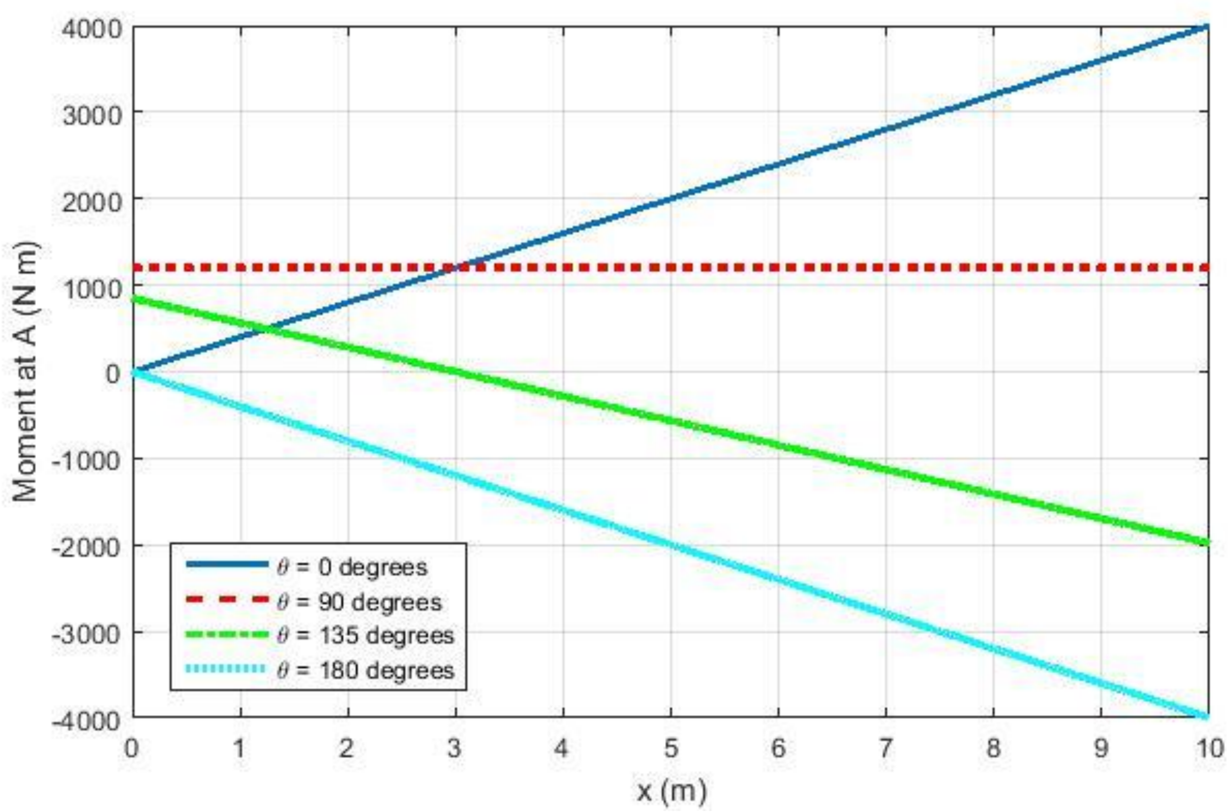
UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING
First Year Program – Core 8 and TrackOne

FIRST YEAR CORE 8 ENGINEERING PROGRAM
ENGINEERING PROBLEM SOLVING LABS

MAT188: Laboratory #8
Parameterization



PARAMETERIZATION

In this lab, you will use MATLAB to model an engineering problem with two input parameters and use matrices to find an approximate solution.

Preparation *(Required doing* ***before*** *you come to the laboratory session)*

1. Read through this lab document.
2. Consider the Engineering Application problem described below. Determine $M(x,\theta)$, the expression for the moment at A in terms of the variables x and θ.

Learning Outcomes: By the end of this lab, students will be able to:

1. Explore nested for loops in the context of linear algebra to solve a civil engineering problem
2. Critically reflect on the validity of their solution and discuss this in reference to the problem(s) identified.

MATLAB Skills and Knowledge:
Nested For Loops

Nested For Loops

In Lab 5, you learned about for loops to repeat sections of code. We will now extend this idea to create ***nested for loops***, which use a loop inside of another loop.

For example, let's use Newton's second law ($F_{net} = ma$) to calculate the value of F_{net} for various values of m and a and store the results in a matrix. The `length` function is used to get the number of values in each vector, loop through each of the indices, and access the value at each index. ***Create a new script and type in the following example to see the result:***

```
F=[];
m=linspace(3,8,11);
a=linspace(0,5,21);

for i_m=1:length(m)
    for i_a=1:length(a)
        F(i_m,i_a)=m(i_m)*a(i_a);
    end
end

F   % Output the result
```

Using `length` to get the number of elements in each vector, `i_m` and `i_a` loop through each index in the corresponding vectors, and vector indexing is used to get and set the actual values in the vectors. In this case, the command inside the loop is executed `length(m) * length(a)` (or 11 * 21 = 231) times.

The resulting matrix `F` will be of the following form, with the value of F_{net} for each pair of m and a values:

$$\begin{array}{cc} & \xrightarrow{\hspace{3em} a\ increases \hspace{3em}} \\ m\ increases \downarrow & \begin{bmatrix} F(1,1) & \cdots & F(1,21) \\ \vdots & \ddots & \vdots \\ F(11,1) & \cdots & F(11,21) \end{bmatrix} \end{array}$$

Engineering Applications: Design Exercise

The Engineer's Toolbox - Parameterization

Numeric computation allows you to create a mathematical model of a physical system or problem to test a wide variety of possible solutions. This virtual testing, or *simulation*, of the problem can be done by ***parameterizing*** a characteristic and calculating the response of the physical system as this parameter is changed.

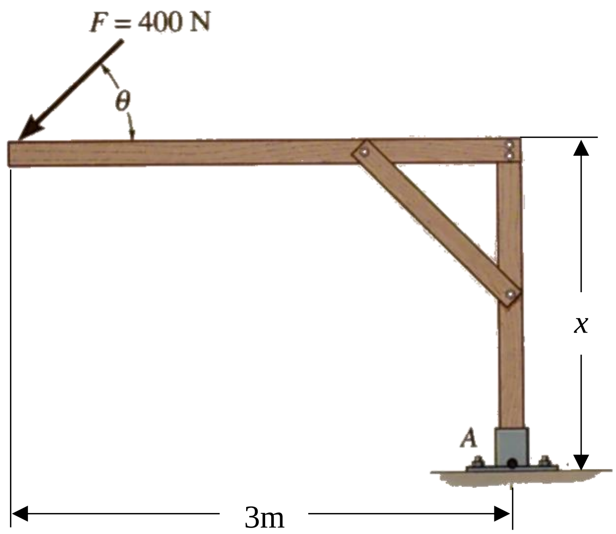


Figure 1: *L*-Shaped Support (Taken from Problem 4-29, R.C. Hibbeler, *Engineering Mechanics: Statics*, 10th Edition, Pearson, 2004, pg. 133.)

Consider the *L*-shaped support in the figure above. It consists of a 3 m long horizontal arm, a vertical support of variable length ($0 \leq x \leq 10$ m), and is subjected to a 400 N force at a variable angle ($0 \leq \theta \leq 180°$).

Part 1: *Understand how the moment at A depends on the parameters x and θ by creating two 2D figures.*

Write a script to calculate the moment at A for each combination of *x* and θ in the range given above, using the expression you determined in your preparation exercise.

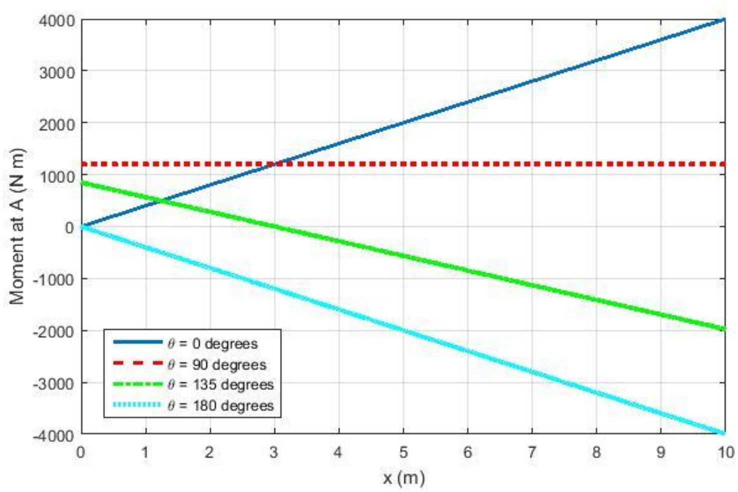
To better understand this situation, we will use MATLAB to generate a matrix with the various values of the moment at A as *x* and θ change. If we allow `theta` to be a vector with *m* elements (let $m = 401$) and `x` to be a vector with *n* elements (let $n = 501$), the *m* x *n* matrix will look like:

$$\begin{array}{cc} & \theta \text{ is constant, } x \text{ varies} \longrightarrow \\ \mathbf{M} = & \begin{bmatrix} M_A(1,1) & \cdots & M_A(1,n) \\ \vdots & \ddots & \vdots \\ M_A(m,1) & \cdots & M_A(m,n) \end{bmatrix} \Big\downarrow \begin{array}{l} x \text{ is constant,} \\ \theta \text{ varies} \end{array} \end{array}$$

Note: θ is in degrees, but the standard trigonometric functions take angles in radians. You will either have to convert θ to radians or use the trigonometric functions that take degrees (refer to the MATLAB Summary or documentation).

Now, add to your script to create two 2D plots that demonstrate how the moment at A depends on x and θ. For each plot, add a title, axis labels, a legend, and different styles/colours for each line.

Plot #1: For four constant values of θ (at indices 1, 201, 301, 401), show how the moment at A varies according to x.



Plot #2: For five constant values of *x* (at indices 1, 201, 301, 401, 501), show how the moment at A varies according to θ, similar to the figure above.

***Hint**:* To keep one parameter constant while the other changes, you will need to select an entire row or column in the matrix. For example, if you just wanted to plot `x` against `M` for a constant θ (at index 1):

```
plot(x,M(1,:));
```

Hint: To create titles or legends that contain values within a vector or matrix, you can use the the `sprintf` command, similar to the following commands:

```
label1=sprintf('x = %2.1f m',x(1));
label2=sprintf('x = %2.1f m',x(201));
legend(label1,label2);
```

`sprintf` contains specifiers using the `%` sign, which substitute values with particular formatting properties into the string. In this case, `%s` means a string and `%2.1f` means a floating-point (decimal) number with two digits before the decimal point and one digit after.

```
label3=sprintf('%s = %2.1f %s','\theta',theta(1),'\circ');
label4=sprintf('%s = %2.1f %s','\theta',theta(201),'\circ');
legend(label3,label4);
```

`\theta` creates a theta symbol (θ), while `\circ` creates a degrees symbol (°). You can use `help` or online documentation to read more about the `sprintf` function and formatting options.

Part 2: *Analyze your figures and explain why you are confident that your calculations and procedure are correct. Hint*: Start by considering the results for the extreme or unique cases (such as $x = 0$ m, $x = 10$ m, ***others?***)

Based on the two plots created in Part 1, you should have a good sense of how the moment at A depends on the two design parameters x and θ. You should also be able to look at the ***edge and special cases*** to assess if your mathematical modeling, i.e., your expression for $M(x, \theta)$, and subsequent numeric calculations are correct. To assess this, answer the following questions:

Edge and Special Case Considerations

1) Do your results make sense when $\theta = 90°$ as x varies? What happens?
2) Do your results make sense when $x = 0$ as θ varies? What happens?
3) As x increases, do the results make sense? Pay particular attention to the location of the maximum moment. If $x \to \infty$, from your understanding of the physical problem where would you expect the maximum moment to occur? Do your results indicate that this will happen?

Interpreting Results

1) If you wanted to have the moment at A be completely independent of x, what would you do based on your results?
2) If you wanted to have the moment of A be zero, what would you do based on your results?

For submission:
Students may work in pairs to answer the questions in this lab, but each student must submit their own work. Each student must put their own name and student number on their submission, and it is their responsibility to ensure that all questions from the lab have been appropriately answered (using relevant figures, code, etc).

Note: It is neither fair nor professional to have one person do all the work; note that the MatLab quiz in this course is NOT a pair-programming exercise, and thus, individual mastery is key (as is communicating that mastery).

One file submission per person, in either .docx or .pdf (preferred) format.

Deadlines:
Monday Practical Sections (PRA 113, 114, 116): no later than Saturday November 10^{th} by 11:59pm
All other Practical Sections, no later than Saturday November 3^{rd} by 11:59pm.