UNIVERSITY OF CALIFORNIA AT BERKELEY
Department of Civil and Environmental Engineering
Structural Engineering, Mechanics and Materials

CE 130N, Spring 2009
Prof. S. Govindjee and Dr. T. Koyama
Lab 15

Mechanics of Structures (CE130N)
Lab 15

1 Objective

The objective of this lab is to combine the use of hat functions with Principle of Virtual Work (Lab 13) and the Newton-Raphson method (Lab 14) to solve mechanical phenomenon governed by a nonlinear differential equation. The specific problem that is treated is the torsion bar. The same set up as Lab 13 is assumed but the material model is replaced with a non-linear model.

Once the form of the ”residual” and ”tangent stiffness matrix” is defined, the rest is just an application of Newton-Raphson to solve the non-linear equations.

2 Steps in applying the Principle of Virtual Work: Non-linear case

The Principle of Virtual Work can be summarized by the following steps.

1. Form the expression for the virtual work.

$$\text{Internal Virtual Work} \quad = \quad \text{External Virtual Work.}$$

   Depending on the problem one has,

$$\text{Internal Virtual Work} \quad = \quad \begin{cases} \int_0^L \delta u'(x) R(x)\, dx & \text{(Tension-compression bar)} \\ \int_0^L \delta\varphi'(z) T(z)\, dz & \text{(Torsion bar)} \end{cases} ,$$

   and depending on the loading one has different forms for the External Virtual Work. For example, the torsion bar with a point torque $\bar{T}$ at the end $x = L$ and distributed torque $t(z)$ takes the form,

$$\text{External Virtual Work} \quad = \quad \int_0^L \delta\varphi(z) t(z)\, dz + \delta\varphi(L)\bar{T} \; .$$

2. Determine the functions $u(x)$ in the solution space $\mathcal{S}$ (form for the approximate solution),

$$\mathcal{S} := \left\{ u(x) | \sum_{i=1}^{N} u_i f_i(x) \right\} \; ,$$

   where u_i are the coefficients to be determined and $f_i(x)$ are determined functions, and the functions $\delta u(x)$ in the test function space $\mathcal{V}$,

$$\mathcal{V} := \left\{ \delta u(x) | \sum_{j=1}^{N_\delta} \delta u_j g_j(x) \right\} \; ,$$

   where δu_j are the arbitrary coefficients and $g_j(x)$ are determined functions.

3. Insert the from for the solution $u(x) \in \mathcal{S}$ and the test function $\delta u(x) \in \mathcal{V}$ into the expression for the virtual work. In Lab 13, we assumed a linear constitutive relationship of the form,

$$\begin{aligned} R(u'(x)) = & \quad AEu'(x), \quad \text{(Tension compression bar)}, \\ T(\varphi'(z)) = & \quad GJ\varphi'(z), \quad \text{(Torsion bar)} \; . \end{aligned}$$

   For this lab we consider a more general form of constitutive relation,

$$\begin{aligned} R = R(u'(x)), & \quad \text{(Tension compression bar)}, \\ T = T(\varphi'(z)), & \qquad\qquad \text{(Torsion bar)} \; . \end{aligned}$$

   This results in the expressions,

$$\begin{aligned} & \sum_{i=1}^{N_\delta} \delta u_i \left\{ F_{I,i}(\mathbf{u}) - F_{E,i} \right\} = 0 \\ \Leftrightarrow \quad & \sum_{i=1}^{N} \delta u_i h_i = 0 \\ \Leftrightarrow \quad & \delta\mathbf{u}^T \mathbf{h} = \mathbf{0} \; . \end{aligned}$$

Here we have defined $\mathbf{h} := \mathbf{F}_I - \mathbf{F}_E$ and $\delta\mathbf{u}$ as the vector with entries δu_i. The entries of the vector $\mathbf{F}_I$ are determined from the Internal Virtual Work and have the form,

$$\boxed{F_{I,i}(\mathbf{u}) = \begin{cases} \displaystyle\int_0^L g_i' R\left(\sum_{j=1}^N u_j f_j'(x)\right) dx & (\mathrm{Tension - compression\ bar}) \\ \displaystyle\int_0^L g_i' T\left(\sum_{j=1}^N \varphi_j f_j'(x)\right) dx & (\mathrm{Torsion\ bar}) \end{cases} .}$$

and the entries of the vector $\mathbf{F}_E$ are determined from the External Virtual Work and have the form,

$$\boxed{F_{E,i} = \begin{cases} \displaystyle\int_0^L g_i b dx + \text{term from point loads} & \text{(Tension-compression bar)} \\ \displaystyle\int_0^L g_i t dz + \text{term from point loads} & \text{(Torsion bar)} \end{cases} .}$$

For the last equation, $\delta\mathbf{u}^T\mathbf{h} = 0$, to hold for arbitrary $\delta\mathbf{u}$, one must have,

$$\mathbf{h} = \mathbf{F}_I(\mathbf{u}) - \mathbf{F}_E = 0 \, ,$$

which determines the *non-linear* system of equations one must solve for $\mathbf{u}$, which are the undetermined coefficients.

4. Solve the *non-linear* system of equations,

$$\mathbf{F}_I(\mathbf{u}) - \mathbf{F}_E = \mathbf{0} \, .$$

to obtain the approximation,

$$u(x) \quad = \quad \sum_{i=1}^N u_i f_i(x) \, .$$

One must employ the Newton-Raphson method in N variables to solve these system of equations for the N unknowns.

It is important to note that when one has a linear constitutive material law, $\mathbf{F}_I(\mathbf{u}) = \mathbf{K}\mathbf{u}$ where $\mathbf{K}$ is the stiffness matrix you have computed in the linear case which does not depend on $\mathbf{u}$.

3 Application of the Newton-Raphson method in N-variables

Let $f_i(u_1, \ldots, u_N)$ $(i = 1, \ldots, N)$ be N non-linear functions. Each function is a function in N variables, $u_1, \ldots, u_N$. One can use the compact notation,

$$\mathbf{f}(\mathbf{u}),$$

to denote this, where we have defined,

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix}$$
$$\mathbf{f} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \, .$$

For the torsion problem this function is defined as,

$$\boxed{\mathbf{f}(\mathbf{u}) := \mathbf{F}_I(\mathbf{u}) - \mathbf{F}_E \; .}$$

We would like to find the solution $\mathbf{u}$ to the equation,

$$\mathbf{f}(\mathbf{u}) = \mathbf{0} \; .$$

We can approach this in the same way as the 1-variable case, such that we would like to iteratively find the exact solution. Given an initial guess of $\mathbf{u}^0$, one would like to fix this guess by adding $\Delta\mathbf{u}$ so that $\mathbf{u}^0 + \Delta\mathbf{u}$ is the exact solution,

$$\mathbf{f}(\mathbf{u}^0 + \Delta\mathbf{u}) = \mathbf{0} \; ,$$

or equivalently,

$$f_i(u_1^0 + \Delta u_1, \ldots, u_N^0 + \Delta u_N) = 0 \quad (i = 1, \ldots, N) \; .$$

Conducting a Taylor series expansion around $\mathbf{u} = \mathbf{u}_0$ and retaining only the linear terms one has,

$$\mathbf{0} = \mathbf{f}(\mathbf{u}^0 + \Delta\mathbf{u}) \approx \mathbf{f}(\mathbf{u}^0) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}^0) \cdot \Delta\mathbf{u} \; ,$$

such that one solves,

$$\mathbf{0} = \mathbf{f}(\mathbf{u}^0) + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}^0) \cdot \Delta\mathbf{u} \; ,$$

to find the update $\Delta\mathbf{u}$ by,

$$\Delta\mathbf{u} \quad = \quad -\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}^0)^{-1}\mathbf{f}(\mathbf{u}^0) \; .$$

For the torsion problem the tangent stiffness is defined as,

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}) = \mathbf{K}(\mathbf{u}) \quad &:= \quad \frac{\partial}{\partial \mathbf{u}}(\mathbf{F}_I(\mathbf{u}) - \mathbf{F}_E) \\ &= \quad \frac{\partial \mathbf{F}_I(\mathbf{u})}{\partial \mathbf{u}} \; . \end{aligned}$$

The (i,j)th entry of this tangent stiffness matrix evaluated at the point $\mathbf{u}^0$ is given as,

$$\boxed{K_{ij}(\mathbf{u}^0) = \begin{cases} \displaystyle\int_0^L g_i' \frac{dR}{ds} \left(\sum_{j=1}^N u_j^0 f_j'(x) \right) f_j' dx & (\text{Tension} - \text{compression bar}) \\ \displaystyle\int_0^L g_i' \frac{dT}{ds} \left(\sum_{j=1}^N u_j^0 f_j'(x) \right) f_j' dx & (\text{Torsion bar}) \end{cases} \; .}$$

Note that the derivative of R and T above is respect to its argument s and the expression dR/ds is evaluated with the argument $\sum_{j=1}^N u_j^0 f_j'(x)$.

This yields a new approximation $\mathbf{u}^1$ for the solution,

$$\begin{aligned} \mathbf{u}^1 \quad &= \quad \mathbf{u}^0 + \Delta\mathbf{u} \\ &= \quad \mathbf{u}^0 - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u}^0)^{-1}\mathbf{f}(\mathbf{u}^0) \; . \end{aligned}$$

One can proceed to find better approximations by repeating the procedure above, replacing $\mathbf{u}^0$ with $\mathbf{u}^1$ and so on, and continue until one has found a solution which is good enough, i.e., $\|\mathbf{f}(\mathbf{u}^n)\| < \mathrm{tolerance}$. The steps can be summarized as follows,

```
1. Guess initial solution $\hat{\mathbf{u}} = \mathbf{u}^0$, and compute residual $\mathbf{f}(\hat{\mathbf{u}}) = \mathbf{F}_I(\hat{\mathbf{u}}) - \mathbf{F}_E$.
2. While $\|\mathbf{f}(\hat{\mathbf{u}})\| > \mathrm{abstol}$,
    (a) Compute tangent stiffness: $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\hat{\mathbf{u}}) = \mathbf{K}(\hat{\mathbf{u}})$.
    (b) Compute update: $\Delta\mathbf{u} = -\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\hat{\mathbf{u}})^{-1}\mathbf{f}(\hat{\mathbf{u}})$.
    (c) Update solution: $\hat{\mathbf{u}} \leftarrow \hat{\mathbf{u}} + \Delta\mathbf{u}$.
    (d) Compute new residual: $\mathbf{f}(\hat{\mathbf{u}}) = \mathbf{F}_I(\hat{\mathbf{u}}) - \mathbf{F}_E$.
```

Thus in order to solve the non-linear torsion problem all that one must do is to supply the residual function $\mathbf{f}(\mathbf{u})$ and the tangent stiffness $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$.

4 Exercise

4.1 Download files

1. Download the file `pvw_nonlinear.zip` into your `ce130n/programs` directory and unzip it.
2. Go to the `ce130n/programs/pvw_nonlinear/exercise/` directory, and execute the file `init.m`. This will set the necessary paths to run the files.

YOU MUST RUN THE FILE init.m EVERYTIME YOU START UP MATLAB.

4.2 Study on material models

4.2.1 Complete the power law function

The power law material model is given as,

$$T(\varphi') \quad := \quad T_c \, \mathrm{sign}(\varphi') \left| \frac{\varphi'}{\varphi'_c} \right|^n ,$$

where the force T depends on the derivative of a given quantity φ. T_c, φ'_c, n are constants that can be adjusted to fit the behavior of specific materials. The simplicity makes it attractive for hand calculations but as you will observe it may not be optimal in terms of numerical properties.

Things to do:

- Complete the function `powerlaw.m` and `powerlaw_stiff.m` function.
- Compute the tangent $\dfrac{dT}{d\varphi'}$.
- Try plotting the constitutive law for the parameters given in your homework. For this case where n is larger than 1, what is the slope of the curve (tangent) at the origin? How do the parameters effect the curve? Draw a sketch of the force and tangent for varying φ'.
- What happens if you make the parameter n smaller than 1 ? what is the slope of the curve (tangent) at the origin for this case? Draw a sketch of the force and tangent for varying φ'.

4.2.2 Plot the Ramberg-Osgood function

The Ramberg-Osgood material model is given as,

$$\varphi' \quad := \quad \frac{T}{(GJ)_e} + K\left(\frac{T}{(GJ)_e}\right)^n ,$$

where $(GJ)_e$, K, $n(\geq 1)$ are constants that can be adjusted to fit the behavior of specific materials. The parameter $(GJ)_e$ defines the behavior of the function close to the origin and essentially represents the linear stiffness. As φ' increases the term with the power n kicks in and one obtains a ”flattening” non-linear type of behavior.

The force T and tangent $\frac{dT}{d\varphi'}$ for a given φ' is computed by the given functions `ramosg.m` and `ramosg_stiff.m`.

Things to do:

- Compute the tangent $\dfrac{dT}{d\varphi'}$.
- Try plotting the constitutive law for the parameters $(GJ)_e = 1$, $K = 0.01$, $n = 5$. For this case, what is the slope of the curve (tangent) at the origin? Is it as badly behaved as the power low? Draw a sketch of the force and tangent for varying φ'. How do the parameters effect the curve?

4.3 Gauss integration

To compute the integral on a fixed interval approximately, one can use a quadrature rule. The value of the integral is computed as a weighted sum of function values at specified points within the domain of integration. Such that the integral of $f(x)$ on the interval $[a, b]$ is approximated as,

$$\int_a^b f(x)\, dx \quad \approx \quad \sum_{i=1}^{N} f(x_i) W_i \ ,$$

where $x_i\ (i = 1, \ldots, N) \in [a, b]$ are points in the interval and $W_i\ (i = 1, \ldots, N)$ are weights.

The Gauss quadrature rule is an effective way to compute integrals and depending on the size of N, it can compute polynomials of order $2N - 1$ exactly, i.e., a polynomial $f(x)$ of order upto $2N - 1$ can be computed exactly with N-point Gauss quadrature.

For our purposes a 2 point quadrature rule will suffice where,

$$\begin{aligned}
N &= 2, \\
x_1 &= \frac{a+b}{2} + \frac{b-a}{2}\frac{1}{\sqrt{3}}, \\
x_2 &= \frac{a+b}{2} - \frac{b-a}{2}\frac{1}{\sqrt{3}}, \\
W_1 &= W_2 = \frac{b-a}{2} \ .
\end{aligned}$$

This is implemented in the function `quad_2pg.m`. This function can be used just like the function `quad.m`.

Things to do:

- Confirm that this function integrates polynomials upto order 3 exactly.

4.4 Torsion problem

In the file `torsion_ex2.m` the input file for running the torsion bar problem with one end fixed is given. There are two materials that you can select, a linear material and a Ramberg-Osgood model. Additionally you will implement your power law model.

A simple explanation of some functions you may need are summarized as follows,

`computeK_nlh.m`	: Computes the tangent stiffness matrix $\mathbf{K}(\mathbf{u})$
`computeFe_nlh.m`	: Computes the external force vector $\mathbf{F}_E$
`computeFi_nlh.m`	: Computes the internal force vector $\mathbf{F}_I$
`plotsol.m`	: Plots solution
`plot_history.m`	: Plots residual history of the Newton-Raphson scheme
`plot_hist.m`	: Plots deformation history of the Newton-Raphson for the bar
`newton_raphson.m`	: The Newton-Raphson algorithm

You have access to all the functions you used in the previous two Labs (13,14).

Things to do:

- Complete the function `computeFi_nlh.m` for computing the internal force vector $\mathbf{F}_I$. Look at the function `computeK_nlh.m` for hints.

  The way to check if your implementation is correct is by running the function using a linear material model. Recall that for a linear material model, $\mathbf{F}_i(\mathbf{u}) = \mathbf{K}\mathbf{u}$ where $\mathbf{K}$ does not depend on $\mathbf{u}$. Check and confirm that for a random $\mathbf{u}$, the quantities $\mathbf{K}\mathbf{u}$ and $\mathbf{F}_i(\mathbf{u})$ are equal to each other.

- Define appropriate functions `f_resid` and `f_stiff` using `computeK_nlh.m`, `computeFe_nlh.m`, and `computeFi_nlh.m`, which you can pass to the function `newton_raphson.m` to compute the nonlinear response of the torsion bar.

  First make sure that your example runs with the linear material model. Then change the model to the power law model. For initial guesses for the Newton-Raphson scheme, you can pick a linearly ramping function for the fixed-free case, and a linear function peaking at the middle for the fixed-fixed case. DO NOT START WITH A ZERO GUESS!!! WHY DO YOU THINK YOU SHOULDN'T?? This is related to how the slope of the power law looks near the origin.