



TUBAF
Die Ressourcenuniversität.
Seit 1765.

PVL Submission for the subject
NLFEM(Nonlinear finite element methods) for
the term summer 2023.



Author name, Matr.-number
Brijraj Jadeja, 67632

Supervisor
Dr.-Ing. Stefan Prüger

Report: Non Linear Finite Element Analysis of an elastic-plastic plate with a hole under uniform farfield tractions

Brijraj Jadeja 67632

July 2023

1 Introduction and Underlying theory

Our developed program makes use of the Finite Element method and the principles of mechanics to perform an analysis of a material under loading(far field tractions) conditions.

The material behaviour is defined using the isotropic hardening plasticity theory. As the material undergoes plastic deformation, isotropic hardening is considered in where the yield stress increases linearly with the accumulated plastic stress.

Next step is the Finite element discretization, The geometry is divided into a number of subdomains called finite elements which are read by the program via the mesh file. Each element is divided by the a set of nodes and contains a certain number of integration points. The displacement field within each element is approximated using shape functions. The shape functions and its derivatives are calculated at the integration points.

Further, based on the principle of virtual work we derive the equilibrium equation which relates the forces, nodal displacements and stiffness matrix. The stiffness matrices, residuals and forces are calculated in what we call a element routine. The non-linear equilibrium equations are then solved using the Newton Raphson method iteratively.

We then account for the boundary conditions. The constrained degree of freedoms are eliminated so that we get a reduced set of equations.

$$u_{red} = [K_{red}]^{-1} \cdot R_{red} \quad (1)$$

1.1 Element Routine

This is where our global stiffness matrix and Residual forces get assembled. The input for the element routine is data extracted from the mesh such as element and nodal connectivity, material properties etc. The Gauss points are defined and for the defined gauss points we calculate the shape functions(N) and their derivatives, The mass matrix(B), The Jacobian matrix(J), strains and stresses are calculated. All of these are required to calculate the element stiffness matrix and the internal and external forces. Stresses are calculated by calling the material routine. The formulae involved are

$$K_{element} = B' \cdot C_t \cdot B \cdot det(J) \cdot w \quad (2)$$

$$Fe_{int} = B' \cdot \sigma \cdot det(J) \cdot w \quad (3)$$

$$Fe_{ext} = N' \cdot \sigma_{\infty} \cdot det(J) \cdot w \quad (4)$$

Code Flow

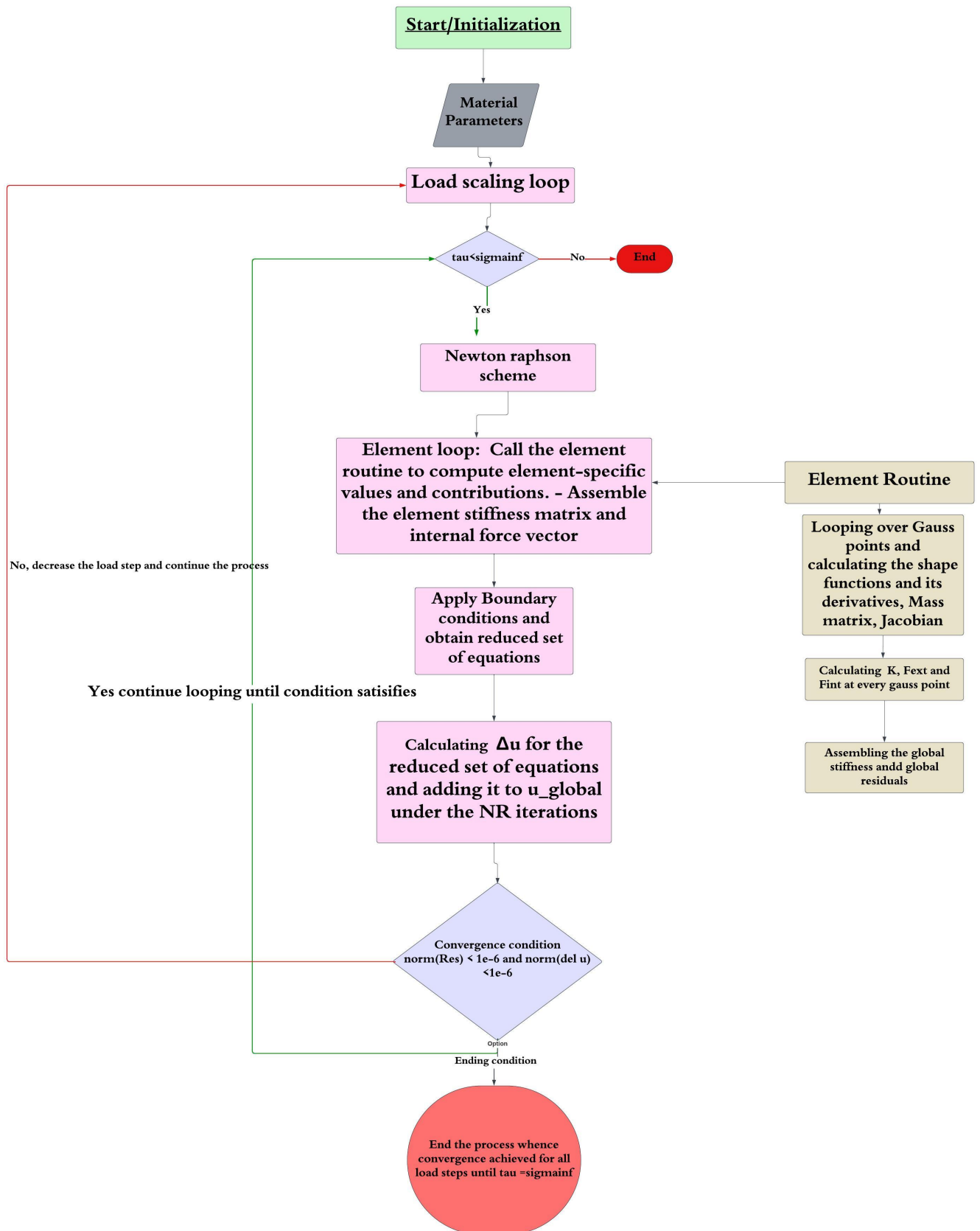


Figure 1: Workflow of coding implementation

Implementation Details

The purpose of the code is to implement a FEM(Finite Element Method) algorithm for solving a load dependent problem. The algorithm consists of a main program , element routine and the material routine.

Structure

The main program serves as the central driver for the finite element method (FEM) implementation. It initializes parameters such as loads and displacements and performs a load scaling loop to incrementally apply the loads and update the initial displacement. It also initializes the global stiffness and residual matrices.

For each element in the structure, the element routine is called from the main program. The element routine computes element-level displacements and loads. It calculates the element stiffness, external forces, and internal forces by integrating them over Gauss points for accuracy. The strains, stresses, and material tangent matrices are computed by invoking the material routine at each Gauss point. The material routine defines the constitutive behavior of the simulated material.

The element routine plays a crucial role in evaluating the behavior within each finite element and capturing its response to applied loads and displacements.

After computing the element-level quantities, the next step is assembly. This process involves using assembly matrices to combine the element contributions and obtain the global stiffness and residual matrices. Eqn (5) and (6) describe the assembly procedure.

The main program then proceeds to apply boundary conditions. This involves removing the rows and columns in the global matrices that are not relevant to the analysis. This results in a reduced system of equations, which is then solved to determine the displacement increments.

The Newton-Raphson loop is performed for each load in the load scaling loop, with a certain number of iterations specified. The loop iteratively refines the solution by updating the displacement increments until convergence is achieved. Convergence conditions are imposed to ensure the accuracy of the solution.

$$K_{global} = Ae' \cdot K_{elem} \cdot Ae \quad (5)$$

$$Res_{global} = Ae' \cdot (Fe_{int} - Fe_{ext}) \quad (6)$$

A short user Manual on How to Run the Program

As mentioned before the assignment dealt with a planar stress problem with a elastic-plastic plate with a hole under farfield tractions. The assignment was already provided with the meshes on which the finite element study was to be performed.

How to start the program

To start the program user needs to execute the code in a Matlab environment. It is to be made sure that all the files(meshes, main program, function files, element routine, material routine etc are in the same folder for successful implementation.

Where does the program get its input from

The input for the for the program is a mesh file, such as the mesh file `Mesh_Tri_Coarse.mat`, which is in the same directory as the program file. The program reads the input mesh information from the mesh file which includes node coordinates and element connectivity. It also sets various material properties and parameters such as the Youngs Modulus, Poisson's ratio, yield stress, hardening parameters, etc. All these properties are defined in the beginning of the code.

What ouput does the program generate and where does it store to

The program performs a nonlinear analysis by iteratively solving the equilibrium equations using the Newton Raphson scheme. It gradually increases the applied load (τ) until it reaches a certain prescribed value(0.335 in our case).

Inside the Newton raphson loop the program assembles the global stiffness and residual matrices by looping over the elements. For each element it calls the element routine function, which performs element level computations and assembles the element contribution to the global matrices.

The program also includes post processing operations. It stores stress and strain values for visualisation purposes and plots stress profiles with respect to the radial distance for three different loading cases (0 degrees, 45 degrees, and 90 degrees). The plots are shown in the verification section(up next) of the report.

The program does not explicitly store any output to external files. But what it does is it generates visual output in terms of plots directly in the MATLAB environment.

Verification:Results

In this section, we perform a convergence study to assess the accuracy of the numerical results. We do this by computing the stresses at the quadrature points located in the vicinity of three rays, defined by $\theta = 0, 45, 90$. The stresses are then transformed according to the equations provided in the assignment and plotted as a function of the radial coordinate r for different mesh refinements.

The convergence study helps assess the accuracy and reliability of the solution, by comparing the stresses obtained at quadrature points located near the three defined rays for different mesh refinement we can observe the behaviour of the solution as the mesh becomes finer

Following are the images for the various cases.

Uniform Mesh

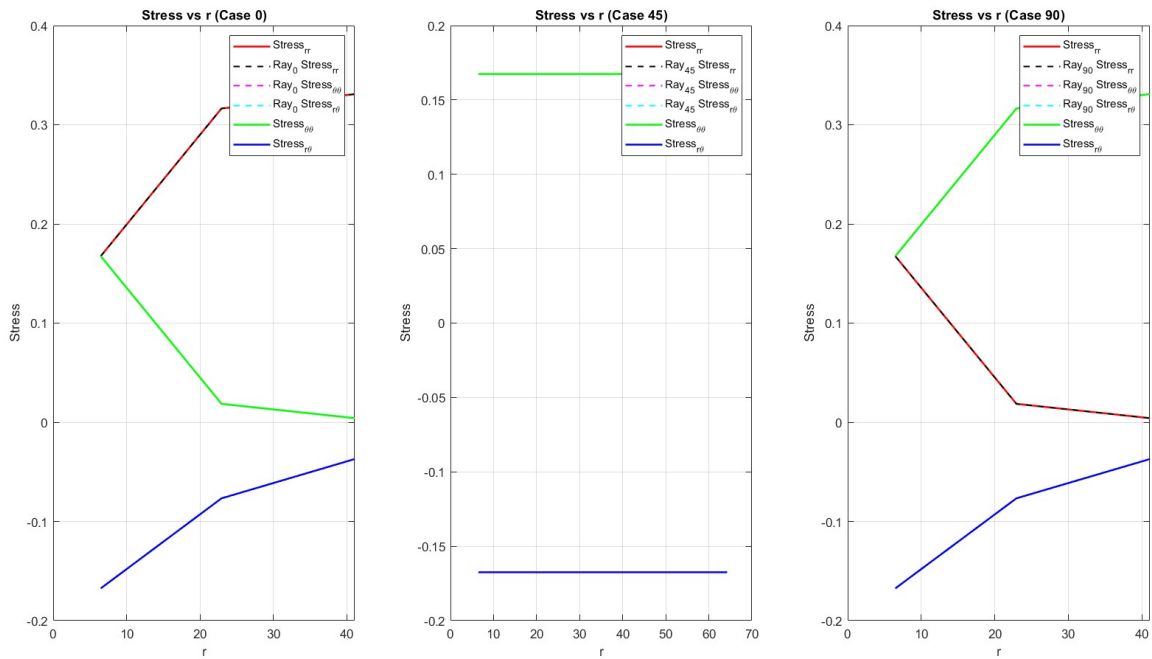


Figure 2: Stress plot with uniform mesh

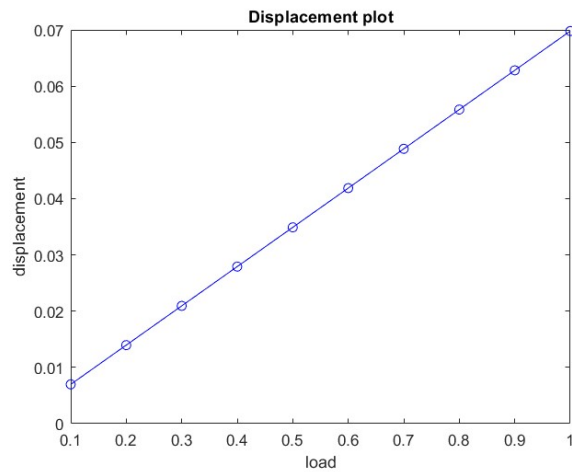


Figure 3: displacement plot

Coarse Mesh

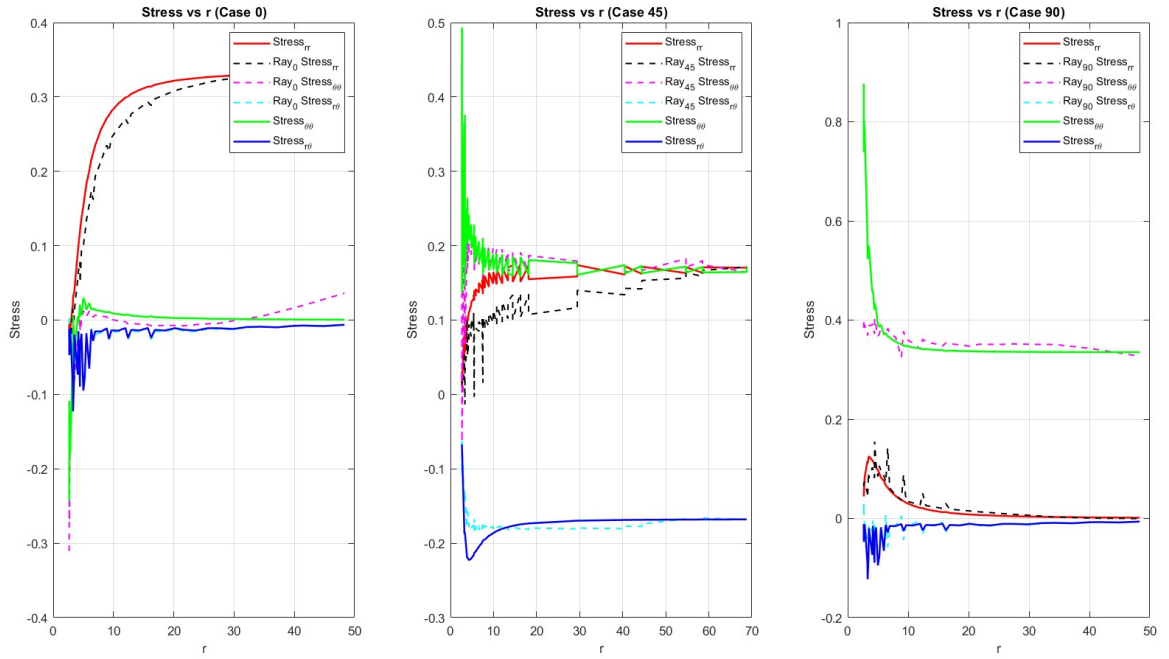


Figure 4: Stress plot with uniform mesh

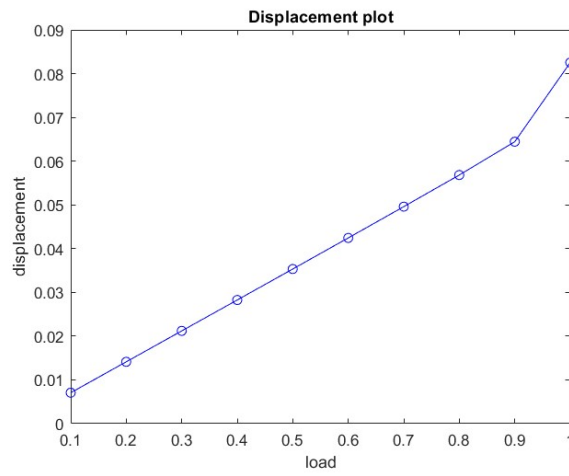


Figure 5: displacement plot

Fine Mesh

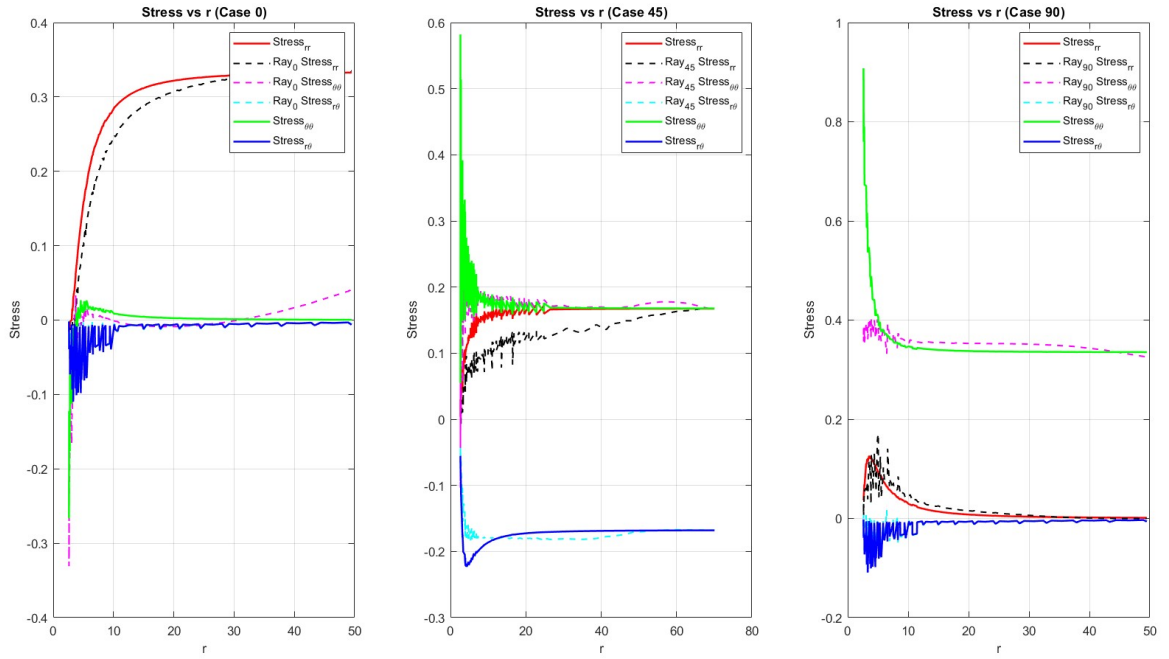


Figure 6: Stress plot with uniform mesh

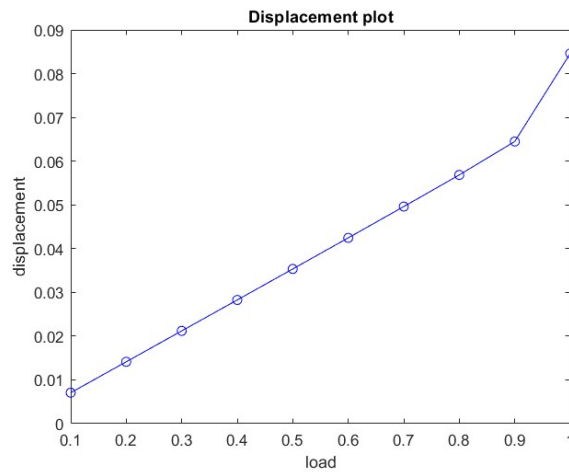


Figure 7: displacement plot

Very Fine Mesh

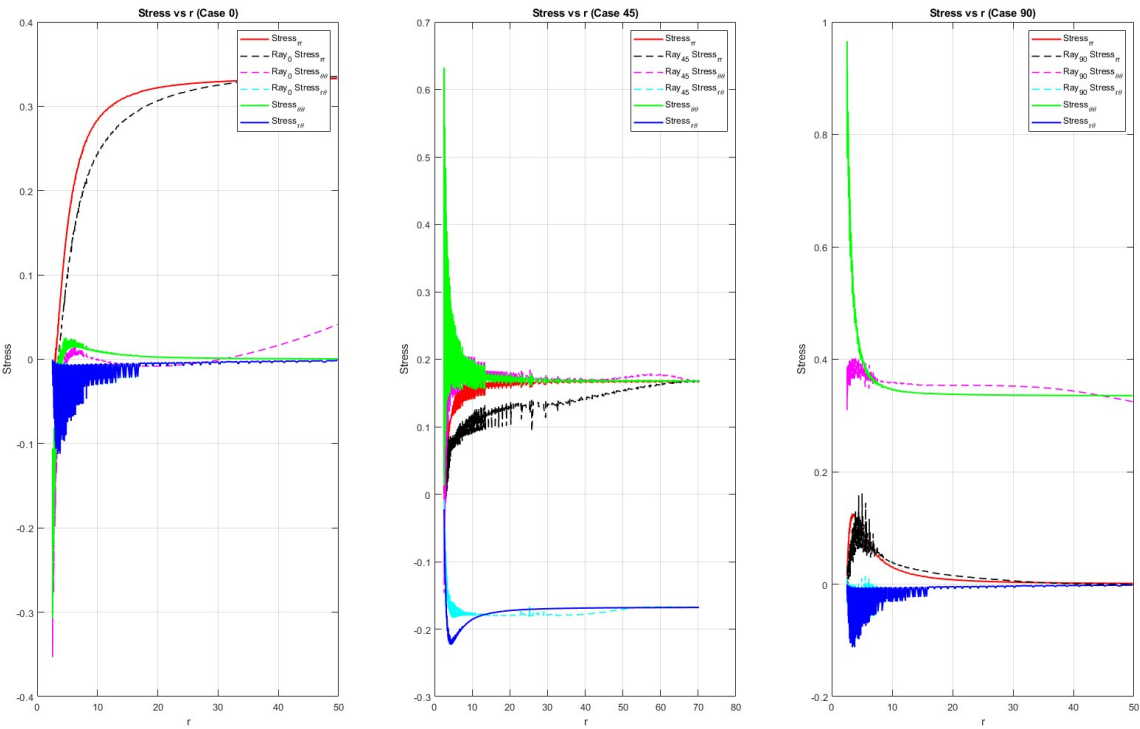


Figure 8: Stress plot with uniform mesh

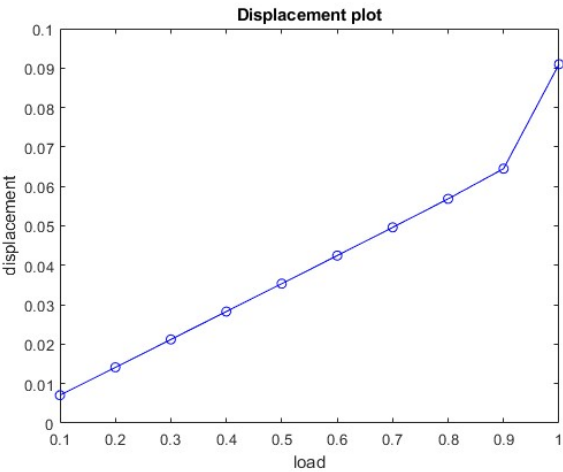


Figure 9: displacement plot