

Numerical Methods for Mechanical Engineering A New Course at Clemson University

Daniel D. Warner

Department of Mathematical Sciences
Clemson University
warner@clemson.edu

November 16, 2009

MATLAB can be used as a calculator by typing expressions into the command window. More importantly, in keeping with its origins as a Matrix Laboratory, it can conveniently do extensive matrix calculations.

Alex, Barry, and Carol, three Bungee jumpers, plan to jump off a bridge

- Alex is tied to the bridge
- Barry is tied to Alex
- Carol is tied to Barry

Goal: determine their final positions.

Let z_A , z_B , and z_C denote the distances, in feet, below the bridge for Alex, Barry, and Carol, respectively.

Assume that the weights of the three jumpers are:

- $w_A = 225$ lbs
- $w_B = 160$ lbs
- $w_C = 105$ lbs

Three short cords, each 15 feet long with Hooke constants

- $c_1 = 75$ lbs/ft
- $c_2 = c_3 = 30$ lbs/ft

First approach, consider the force on each bungee cord separately.

- First bungee cord is being stretched by the weight of all three jumpers;
- Second bungee cord is only being stretched by the weight of the Barry and Carol; and
- Third bungee cord is only being stretched by the weight of Carol.

Apply Hooke's Law, to get these equations.

$$c_1 e_1 = w_A + w_B + w_C$$

$$c_2 e_2 = w_B + w_C$$

$$c_3 e_3 = w_C$$

Exercise 1. Using MATLAB as a calculator,

- 1 solve these equations to find the elongation for each bungee cord.
- 2 add up the elongations and the original lengths of the cords to calculate z_A , z_B , and z_C .
- 3 If the bridge is 80 feet above the river, will Carol be underwater when all the bouncing stops? If not, do you think that there is a good chance that she might be wet?

But MATLAB is a Matrix Laboratory, let

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad \text{and} \quad \mathbf{w} = \begin{bmatrix} w_A \\ w_B \\ w_C \end{bmatrix}$$

Using the vectors \mathbf{e} and \mathbf{w} we get the matrix equation

$$\mathbf{C}\mathbf{e} = \mathbf{A}\mathbf{w}$$

where \mathbf{C} is the diagonal matrix

$$\mathbf{C} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_3 \end{bmatrix} = \begin{bmatrix} 75 & 0 & 0 \\ 0 & 30 & 0 \\ 0 & 0 & 30 \end{bmatrix}$$

and \mathbf{A} is a 3×3 matrix that the student should figure out.

Then

$$\mathbf{e} = \mathbf{C}^{-1}\mathbf{A}\mathbf{w}$$

Given, \mathbf{A} and \mathbf{e} we can find the positions

$$\mathbf{z} = \begin{bmatrix} z_A \\ z_B \\ z_C \end{bmatrix}$$

from the equation

$$\mathbf{z} = \mathbf{A}^T(\mathbf{e} + \mathbf{L})$$

Exercise 2. Using MATLAB,

- 1 Calculate the matrices **C** and **A**
- 2 Solve the matrix equation for **e**.
- 3 Calculate **z**.

Some relevant key MATLAB idioms

$$C = \text{diag}(c)$$

$$e = \text{inv}(C) * A * w$$

$$e = C^{-1} * A * w$$

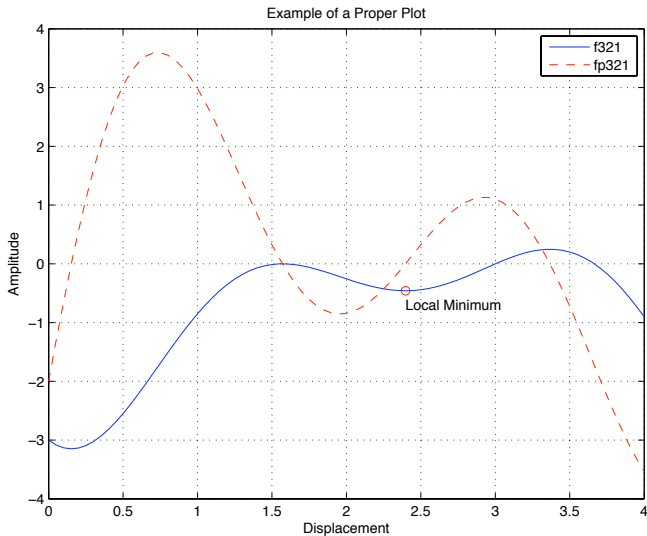
$$e = C \setminus (A * W)$$

$$z = A' * (e + L)$$

One of the most important pedagogical features of MATLAB is its simple but powerful plot commands.

Workshop Example 3 generates a proper plot of a simple oscillating function and its derivative. It includes

- 1 a title
- 2 labels for the x and y axes
- 3 a legend
- 4 a background grid.
- 5 and a highlighted and labeled indication of a local minimum.



Exercise 3. Modify the parameters in `f321` and `fp321`, then adjust the commands in `WorkshopEx03.m` so that the display of the new plot is appropriate.

Avoid unnecessary loops by using MATLAB's vector, matrix, and array operations. This Proper Plot provides several examples

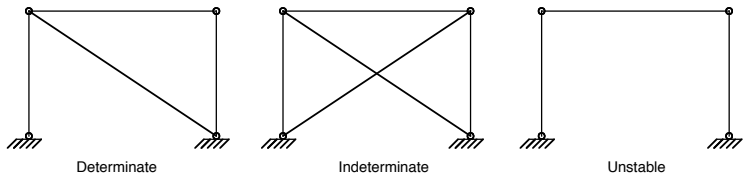
```
x = 0:0.05:4    % a row vector  
y = f321(x)    % vector valued functions  
yp = fp321(x)  % of a vector argument
```

Matrices and vectors, then arrays

```
y = (x-a).*(cos(b*x) + sin(c*x));
```

Drawing Trusses with MATLAB

- A truss is an idealized structure that is built from elastic bars
- Each bar resists a change in its length
- The bars are connected at points called nodes with pin joints, which can only transmit forces *along* the bars
- The pin joints allow the bars to point in any direction
- Bars are simpler than beams, plates, and shells.
 - 1 Unlike beams, they do not bend
 - 2 Unlike plates, they are one-dimensional
 - 3 Unlike shells, they are simple and straight



- Three trusses with 4 nodes.
- The nodes at the bottom are immovable.
- First Truss has 4 bars – a Determinate Truss
- Second Truss has 5 bars – an Indeterminate Truss
- Third Truss has 3 bars – an Unstable Truss

Specifying the Truss

Specify the coordinates of the nodes

```
nodes = [ 0.0  0.0  
          3.0  0.0  
          0.0  2.0  
          3.0  2.0 ]
```

Specify the nodes connected by each bar

```
bars = [ 1  3  
         2  3  
         2  4  
         3  4 ]
```

Add comments about units and other information.

A MATLAB code to input the truss specifications

```
name = input('Enter the file name: ', 's');  
eval(name);
```

The nodes and bars arrays are now defined. Plot the nodes

```
xn = nodes(:,1);  
yn = nodes(:,2);  
plot(xn,yn,'ok')
```

and plot the bars, where nb is the number of bars

```
for k=1:nb  
    s = bars(k,1);  
    t = bars(k,2);  
    plot([xn(s) xn(t)], [yn(s) yn(t)])  
end
```

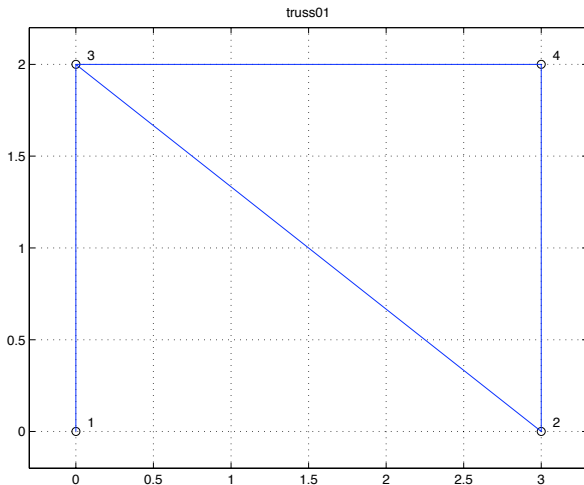
Adding A Little Splash

WorkshopEx04.m implements the preceding plot, but the figure is not entirely satisfactory.

- 1 It needs a small margin around the figure of the truss,
- 2 The nodes really need to be labeled.

WorkshopEx05.m includes two additional blocks of code to accomplish this.

WorkshopEx05 Rendition of Truss 1



Exercise 4. Construct a file, `truss02.m`, which specifies a triangular truss with three nodes and three bars. The nodes should be located at $(0, 0)$, $(5, 0)$, and $(10, 5)$. Generate a plot of this truss using `WorkshopEx05.m`.

Exercise 5. Construct a file, `truss03.m`, which specifies a truss of your own design with five or more nodes. Generate a plot of this truss using `WorkshopEx05.m`.

- A **Graph** is an abstraction of a truss that simply represents a set of points and the connections between pairs of points.
- The points are called nodes
- The connections between a pair of nodes are called edges
- The Edge-Node Matrix
 - 1 one row for each edge
 - 2 one column for each node
 - 3 source node assigned -1
 - 4 target node assigned $+1$

The Edge-Node Matrix captures the structure. For Truss 1

$$\mathbf{E}_1 = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

For Truss 2

$$\mathbf{E}_2 = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

The programming to build the Edge-Node Matrix is a good first step in learning how to build more complicated structural matrices.

The Laplacian Matrix is the transpose of the Edge-Node Matrix with itself. For Truss 1

$$\mathbf{L}_1 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

For Truss 2

$$\mathbf{L}_2 = \begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

The Laplacian Matrix has some neat properties.

Exercise 6. Add the two lines of code to the file WorkshopEx06.m that are needed to complete the calculation of the edge-node matrix.

Exercise 7. Run your modified version of WorkshopEx06.m on the files truss01.m, truss02.m, and truss03.m. The number of spanning trees for truss01.m and truss02.m should be 3. Depending on how creative your truss was, the number of spanning trees for truss03.m could be quite large and finding all of them could be a challenge. Take a stab at that challenge.

We can analyze a determinate truss, by assuming that the forces on each node are balanced – two equations for each node.

- The sum of the horizontal forces equals zero
- The sum of the vertical forces equals zero
- n nodes implies $2n$ equations in $2n$ unknowns.
- Forces on each bar constitute nb of the unknowns.
- The remaining unknowns are the reactive forces at the fixed nodes.

Suppose bar k connects node 1 at (x_1, y_1) with node 3 at (x_3, y_3) . If the bar is under tension, then the bar exerts a force, F_k , pulling nodes 1 and 3 towards each other.

Then the contribution of bar k is $+c_k F_k$ in the horizontal direction and $+s_k F_k$ in the vertical direction for node 1, where

$$c_k = \cos(\theta_k) = \frac{x_3 - x_1}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}}$$

$$s_k = \sin(\theta_k) = \frac{y_3 - y_1}{\sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}}$$

and θ_k is the angle of bar k relative to the normal coordinate system centered at node 1.

The angle of bar k relative to node 3 is $(\theta_k + \pi)$ so the contribution of bar k is $-c_k F_k$ in the horizontal direction and $-s_k F_k$ in the vertical direction.

Label the reactive forces H_1 , V_1 , H_2 , and V_2 and include vertical loads, L_3 and L_4 . This gives us the 8 equations:

$$F_1 \cdot 0 + H_1 = 0$$

$$F_1 \cdot 1 + V_1 = 0$$

$$F_2 \cdot c_2 + F_3 \cdot 0 + H_2 = 0$$

$$F_2 \cdot s_2 + F_3 \cdot 1 + V_2 = 0$$

$$-F_1 \cdot 0 - F_2 \cdot c_2 + F_4 \cdot 1 = 0$$

$$-F_1 \cdot 1 - F_2 \cdot s_2 + F_4 \cdot 0 = -L_3$$

$$-F_3 \cdot 0 - F_4 \cdot 1 = 0$$

$$-F_3 \cdot 1 - F_4 \cdot 0 = -L_4$$

We can write this system of equations as the matrix equation

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & c_2 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & s_2 & 1 & 0 & 0 & 0 & 0 & 1 \\
 0 & -c_2 & 0 & 1 & 0 & 0 & 0 & 0 \\
 -1 & -s_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 F_1 \\
 F_2 \\
 F_3 \\
 F_4 \\
 H_1 \\
 V_1 \\
 H_2 \\
 V_2
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 -L_3 \\
 0 \\
 -L_4
 \end{bmatrix}$$

Let \mathbf{A} denote the 8×8 matrix; let \mathbf{f} denote the vector of forces (internal and reactive); and let \mathbf{y} denote the vector of loads. This gives us the equation

$$\mathbf{A}\mathbf{f} = \mathbf{y}$$

Then once we have set up the matrix \mathbf{A} and the vector of loads, \mathbf{y} , we can easily solve for the forces.

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{y}$$

Extend the Truss specification file to include the fixed nodes and the load with the following two MATLAB code blocks

```
fixed = [ 1  1  
         1  1  
         0  0  
         0  0 ]
```

```
loads = [ 0  0  
         0  0  
         0 -50  
         0 -100 ]
```

The specification now includes all the information for **A** and **y**.

Exercise 8. Add the nine lines of code to the file WorkshopEx07.m that are needed to complete the calculation of the matrix **A**. Run your modified version on the truss04.m.

Exercise 9. Modify your truss02.m file, so that Node 1 is fixed both horizontally and vertically, Node 2 is fixed vertically, and Node 3 has a load of -100 pounds. Run your version of WorkshopEx07.m on this truss.

Exercise 10. Copy lines 74 through 80 of WorkshopEx08.m (or your modified version of WorkshopEx07.m) and adjust them so that the truss is redrawn with bars under tension plotted in blue and bars under compression plotted in red.

Newton's Second Law for a spring-mass system

$$m \frac{d^2x}{dt^2} = -cx \quad (1)$$

Newton's Second Law for a spring-mass system

$$m \frac{d^2x}{dt^2} = -cx \quad (1)$$

$$m \frac{dv}{dt} = -cx \quad (2)$$

Newton's Second Law for a spring-mass system

$$m \frac{d^2x}{dt^2} = -cx \quad (1)$$

$$m \frac{dv}{dt} = -cx \quad (2)$$

$$mv \frac{dv}{dt} = -cx \frac{dx}{dt} \quad (3)$$

Newton's Second Law for a spring-mass system

$$m \frac{d^2x}{dt^2} = -cx \quad (1)$$

$$m \frac{dv}{dt} = -cx \quad (2)$$

$$mv \frac{dv}{dt} = -cx \frac{dx}{dt} \quad (3)$$

$$m \int_0^T v \frac{dv}{dt} dt = -c \int_0^T x \frac{dx}{dt} dt \quad (4)$$

Newton's Second Law for a spring-mass system

$$m \frac{d^2 x}{dt^2} = -cx \quad (1)$$

$$m \frac{dv}{dt} = -cx \quad (2)$$

$$mv \frac{dv}{dt} = -cx \frac{dx}{dt} \quad (3)$$

$$m \int_0^T v \frac{dv}{dt} dt = -c \int_0^T x \frac{dx}{dt} dt \quad (4)$$

$$m \int_{v(0)}^{v(T)} v dv = -c \int_{x(0)}^{x(T)} x dx \quad (5)$$

This derivation establishes the Conservation of Energy for the spring-mass system.

$$\frac{1}{2}mv^2(T) + \frac{1}{2}cx^2(T) = \frac{1}{2}mv^2(0) + \frac{1}{2}cx^2(0)$$

- $\frac{1}{2}mv^2(T)$ is the kinetic energy of the moving mass
- $\frac{1}{2}cx^2(T)$ is the potential energy stored in the spring.

Back to Alex, Barry, and Carol, with masses, m_A , m_B , and m_C , and equilibrium positions z_A , z_B , and z_C .

Let x_1 , x_2 , and x_3 denote the displacements of the jumpers.

Then the elongation of each cord is given by

$$e_1 = x_1$$

$$e_2 = x_2 - x_1$$

$$e_3 = x_3 - x_2$$

and this can be written as the matrix equation

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mathbf{Ax}$$

Let \mathbf{y} denote the force acting on each spring, and by Hooke's Law

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_3 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \mathbf{C}\mathbf{e}$$

The external forces, \mathbf{f} , on the masses are the weights, so

$$\mathbf{f} = \begin{bmatrix} gm_A \\ gm_B \\ gm_C \end{bmatrix} = \begin{bmatrix} w_A \\ w_B \\ w_C \end{bmatrix} = \begin{bmatrix} 225 \\ 160 \\ 105 \end{bmatrix}$$

At equilibrium, the forces acting on each node must balance.

$$f_1 - (y_1 - y_2) = 0$$

$$f_2 - (y_2 - y_3) = 0$$

$$f_3 - y_3 = 0$$

As a matrix equation we get

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathbf{A}^T \mathbf{y}$$

- \mathbf{A} , relates displacements, \mathbf{x} , to elongations, \mathbf{e} .
- \mathbf{A}^T relates internal forces \mathbf{y} to external forces, \mathbf{f} .

The bungee jumper problem can be summarized by the three equations

$$\mathbf{e} = \mathbf{A}\mathbf{x} \quad (6)$$

$$\mathbf{y} = \mathbf{C}\mathbf{e} \quad (7)$$

$$\mathbf{f} = \mathbf{A}^T \mathbf{y} \quad (8)$$

We can solve this problem in three steps

- 1 Solve Equation 8 for \mathbf{y} ,
- 2 Solve Equation 7 for \mathbf{e} ,
- 3 Solve Equation 6 for \mathbf{x} .

A more general solution is to minimize the potential energy of the system. There are two components.

- 1 The work needed to recover the displacements

$$-(x_1 f_1 + x_2 f_2 + x_3 f_3) = -\mathbf{x}^T \mathbf{f}.$$

- 2 The potential energy stored in the stretched cords.

$$\frac{1}{2}c_1 e_1^2 + \frac{1}{2}c_2 e_2^2 + \frac{1}{2}c_3 e_3^2 = \frac{1}{2}\mathbf{e}^T \mathbf{C}\mathbf{e} = \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C}\mathbf{A}\mathbf{x}$$

Consequently the total potential energy of the system is

$$P(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C}\mathbf{A}\mathbf{x} - \mathbf{x}^T \mathbf{f}$$

Since \mathbf{C} is a diagonal matrix with positive entries, and since $\mathbf{Ax} \neq \mathbf{0}$ if $\mathbf{x} \neq \mathbf{0}$, then

$$\left(\mathbf{A}^T \mathbf{C} \mathbf{A}\right)^T = \mathbf{A}^T \mathbf{C} \mathbf{A}$$

and

$$\left(\mathbf{Ax}\right)^T \mathbf{C} \left(\mathbf{Ax}\right) > 0$$

In short, the stiffness matrix $\mathbf{K} = \mathbf{A}^T \mathbf{C} \mathbf{A}$ is symmetric and positive definite, and the Potential Energy function, $P(\mathbf{x})$, is minimized when \mathbf{x} satisfies

$$\mathbf{Kx} = \mathbf{f}$$

The minimization argument is elementary: let $\mathbf{u} = \mathbf{v} + \mathbf{x}$ then

$$P(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f}$$

The minimization argument is elementary: let $\mathbf{u} = \mathbf{v} + \mathbf{x}$ then

$$\begin{aligned} P(\mathbf{u}) &= \frac{1}{2} \mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f} \\ &= \frac{1}{2} (\mathbf{v} + \mathbf{x})^T \mathbf{A}^T \mathbf{C} \mathbf{A} (\mathbf{v} + \mathbf{x}) - (\mathbf{v} + \mathbf{x})^T \mathbf{f} \end{aligned}$$

The minimization argument is elementary: let $\mathbf{u} = \mathbf{v} + \mathbf{x}$ then

$$\begin{aligned}P(\mathbf{u}) &= \frac{1}{2}\mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f} \\&= \frac{1}{2}(\mathbf{v} + \mathbf{x})^T \mathbf{A}^T \mathbf{C} \mathbf{A} (\mathbf{v} + \mathbf{x}) - (\mathbf{v} + \mathbf{x})^T \mathbf{f} \\&= \frac{1}{2}\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \frac{1}{2}\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} \\&\quad + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{v}^T \mathbf{f} - \mathbf{x}^T \mathbf{f}\end{aligned}$$

The minimization argument is elementary: let $\mathbf{u} = \mathbf{v} + \mathbf{x}$ then

$$\begin{aligned}P(\mathbf{u}) &= \frac{1}{2}\mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f} \\&= \frac{1}{2}(\mathbf{v} + \mathbf{x})^T \mathbf{A}^T \mathbf{C} \mathbf{A} (\mathbf{v} + \mathbf{x}) - (\mathbf{v} + \mathbf{x})^T \mathbf{f} \\&= \frac{1}{2}\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \frac{1}{2}\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} \\&\quad + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{v}^T \mathbf{f} - \mathbf{x}^T \mathbf{f} \\&= \frac{1}{2}\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{v}^T \mathbf{f} + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{f}\end{aligned}$$

The minimization argument is elementary: let $\mathbf{u} = \mathbf{v} + \mathbf{x}$ then

$$\begin{aligned}
 P(\mathbf{u}) &= \frac{1}{2} \mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f} \\
 &= \frac{1}{2} (\mathbf{v} + \mathbf{x})^T \mathbf{A}^T \mathbf{C} \mathbf{A} (\mathbf{v} + \mathbf{x}) - (\mathbf{v} + \mathbf{x})^T \mathbf{f} \\
 &= \frac{1}{2} \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \frac{1}{2} \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} \\
 &\quad + \frac{1}{2} \mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{v}^T \mathbf{f} - \mathbf{x}^T \mathbf{f} \\
 &= \frac{1}{2} \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{v}^T \mathbf{f} + \frac{1}{2} \mathbf{x}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{f} \\
 &= \frac{1}{2} \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + \mathbf{v}^T (\mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{f}) + P(\mathbf{x})
 \end{aligned}$$

Since $\mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{x} - \mathbf{f} = 0$, and we get

$$P(\mathbf{u}) = \frac{1}{2} \mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} + P(\mathbf{x})$$

If we let $\mathbf{w} = \mathbf{A} \mathbf{v}$, then

$$\mathbf{v}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{v} = \mathbf{w}^T \mathbf{C} \mathbf{w} = c_1 w_1^2 + c_2 w_2^2 + c_3 w_3^2 + c_4 w_4^2$$

Since all the c_i 's are positive this is a positive number, and we can conclude that $P(\mathbf{x})$ must be the minimum.

Exercise 11. The file WorkshopEx09.m finds the equilibrium positions of the bungee jumpers by minimizing the potential energy of the system. Run it and verify that it arrives at the same positions for the jumpers that we obtained earlier by balancing the forces.

Exercise 12. Run WorkshopEx09.m so that \mathbf{K} , the stiffness matrix, is available. Using displacement vectors \mathbf{u} of your own choosing, calculate the potential energy associated with that displacement using the statement

$$p = 0.5 * u' * K * x - x' * w$$

- The Analysis of Determinate Trusses is based on the premise that the sum of the forces is zero at each node.
- Decomposing the forces into horizontal components vertical components creates two equations for each of the n nodes.
- Does not work if the truss includes additional bars for extra strength.
- Assume that each bar is compressed or stretched slightly
- The internal force is proportional to the elongation of the bar
- Then we can calculate and minimize the potential energy

Consider a bar connecting node 1 at (x_1, y_1) to node 2 at (x_2, y_2) .
Under load the nodes are displaced

$$(x_1 + \Delta x_1, y_1 + \Delta y_1) \quad \text{and} \quad (x_2 + \Delta x_2, y_2 + \Delta y_2)$$

The length of the bar before it is displaced

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The length of the bar under load is

$$\begin{aligned} \bar{L} &= \sqrt{[(x_2 + \Delta x_2) - (x_1 + \Delta x_1)]^2 + [(y_2 + \Delta y_2) - (y_1 + \Delta y_1)]^2} \\ &= \sqrt{[(x_2 - x_1) + (\Delta x_2 - \Delta x_1)]^2 + [(y_2 - y_1) + (\Delta y_2 - \Delta y_1)]^2} \end{aligned}$$

Consider what happens if $\Delta x_1 \neq 0$ and $\Delta y_1 = \Delta x_2 = \Delta y_2 = 0$.

$$\bar{L} = \sqrt{[(x_2 - x_1) + (-\Delta x_1)]^2 + (y_2 - y_1)^2}$$

Consider what happens if $\Delta x_1 \neq 0$ and $\Delta y_1 = \Delta x_2 = \Delta y_2 = 0$.

$$\begin{aligned}\bar{L} &= \sqrt{[(x_2 - x_1) + (-\Delta x_1)]^2 + (y_2 - y_1)^2} \\ &= \sqrt{(x_2 - x_1)^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2 + (y_2 - y_1)^2}\end{aligned}$$

Consider what happens if $\Delta x_1 \neq 0$ and $\Delta y_1 = \Delta x_2 = \Delta y_2 = 0$.

$$\begin{aligned}\bar{L} &= \sqrt{[(x_2 - x_1) + (-\Delta x_1)]^2 + (y_2 - y_1)^2} \\ &= \sqrt{(x_2 - x_1)^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2 + (y_2 - y_1)^2} \\ &= \sqrt{L^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2}\end{aligned}$$

Consider what happens if $\Delta x_1 \neq 0$ and $\Delta y_1 = \Delta x_2 = \Delta y_2 = 0$.

$$\begin{aligned}\bar{L} &= \sqrt{[(x_2 - x_1) + (-\Delta x_1)]^2 + (y_2 - y_1)^2} \\ &= \sqrt{(x_2 - x_1)^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2 + (y_2 - y_1)^2} \\ &= \sqrt{L^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2} \\ &= \sqrt{L^2 - 2L\frac{(x_2 - x_1)}{L}\Delta x_1 + \Delta x_1^2}\end{aligned}$$

Consider what happens if $\Delta x_1 \neq 0$ and $\Delta y_1 = \Delta x_2 = \Delta y_2 = 0$.

$$\begin{aligned}\bar{L} &= \sqrt{[(x_2 - x_1) + (-\Delta x_1)]^2 + (y_2 - y_1)^2} \\ &= \sqrt{(x_2 - x_1)^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2 + (y_2 - y_1)^2} \\ &= \sqrt{L^2 - 2(x_2 - x_1)\Delta x_1 + \Delta x_1^2} \\ &= \sqrt{L^2 - 2L \frac{(x_2 - x_1)}{L} \Delta x_1 + \Delta x_1^2} \\ &= \sqrt{L^2 - 2L \cos(\theta)\Delta x_1 + \Delta x_1^2}\end{aligned}$$

where θ is the angle the bar originally made with the x -axis and $\cos(\theta) = (x_2 - x_1)/L$.

- Neglect Δx_1^2
- Use Taylor's Theorem to approximate $\bar{L}(\Delta x_1)$, with its first order Taylor polynomial.

$$\begin{aligned}\bar{L}(\Delta x_1) &\approx L - \frac{1}{2L} (2L \cos(\theta)) \Delta x_1 \\ &\approx L - \cos(\theta) \Delta x_1\end{aligned}$$

- Repeat for Δx_2 , Δy_1 , and Δy_2 to get

$$\bar{L} \approx L - \cos(\theta) \Delta x_1 + \cos(\theta) \Delta x_2 - \sin(\theta) \Delta y_1 + \sin(\theta) \Delta y_2$$

- The linearized elongation, e , is

$$e = \bar{L} - L \approx -\cos(\theta) \Delta x_1 + \cos(\theta) \Delta x_2 - \sin(\theta) \Delta y_1 + \sin(\theta) \Delta y_2$$

Given a truss with n nodes and m bars, construct \mathbf{A}_0 . \mathbf{A}_0 will have m rows, one for each bar, and it will have $2n$ columns, two for each node, the first column in the pair corresponding to the horizontal displacement and the second corresponding to the vertical displacement.

For each bar we will compute $c_k = \frac{(x_2 - x_1)}{L}$ and $s_k = \frac{(y_2 - y_1)}{L}$

Let \mathbf{u}_0 denote the vector of displacements with $u_1 = \Delta x_1$, etc. Also let \mathbf{e} denote the vector of elongations. For small displacements elongations can be well approximated by

$$\mathbf{e} = \mathbf{A}_0 \mathbf{u}_0$$

The preliminary structural matrix for the determinate truss is

$$\mathbf{A}_0 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -c_2 & -s_2 & c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \end{bmatrix},$$

and the preliminary structural matrix for the indeterminate truss is

$$\mathbf{A}_0 = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -c_2 & -s_2 & c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ -c_5 & -s_5 & 0 & 0 & 0 & 0 & c_5 & s_5 \end{bmatrix}.$$

For both trusses, nodes 1 and 2 are fixed, which means that $u_1 = \Delta x_1$, $u_2 = \Delta y_1$, $u_3 = \Delta x_2$, and $u_4 = \Delta y_2$, can only be zero. Consequently, the first four columns of these two matrices can not contribute to the calculation of \mathbf{e} , and we can write

$$\mathbf{e} = \mathbf{A}\mathbf{u} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \end{bmatrix} \mathbf{u}$$

and

$$\mathbf{e} = \mathbf{A}\mathbf{u} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ c_2 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & c_5 & s_5 \end{bmatrix} \mathbf{u}$$

The potential energy analysis for the three bungee jumpers also applies to trusses. But now \mathbf{A} reflects the truss's 2D geometry.

The elongations of the bars are given by

$$\mathbf{e} = \mathbf{A}\mathbf{u}$$

The internal forces acting on the bars are given by

$$\mathbf{y} = \mathbf{C}\mathbf{e}$$

The balance between the external forces and the internal forces is

$$\mathbf{f} = \mathbf{A}^T\mathbf{y}$$

In equilibrium, the truss will minimize the total potential energy

$$P(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{A}^T \mathbf{C} \mathbf{A} \mathbf{u} - \mathbf{u}^T \mathbf{f} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u} - \mathbf{u}^T \mathbf{f}$$

and, as before, this function is minimized when \mathbf{u} satisfies

$$\mathbf{K} \mathbf{u} = \mathbf{f}$$

From the displacements we calculate the internal forces on the bars

$$\mathbf{y} = \mathbf{C} \mathbf{A} \mathbf{u}$$

Finally, we can calculate the vector, \mathbf{f}_0 , of all external forces, including the reactive forces, from

$$\mathbf{f}_0 = \mathbf{A}_0^T \mathbf{y}$$

Exercise 13. Add the 9 lines of code to WorkshopEx10.m that are needed to complete the calculation of the matrix **A0**. Run your modified version of WorkshopEx10.m on the file truss06.m, the determinate truss specification with loads and Hooke's constants.

Exercise 14. Modify your truss05.m file, call it truss07.m, to include Hooke's constants for the bars and run your modified WorkshopEx10.m on truss07.m

Exercise 15. Modify your truss03.m file, call it truss08.m, to include loading information and Hooke's constants for the bars and run your modified WorkshopEx10.m on truss08.m

Class Project

