Non Linear Finite Element Analysis

Prakash Chinnasamy, Arun Prakash Ganapathy

July 11, 2019

Problem Description

The problem provided contains a spherical inclusion of radius r_i which undergoes a phase transformation inside an ideally elastic-plastic matrix material. The matrix material is modeled as concentric sphere of radius r_o. The phase transformation of the inclusion only leads only to the volumetric strain without change of shape. From the model it is observed that, it exhibits spherical symmetry with respect to the center of the inclusion.

The non-trivial equilibrium condition for the spherical coordinate system r-ϕ-θ is given by

$$0 = \frac{\partial(r^2\sigma_{rr})}{\partial r} - (\sigma_{\phi\phi} + \sigma_{\theta\theta}) \tag{1}$$

The weak form of Eq(1) reads

$$0 = \delta W = \int_{r_i}^{r_o} \delta\epsilon^T.\sigma r^2 dr - [r^2\sigma_{rr}\delta u_r]_{r=r_i}^{r_o} \tag{2}$$

The stress components σ_{rr},$\sigma_{\phi\phi}$ and $\sigma_{\theta\theta}$ are non-vanishing and the displacement in radial direction $u_r(r)$ is the only non-vanishing displacement component. Since the problem exhibits spherical symmetry, strains in each axis is connected to each other by the radial displacement u_r and the relation reads

$$\epsilon = \begin{bmatrix} \epsilon_{rr} = \frac{\partial u_r}{\partial r} \\ \epsilon_{\phi\phi} = \frac{u_r}{r} \\ \epsilon_{\theta\theta} = \frac{u_r}{r} \end{bmatrix} \tag{3}$$

The boundary conditions for this problem are σ_{rr}(r=r_o) $= 0$ and u_r(r=r_i)$=$ $\frac{1}{3}\tau\epsilon_v r_i$

Derivation

In Finite element method the governing equation used to solve for displacements in static condition is given by

$$G = F_{int} - F_{ext} = 0$$

From the given weak form(eq(2)), the required parameters to solve the problem in finite element method has been derived. The term that corresponds to the internal force is

$$\int_{r_i}^{r_o} \underline{\delta\epsilon}^T.\underline{\sigma} r^2 dr$$

From this term, the parameters internal force(F_{int}) and stiffness matrix can be derived. The internal force equation reads

$$F_{int} = \int_{r_i}^{r_o} \underline{B}^T.\underline{\sigma} r^2 dr$$

by applying gauss quadrature with single gauss point we get

$$F_{int} = 2.\underline{B}^T.\underline{\sigma}(\frac{r_1 + r_2}{2})^2.J$$

and the equation that computes stiffness matrix for each element reads

$$K_e = \int_{r_i}^{r_o} \underline{B}^T.\underline{C}.\underline{B}.r^2 dr$$

by applying gauss quadrature we get

$$K_e = 2.\underline{B}^T.\underline{C}.\underline{B}.(\frac{r_1 + r_2}{2})^2.J$$

where J is the Jacobian.
The strain-displacement relation (B matrix) can be computed from equation(3) using the relation $\epsilon = [B].\underline{u}^e$. The derivation of [B] matrix follows

$$\epsilon = \begin{bmatrix} \frac{\partial u_r}{\partial r} \\ \frac{u_r}{r} \\ \frac{u_r}{r} \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \frac{\partial}{\partial r} \\ \frac{1}{r} \\ \frac{1}{r} \end{bmatrix}.N.\underline{u}$$

where N is linear shape function matrix for 1D element and it is given by

$$[N](\xi) = \left[\tfrac{1}{2}(1-\xi), \quad \tfrac{1}{2}(1+\xi)\right]^T$$

From the above relations the [B]matrix can be obtained and it is given as

$$[B] = \begin{bmatrix} -\frac{1}{2}\frac{\partial \xi}{\partial r} & \frac{1}{2}\frac{\partial \xi}{\partial r} \\ \frac{1-\xi}{2(N_1 r_1 + N_2 r_2)} & \frac{1+\xi}{2(N_1 r_1 + N_2 r_2)} \\ \frac{1-\xi}{2(N_1 r_1 + N_2 r_2)} & \frac{1+\xi}{2(N_1 r_1 + N_2 r_2)} \end{bmatrix}$$

by applying gauss quadrature we get

$$[B] = \begin{bmatrix} -\frac{1}{2}\frac{\partial \xi}{\partial r} & \frac{1}{2}\frac{\partial \xi}{\partial r} \\ \frac{1-\xi}{(r_1 + r_2)} & \frac{1+\xi}{(r_1 + r_2)} \\ \frac{1-\xi}{(r_1 + r_2)} & \frac{1+\xi}{(r_1 + r_2)} \end{bmatrix}$$

where $\frac{\partial \xi}{\partial r}$ is inverse of Jacobian and r_1 and r_2 are position of each nodes.

Program Structure

With the required parameters derived, the finite element method can now be programmed. The program structure is shown in the following flowchart

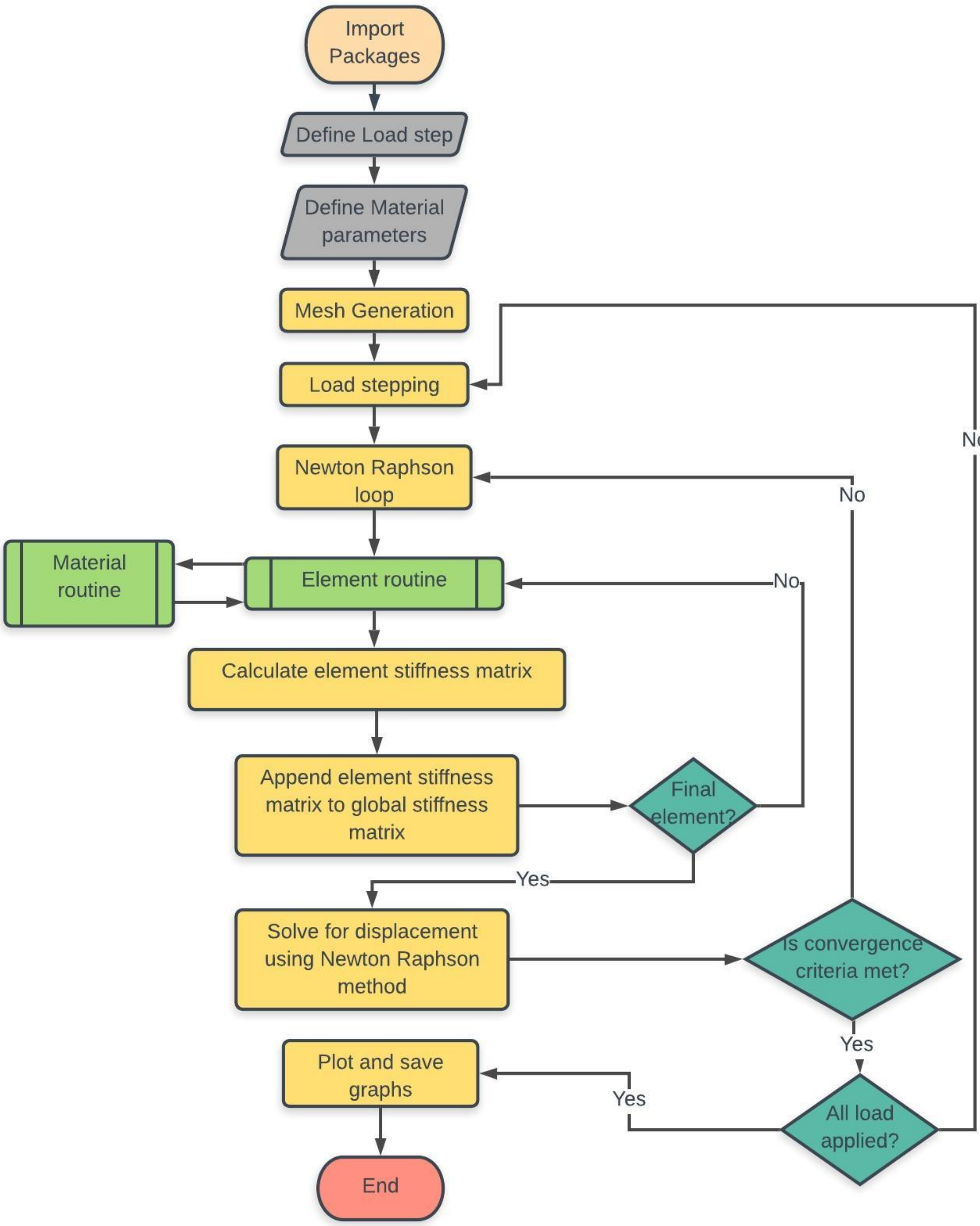

Figure 1: Structure of the program

The flow of the program is described below

- The necessary python packages are imported
- The loadstep is defined in order to analyze the behaviour of the model incrementally

- Material parameters and boundary conditions are defined
- Mesh is generated as per the given code snippet
- For a given loadstep the program enters the Newton-Raphson scheme
- Inside the Newton-Raphson scheme for each element the element routine and material routine are processed
- The element routine computes the stiffness matrix and internal force for each element. In order to compute these C matrix and Internal stress are needed, this is performed by the material routine
- The workflow of the material routine is given in the following flowchart

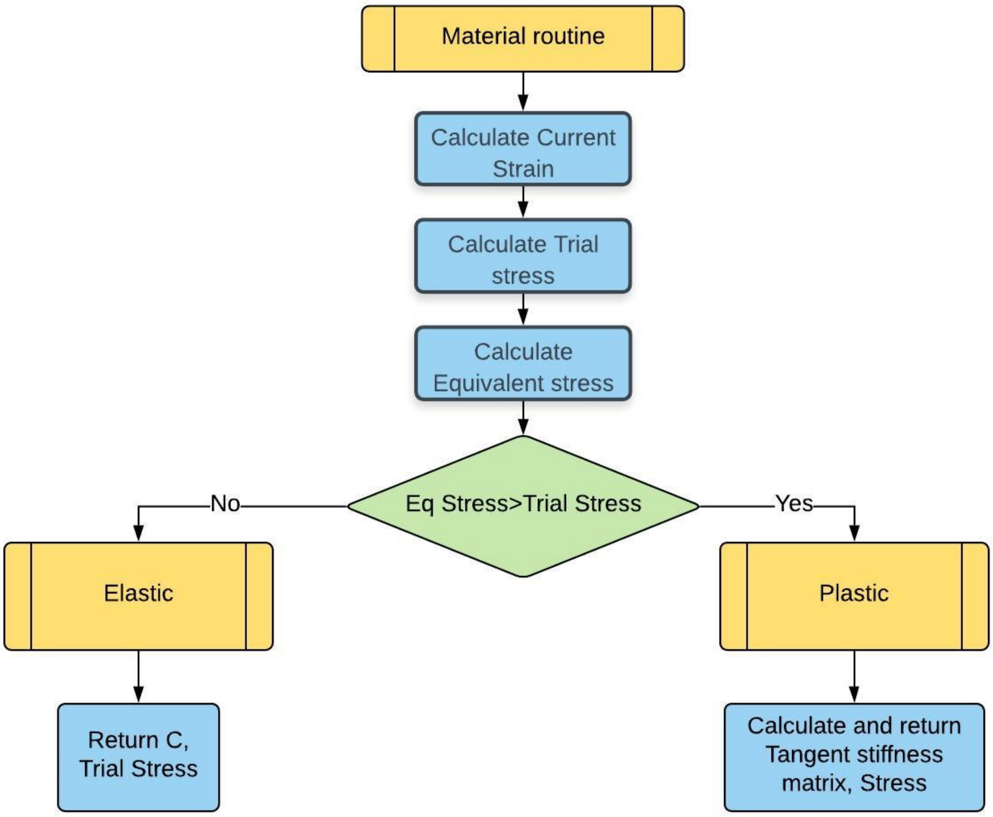

Figure 2: Material routine

- The material routine returns the algorthimically consistent material tangent stiffness matrix and the current stress
- The element routine now computes the stiffness matrix and internal force for each element
- From the elemental matrices the global matrices is assembeled using the assignment matrix

- The governing equation can now be solved using the Newton-Raphson scheme and the error can be calculated
- This error is checked against the convergence criteria if the criteria is met the program proceeds to the next load step
- The above process is repeated until the whole load is applied
- The final results are the plotted and saved in the working directory