TaperedBeam Static Analysis

Amir Baharvand

1 Problem Statement

A cantilever tapered beam is shown in Figure. 1a. As is seen, the beam undergoes a uniaxial compressive pressure at its right-end and its radius decreases linearly from 20mm to 10mm along the x-axis. The material behavior is elastoplastic and is given in Eq. 1 and illustrated in Figure. 2.

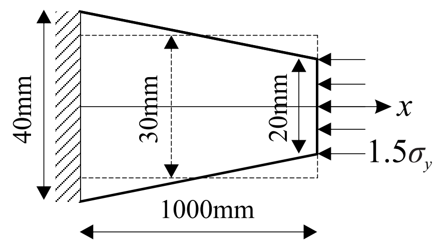
$$\sigma(\epsilon) = \begin{cases} E\epsilon & \sigma(\epsilon) \leq \sigma_y \\ \sigma_y \left(\dfrac{E\epsilon}{\sigma_y}\right)^{0.4} & \sigma(\epsilon) > \sigma_y \end{cases} \tag{1}$$

where σ and ϵ denote stress and strain, respectively. E is Young's modulus and σ_y is the yielding stress. The corresponding values are E=210GPa and σ=200MPa. The applied load magnitude is $1.5\sigma_y$

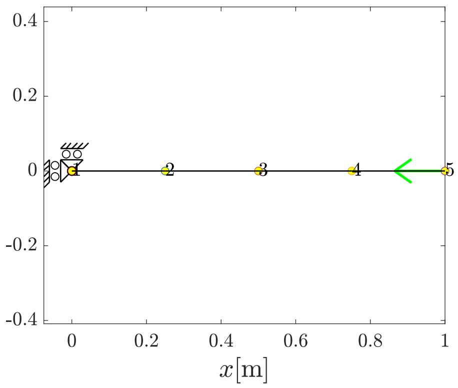
Figure. 1b represents the numerical model of the tapered beam. The numerical model is built upon the following hypothesis.

1. The beam undergoes uniaxial loading; hence, the beam element is reduced to a bar element.
2. The tapered beam is reduced to a beam with a constant cross-section over the x-axis. A schematic representation is given with dashed lines in Figure. 1b.
3. Boundary conditions are further simplified by removing the displacement in the y-direction.
4. The applied stress is replaced by its equivalent point load.
5. The effect of body force and traction are neglected and the numerical model only supports point loads.

As a result, the tapered beam reduces to a beam with a constant cross-section that can be solved numerically using a bar element.



(a) The cantilever tapered beam.



(b) The Numerical model with constant radius of 30 mm.

Figure 1: The geometry, coordinate system and boundary condition of the cantilever tapered beam under uniaxial loading.

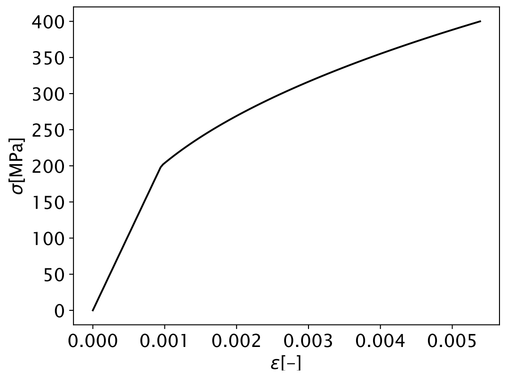


Figure 2: The stress-strain curve from Eq. 1.

2 Finite Element Method

The numerical model utilized in the present problem is Finite Element Method(FEM) and is developed base on the based on the virtual work principle. Finally, the results from the self-developed code, `mnb1c.py`[1] are compared with the commercial FEM code, Abaqus at the end of this report.

[1]Material Nonlinearity Beam element one-dimensional Code

2.1 `mnb1c.py` Solver

To satisfy the equilibrium `mnb1c.py` is combined with an iterative algorithm known as Implicit method. Here, the Newton-Raphson (NR) the Modified Newton-Raphson (MNR) are invoked. Such methods are invoked when one needs to derive the displacement from the applied load. Such methods do not depend on the increment size; however, implicit methods are computationally expensive due to the stiffness matrix factorization which is enhanced in the MNR.

2.1.1 Newton-Raphson Method

The pseudocode of the NR method is provided in algorithm 1 and depicted in Figure. 3 for two load increments. The stiffness matrix factorization is performed in each iteration within an individual increment. To solve for displacement, different methods, for instance the LU factorization (as used in the present report) can be utilized.

Algorithm 1 Newton-Raphson pseudocode.

```
Ensure: $e = 1E-11$, $\boldsymbol{R} = 0$, $\boldsymbol{dD} = 0$, $\boldsymbol{D} = 0$
  for Load increment ($n$) do
    for Iteration do
      $\boldsymbol{K} = 0$
      $\boldsymbol{R_{int}} = 0$
      Compute $\boldsymbol{R_{int}}$
      Compute $R = \boldsymbol{R_{int}} - \boldsymbol{n\Delta P}$
      Apply B.C on $\boldsymbol{R}$
      if norm($\boldsymbol{R}$) $\leq \epsilon$ norm($P$) then
        break
      end if
      Compute $\boldsymbol{K}$
      Apply B.C on $\boldsymbol{K}$
      Solve $\boldsymbol{dD} = \boldsymbol{K}^{-1}\boldsymbol{R}$
      Factorize $\boldsymbol{K}$
      $\boldsymbol{D} = D + dD$
    end for
    Write $\boldsymbol{D}$
    Write $n\Delta P$
  end for
```

2.1.2 Modified Newton-Raphson Method

In this method, the expensive process of stiffness matrix factorization is boosted by removing this step from each iteration. The MNR method performs the stiffness matrix factorization outside the iteration loop and use it later for solving the displacement. As a result, MNR requires more iteration in comparison to the NR. The MNR pseudocode is given in algorithm 2 and illustrated in Figure. 4 for two load increments.

Algorithm 2 Modified Newton-Raphson pseudocode.

```
Ensure: $e = 1E-11$, $\boldsymbol{R} = 0$, $\boldsymbol{dD} = 0$, $\boldsymbol{D} = 0$
  for Load increment $(n)$ do
    Compute $\boldsymbol{K_{inc}}$
    Factorize $\boldsymbol{K_{inc}}$
    Apply B.C on $\boldsymbol{K_{inc}}$
    for Iteration do
      $\boldsymbol{K} = 0$
      $\boldsymbol{R_{int}} = 0$
      Compute $\boldsymbol{R_{int}}$
      Compute $R = \boldsymbol{R_{int} - n\Delta P}$
      Apply B.C on $\boldsymbol{R}$
      if norm($\boldsymbol{R}$) $\leq \epsilon$ norm($P$) then
        break
      end if
      Solve $\boldsymbol{dD} = \boldsymbol{K_{inc}}^{-1}\boldsymbol{R}$
      $\boldsymbol{D} = D + dD$
    end for
    Write $\boldsymbol{D}$
    Write $n\Delta P$
  end for
```

2.1.3 Newton-Raphson vs. Modified Newton-Raphson

Table. 1 compares the number of iterations required for each load increment from the three NR, MNR and Abaqus NR for the beam problem in Figure. 1b. As expected, the MNR needs more iteration for each load increment compared to the NR method. The maximum number of iteration for the MNR occurs at the load increment 7 where the problem migrates from the elastic regime to the plastic and thus the codes requires more iterations to fulfill the equilibrium.

Table 1: Non-tapered beam: Comparison between the number of iterations required for a 10-load increment (Fixed load increment activated in Abaqus).

Increment Number	Number of iterations		
	Newton-Raphson	Modified Newton-Raphson	Abaqus (Newton-Raphson)
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	4	52	2
8	4	15	1
9	4	14	2
10	4	13	1

Table. 2 compares the elapsed time for both the NR and MNR methods. The advantage of stiffness matrix factorization outside the iteration loop makes the MNR a suitable method for numerical problems with higher number of load increments.

Table 2: Non-tapered beam: Newton-Raphson vs. Modified Newton-Raphson benchmark

Number of increments	Elapsed time[s]	
	Newton-Raphson	Modified Newton-Raphson
10	0.024	0.050
100	0.161	0.175
1000	1.33	1.32
10000	12.7	11.9
100000	126	105

2.2 Abaqus

In this section, necessary steps for the numerical modeling in Abaqus are mentioned.

2.2.1 Mechanical Properties

The numerical model unit is set to SI unit in meters. Table. 3 summarizes the mechanical properties of the beam.

Table 3: Mechanical properties of the beam.

Property	Symbol	Unit	Value
Young's modulus	E	Pa	2.1E11
Poisson's ratio	ν	-	0.3
Yielding stress	σ_y	Pa	2E08

The plastic behavior is imported as a table in Abaqus. A Python script is developed which can generate a table of plastic properties which can be imported into Abaqus. Table. 4 summarizes the input file for the plastic behavior in Abaqus.

Table 4: Plastic behavior of the beam defined in Abaqus.

Yield stress[Pa]	Plastic strain[-]
2E08	0
2.5E08	0.0007114
3E08	0.0016721
3.5E08	0.0029060
4E08	0.0044351

2.2.2 Solver

The solver for this problem is set to **Static, General** because the problem is linear static (no contact, boundary conditions do not change with time). As mentioned, the solver invokes the Newton-Raphson method.

2.2.3 Element Type

The B21 element (a 2-node linear beam with one integration point in middle in two-dimension) and B31 element (a 2-node linear beam with one integration point in middle in three-dimension) with shear-flexible (based on Timoshenko beam theory) formulation are selected for the Abaqus numerical modeling in the two and three-dimension, respectively. The B21, B31 and the bar element from `mnb1c.py` are illustrated for convenience in Figure. 5

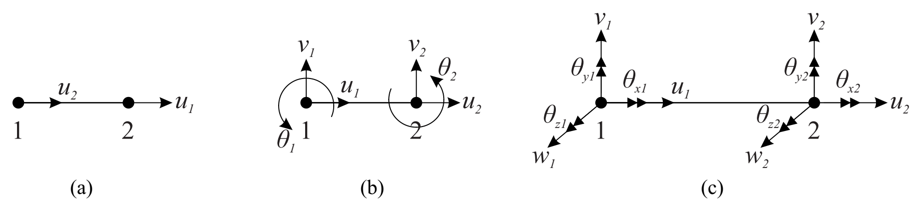


Figure 5: (a) The bar element used in `mnb1c.py`. (b) The B21 element. (c) The B31 element. u, v and w are the nodal displacement in the x, y and z-direction and θ is the rotation. Subscripts 1 and 2 denote the node number.

2.3 Result and Discussion

2.3.1 Non-tapered Beam

Figure. 6 shows the load-displacement plot for the right end of the beam (node 5 in Figure. 1b). For the FEM analysis using Abaqus, two numerical models are developed. One uses 5 points for the definition of plastic behavior (see Table. 4) while another uses 50 points. The results from both the NR and MNR coincide with the analytical model as both the NR and MNR utilizes Eq. 1 for the relation between the stress and strain and Eq. 2 tangential stiffness in the plastic region.

$$\frac{\mathrm{d}\sigma(\epsilon)}{\mathrm{d}\epsilon} = \begin{cases} E & \sigma \leq \sigma_y \\ \dfrac{0.4E}{\left(\dfrac{E\epsilon}{\sigma_y}\right)^{0.6}} & \sigma > \sigma_y \end{cases} \tag{2}$$

Abaqus follows the analytical solution in the elastic regime; nevertheless, it overestimates the displacement for a certain load increment. Increasing the number of plastic behavior point to 50 enables Abaqus to perform a better interpolation between the given points in the transition zone (from elastic to plastic). Abaqus is incapable of solving the problem with 10 load increments and crashes for fixed load increment. Thus, the increment size is set to "Automatic". It appears that higher number of points for the plastic regime only improves the result in the elasto-plastic transition zone and does not affect the solution.

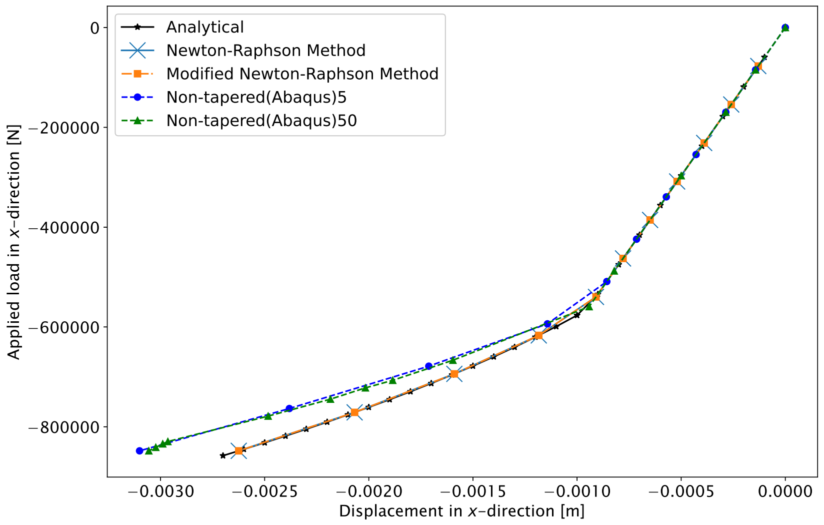


Figure 6: Non-tapered beam: Load vs. displacement (along x-axis) plot of right-end of the beam (node 5 in Figure. 1b).

2.3.2 Tapered Beam

The load-displacement plot for the tapered beam (see Figure. 7) is shown in Figure. 8. `mnb1c.py` requires more load increment than the non-tapered version and the NR method crashes for load increments less than 10. Table. 5 list the iterations for each load increments in `mnb1c.py`.

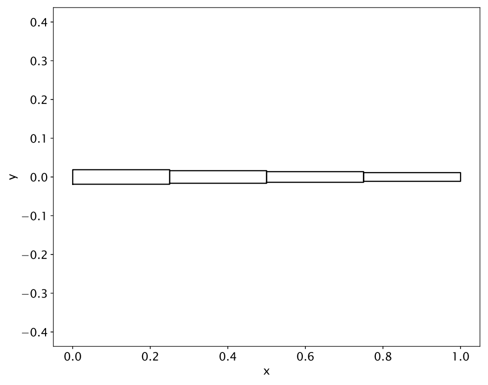


Figure 7: The simplified version of the tapered beam from Figure. 1a (Axes units depends on the initial parameter definition. Here both axes units are meters.)

Table 5: Tapered beam: Comparison between the number of iterations required for an 11-load increment in `mnb1c.py`.

Increment Number	Number of iterations	
	Newton-Raphson	Modified Newton-Raphson
1	1	1
2	1	1
3	8	224
4	10	110
5	10	42
6	13	72
7	10	35
8	9	93
9	4	14
10	4	13
11	4	13

Unlike Abaqus, `mnb1c.py` is able to solve the problem (see Figure. 8) to the last load increment which may be attributed to the analytical equations for and the stress-strain (Eq. 1) tangential stiffness matrix (Eq. 2). Abaqus solver crashes right before the transition zone and increasing the number of plastic behavior to 50 results in crashing the solution procedure.

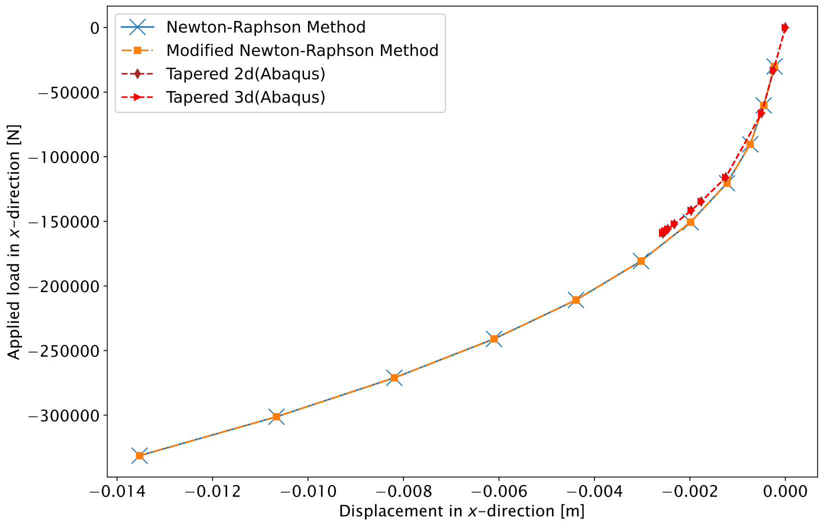
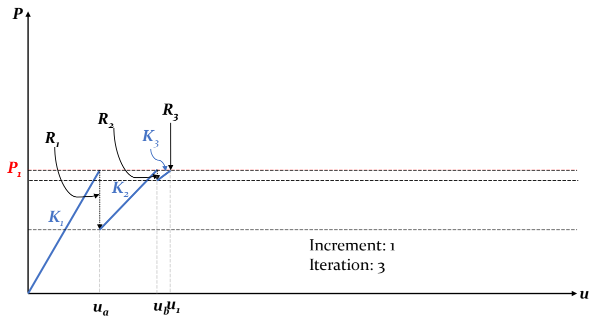
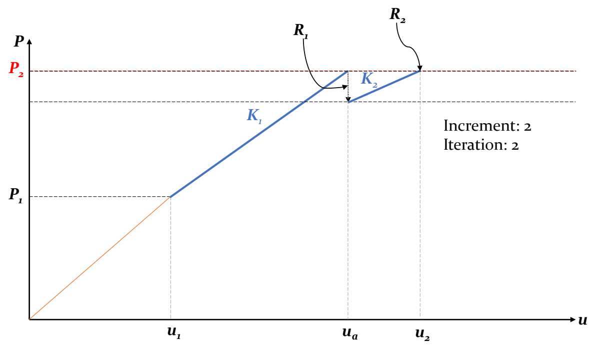


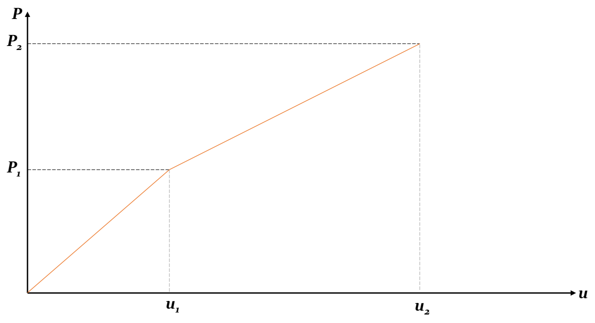
Figure 8: Tapered beam: Load vs. displacement (along x-axis) plot of right-end of the beam (node 5 in Figure. 1b). Abaqus results are provided up to the last solve increment.



(a) The first load increment with three iterations

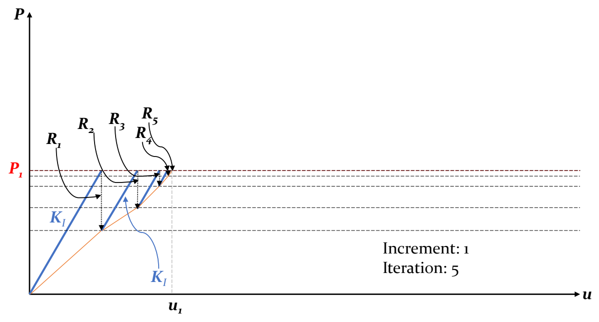


(b) The second load increment with two iterations

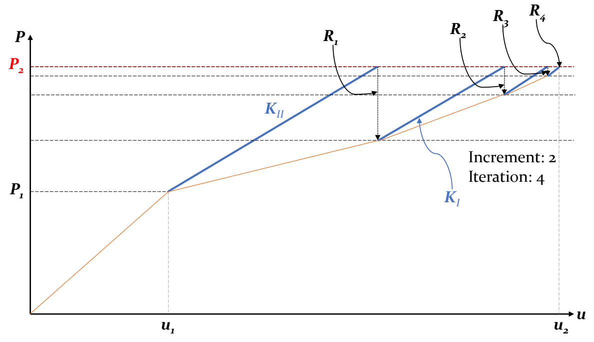


(c) The first two load increments with their corresponding displacements.

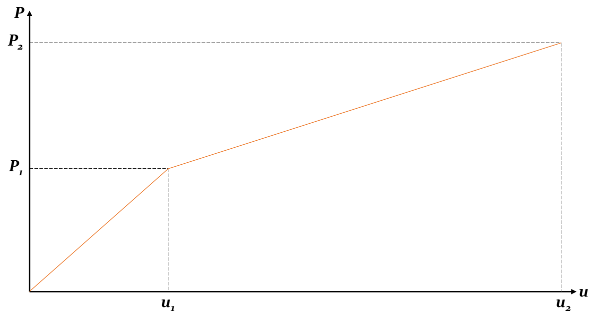
Figure 3: An schematic representation of the Newton-Raphson method for two load increments. K is the stiffness matrix, R is the residual vector from equilibrium, P is the load and u is the displacement. The subscripts a and b denote the trial displacement while the numbered subscripts denote the final displacement for the corresponding load increment.



(a) The first load increment with five iterations



(b) The second load increment with four iterations



(c) The first two load increments with their corresponding displacements.

Figure 4: An schematic representation of the Modified Newton-Raphson method for two load increments. K is the stiffness matrix, R is the residual vector from equilibrium, P is the load and u is the displacement. The Roman subscripts in K denote the initial tangential stiffness matrix for each load increment. The numbered subscripts denote the final displacement for the corresponding load increment.