Automating Structural Stress Analysis: Beam Deflection, Shear, and Moment Diagram Generator for Single and Multi-Span Beams

by

Cody Jacobucci



Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

June 2017



Signature redacted

Signature of Author: ________________________________
Department of Mechanical Engineering
May 12, 2017

Signature redacted

Certified by: ________________________________
Brian Yutko
MIT Research Engineer, Department of Aeronautics and Astronautics
Aurora Flight Sciences, X-Plane Program Manager
Signature redacted Thesis Supervisor

Accepted by: ________________________________
Rohit Karnik
Associate Professor of Mechanical Engineering
Undergraduate Officer

# Automating Structural Stress Analysis: Beam Deflection, Shear, and Moment Diagram Generator for Single and Multi-Span Beams

by

Cody Jacobucci

Submitted to the Department of Mechanical Engineering
on May 12, 2017 in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

ABSTRACT

A new tool has been developed with Aurora Flight Sciences to automate stress analysis of beams under loading. It is a Microsoft Excel based tool to be consistent with Aurora's other analysis tools and analyst preference, and is coded in Visual Basic. The tool can generate the shear, moment, and deflection diagram of a single span or multi-span beam in less than 10 seconds for any combination of edge constraints and applied loads, as well as output the reaction force at each support. The tool can also analyze beams that change material or shape after reaching a support to account for changes in longer beams across the airframe.

Thesis Supervisor: Brian Yutko
Tile: MIT Research Engineer, Department of Aeronautics and Astronautics
Aurora Flight Sciences, X-Plane Program Manager

# Table of Contents

# List of Figures

## List of Tables

## 1. Introduction

Stress Analysis is fundamental to the design of every product and structure. The role of an analyst is to model a structure and the loads applied to it, and determine if the part can withstand the ultimate loads with appropriate factors of safety built in. Analysts use a variety of tools to assist them, from Finite Element Modeling all the way to classic hand calculations. Currently, Aurora Flight Sciences is developing a new aircraft and is beginning to size the first version of the airframe structure. Topology optimization software like HyperSizer has revolutionized airframe sizing to quickly give structures based on assigned loads. However, the FAA has strict guidelines on acceptable forms of analysis for aero-structures and many of the new programs have yet to be approved by the FAA. Even with all of the software available, it is often quicker to have basic first order principle tools based on hand calculations to validate designs.

For this phase in the design process, it is assumed that the airframe is a rigid structure and that deflections of individual beams are small. Classic Euler beam theory can be applied to analyze the structure and determine if the design is sufficiently strong. However, there are numerous boundary conditions and loads that could be applied anywhere on the beam, making the calculations extremely time consuming after starting with a fourth order differential equations. This becomes even more complicated when working with multispan beams given the additional geometric constraint in the middle. The goal of this work was to create a user-friendly tool that can provide the analyst with the shear, moment, and deflection diagram for a beam, given any boundary condition or loading condition.

This paper will outline open-form beam bending solutions found in popular airframe structural design handbooks, as well as discuss the steps towards forming closed form solutions for various loads and boundary conditions to create a robust analysis tool to improve the efficiency of stress analysts during the early airframe development phase. The tool is Excel based and automatically generates graphs of the shear, moment, and deflection of a beam based on edge constrains and loading conditions. The entire code is also included so that this tool can be used by others as an educational tool.

## 2. Background

This section will walk through all the beam bending theory needed to create the analysis tool to automate the generation of shear, moment, and deflection diagrams for various beams.

### 2.1 Euler-Bernoulli Beam Theory

Euler-Bernoulli Beam Theory is the most basic beam theory that there is, relating deflection to the applied load in the classic fourth order differential Equation [1]:

$$\frac{d^2}{dx^2}\left(EI\frac{d^2w}{dx^2}\right) = q(x) \tag{1}$$

Here, E is the young's modulus, I is the second moment of inertia, $w$ is deflection, and $q$ is the distributed load on the beam as a function of $x$ [N/m]. $x$ moves in the direction of the length of the beam. This equation needs to be integrated 4 times to finally arrive at the displacement of the beam $w(x)$, more commonly seen as $\delta(x)$. The first integration will give the shear load $V(x)$, followed by the moment $M(x)$, then the angle of the beam $\theta(x)$, and finally the displacement $\delta(x)$. These factors are important to consider when thinking about which fasteners should be used at joints, how much brackets will be displaced, if the beam will displace enough to run into geometric constraints (like other beams or electrical components), or even if the beam in question is sized correctly.

## 2.2 Common Edge Constraints

The most common constraints that are seen are fixed ends, simply supported ends, hinged ends, or free ends. Each constraint has a new set of appropriate boundary condition to be used with the Euler-Bernoulli equation.

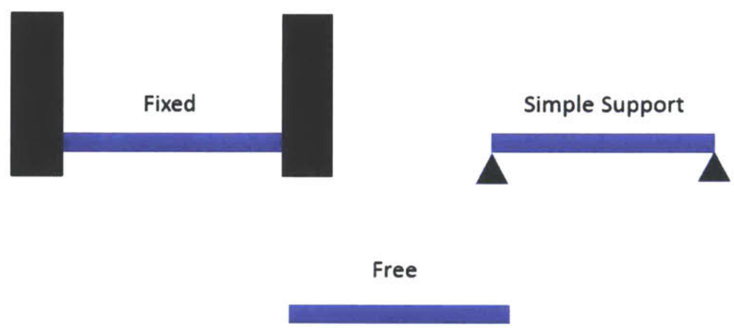



**Figure 2-1:** The three most common edge constraints.

For example, a beam with the left and right end simply supported would have $\delta(0)| = 0$, and $\delta(L) = 0$ for a beam of length L. Regardless of the load applied or the location of the load, $M(0) = M(L) = 0$. With these 4 boundary conditions the equation can be solved for in terms of variables needed to depict different loading styles to keep the solutions as robust as possible.

## 2.3 Common Loading Conditions

Common loads applied to beams in an airframe structure are distributed loads (often coming from aerodynamic loads transferred from the skin to the frame), point loads (load paths distributing loads from the wings to the body of the aircraft), and applied moments (also from the wing and other internal loads).

## 2.4 Superposition Principle

A beam in an aircraft will often have multiple loads going through it. For example, frame beams at the bottom of the plane will be subjected to a distributed load from the aerodynamic forces applied from drag, point loads from vertical supports that tie into the floor above, as well as applied moments at the end resulting from forces on the wings that is transferred through the other beams. The superposition principle states that the beam can be analyzed based on each individual load, and the resulting shear, moment, and deflection will be the sum of the individual contributions. For example, a cantilevered beam with a linear distributed load, a point load, and an applied moment can be broken down an analyzed as three separate beams as shown in Figure 2-2.

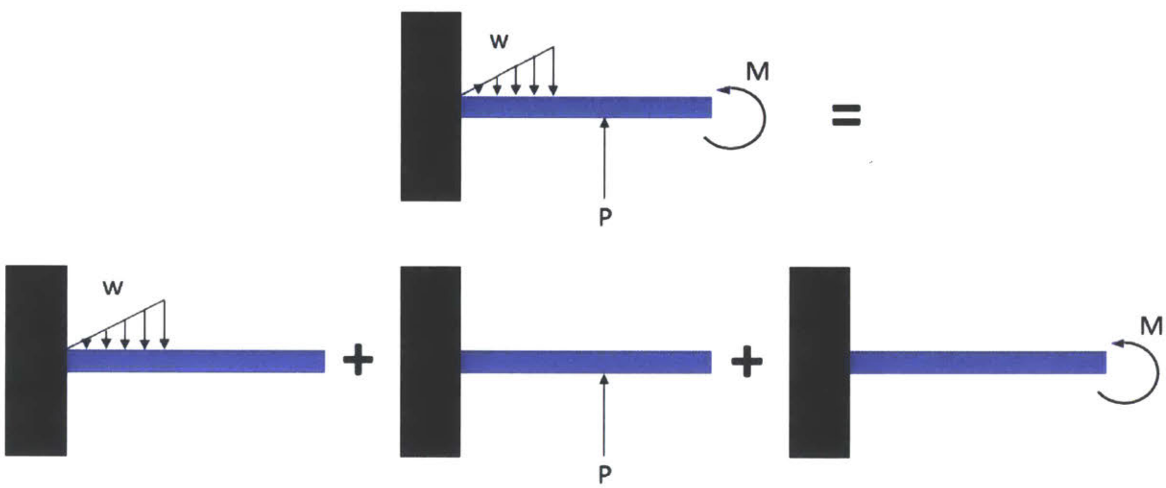



**Figure 2-2:** The top beam can be modeled as the sum of the shear, moment, and deflections calculated for a cantilevered beam with a linear distributed load, another with a point load P, and another with an applied moment M.

All of the results are added together to compute the final equations modeling $V(x)$, $M(x)$, $\theta(x)$, and $\delta(x)$.

## 2.5 Multi-Span Beams and the 3-Moment Equation

While single beam cases are relatively simple, multi-span beams that are broken up by rigid supports are much more difficult to handle.

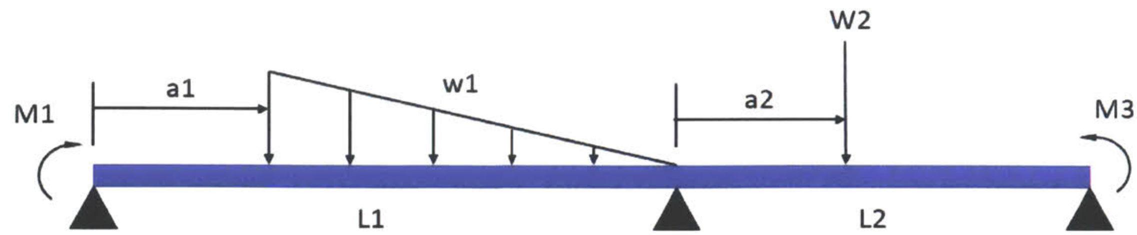



**Figure 2-3:** A sample multi-span beam with multiple loads.

The most common method to analyze multi-span continuous beams consisting of *n* segments is the 3-Moment Equation. The 3-Moment Equation refers to the equation created by equating the angle of the left segment of the beam to the right segment of the beam where they meet at the support. [2] Physically speaking, the angle of these beams must be equal for the beam to be continuous.

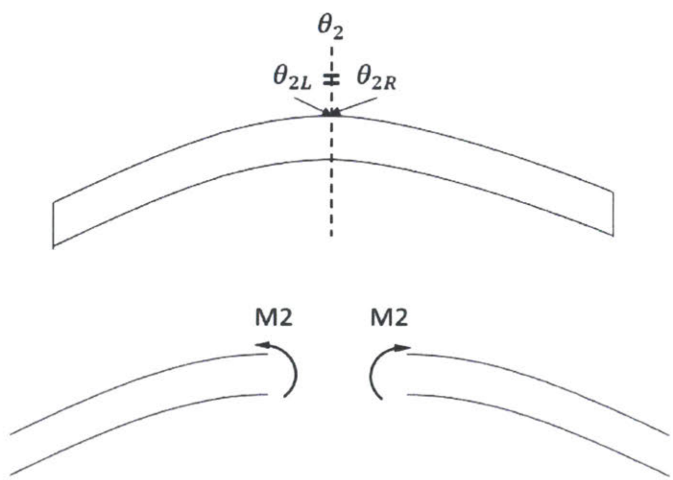



**Figure 2-4:** The top beam shows that $\theta_{2L} = \theta_{2R}$, validating that the central angle can be used to create an additional equation and solve for the $M_2$ found inside of the beam over the center support.

The angle of the beam is expressed in terms of the applied loads (using superposition) and any possible applied moments for each beam. This includes the internal moment created where the left section of the beam meets the right section at the joint. By setting $\theta_2$ in terms of known applied loads and the unknown moment in the middle, the middle Moment $M_2$ can be solved for by completing superposition of angles created for each side of the beam. As an example, the beam in Figure 2-3 would have:

$$\theta_{2L} = \frac{w_1(l_1^2 - a_1^2)^2}{24E_1I_1l_1} - \frac{w_1(l_1 - a_1)^2}{360E_1I_1l_1}(8l_1^2 + 9a_1l_1 + 3a_1^2) + \frac{M_1l_1^2}{6E_1I_1l_1} - \frac{M_2(l_1^2 - 3a_1^2)}{6I_1l_1}$$

$$\theta_{2R} = \frac{W_2a_2}{6E_2I_2l_2}(2l_2 - a_2)(l_2 - a_2) - \frac{M_2 2l_2^2}{6E_2I_2l_2} - \frac{M_3(2l_2^2 - 6l_2^2 + 3l_2^2)}{6E_2I_2l_2}$$

Once $M_2$ is solved for, the superposition principle can be used to add the effects of the this applied moment to each beam to compute the effective shear, moment, and deflection of the beam entire beam. This can be done with any size beam with *n* sections.

### 2.6 Singularity Variable

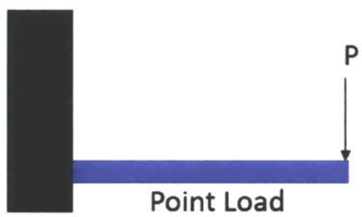
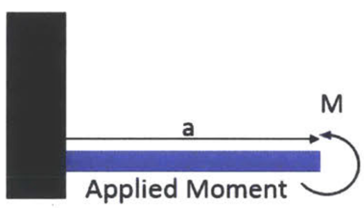
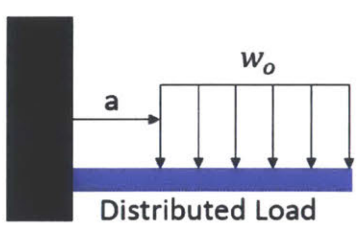
For ease of derivations, all of the equations used to develop this tool made use of singularity functions. [3] The singularity function can be seen as $\langle x - a \rangle^n$ where

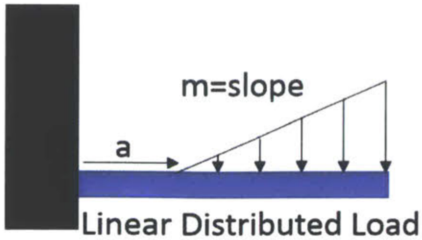
$$if\ x \geq a;\ \langle x - a \rangle^n = (x - a)^n$$

$$else\ if\ x < a;\ \langle x - a \rangle^n = 0$$

Here, *a* is equal to the location of a load from the origin, and *x* is the distance from the origin. Below is a table displaying different loading conditions and the singularity function applied to the fourth order differential equation. With each consequent integration, the exponent of the singularity function will increase by 1.

**Table 2-1:** Singularity functions based on the applied load style. Note that a second order distributed load would have $n = 2$, and $n$ would increase by 1 for each additional order of the distributed load.

| Load Type | Singularity Function |
| --- | --- |
|  | $-P\langle x-a\rangle^{-1}$ |
|  | $-M\langle x-a\rangle^{-2}$ |
|  | $-w_o\langle x-a\rangle^{0}$ |

| | |
|---|---|
|  | $-m\langle x-a\rangle^1$ |

**3. Design**

In this section, the design decisions for this tool are explained.

**3.1 Use of Excel**

It is important to keep the user in mind when designing anything. After surveying the stress analyst group on which programs they use most often and where their strengths lied, the overwhelming result was Excel. Excel is used across many of the big OEMs because of the ability to easily create macros and organize the data in a visual way. They also did not have access to MATLAB without purchasing more licenses, so Excel was the most cost effective solution.

**3.2 Iterations**

The design of this tool went through multiple iterations as time went on and further study revealed elegant solutions to solve indeterminate systems.

**3.2.1 Iteration 1: Single Span**

The original task was just to create a tool that created shear, moment, and deflection diagrams for common single span loading conditions (like a cantilevered beam with a point load at the end). *Roark's Stress and Strain* contained closed form solutions for the most common boundary conditions and load cases. However, further study revealed that some of the equations are actually wrong and the publisher has been contacted by me with the support of my group for changes to be made for future iterations. As a result, all of the equations had to be hand validated, which consumed a significant amount of time for creating this tool. Please reference Roark's text for outlines of equations to solve for $V(x)$, $M(x)$, $\theta(x)$, and $\delta(x)$ for various load types and edge constraints. In the end, a tool was created to graph $V(x)$, $M(x)$, and $\delta(x)$ for point and distributed loads, for simple, fixed, and free ends.

### 3.2.2 Iteration 2: Single Span with Superposition

After completing iteration 1, it was fairly simple to add logic that would rerun the body of the code if there was a second load input detected. Currently, the single beam is modeled to take up to 3 loads of any kind at the same time using the superposition principal. Additionally, this iteration could solve for $V(x)$, $M(x)$, and $\delta(x)$ with applied moments.

### 3.2.3 Iteration 3: Multi-Span with Superposition

Building off iteration 2, iteration 3 was created for a multi-span beam. Using the concepts discussed in 2.5, new equations were derived to solve for $\theta_2$ (and consequentially $M_2$). Reference Appendix B to find the result of applying the 3 Moment Equation for every combination of end constraints and load types. Typically, the 3 Moment equation must be solved on a case by case basis, but this tool takes advantage of symmetry, superposition, and equations of $\theta(x)$ that contain all of the singularity variables to apply to any load type. This function of the tool saves a significant portion of time for the analyst and is the highlight of the tool.

### 3.2.4 Iteration 4: Final

In the final iteration, things like usability, intuitive layout, visibility, safe proofing, and additional useful data points were incorporated to assist the analyst with using the tool. The number of points per graph is automated at this point based on the length of the beam to reduce the computation time while ensuring adequate resolution that discontinuities do not appear on the graphs as artifacts. It is also possible to give the different beams different properties, including E, I, and L.

#### 3.2.4.1 Final User Features

The final tool was designed with usability and convenience in mind.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Instructions | Beam Conditions | | | Multiple Loads | | | |
| 2 | Step 1: Select appropriate Beam Type | Step 1 | Beam Case | Fixed-Fixed | | | | |
| 3 | Step 2: Select Load Case | Step 2 | Load Case | Point Load | Load Case 2 | None | Load Case 3 | None |
| 4 | Step 3: Fill out loading values based on images Below | Step 3 | W | 100 | No Input Required | | No Input Required | |
| 5 | | | a | 3 | No Input Required | | No Input Required | |
| 6 | | | No Input Required | | No Input Required | | No Input Required | |
| 7 | | | No Input Required | | No Input Required | | No Input Required | |
| 8 | Step 4: Fill out Property Table **Note:** You can override the Points value if you want high resolutution (up to 5000) | Step 4 | Property Table | | | | | |
| 9 | | | Material Properties | | Beam Properties | | Graph Properties | |
| 10 | | | E | 200000000 | I | 0.083333333 | Points | 600 |
| 11 | | | | | L | 15 | | |
| 12 | | | | | | | | |
| 13 | Step 5: Click Graph Button | Graph | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |
| 16 | Step 6: For detailed values of V, M, and delta, refer to the calculations Tab | | | | | | | |

**Figure 3-1:** Fields are colored according to how they should be interacted with. Orange is an output value, yellow is an input, light blue are instructions, and grey indicates that nothing should be filled out.

From the very start of using the tool, the bright yellow attracts the user and brings them to a dropdown menu. Conditional formatting is then used to populate the appropriate fields that need to be filled out for a specific load. There are steps located on the left side of the inputs to guide the user, and a large graph button to press when ready to perform the calculations.

There is also an output table for analysts to get quick values that should be useful, as shown in Figure 3-2.

| Outputs | | |
|---|---|---|
| Deflection | Max deflection | Location |
| | -4.08994E-05 | 5.775000095 |
| Shear | Max Shear | Location |
| | 89.59999847 | 0 |
| Moment | Max Moment | Location |
| | -192 | 0 |
| Reactions | Ra | Rb |
| | 89.59999847 | 10.40000153 |
| | Ma | Mb |
| | -192 | 48.00002289 |
| | delta_a | delta_b |
| | 0.00000000 | 0.00000000 |

**Figure 3-2**: This table outputs the maximum deflection, shear, moment, and which locations they occur at. It also outputs the reactions.

Also included is a chart of all of the different load cases and variables that correspond to the appropriate input fields in the top section of the sheet. The chart is shown in Figure 3-3.

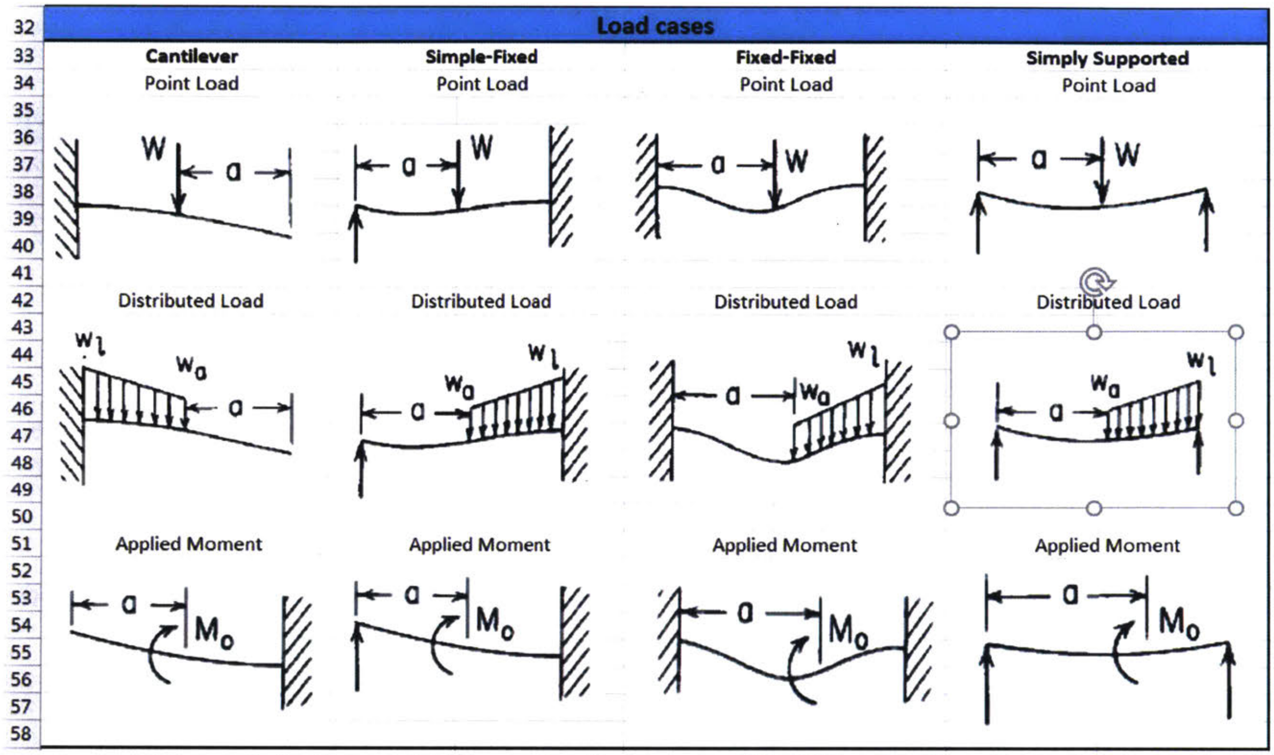



**Figure 3-3:** Chart of different loading types with different edge constraints.

Finally, the tool will output three graphs to be examined by the analyst. Here is an example of the inputs for a multi-span simply supported beam with 6 different loads. Inputs are shown in Figure 3-4, and the following figures 3-5 through 3-7 are the shear, moment, and deflection diagrams.

| Intructions | | Beam Conditions | | Multiple Loads | | | |
|---|---|---|---|---|---|---|---|
| Step 1: Select appropriate Beam Type | Setp 1 | Beam Case | Multi-Span | | | | |
| Step 2: Select Load Cases for Left Beam | Step 2 | Load Case | Point Load | Load Case 2 | Distributed Load | Load Case 3 | Applied Moment |
| Step 3: Fill out loading values based on images below (Row 37) | Step 3 | W | 100 | No Input Required | | No Input Required | |
| | | a | 3 | a | 5 | a | 50 |
| | | No Input Required | | wa | 100 | No Input Required | |
| | | No Input Required | | wl | 200 | No Input Required | |
| Step 4: Select Load Cases for Right Beam | Step 4 | Load Case | Point Load | Load Case 2 | Distributed Load | Load Case 3 | Applied Moment |
| Step 5: Fill out loading values based on images below. Note: a is the distance from the right end of beam, not left end as in step 3 | Step 5 | W | 100 | No Input Required | | No Input Required | |
| | | a | 3 | a | 5 | a | 50 |
| | | No Input Required | | wa | 100 | No Input Required | |
| | | No Input Required | | wl | 200 | No Input Required | |
| Step 6: Fill out Property Table. Note: You can override the Points value if you want high resolutution (up to 5000)- must be even number | Step 6 | Property Table | | | | | |
| | | Material Properties | | Beam Properties | | Graph Properties | |
| | | E1 (Left beam) | 200000000 | I1 | 0.083333333 | Points | |
| | | E2 (Right beam) | 200000000 | I2 | 0.083333333 | | |
| | | | | L1 | 20 | | |
| Step 7: Click Graph Button | | Graph | | L2 | 20 | | |

**Figure 3-4:** Screen shot of example inputs for multispan case.

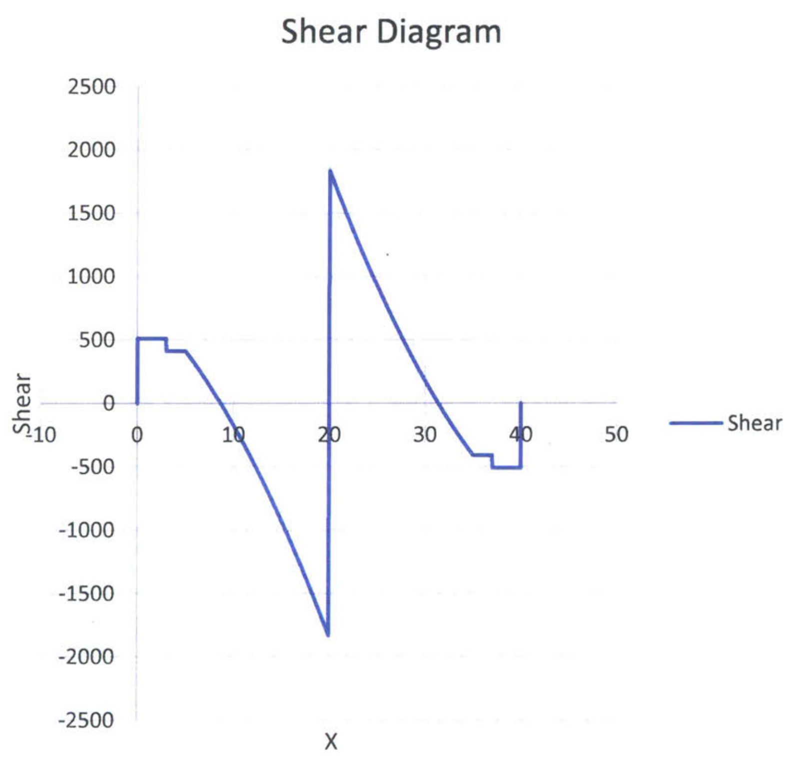



**Figure 3-5:** Shear Diagram for the multi-span beam

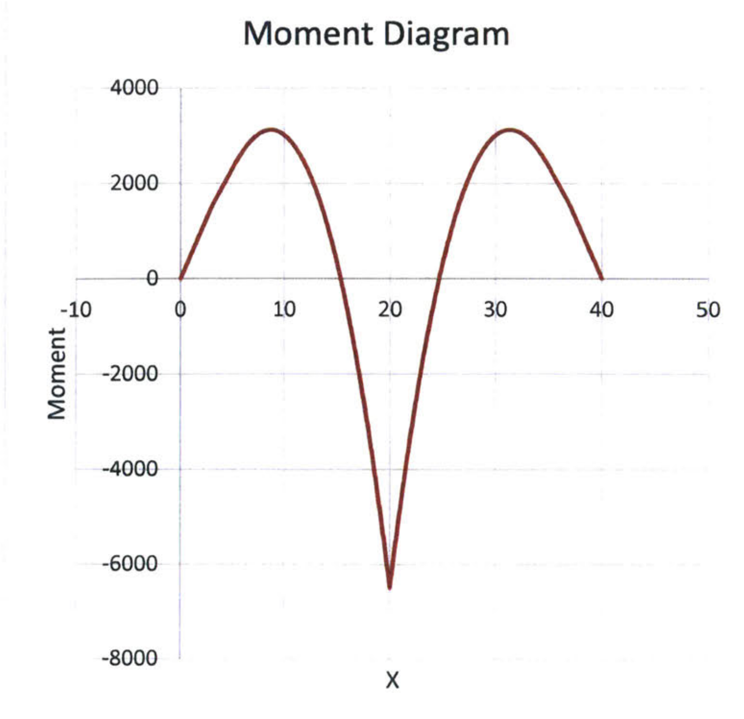



**Figure 3-6:** Moment Diagram for the multispan beam

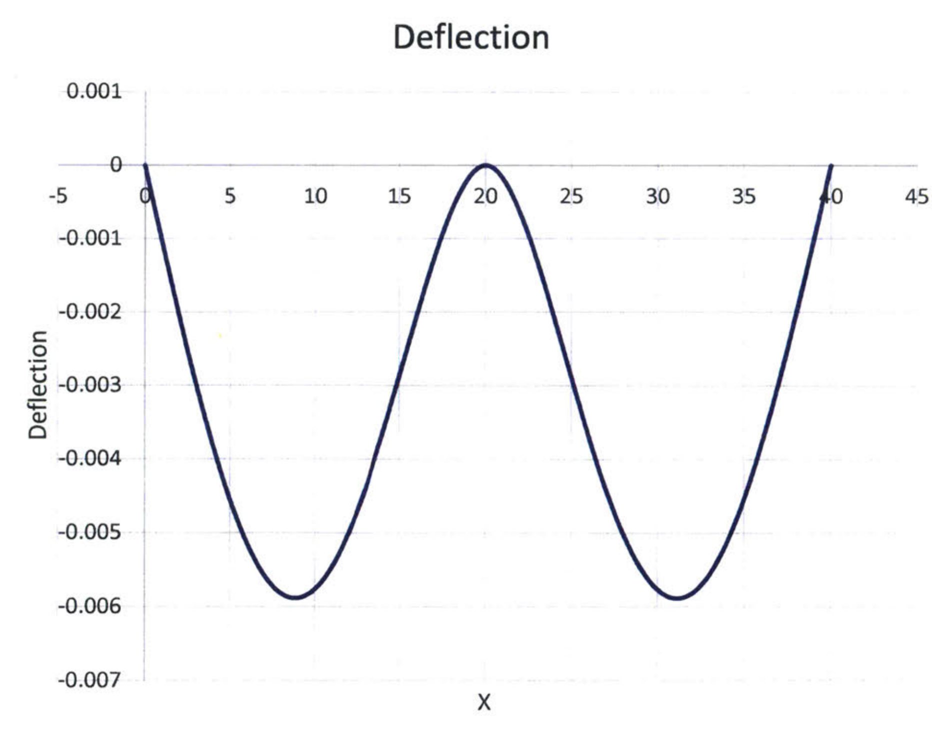



**Figure 3-7:** Deflection Diagram for the multi-span beam

## 4. Future Applications, Improvements, and Conclusions

### 4.1 Education

This tool is extremely useful for engineering students to look through. Not only does this help students taking mechanics classes that are learning the basics of beam bending, but it could be used by aspiring engineers without the calculus background to do these calculations. Upon a visit to my high school, I talked with one of the teachers leading an engineering seminar. She loved the tool and wanted to incorporate something similar to it with her class so that they could focus on learning about safety factors, thinking about how the shape (and thus moment of inertia) impacts the load carrying capabilities of a beam, and explore how real analysts look at engineering problems with this tool.

#### 4.1.1 Improvements for Educational Use

To make it easier for high school and middle school students to use, this tool could easily have a page that explains the concepts of different loads, as well as a built in moment of inertia calculator to give them I to use in their calculations. If this was combined with a solver to pull loads from trusses, it would be the perfect introductory engineering tool to assist with bridge design.

### 4.2 Improvements for Industry

The current tool only allows for up to 3 loads per beam, so a future tool should allow the user to input the number of loads needed and keep that as a variable in the code to run the loop as many times as needed to superimpose the different shear, moments, and deflections from each load.

Another simple feature that may be helpful is an input for geometric constraints. If it is known that a beam cannot displace more than .002 inches in a location, a line could be drawn across the displacement graph to show where the displacement crosses the line and thus does not meet the design requirements. A similar line could be drawn for critical loads on the shear or moment diagram based on any additional fatigue analysis or material properties of the beam.

Further, a built in moment of inertia calculator would help make this tool a little more all-inclusive so other tools don't have to used to calculate moment of inertia.

Additionally, the Reaction Forces that the analyst gets from the tool will not include any load applied directly on the support. Because the beam does not displace, at all if a load is applied directly to a point over a support, the load is not transferred across the beam and not picked up by the code. A stepwise function could be incorporated to fix this issue.

Finally, the computation time was targeted to be under 10 seconds, which does hold true. However, with more VBA experience the code could easily be optimized to reduce computation time.

### 4.3 Conclusions

This tool can be used to improve the efficiency of stress analysts by automatically generating shear, moment, and deflection diagrams for all edge constraints and load types. By solving the three moment equation for every combination of loads and edge constraints, the tool can drastically decrease analysis time for multi-span beams. The tool has applications both in education, as well as with industry. By focusing on usability, this tool can and already has been used by stress analysts across the company with no training.

## 5. Appendices

### A. Single Span Code

```
Sub master_graph()

' Clear Old Data

Sheets("Calculations").Range("A2:W5005").ClearContents

'Load 1 variables
Dim F As Single
Dim L As Single
Dim a As Single
Dim E As Single
Dim I As Single
Dim Points As Single
Dim wa As Single
```

```
Dim wl As Single
Dim Mo As Single

'Arrays
Dim X() As Single
Dim V1() As Single
Dim M1() As Single
Dim d1() As Single
Dim V2() As Single
Dim M2() As Single
Dim d2() As Single
Dim V3() As Single
Dim M3() As Single
Dim d3() As Single

'Load Contsants
Dim Ra As Single
Dim Ma As Single
Dim theta_a As Single
Dim delta_a As Single
Dim EI As Single

'Load 1 Values
F = Cells(4, 4).Value
L = Cells(11, 6).Value
a = Cells(5, 4).Value
E = Cells(10, 4).Value
I = Cells(10, 6).Value
wa = Cells(6, 4).Value
wl = Cells(7, 4).Value
EI = E * I
```

```
Mo = Cells(4, 4).Value


' Setting graph resolution
If L > 150 Then
Points = 4000
ElseIf L < 20 Then
Points = L * 40
ElseIf L < 50 And L >= 20 Then
Points = L * 30
ElseIf L >= 50 And L < 75 Then
Points = L * 20
ElseIf Cells(10, 8).Value > L * 10 Then
Points = Cells(10, 8).Value
Else
Points = 10 * L
End If

'Update Points in doc
Cells(10, 8).Value = Points

BeamType = Cells(2, 4).Text
LoadType = Cells(3, 4).Text


' Set up arrays
If Points > 1 Then
   ReDim Preserve X(1 To (Points + 1))
   ReDim Preserve V1(1 To (Points + 1))
   ReDim Preserve M1(1 To (Points + 1))
   ReDim Preserve d1(1 To (Points + 1))
```

```
    ReDim Preserve V2(1 To (Points + 1))
    ReDim Preserve M2(1 To (Points + 1))
    ReDim Preserve d2(1 To (Points + 1))
    ReDim Preserve V3(1 To (Points + 1))
    ReDim Preserve M3(1 To (Points + 1))
    ReDim Preserve d3(1 To (Points + 1))
End If

'Set up X
For j = 1 To (Points + 1) _

    X(j) = (j - 1) * L / Points
    Worksheets("Calculations").Cells((j + 2), 20) = X(j)

Next j


' Select proper Ra, Ma, theta_a, and delta_a values

If BeamType = "Cantilever" And LoadType = "Point Load" Then _

    Ra = 0
    Ma = 0
    theta_a = F / (2 * EI) * (L - a) ^ 2
    delta_a = -F / (6 * EI) * (2 * L ^ 3 - 3 * a * L ^ 2 + a ^ 3)

ElseIf BeamType = "Cantilever" And LoadType = "Distributed Load" Then _

    Ra = 0
    Ma = 0
    theta_a = (wa / (6 * EI) * (L - a) ^ 3) + (wl - wa) / (24 * EI) * (L - a) ^ 3
```

```
    delta_a = -wa / (24 * EI) * ((L - a) ^ 3) * (3 * L + a) - (wl - wa) / (120 * EI) * ((L - a) ^ 3) * (4 * L + a)

ElseIf BeamType = "Cantilever" And LoadType = "Applied Moment" Then _

    Ra = 0
    Ma = 0
    theta_a = (-Mo * (L - a)) / (EI)
    delta_a = Mo * (L ^ 2 - a ^ 2) / (2 * EI)

ElseIf BeamType = "Fixed-Fixed" And LoadType = "Point Load" Then _

    Ra = F / (L ^ 3) * (L + 2 * a) * (L - a) ^ 2
    Ma = -F * a / (L ^ 2) * (L - a) ^ 2
    theta_a = 0
    delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType = "Distributed Load" Then _

    Ra = wa / (2 * L ^ 3) * ((L - a) ^ 3) * (L + a) + (wl - wa) / (20 * L ^ 3) * ((L - a) ^ 3) * (3 * L + 2 * a)
    Ma = -wa / (12 * L ^ 2) * ((L - a) ^ 3) * (L + 3 * a) - (wl - wa) / (60 * L ^ 2) * ((L - a) ^ 3) * (2 * L + 3 * a)
    theta_a = 0
    delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType = "Applied Moment" Then _

    Ra = -6 * Mo * a / (L ^ 3) * (L - a)
    Ma = -Mo / L ^ 2 * (L ^ 2 - 4 * a * L + 3 * a ^ 2)
    theta_a = 0
```

```
    delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType = "Point Load" Then _

    Ra = F / (2 * L ^ 3) * ((L - a) ^ 2) * (2 * L + a)
    Ma = 0
    theta_a = -F * a / (4 * EI * L) * (L - a) ^ 2
    delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType = "Distributed Load" Then _

    Ra = wa / (8 * L ^ 3) * (3 * L + a) * ((L - a) ^ 3) + (wl - wa) / (40 * L ^ 3) * (4 * L + a) * (L -
a) ^ 3
    Ma = 0
    theta_a = -wa / (48 * EI * L) * (L + 3 * a) * ((L - a) ^ 3) - (wl - wa) / (240 * EI * L) * (2 * L +
3 * a) * (L - a) ^ 3
    delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType = "Applied Moment" Then _

    Ra = -3 * Mo / (2 * L ^ 3) * (L ^ 2 - a ^ 2)
    Ma = 0
    theta_a = Mo * (L - a) * (3 * a - L) / (4 * EI * L)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType = "Point Load" Then _

    Ra = F / L * (L - a)
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
```

```
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType = "Distributed Load" Then _

    Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
    Ma = 0
    theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
    - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType = "Applied Moment" Then _

    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

End If

If BeamType = "Cantilever" Or BeamType = "Simply Supported" Or BeamType = "Fixed-
Fixed" Or BeamType = "Simple-Fixed" Then
'calculations for all beam types

    If LoadType = "Point Load" Then _

        For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then
                V1(j) = Ra
                M1(j) = Ma + Ra * X(j)
                d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
```

```
        Else
          V1(j) = Ra - F * (X(j) - a) ^ 0
          M1(j) = Ma + Ra * X(j) - F * (X(j) - a)
          d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 - F
/ (6 * EI) * (X(j) - a) ^ 3
        End If
      Next j


    ElseIf LoadType = "Distributed Load" Then
' Distributed Load for all beam types
      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then
          V1(j) = Ra
          M1(j) = Ma + Ra * X(j)
          d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
        Else
          V1(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2
          M1(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a) ^ 3
          d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
          - wa / (24 * EI) * (X(j) - a) ^ 4 _
          - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5
        End If
      Next j

    ElseIf LoadType = "Applied Moment" Then
' Moment Load for all beam types
      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then
```

```
            V1(j) = Ra
            M1(j) = Ma + Ra * X(j)
            d1(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI)
        Else
            V1(j) = Ra
            M1(j) = Ma + Ra * X(j) + Mo
            d1(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI) +
Mo / (2 * EI) * (X(j) - a) ^ 2
        End If
    Next j
  End If
End If

' Will run if there is a second Load Case
If Cells(3, 6).Text = "Point Load" Or Cells(3, 6).Text = "Distributed Load" Or Cells(3, 6).Text =
"Applied Moment" Then

  'Update values
  F = Cells(4, 6).Value
  a = Cells(5, 6).Value
  wa = Cells(6, 6).Value
  wl = Cells(7, 6).Value
  Mo = Cells(4, 6).Value
  LoadType2 = Cells(3, 6).Text

'Update loading constants
If BeamType = "Cantilever" And LoadType2 = "Point Load" Then _

  Ra = 0
  Ma = 0
  theta_a = F / (2 * EI) * (L - a) ^ 2
```

```
   delta_a = -F / (6 * EI) * (2 * L ^ 3 - 3 * a * L ^ 2 + a ^ 3)

ElseIf BeamType = "Cantilever" And LoadType2 = "Distributed Load" Then _

   Ra = 0
   Ma = 0
   theta_a = (wa / (6 * EI) * (L - a) ^ 3) + (wl - wa) / (24 * EI) * (L - a) ^ 3
   delta_a = -wa / (24 * EI) * ((L - a) ^ 3) * (3 * L + a) - (wl - wa) / (120 * EI) * ((L - a) ^ 3) * (4
* L + a)

ElseIf BeamType = "Cantilever" And LoadType2 = "Applied Moment" Then _

   Ra = 0
   Ma = 0
   theta_a = (-Mo * (L - a)) / (EI)
   delta_a = Mo * (L ^ 2 - a ^ 2) / (2 * EI)

ElseIf BeamType = "Fixed-Fixed" And LoadType2 = "Point Load" Then _

   Ra = F / (L ^ 3) * (L + 2 * a) * (L - a) ^ 2
   Ma = -F * a / (L ^ 2) * (L - a) ^ 2
   theta_a = 0
   delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType2 = "Distributed Load" Then _

   Ra = wa / (2 * L ^ 3) * ((L - a) ^ 3) * (L + a) + (wl - wa) / (20 * L ^ 3) * ((L - a) ^ 3) * (3 * L
+ 2 * a)
   Ma = -wa / (12 * L ^ 2) * ((L - a) ^ 3) * (L + 3 * a) - (wl - wa) / (60 * L ^ 2) * ((L - a) ^ 3) * (2
* L + 3 * a)
   theta_a = 0
```

```
    delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType2 = "Applied Moment" Then _

    Ra = -6 * Mo * a / (L ^ 3) * (L - a)
    Ma = -Mo / L ^ 2 * (L ^ 2 - 4 * a * L + 3 * a ^ 2)
    theta_a = 0
    delta_a = 0


ElseIf BeamType = "Simple-Fixed" And LoadType2 = "Point Load" Then _

    Ra = F / (2 * L ^ 3) * ((L - a) ^ 2) * (2 * L + a)
    Ma = 0
    theta_a = -F * a / (4 * EI * L) * (L - a) ^ 2
    delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType2 = "Distributed Load" Then _

    Ra = wa / (8 * L ^ 3) * (3 * L + a) * ((L - a) ^ 3) + (wl - wa) / (40 * L ^ 3) * (4 * L + a) * (L -
a) ^ 3
    Ma = 0
    theta_a = -wa / (48 * EI * L) * (L + 3 * a) * ((L - a) ^ 3) - (wl - wa) / (240 * EI * L) * (2 * L +
3 * a) * (L - a) ^ 3
    delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType2 = "Applied Moment" Then _

    Ra = -3 * Mo / (2 * L ^ 3) * (L ^ 2 - a ^ 2)
    Ma = 0
    theta_a = -Mo * (L - a) * (3 * a - L) / (4 * EI * L)
```

```
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType2 = "Point Load" Then _

    Ra = F / L * (L - a)
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType2 = "Distributed Load" Then _

    Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
    Ma = 0
    theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
    - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType2 = "Applied Moment" Then _

    Ra = -Mo / L
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

End If
    If LoadType2 = "Point Load" Then _

        For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then
                V2(j) = Ra
                M2(j) = Ma + Ra * X(j)
```

```
        d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
      Else
        V2(j) = Ra - F * (X(j) - a) ^ 0
        M2(j) = Ma + Ra * X(j) - F * (X(j) - a)
        d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 - F
/ (6 * EI) * (X(j) - a) ^ 3
      End If

  Next j

  ElseIf LoadType2 = "Distributed Load" Then

    For j = 1 To (Points + 1)

      If (X(j) - a) < 0 Then
        V2(j) = Ra
        M2(j) = Ma + Ra * X(j)
        d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
      Else
        V2(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2
        M2(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a) ^ 3
        d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
        - wa / (24 * EI) * (X(j) - a) ^ 4 _
        - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5
         End If

    Next j
  ElseIf LoadType2 = "Applied Moment" Then
' Moment Load for all beam types
    For j = 1 To (Points + 1)
```

```
        If (X(j) - a) < 0 Then
          V2(j) = Ra
          M2(j) = Ma + Ra * X(j)
          d2(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI)
        Else
          V2(j) = Ra
          M2(j) = Ma + Ra * X(j) + Mo
          d2(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI) +
Mo / (2 * EI) * (X(j) - a) ^ 2
        End If
      Next j
    End If


End If

' Will run if there is a third Load Case
If Cells(3, 8).Text = "Point Load" Or Cells(3, 8).Text = "Distributed Load" Or Cells(3, 8).Text =
"Applied Moment" Then

    'Update values
    F = Cells(4, 8).Value
    a = Cells(5, 8).Value
    wa = Cells(6, 8).Value
    wl = Cells(7, 8).Value
    Mo = Cells(4, 8).Value
    LoadType3 = Cells(3, 8).Text

'Update loading constants
If BeamType = "Cantilever" And LoadType3 = "Point Load" Then _
```

```
    Ra = 0
    Ma = 0
    theta_a = F / (2 * EI) * (L - a) ^ 2
    delta_a = -F / (6 * EI) * (2 * L ^ 3 - 3 * a * L ^ 2 + a ^ 3)

ElseIf BeamType = "Cantilever" And LoadType3 = "Distributed Load" Then _

    Ra = 0
    Ma = 0
    theta_a = (wa / (6 * EI) * (L - a) ^ 3) + (wl - wa) / (24 * EI) * (L - a) ^ 3
    delta_a = -wa / (24 * EI) * ((L - a) ^ 3) * (3 * L + a) - (wl - wa) / (120 * EI) * ((L - a) ^ 3) * (4 * L + a)

ElseIf BeamType = "Cantilever" And LoadType3 = "Applied Moment" Then _

    Ra = 0
    Ma = 0
    theta_a = (-Mo * (L - a)) / (EI)
    delta_a = Mo * (L ^ 2 - a ^ 2) / (2 * EI)

ElseIf BeamType = "Fixed-Fixed" And LoadType3 = "Point Load" Then _

    Ra = F / (L ^ 3) * (L + 2 * a) * (L - a) ^ 2
    Ma = -F * a / (L ^ 2) * (L - a) ^ 2
    theta_a = 0
    delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType3 = "Distributed Load" Then _

    Ra = wa / (2 * L ^ 3) * ((L - a) ^ 3) * (L + a) + (wl - wa) / (20 * L ^ 3) * ((L - a) ^ 3) * (3 * L + 2 * a)
```

```
   Ma = -wa / (12 * L ^ 2) * ((L - a) ^ 3) * (L + 3 * a) - (wl - wa) / (60 * L ^ 2) * ((L - a) ^ 3) * (2
* L + 3 * a)
   theta_a = 0
   delta_a = 0

ElseIf BeamType = "Fixed-Fixed" And LoadType3 = "Applied Moment" Then _

   Ra = -6 * Mo * a / (L ^ 3) * (L - a)
   Ma = -Mo / L ^ 2 * (L ^ 2 - 4 * a * L + 3 * a ^ 2)
   theta_a = 0
   delta_a = 0


ElseIf BeamType = "Simple-Fixed" And LoadType3 = "Point Load" Then _

   Ra = F / (2 * L ^ 3) * ((L - a) ^ 2) * (2 * L + a)
   Ma = 0
   theta_a = -F * a / (4 * EI * L) * (L - a) ^ 2
   delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType3 = "Distributed Load" Then _

   Ra = wa / (8 * L ^ 3) * (3 * L + a) * ((L - a) ^ 3) + (wl - wa) / (40 * L ^ 3) * (4 * L + a) * (L -
a) ^ 3
   Ma = 0
   theta_a = -wa / (48 * EI * L) * (L + 3 * a) * ((L - a) ^ 3) - (wl - wa) / (240 * EI * L) * (2 * L +
3 * a) * (L - a) ^ 3
   delta_a = 0

ElseIf BeamType = "Simple-Fixed" And LoadType3 = "Applied Moment" Then _
```

```
    Ra = -3 * Mo / (2 * L ^ 3) * (L ^ 2 - a ^ 2)
    Ma = 0
    theta_a = -Mo * (L - a) * (3 * a - L) / (4 * EI * L)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType3 = "Point Load" Then _

    Ra = F / L * (L - a)
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType3 = "Distributed Load" Then _

    Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
    Ma = 0
    theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
    - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
    delta_a = 0

ElseIf BeamType = "Simply Supported" And LoadType3 = "Applied Moment" Then _

    Ra = -Mo / L
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

End If
    If LoadType3 = "Point Load" Then _

        For j = 1 To (Points + 1)
```

```
        If (X(j) - a) < 0 Then
            V3(j) = Ra
            M3(j) = Ma + Ra * X(j)
            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
        Else
            V3(j) = Ra - F * (X(j) - a) ^ 0
            M3(j) = Ma + Ra * X(j) - F * (X(j) - a)
            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 - F
/ (6 * EI) * (X(j) - a) ^ 3
        End If

    Next j

    ElseIf LoadType3 = "Distributed Load" Then

      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then
            V3(j) = Ra
            M3(j) = Ma + Ra * X(j)
            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3
        Else
            V3(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2
            M3(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a) ^ 3
            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5
              End If

      Next j
    ElseIf LoadType3 = "Applied Moment" Then
```

```
' Moment Load for all beam types
    For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then
            V3(j) = Ra
            M3(j) = Ma + Ra * X(j)
            d3(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI)
        Else
            V3(j) = Ra
            M3(j) = Ma + Ra * X(j) + Mo
            d3(j) = delta_a + theta_a * X(j) + Ma * X(j) ^ 2 / (2 * EI) + Ra * X(j) ^ 3 / (6 * EI) +
Mo / (2 * EI) * (X(j) - a) ^ 2
        End If
    Next j
  End If


End If


If BeamType = "Cantilever" Then _
' Cantilever beam calculations were done with reflected coordinated of those shown in image-
this corrects that
  For j = 1 To (Points + 1)
    ' Used for Graph
    Worksheets("Calculations").Cells((Points + 4 - j), 21) = (-V1(j) - V2(j) - V3(j))
    Worksheets("Calculations").Cells((Points + 4 - j), 22) = (M1(j) + M2(j) + M3(j))
    Worksheets("Calculations").Cells((Points + 4 - j), 23) = (d1(j) + d2(j) + d3(j))
    'Used for reference
    Worksheets("Calculations").Cells((j + 1), 1) = X(j)
    Worksheets("Calculations").Cells((Points + 3 - j), 2) = (-V1(j) - V2(j) - V3(j))
```

```
        Worksheets("Calculations").Cells((Points + 3 - j), 3) = (M1(j) + M2(j) + M3(j))
        Worksheets("Calculations").Cells((Points + 3 - j), 4) = (d1(j) + d2(j) + d3(j))
    Next j
Else
'For everything else
    For j = 1 To (Points + 1)
        Worksheets("Calculations").Cells((j + 2), 21) = (V1(j) + V2(j) + V3(j))
        Worksheets("Calculations").Cells((j + 2), 22) = (M1(j) + M2(j) + M3(j))
        Worksheets("Calculations").Cells((j + 2), 23) = (d1(j) + d2(j) + d3(j))
        Worksheets("Calculations").Cells((j + 1), 1) = X(j)
        Worksheets("Calculations").Cells((j + 1), 2) = (V1(j) + V2(j) + V3(j))
        Worksheets("Calculations").Cells((j + 1), 3) = (M1(j) + M2(j) + M3(j))
        Worksheets("Calculations").Cells((j + 1), 4) = (d1(j) + d2(j) + d3(j))
    Next j

End If

' Placing arbitrary points to show reaction forces at Boundary

Worksheets("Calculations").Cells(2, 20) = -0.000000000000001
Worksheets("Calculations").Cells(2, 21) = 0
Worksheets("Calculations").Cells(2, 22) = 0
Worksheets("Calculations").Cells(2, 23) = 0
Worksheets("Calculations").Cells((Points + 4), 20) = (L + 0.00000000000001)
Worksheets("Calculations").Cells((Points + 4), 21) = 0
Worksheets("Calculations").Cells((Points + 4), 22) = 0

' Update outputs

Cells(25, 4).Value = -1 * (Worksheets("Calculations").Cells(Points + 2, 2).Value)
Cells(27, 4).Value = -1 * (Worksheets("Calculations").Cells(Points + 2, 3).Value)
```

```
Cells(29, 4).Value = (Worksheets("Calculations").Cells(Points + 2, 4).Value)

End Sub
```

**B.** Multi-Span Code

```
Sub Two_side_Multi_Span_2EI_w_moments()

' Clear Old Data

Sheets("Calculations").Range("A2:W5005").ClearContents

'Load 1 variables
Dim F As Single
Dim L As Single
Dim a As Single
Dim E As Single
Dim E2 As Single
Dim I As Single
Dim I2 As Single
Dim Points As Single
```

```
Dim wa As Single
Dim wl As Single


'Arrays
Dim X() As Single
Dim V1() As Single
Dim M1() As Single
Dim d1() As Single
Dim V2() As Single
Dim M2() As Single
Dim d2() As Single
Dim V3() As Single
Dim M3() As Single
Dim d3() As Single
Dim V4() As Single
Dim M4() As Single
Dim d4() As Single
Dim V5() As Single
Dim M5() As Single
Dim d5() As Single
Dim V6() As Single
Dim M6() As Single
Dim d6() As Single


'Load Contsants
Dim Ra As Single
Dim Ma As Single
Dim theta_a As Single
Dim delta_a As Single
```

```
Dim EI As Single
Dim EI2 As Single
Dim R__l_moment As Single
Dim theta_l_moment As Single
Dim M_l_moment As Single
Dim delta_l_moment As Single
Dim R__r_moment As Single
Dim theta_r_moment As Single
Dim M_r_moment As Single
Dim delta_r_moment As Single
Dim Mo_L As Single
Dim Mo_r As Single
Dim Mo As Single
'Logic
Dim BeamType As String
Dim LoadType As String
Dim LoadType2 As String
Dim LoadType3 As String
Dim LoadType4 As String
Dim LoadType5 As String
Dim LoadType6 As String



'Begin Analysis for Left side'
'Load 1 Values
F = Cells(4, 4).Value
L = Cells(17, 6).Value
L2 = Cells(18, 6).Value
a = Cells(5, 4).Value
E = Cells(15, 4).Value
```

```
E2 = Cells(16, 4).Value
I = Cells(15, 6).Value
I2 = Cells(16, 6).Value
Points = Cells(15, 8).Value
wa = Cells(6, 4).Value
wl = Cells(7, 4).Value
EI = E * I
EI2 = E2 * I2
Mo = Cells(4, 4).Value


BeamType = Cells(2, 4).Text
LoadType = Cells(3, 4).Text

'Setting Graph resolution

If (L + L2) > 150 Then
Points = 4000
ElseIf (L + L2) < 20 Then
Points = (L + L2) * 40
ElseIf (L + L2) < 50 And (L + L2) >= 20 Then
Points = (L + L2) * 30
ElseIf (L + L2) >= 50 And (L + L2) < 75 Then
Points = (L + L2) * 20
ElseIf Cells(15, 8).Value > (L + L2) * 10 Then
Points = Cells(15, 8).Value
Else
Points = 10 * (L + L2)
End If

' Set up arrays
```

```
If Points > 1 Then
    ReDim Preserve X(1 To (Points + 1))
    ReDim Preserve V1(1 To (Points + 1))
    ReDim Preserve M1(1 To (Points + 1))
    ReDim Preserve d1(1 To (Points + 1))
    ReDim Preserve V2(1 To (Points + 1))
    ReDim Preserve M2(1 To (Points + 1))
    ReDim Preserve d2(1 To (Points + 1))
    ReDim Preserve V3(1 To (Points + 1))
    ReDim Preserve M3(1 To (Points + 1))
    ReDim Preserve d3(1 To (Points + 1))
    ReDim Preserve V4(1 To (Points + 1))
    ReDim Preserve M4(1 To (Points + 1))
    ReDim Preserve d4(1 To (Points + 1))
    ReDim Preserve V5(1 To (Points + 1))
    ReDim Preserve M5(1 To (Points + 1))
    ReDim Preserve d5(1 To (Points + 1))
    ReDim Preserve V6(1 To (Points + 1))
    ReDim Preserve M6(1 To (Points + 1))
    ReDim Preserve d6(1 To (Points + 1))
End If

'Set up X
For j = 1 To (Points + 1) _

    X(j) = (j - 1) * (L + L2) / Points
    Worksheets("Calculations").Cells((j + 2), 20) = X(j)

Next j
```

```
' Select proper Ra, Ma, theta_a, and delta_a values

If BeamType = "Multi-Span" And LoadType = "Point Load" Then _

    Ra = F / L * (L - a)
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
    delta_a = 0

    Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
    * (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType = "Distributed Load" Then _

    Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
    Ma = 0
    theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
    - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
```

```
    delta_a = 0

    Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 * (8 *
L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
    * ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType = "Applied Moment" Then
    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

    Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L) *
(L ^ 2 - 3 * a ^ 2)
    Mo_r = -Mo_L


    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
```

```
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

End If

If BeamType = "Multi-Span" Then
'calculations multispan

    If LoadType = "Point Load" Then _

        For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then

                V1(j) = Ra _
                + R_l_moment

                M1(j) = Ma + Ra * X(j) _
                + M_l_moment + R_l_moment * (X(j))

                d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
                + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
```

```
        ElseIf X(j) >= a And X(j) < L Then

            V1(j) = Ra - F _
            + R_l_moment

            M1(j) = Ma + Ra * X(j) - F * (X(j) - a) _
            + M_l_moment + R_l_moment * (X(j))

            d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 - F
/ (6 * EI) * (X(j) - a) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        Else

            V1(j) = R_r_moment

            M1(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d1(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L) ^ 2
/ (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

        End If
    Next j

  ElseIf LoadType = "Distributed Load" Then
' Distributed Load
    For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

            V1(j) = Ra _
```

```
            + R_l_moment

            M1(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        ElseIf X(j) >= a And X(j) <= L Then

            V1(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
            + R_l_moment

            M1(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a) ^ 3
_
            + M_l_moment + R_l_moment * (X(j))

            d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) > L Then

            V1(j) = R_r_moment

            M1(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d1(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L) ^ 2
/ (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2
```

```
            End If

        Next j

    ElseIf LoadType = "Applied Moment" Then

' Distributed Load
        For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then

                V1(j) = Ra _
                 + R_l_moment

                M1(j) = Ma + Ra * X(j) _
                + M_l_moment + R_l_moment * (X(j))

                d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
                + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
            ElseIf X(j) >= a And X(j) < L Then

                V1(j) = Ra + R_l_moment

                M1(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

                d1(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
                + Mo / (2 * EI) * (X(j) - a) ^ 2 _
```

```
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V1(j) = R_r_moment

            M1(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d1(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L) ^ 2
/ (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


        End If

      Next j

    End If
End If

' Will run if there is a second Load Case
If Cells(3, 6).Text = "Point Load" Or Cells(3, 6).Text = "Distributed Load" Then

    'Update values
    F = Cells(4, 6).Value
    a = Cells(5, 6).Value
    wa = Cells(6, 6).Value
    wl = Cells(7, 6).Value
    Mo = Cells(4, 6).Value

    LoadType2 = Cells(3, 6).Text
```

```
'Update loading constants
If BeamType = "Multi-Span" And LoadType2 = "Point Load" Then _

    Ra = F / L * (L - a)
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
    delta_a = 0

    Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
    * (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType2 = "Distributed Load" Then _

    Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
    Ma = 0
    theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
    - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
    delta_a = 0
```

```
    Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 *
(8 * L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
    * ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0


  ElseIf BeamType = "Multi-Span" And LoadType2 = "Applied Moment" Then
    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

    Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L)
* (L ^ 2 - 3 * a ^ 2)
    Mo_r = -Mo_L


    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0
```

```
    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

  End If

  If BeamType = "Multi-Span" Then
  'calculations multispan

    If LoadType2 = "Point Load" Then _

      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

          V2(j) = Ra _
          + R_l_moment

          M2(j) = Ma + Ra * X(j) _
          + M_l_moment + R_l_moment * (X(j))

          d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
          + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= a And X(j) < L Then

          V2(j) = Ra - F _
          + R_l_moment
```

```
                M2(j) = Ma + Ra * X(j) - F * (X(j) - a) _
                + M_l_moment + R_l_moment * (X(j))

                d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 -
F / (6 * EI) * (X(j) - a) ^ 3 _
                + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
            Else

                V2(j) = R_r_moment

                M2(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

                d2(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

            End If
        Next j

      ElseIf LoadType2 = "Distributed Load" Then
   ' Distributed Load
        For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then

                V2(j) = Ra _
                 + R_l_moment

                M2(j) = Ma + Ra * X(j) _
                + M_l_moment + R_l_moment * (X(j))
```

```
            d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        ElseIf X(j) >= a And X(j) < L Then

            V2(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
            + R_l_moment

            M2(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a)
^ 3 _
            + M_l_moment + R_l_moment * (X(j))

            d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V2(j) = R_r_moment

            M2(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d2(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


        End If
```

```
        Next j

      ElseIf LoadType2 = "Applied Moment" Then

    ' Applied Moment Load
        For j = 1 To (Points + 1)

          If (X(j) - a) < 0 Then

            V2(j) = Ra _
             + R_l_moment

            M2(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
          ElseIf X(j) >= a And X(j) < L Then

            V2(j) = Ra + R_l_moment

            M2(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

            d2(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + Mo / (2 * EI) * (X(j) - a) ^ 2 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

          ElseIf X(j) >= L Then
```

```
            V2(j) = R_r_moment

            M2(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d2(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


          End If

        Next j
      End If
    End If
End If

' Will run if there is a third Load Case
If Cells(3, 8).Text = "Point Load" Or Cells(3, 8).Text = "Distributed Load" Then


    'Update values
    F = Cells(4, 8).Value
    a = Cells(5, 8).Value
    wa = Cells(6, 8).Value
    wl = Cells(7, 8).Value

    LoadType3 = Cells(3, 8).Text

'Update loading constants
    If BeamType = "Multi-Span" And LoadType3 = "Point Load" Then _

    Ra = F / L * (L - a)
```

```
    Ma = 0
    theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
    delta_a = 0

    Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
    * (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType3 = "Distributed Load" Then _

        Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
        Ma = 0
        theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
        - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
        delta_a = 0

        Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 *
(8 * L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
        * ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
        Mo_r = -Mo_L
```

```
    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

  ElseIf BeamType = "Multi-Span" And LoadType3 = "Applied Moment" Then
    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

    Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L)
* (L ^ 2 - 3 * a ^ 2)
    Mo_r = -Mo_L


    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0
```

```
   End If

If BeamType = "Multi-Span" Then
'calculations multispan

   If LoadType3 = "Point Load" Then _

      For j = 1 To (Points + 1)

         If (X(j) - a) < 0 Then

            V3(j) = Ra _
            + R_l_moment

            M3(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

         ElseIf X(j) >= a And X(j) < L Then

            V3(j) = Ra - F _
            + R_l_moment

            M3(j) = Ma + Ra * X(j) - F * (X(j) - a) _
            + M_l_moment + R_l_moment * (X(j))

            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 - F
/ (6 * EI) * (X(j) - a) ^ 3 _
```

```
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        Else

            V3(j) = R_r_moment

            M3(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d3(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L) ^ 2
/ (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

        End If
      Next j

   ElseIf LoadType3 = "Distributed Load" Then
' Distributed Load
      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

            V3(j) = Ra _
             + R_l_moment

            M3(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        ElseIf X(j) >= a And X(j) < L Then
```

```
            V3(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
            + R_l_moment

            M3(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a) ^ 3
_
            + M_l_moment + R_l_moment * (X(j))

            d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V3(j) = R_r_moment

            M3(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d3(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L) ^ 2
/ (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


        End If

    Next j
    ElseIf LoadType3 = "Applied Moment" Then

  ' Applied Moment Load
        For j = 1 To (Points + 1)
```

```
If (X(j) - a) < 0 Then

    V3(j) = Ra _
     + R_l_moment

    M3(j) = Ma + Ra * X(j) _
    + M_l_moment + R_l_moment * (X(j))

    d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
    + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
ElseIf X(j) >= a And X(j) < L Then

    V3(j) = Ra + R_l_moment

    M3(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

    d3(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
    + Mo / (2 * EI) * (X(j) - a) ^ 2 _
    + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

ElseIf X(j) >= L Then

    V3(j) = R_r_moment

    M3(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

    d3(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2
```

```
            End If

        Next j

    End If

End If
End If
' End Left Beam Analysis
'Begin Right Beam Analysis
'Be sure that 'a' is distrance from right end of beam, rather than the left beam
'This script simply runs the same as left side analysis, and will reflect across the middle span and multiply by negative one to add to the other loads

' Note that I switched L and L2 here intentionally.  That way when the arrays are inverted later on, everything will match up!
L = Cells(18, 6).Value
L2 = Cells(17, 6).Value

If Cells(8, 4).Text = "Point Load" Or Cells(8, 4).Text = "Distributed Load" Then

    'Update values
    F = Cells(9, 4).Value
    a = Cells(10, 4).Value
    wa = Cells(11, 4).Value
    wl = Cells(12, 4).Value


    LoadType4 = Cells(8, 4).Text
```

```
'Update loading constants
If BeamType = "Multi-Span" And LoadType4 = "Point Load" Then _

Ra = F / L * (L - a)
Ma = 0
theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
delta_a = 0

Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
* (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
Mo_r = -Mo_L

R_l_moment = -Mo_L / L
M_l_moment = 0
theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
delta_l_moment = 0

R_r_moment = -Mo_r / L2
M_r_moment = 0
theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType4 = "Distributed Load" Then _

  Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
  Ma = 0
  theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
  - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
  delta_a = 0
```

```
    Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 *
(8 * L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
    * ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
    Mo_r = -Mo_L

    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

  ElseIf BeamType = "Multi-Span" And LoadType4 = "Applied Moment" Then
    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

    Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L)
* (L ^ 2 - 3 * a ^ 2)
    Mo_r = -Mo_L


    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0
```

```
    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

    End If

  If BeamType = "Multi-Span" Then
  'calculations multispan

    If LoadType4 = "Point Load" Then _

      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

          V4(j) = Ra _
          + R_l_moment

          M4(j) = Ma + Ra * X(j) _
          + M_l_moment + R_l_moment * (X(j))

          d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
          + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= a And X(j) < L Then

          V4(j) = Ra - F _
          + R_l_moment
```

```
            M4(j) = Ma + Ra * X(j) - F * (X(j) - a) _
            + M_l_moment + R_l_moment * (X(j))

            d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 -
F / (6 * EI) * (X(j) - a) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
          Else

            V4(j) = R_r_moment

            M4(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d4(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

          End If
        Next j

      ElseIf LoadType4 = "Distributed Load" Then
    ' Distributed Load
        For j = 1 To (Points + 1)

          If (X(j) - a) < 0 Then

            V4(j) = Ra _
             + R_l_moment

            M4(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))
```

```
            d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        ElseIf X(j) >= a And X(j) < L Then

            V4(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
            + R_l_moment

            M4(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a)
^ 3 _
            + M_l_moment + R_l_moment * (X(j))

            d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V4(j) = R_r_moment

            M4(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d4(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


        End If

    Next j
```

```
ElseIf LoadType4 = "Applied Moment" Then

' Applied Moment Load
    For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

            V4(j) = Ra _
             + R_l_moment

            M4(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        ElseIf X(j) >= a And X(j) < L Then

            V4(j) = Ra + R_l_moment

            M4(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

            d4(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + Mo / (2 * EI) * (X(j) - a) ^ 2 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V4(j) = R_r_moment
```

```
            M4(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d4(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


          End If

        Next j

      End If

    End If
End If

' Will run if there is a 5th Load Case (second load on right beam)
If Cells(8, 6).Text = "Point Load" Or Cells(8, 6).Text = "Distributed Load" Then


    'Update values
    F = Cells(9, 6).Value
    a = Cells(10, 6).Value
    wa = Cells(11, 6).Value
    wl = Cells(12, 6).Value


    LoadType5 = Cells(8, 6).Text

    If BeamType = "Multi-Span" And LoadType5 = "Point Load" Then _
```

```
        Ra = F / L * (L - a)
        Ma = 0
        theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
        delta_a = 0

        Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
        * (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
        Mo_r = -Mo_L

        R_l_moment = -Mo_L / L
        M_l_moment = 0
        theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
        delta_l_moment = 0

        R_r_moment = -Mo_r / L2
        M_r_moment = 0
        theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
        delta_r_moment = 0

    ElseIf BeamType = "Multi-Span" And LoadType5 = "Distributed Load" Then _

        Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
        Ma = 0
        theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
        - (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
        delta_a = 0

        Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 *
(8 * L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
        * ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
        Mo_r = -Mo_L
```

```
        R_l_moment = -Mo_L / L
        M_l_moment = 0
        theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
        delta_l_moment = 0

        R_r_moment = -Mo_r / L2
        M_r_moment = 0
        theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
        delta_r_moment = 0

    ElseIf BeamType = "Multi-Span" And LoadType5 = "Applied Moment" Then
        Ra = -Mo / L
        Ma = 0
        theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
        delta_a = 0

        Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L)
* (L ^ 2 - 3 * a ^ 2)
        Mo_r = -Mo_L


        R_l_moment = -Mo_L / L
        M_l_moment = 0
        theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
        delta_l_moment = 0

        R_r_moment = -Mo_r / L2
        M_r_moment = 0
        theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
        delta_r_moment = 0
```

```
End If

If BeamType = "Multi-Span" Then
'calculations multispan

 If LoadType5 = "Point Load" Then _

  For j = 1 To (Points + 1)

   If (X(j) - a) < 0 Then

    V5(j) = Ra _
    + R_l_moment

    M5(j) = Ma + Ra * X(j) _
    + M_l_moment + R_l_moment * (X(j))

    d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
    + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

   ElseIf X(j) >= a And X(j) < L Then

    V5(j) = Ra - F _
    + R_l_moment

    M5(j) = Ma + Ra * X(j) - F * (X(j) - a) _
    + M_l_moment + R_l_moment * (X(j))
```

```
            d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 -
F / (6 * EI) * (X(j) - a) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
        Else

            V5(j) = R_r_moment

            M5(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d5(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

        End If
      Next j

    ElseIf LoadType5 = "Distributed Load" Then
  ' Distributed Load
      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then

            V5(j) = Ra _
             + R_l_moment

            M5(j) = Ma + Ra * X(j) _
            + M_l_moment + R_l_moment * (X(j))

            d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
```

```
            ElseIf X(j) >= a And X(j) < L Then

                V5(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
                + R_l_moment

                M5(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a)
^ 3 _
                + M_l_moment + R_l_moment * (X(j))

                d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
                - wa / (24 * EI) * (X(j) - a) ^ 4 _
                - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
                + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

            ElseIf X(j) >= L Then

                V5(j) = R_r_moment

                M5(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

                d5(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


            End If

        Next j
    ElseIf LoadType5 = "Applied Moment" Then

  ' Applied Moment Load
```

```
For j = 1 To (Points + 1)

    If (X(j) - a) < 0 Then

        V5(j) = Ra _
         + R_l_moment

        M5(j) = Ma + Ra * X(j) _
        + M_l_moment + R_l_moment * (X(j))

        d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
        + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
    ElseIf X(j) >= a And X(j) < L Then

        V5(j) = Ra + R_l_moment

        M5(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

        d5(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
        + Mo / (2 * EI) * (X(j) - a) ^ 2 _
        + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

    ElseIf X(j) >= L Then

        V5(j) = R_r_moment

        M5(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r
```

```
            d5(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


          End If

        Next j

      End If

   End If
End If

' Will run if there is a 6th Load Case (3rd load on right beam)
If Cells(8, 8).Text = "Point Load" Or Cells(8, 8).Text = "Distributed Load" Then


'Update loading constants
   F = Cells(9, 8).Value
   a = Cells(10, 8).Value
   wa = Cells(11, 8).Value
   wl = Cells(12, 8).Value


   LoadType6 = Cells(8, 8).Text

   If BeamType = "Multi-Span" And LoadType6 = "Point Load" Then _

      Ra = F / L * (L - a)
      Ma = 0
      theta_a = -F * a / (6 * EI * L) * (2 * L - a) * (L - a)
```

```
delta_a = 0

Mo_L = (F * a * (L ^ 2 - a ^ 2) / (6 * EI * L)) _
* (2 * L2 ^ 2 / (6 * EI2 * L2) - (L ^ 2 - 3 * L ^ 2) / (6 * EI * L)) ^ (-1)
Mo_r = -Mo_L

R_l_moment = -Mo_L / L
M_l_moment = 0
theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
delta_l_moment = 0

R_r_moment = -Mo_r / L2
M_r_moment = 0
theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
delta_r_moment = 0

ElseIf BeamType = "Multi-Span" And LoadType6 = "Distributed Load" Then _

Ra = wa / (2 * L) * ((L - a) ^ 2) + (wl - wa) / (6 * L) * (L - a) ^ 2
Ma = 0
theta_a = -wa / (24 * EI * L) * ((L - a) ^ 2) * ((L ^ 2) + 2 * a * L - a ^ 2) _
- (wl - wa) / (360 * EI * L) * ((L - a) ^ 2) * (7 * (L ^ 2) + 6 * a * L - 3 * a ^ 2)
delta_a = 0

Mo_L = (wa * (L ^ 2 - a ^ 2) ^ 2 / (24 * EI * L) + (wl - wa) / (360 * EI * L) * (L - a) ^ 2 *
(8 * L ^ 2 + 9 * a * L + 3 * a ^ 2)) _
* ((2 * L2 ^ 2) / (6 * EI2 * L2) + (2 * L ^ 2) / (6 * EI * L)) ^ (-1)
Mo_r = -Mo_L

R_l_moment = -Mo_L / L
M_l_moment = 0
```

```
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

  ElseIf BeamType = "Multi-Span" And LoadType6 = "Applied Moment" Then
    Ra = -Mo / L
    Ma = 0
    theta_a = -Mo * (2 * L ^ 2 - 6 * a * L + 3 * a ^ 2) / (6 * EI * L)
    delta_a = 0

    Mo_L = ((-2 * L ^ 2) / (6 * EI * L) - (2 * L2 ^ 2) / (6 * EI2 * L2)) ^ -1 * -Mo / (6 * EI * L)
* (L ^ 2 - 3 * a ^ 2)
    Mo_r = -Mo_L



    R_l_moment = -Mo_L / L
    M_l_moment = 0
    theta_l_moment = -Mo_L / (6 * EI * L) * (2 * L ^ 2 - 6 * L ^ 2 + 3 * L ^ 2)
    delta_l_moment = 0

    R_r_moment = -Mo_r / L2
    M_r_moment = 0
    theta_r_moment = -Mo_r / (6 * EI2 * L2) * (2 * L2 ^ 2)
    delta_r_moment = 0

  End If
```

```
If BeamType = "Multi-Span" Then
'calculations multispan

  If LoadType6 = "Point Load" Then _

    For j = 1 To (Points + 1)

      If (X(j) - a) < 0 Then

        V6(j) = Ra _
        + R_l_moment

        M6(j) = Ma + Ra * X(j) _
        + M_l_moment + R_l_moment * (X(j))

        d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
        + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

      ElseIf X(j) >= a And X(j) < L Then

        V6(j) = Ra - F _
        + R_l_moment

        M6(j) = Ma + Ra * X(j) - F * (X(j) - a) _
        + M_l_moment + R_l_moment * (X(j))

        d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 -
F / (6 * EI) * (X(j) - a) ^ 3 _
        + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
```

```
            Else

              V6(j) = R_r_moment

              M6(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

              d6(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2

            End If
          Next j

        ElseIf LoadType6 = "Distributed Load" Then
    ' Distributed Load
          For j = 1 To (Points + 1)

            If (X(j) - a) < 0 Then

              V6(j) = Ra _
               + R_l_moment

              M6(j) = Ma + Ra * X(j) _
              + M_l_moment + R_l_moment * (X(j))

              d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
              + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
            ElseIf X(j) >= a And X(j) < L Then

              V6(j) = Ra - wa * (X(j) - a) - (wl - wa) / (2 * (L - a)) * (X(j) - a) ^ 2 _
              + R_l_moment
```

```
            M6(j) = Ma + Ra * X(j) - wa / 2 * (X(j) - a) ^ 2 - (wl - wa) / (6 * (L - a)) * (X(j) - a)
^ 3 _
            + M_l_moment + R_l_moment * (X(j))

            d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
            - wa / (24 * EI) * (X(j) - a) ^ 4 _
            - (wl - wa) / (120 * EI * (L - a)) * (X(j) - a) ^ 5 _
            + delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

        ElseIf X(j) >= L Then

            V6(j) = R_r_moment

            M6(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

            d6(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2


        End If

      Next j

    ElseIf LoadType6 = "Applied Moment" Then

  ' Applied Moment Load
      For j = 1 To (Points + 1)

        If (X(j) - a) < 0 Then
```

```
V6(j) = Ra _
 + R_l_moment

M6(j) = Ma + Ra * X(j) _
+ M_l_moment + R_l_moment * (X(j))

d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
+ delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)
ElseIf X(j) >= a And X(j) < L Then

V6(j) = Ra + R_l_moment

M6(j) = Ma + Ra * X(j) + Mo + M_l_moment + R_l_moment * (X(j))

d6(j) = delta_a + theta_a * X(j) + Ma / (2 * EI) * X(j) ^ 2 + Ra / (6 * EI) * X(j) ^ 3 _
+ Mo / (2 * EI) * (X(j) - a) ^ 2 _
+ delta_l_moment + theta_l_moment * X(j) + M_l_moment * X(j) ^ 2 / (2 * EI) +
R_l_moment * X(j) ^ 3 / (6 * EI)

ElseIf X(j) >= L Then

V6(j) = R_r_moment

M6(j) = M_r_moment + R_r_moment * (X(j) - L) + Mo_r

d6(j) = delta_r_moment + theta_r_moment * (X(j) - L) + M_r_moment * (X(j) - L)
^ 2 / (2 * EI2) + R_r_moment * (X(j) - L) ^ 3 / (6 * EI2) + Mo_r / (2 * EI2) * (X(j) - L) ^ 2
```

```
            End If

          Next j

        End If

      End If
    End If

      For j = 1 To (Points + 1)
        Worksheets("Calculations").Cells((j + 2), 21) = (V1(j) + V2(j) + V3(j) - V4(Points + 2 - j) -
    V5(Points + 2 - j) - V6(Points + 2 - j))
        Worksheets("Calculations").Cells((j + 2), 22) = (M1(j) + M2(j) + M3(j) + M4(Points + 2 -
    j) + M5(Points + 2 - j) + M6(Points + 2 - j))
        Worksheets("Calculations").Cells((j + 2), 23) = (d1(j) + d2(j) + d3(j)) + d4(Points + 2 - j) +
    d5(Points + 2 - j) + d6(Points + 2 - j)
        Worksheets("Calculations").Cells((j + 1), 1) = X(j)
        Worksheets("Calculations").Cells((j + 1), 2) = (V1(j) + V2(j) + V3(j) - V4(Points + 2 - j) -
    V5(Points + 2 - j) - V6(Points + 2 - j))
        Worksheets("Calculations").Cells((j + 1), 3) = (M1(j) + M2(j) + M3(j) + M4(Points + 2 - j)
    + M5(Points + 2 - j) + M6(Points + 2 - j))
        Worksheets("Calculations").Cells((j + 1), 4) = (d1(j) + d2(j) + d3(j) + d4(Points + 2 - j) +
    d5(Points + 2 - j) + d6(Points + 2 - j))
      Next j


    ' Placing arbitrary points to show reaction forces at Boundary

    Worksheets("Calculations").Cells(2, 20) = -0.000000000000001
    Worksheets("Calculations").Cells(2, 21) = 0
    Worksheets("Calculations").Cells(2, 22) = 0
```

```
Worksheets("Calculations").Cells(2, 23) = 0
Worksheets("Calculations").Cells((Points + 4), 20) = (L + L2 + 0.00000000000001)
Worksheets("Calculations").Cells((Points + 4), 21) = 0
Worksheets("Calculations").Cells((Points + 4), 22) = 0


Cells(30, 4) = -1 * Worksheets("Calculations").Cells(Points / (L + L2) * L + 1, 2) +
Worksheets("Calculations").Cells(Points / (L + L2) * L + 2, 2)
Cells(30, 5).Value = -1 * Worksheets("Calculations").Cells(Points + 2, 2)
MsgBox Points / (L + L2) * L
End Sub
```

## 6. Bibliography


[1] E. Bruhn, Analysis and Design of Flight Vehicle Structures, Indianapolis, Indiana: S.R. Jacobs and Associates Inc., 1973.

[2] W. Young, Roark's Formulas for Stress and Strain, McGraw-Hill, 2002.

[3] W. Thomson, Deflection of Beams by the Operational Method, J. Franklin Institute, 1949.