Modify the linear static truss analysis program to do a Newton iteration to establish the nodal displacements that are consistent with the nonlinear geometry of motion and an elasto-plastic constitutive model in each element. Define the residual as

$$\mathbf{g}(\mathbf{u}) = \sum_{e=1}^{M} \ell_e \mathbf{B}_e \mathbf{E}_e(\mathbf{u}) N_e(\varepsilon_e) - \mathbf{p}$$

where ℓ_e is the original length of member e, $\mathbf{p}$ is the vector of applied nodal loads, $N_e(\varepsilon_e)$ is the axial force constitutive function (i.e., the magnitude of the axial force vector is $\lambda_e N_e$ where λ_e is the ratio of current length of the member to original length), ε_e is the Lagrangian strain given by

$$\varepsilon_e = \tfrac{1}{2}(\lambda_e^2 - 1)$$

and the difference matrix $\mathbf{B}$ is defined as

$$\mathbf{B}_e = \frac{1}{\ell_e}\begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \end{bmatrix} \qquad \mathbf{B}_e\mathbf{u} = \frac{1}{\ell_e}\left(\mathbf{u}_{j(e)} - \mathbf{u}_{i(e)}\right)$$

with the negative 3 by 3 identity in the place associated with the *i-node* and the positive identity in the place associated with the *j-node*. The strain-displacement matrix

$$\mathbf{E}_e(\mathbf{u}) = \mathbf{n}_e + \mathbf{B}_e\mathbf{u}$$

The tangent matrix can be computed (assembled) as

$$\mathbf{A} = \sum_{e=1}^{M} \ell_e \mathbf{B}_e \left(\mathbf{E}_e \frac{\partial N_e}{\partial \varepsilon_e} \mathbf{E}_e^T + N_e \mathbf{I} \right) \mathbf{B}_e^T$$

Modify the linear truss analysis code to assemble the tangent and residual in the *assemble* function and pass those back to the main program for use in the Newton iteration loop. Note that currently the *assemble* function only assembles the stiffness matrix.

Modify the *elem* function to model elasto-plastic response of the member (i.e., compute the trial elastic force, test for yielding, reset the force to the plastic value if yielding has occurred, and set the tangent to zero if the element is yielding). Update the inelastic strain in the function *stresses* (which gets called after the Newton iteration converges. While the static code will only do one load step this update is preparatory for implementing the model in the dynamic version (coming soon to HW near you!). Note also, that you sill need to augment the *d* array to include the yield value (i.e., *d*(2), since *d*(1) is the *EA* value) and you will need to establish an array to hold the plastic strain for each element.

Note that this assignment is essentially about reorganizing the static code to do nonlinear analysis. Once this task is complete we can implement dynamics with very little modification.