

You will need to trap for the special case of $\varepsilon_{ij} = 0$ - in this limit $\sigma_{ij} = 0$, and $E_t = E_s$ so the first term in *D* vanishes, and you can set $E_s = d\sigma_e / d\varepsilon_e$ in the second and third terms.

3. Use the calculated [D] and $[\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}]$ to calculate the element stiffness and residual (you won't need to change the code in any way other than to calculate [D] and the stress correctly). Use your B-bar matrix coded last week, since you are interested[1] in solving problems for near-incompressible materials.
4. You will need to modify your 'field variables' subroutine to compute the stresses for the new material model.
5. Create an input file to test your element/material model. Run a single element test (you can copy the simple 1 element 3D linear elasticity input file) . The material properties for your new element are the values of $\{\sigma_0, \varepsilon_0, n, K\}$. You will need to use the nonlinear solver. For this you will need to changes the lines in the input file that reads

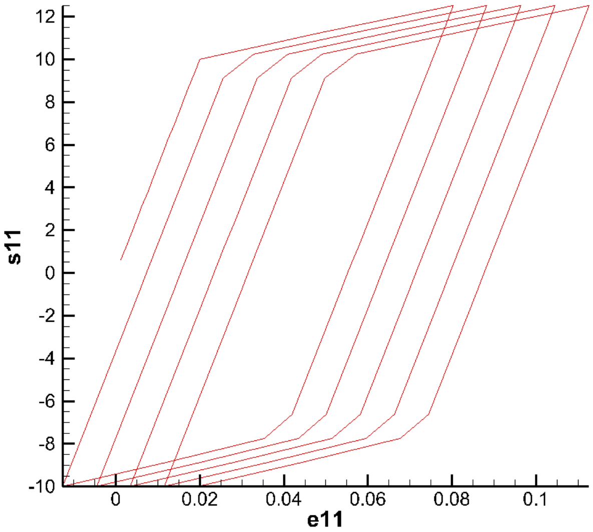
   SOLVER, DIRECT, LINEAR to SOLVER, DIRECT, NONLINEAR, 1.d-06, 15

   (the first number is the tolerance that will be used to decide that the iterations have converged, the second is the maximum Newton iterations that should be attempted before reducing the time step size and starting a second attempt to find a solution).
6. Run the 'check stiffness' procedure (stop the calculation just after the CHECK STIFFNESS) to check that your material stiffness is correct. Use $\sigma_0 = 1,\ \varepsilon_0 = 0.01,\ n = 10,\ K = 1000$. You will need to make a small change in the way the check stiffness procedure computes the numerical derivative for this problem: in the file named checkstiffness.f90, find the lines that read

   ```
   element_dof_increment(icount) = element_dof_increment(icount) + 1.D-07
   numerical_stiffness(1:iu,icount) = -(resid1(1:iu)-resid0(1:iu))/1.D-07
   element_dof_increment(icount) = element_dof_increment(icount) - 1.D-07
   ```

   and change the 1.2-07 to 1.d-12. The analytical and numerical stiffness should then agree to all the decimal places printed.
7. Run a test in which a single element is subjected to a prescribed displacement, and plot the predicted stress-v-strain behavior for the element. You can use the subroutine provided in the 'user_print.f90' file that computes the volume average of the stress and strain in an element for this purpose. Try the following material parameters: $\sigma_0 = 1,\ \varepsilon_0 = 0.01,\ n = 10,\ K = 1000$. Apply a cycle of displacement to the element that cycles the strain between limits of ± 0.1. You can check your calculations against the analytical solution. You could use the 'user print' subroutine to print a stress-v-strain curve (but you will have to modify the '`compute_element_volume_average_3D`' subroutine to calculate the stress state, since for a hypoelastic material stress is not stored as a state variable)
8. You can run other more complicated tests at your discretion; you could also code a 2D version (for plane strain or axisymmetry this is fairly simple; plane stress is more challenging).

[1] Perhaps 'compelled' is a better word than 'interested'



2. In this problem, your mission (should you choose to accept it) is to implement a nonlinear kinematic hardening plasticity model that is often used to idealize the cyclic plastic response of polycrystalline metals. Its main feature is that it is able to predict 'cyclic creep' or 'ratchetting' behavior, i.e. the tendency of a tensile specimen to progressively increase in length when subjected to a tension/compression cycle with a tensile mean stress, as shown in the figure. For simplicity, we focus only on small strains.

The 'Armstrong-Frederick' nonlinear kinematic hardening model has stress-strain relations given by

$$\dot{\varepsilon}_{ij} = \dot{\varepsilon}_{ij}^{e} + \dot{\varepsilon}_{ij}^{p}$$

$$\dot{\varepsilon}_{ij}^{e} = \frac{1+\nu}{E}\left(\dot{\sigma}_{ij} - \frac{\nu}{1+\nu}\dot{\sigma}_{kk}\delta_{ij}\right) \qquad \dot{\varepsilon}_{ij}^{p} = \dot{\varepsilon}_{e}\frac{3}{2}\frac{\left(S_{ij}-\alpha_{ij}\right)}{\sigma_e}$$

$$S_{ij} = \sigma_{ij} - \sigma_{kk}\delta_{ij}/3 \qquad \sigma_e = \sqrt{\frac{3}{2}\left(S_{ij}-\alpha_{ij}\right)\left(S_{ij}-\alpha_{ij}\right)} \qquad \dot{\varepsilon}_e = \sqrt{\frac{2}{3}\dot{\varepsilon}_{ij}^{p}\dot{\varepsilon}_{ij}^{p}}$$

Here, α_{ij} is a material state variable that describes how the material hardens. The yield criterion is

$$\sigma_e - Y_0 \le 0$$

and α_{ij} obeys a hardening law

$$\dot{\alpha}_{ij} = \frac{2}{3}c\dot{\varepsilon}_{ij}^{p} - \gamma\dot{\varepsilon}_e\alpha_{ij}$$

You can find more background reading on plasticity and hardening laws here.

Your solution should include the following steps:

Stress Update Algorithm:

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

1. After computing the elastic predictor

$$S_{ij}^{*} = S_{ij}^{n} + \frac{E}{1+\nu}\Delta e_{ij} \qquad \Delta e_{ij} = \Delta\varepsilon_{ij} - \frac{1}{3}\Delta\varepsilon_{kk}\delta_{ij}$$

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.

2. If the elastic predictor exceeds yield, the plastic strain magnitude $\Delta\varepsilon_e$ must be calculated by solving (using Newton-Raphson iteration inside your stress calculation subroutine)

$$F(\Delta\varepsilon_e, S_{ij}^*) = Y_0 + \left(\frac{3}{2}\frac{E}{(1+\nu)} + \frac{c}{1+\gamma\Delta\varepsilon_e}\right)\Delta\varepsilon_e - \sqrt{\frac{3}{2}\left(S_{ij}^* - \frac{\alpha_{ij}^n}{1+\gamma\Delta\varepsilon_e}\right)\left(S_{ij}^* - \frac{\alpha_{ij}^n}{1+\gamma\Delta\varepsilon_e}\right)} = 0$$

Please derive this result. You can use the approach discussed in class – start by showing

$$\begin{aligned} S_{ij}^{n+1} &= S_{ij}^* - \Delta\varepsilon_e \frac{E}{1+\nu}\frac{3}{2}\frac{\left(S_{ij}^{n+1} - \alpha_{ij}^{n+1}\right)}{\sigma_e^{n+1}} \\ \Rightarrow S_{ij}^{n+1} - \alpha_{ij}^{n+1} &= S_{ij}^* - \alpha_{ij}^{n+1} - \Delta\varepsilon_e \frac{E}{1+\nu}\frac{3}{2}\frac{\left(S_{ij}^{n+1} - \alpha_{ij}^{n+1}\right)}{\sigma_e^{n+1}} \end{aligned} \qquad (1)$$

You can then solve the hardening relation

$$\alpha_{ij}^{n+1} = \alpha_{ij}^n + \Delta\varepsilon_e c \frac{\left(S_{ij}^{n+1} - \alpha_{ij}^{n+1}\right)}{\sigma_e^{n+1}} - \gamma\Delta\varepsilon_e \alpha_{ij}^{n+1}$$

for α_{ij}^{n+1}. Substituting this result for α_{ij}^{n+1} on the right hand side of the second equation in (1) and rearranging the result (collect $S_{ij}^{n+1} - \alpha_{ij}^{n+1}$ on the left hand side, square both sides and multiply by 3/2, and simplify using the yield criterion) gives the answer you need.

3. Next, show that

$$S_{ij}^{n+1} = \lambda\left(S_{ij}^* + \eta\alpha_{ij}^n\right) \qquad \lambda = \left(1 + \left[\frac{3}{2}\frac{E}{Y_0(1+\nu)} - \eta\frac{c}{Y_0}\right]\Delta\varepsilon_e\right)^{-1} \qquad \eta = \frac{3}{2}\frac{E\Delta\varepsilon_e}{(1+\nu)\left(Y_0 + (\gamma Y_0 + c)\Delta\varepsilon_e\right)}$$

$$\alpha_{ij}^{n+1} = \frac{1}{\left(1 + (\gamma + c/Y_0)\Delta\varepsilon_e\right)}\left(\alpha_{ij}^n + \frac{\Delta\varepsilon_e c}{Y_0} S_{ij}^{n+1}\right)$$

so that

$$\sigma_{ij}^{n+1} = \lambda S_{ij}^* + \lambda\eta\alpha_{ij}^n + \frac{1}{3}\sigma_{kk}^{(n)}\delta_{ij} + \frac{E}{3(1-2\nu)}\Delta\varepsilon_{kk}\delta_{ij}$$

4. Finally, calculate the elastic-plastic tangent matrix. This is a tedious, but straightforward calculation. The following results are helpful:

$$\frac{\partial S_{ij}^*}{\partial\Delta\varepsilon_{kl}} = \frac{E}{(1+\nu)}\left(\frac{\delta_{ik}\delta_{jl} + \delta_{jk}\delta_{il}}{2} - \frac{1}{3}\delta_{ij}\delta_{kl}\right)$$

$$F(\Delta\varepsilon_e, S_{ij}^*) = 0 \qquad \Rightarrow \frac{\partial\Delta\varepsilon_e}{\partial S_{ij}^*} = -\left(\frac{\partial F}{\partial\Delta\varepsilon_e}\right)^{-1}\frac{\partial F}{\partial S_{ij}^*}$$

It is also helpful to note that $\alpha_{kk} = 0$. If I did the calculation correctly, you should find that

$$\begin{aligned}\frac{\partial \sigma_{ij}^{(n+1)}}{\partial \Delta\varepsilon_{kl}} &= \frac{\lambda E}{(1+\nu)}\left(\frac{\delta_{ik}\delta_{jl}+\delta_{jk}\delta_{il}}{2}-\frac{1}{3}\delta_{ij}\delta_{kl}\right)+\frac{E}{3(1-2\nu)}\delta_{ij}\delta_{kl} \\ &+\frac{3}{2}\beta\frac{E}{(1+\nu)}\left(\frac{\partial F}{\partial \Delta\varepsilon_e}\right)^{-1}\left\{\frac{\partial\lambda}{\partial\Delta\varepsilon_e}S_{kl}^{*}S_{ij}^{*}-\left(\frac{\partial\lambda}{\partial\Delta\varepsilon_e}\eta+\lambda\frac{\partial\eta}{\partial\Delta\varepsilon_e}\right)\frac{\alpha_{kl}^{n}\alpha_{ij}^{n}}{1+\gamma\Delta\varepsilon_e}\right. \\ &\qquad\left.+\left(\frac{\partial\lambda}{\partial\Delta\varepsilon_e}\eta+\lambda\frac{\partial\eta}{\partial\Delta\varepsilon_e}\right)S_{kl}^{*}\alpha_{ij}^{n}-\frac{\partial\lambda}{\partial\Delta\varepsilon_e}\frac{\alpha_{kl}^{n}S_{ij}^{*}}{1+\gamma\Delta\varepsilon_e}\right\}\end{aligned}$$

where

$$\beta=\left(Y_0+\left(\frac{3}{2}\frac{E}{(1+\nu)}+\frac{c}{1+\gamma\Delta\varepsilon_e}\right)\Delta\varepsilon_e\right)^{-1}$$

This can probably be simplified a lot more, but I am too lazy….

Implementing the new material model in your code

The next step is to add this material model to your EN234FEA fork. Some suggestions:

- If you like, you can add the code to your 3D small-strain B-bar element. To do this, you could add an integer valued flag to the list of element properties that will allow the user to select either a linear elastic material model, or your new plasticity model; alternatively, you could simply check the number of properties to control the selection.
- Your material model is now history dependent, so you will need to store values for state variables that evolve with the deformation. You will need to store the deviatoric stress; the hydrostatic stress; and the components of α_{ij} . You will need to store values for these variables *for each integration point in the element*. Note that $\alpha_{ij}=\alpha_{ji}$ so it can be stored as a 1-D vector using the same convention that is used to store stresses $[\alpha_{11},\alpha_{22},\alpha_{33},\alpha_{12},\alpha_{13},\alpha_{23}]$. If you store deviatoric stress, pressure, and α_{ij} , you will have 13 state variables per integration point – for a 4 noded hexahedron with 8 integration points this would come to 104 total state variables per element.

  In EN234FEA, all state variables for an element are stored in two 1-D vectors: the user subroutine will provide the values of state variables at the start of the step in a vector called initial_state_variables; your user-subroutine must return their values at the end of the step in a variable called updated_state_variables. You can use any storage scheme you like for your variables as long as they are consistent. For example, in my code I extract the state variables for the first integration point as

  ```
      sdev0 = initial_state_variables(1:6)
      press0 = initial_state_variables(7)
      alpha0 =  initial_state_variables(8:13)
  ```

  The second integration point starts at initial_state_variables(14), and so on.
- You must also specify the number of state variables *per element* in the list of element parameters in the input file.
- You will need to write a subroutine that calculates the stress vector and the material tangent matrix. To calculate the stress and tangent, first compute the elastic predictor; then check for yield; then update the stress and state variables and finally compute the material tangent matrix **D**. Although the expression for **D** is quite long you can assemble it very quickly by using outer

products of the stress and α_{ij} vectors (eg $S^*_{ij}\alpha_{kl} \equiv \mathbf{S}^* \otimes \boldsymbol{\alpha}$). Note that in Fortran90 you can calculate the outer product of two 6 dimensional vectors $\mathbf{S} \otimes \boldsymbol{\alpha}$ as follows

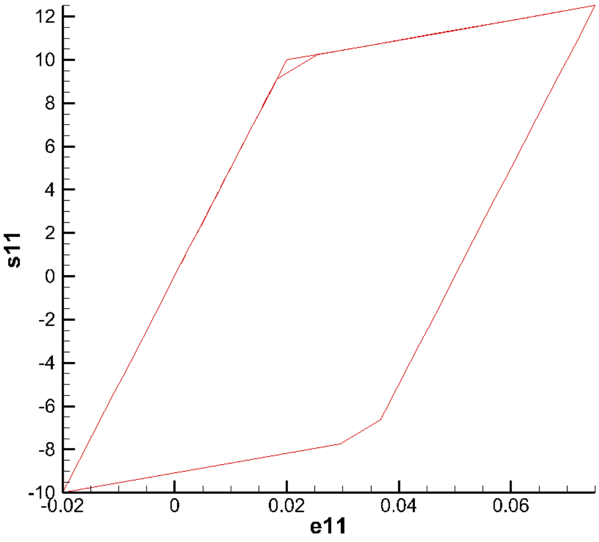
```
    spread(S,dim=2,ncopies=6)*spread(alpha,dim=1,ncopies=6)
```

Coding this element is quite tricky – a small sign error will cause everything to blow up (if it is any consolation, it took me an entire day to find a sign error in the formula given above for λ).

- Check your code with the CHECK STIFFNESS in the input file.
- I found it helpful to add a few lines in the code that printed the value of

  $$\sigma_e = \sqrt{\frac{3}{2}\left(S_{ij}^{n+1} - \alpha_{ij}^{n+1}\right)\left(S_{ij}^{n+1} - \alpha_{ij}^{n+1}\right)}$$

  If yield is exceeded, the result should be *exactly* equal to Y_0 if you typed everything in correctly.
- Be very careful with calculating terms like $S_{ij}\alpha_{ij}$ - it is tempting to interpret this as the dot product of the stress and $\underline{\alpha}$ vectors, but this is not the case, because the off-diagonal terms only appear once in the dot product but twice in the tensor inner product.
- Test your code by calculating the stress-strain response of one or two elements to a cycle of uniaxial tension. A sample input file for this purpose (for 3D elements) has been provided in cyclic_plastic_3d.in. Also, a user-subroutine (in user_print.f90) has been provided that calculates the volume averaged strain and volume averaged state variables in an element (you can choose the element in the input file), and prints the time history of these volume averaged quantities to a file that can be read with TECPLOT (use the XY-line option to plot the results). Run this test in stages: first choose Y_0 to be much larger than the applied stress, and check that you get the correct elastic behavior. Then run a simulation with $E = 500, \nu = 0.3, Y_0 = 10, c = 50, \gamma = 0$, and stresses varying from $-10 < \sigma_{11} < 12.5$. You should see that the stress-strain curve is a closed cycle of plastic strain, and the hardening rate is linear, as shown in the figure. Finally, run a simulation with $E = 500, \nu = 0.3, Y_0 = 10, c = 50, \gamma = 3$: this should produce the ratchetting behavior shown at the top of the homework.



- If this works, you could set up some more interesting boundary value problems to explore as well (e.g. cyclic behavior ahead of a fatigue crack tip; cyclic creep in a pressure vessel, etc).