Cow-tail Biomechanics
Data Analysis

by **Tim Bishop**
4/13/2010

for
Dr. Gove Allen, MBA 614

Background

Biomechanics is a field of science exploring the mechanical properties and functions of the body. In the BYU Applied Biomechanical Engineering Laboratory (BABEL) properties of spinal intervertebral discs are studied. Research includes studying the degeneration of discs as induced by chemical and mechanical initiation. For the research associated with this project, bovine tail segments were dissected and put into a machine that cycles the specimen through three different modes of loading: compression/tension, flexion/extension, and axial rotation. Throughout the test, data points consisting of angular position and force/torque data were collected at 1000Hz and smoothed to 100Hz. Raw data for a single test consists, on average, of approximately 6000 data points representing dead time on either side of fifteen 100-data-point cycles. Then there were 50 samples with two runs of 3 tests each, or approximately 300 data sets.

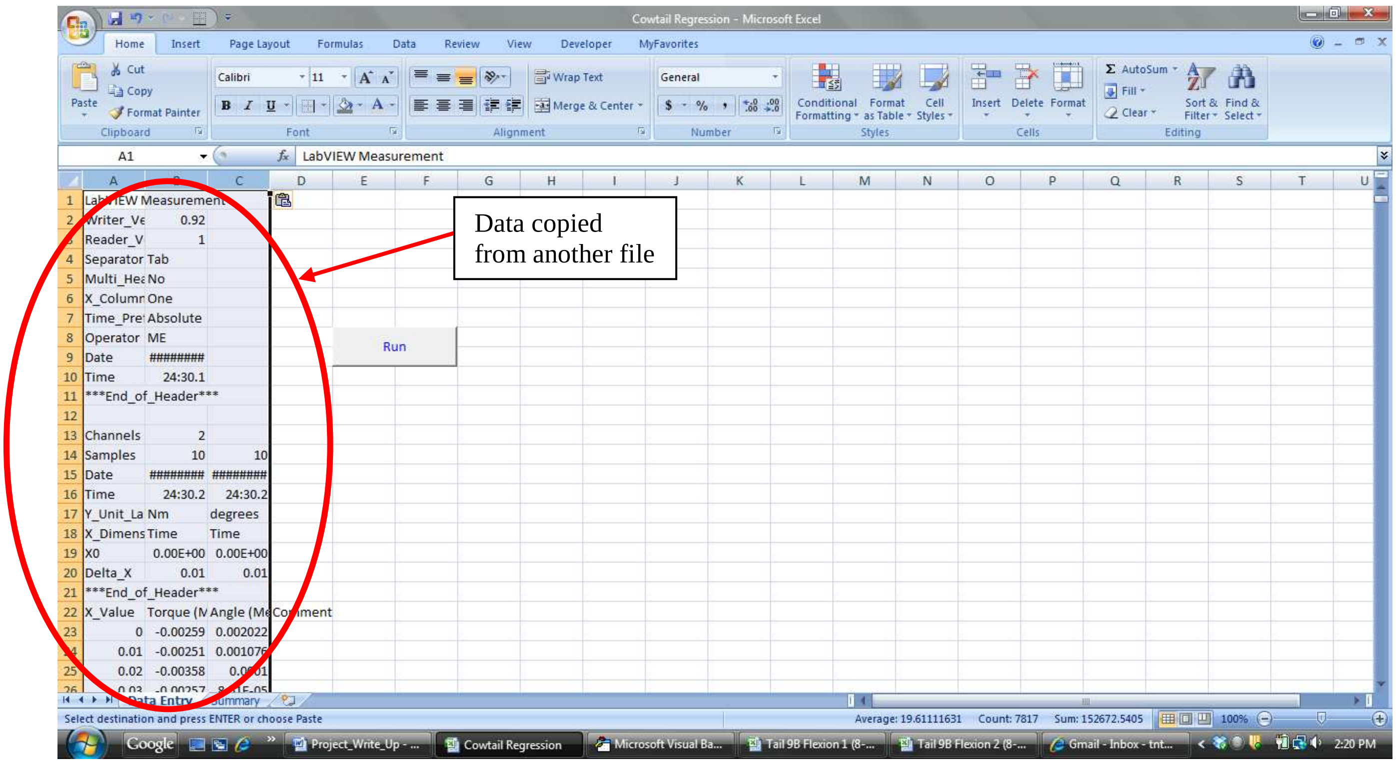
For each data set, only the last of the fifteen cycles is needed. It needs to have a few characteristic pieces pulled from the data including maximum, minimum, and an exponential, least-squares regression model. There is also a neutral zone which is a quasi-linear portion of the curve in the center allowing for more displacement with less force resistance. The length of that neutral zone is a key differentiator in intervertebral disc degeneration levels. For this research, the neutral zone is determined as being between the two points that first break from the linear pattern. Then, statistical analyses can be performed after the characteristic components are pulled from each data set.

The VBA routines assembled to analyze the data go through the following steps:

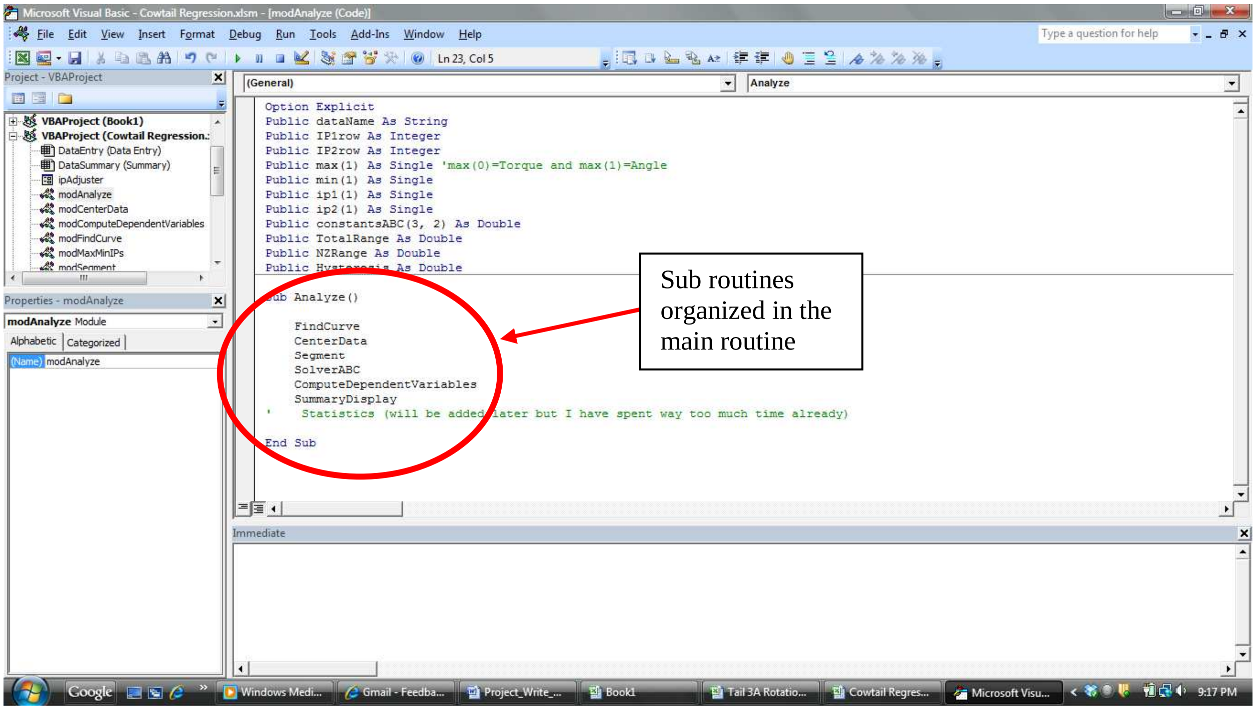
- Finds and copies the 15th cycle of data to a new worksheet named for the data set
- Calls a userform with a Microsoft Chart Object to allow a user to indicate the inflection points which define the boundaries of the neutral zone
- Centers the neutral zone on the origin of the graph and offsets all of the data to match
- Calls the Solver add-in routines to provide three constants for four exponential best-fit models
- Calculates the total range, neutral zone range and the hysteresis in each sample
- Summarizes the Data on a separate worksheet for ease in statistical analysis

Implementation

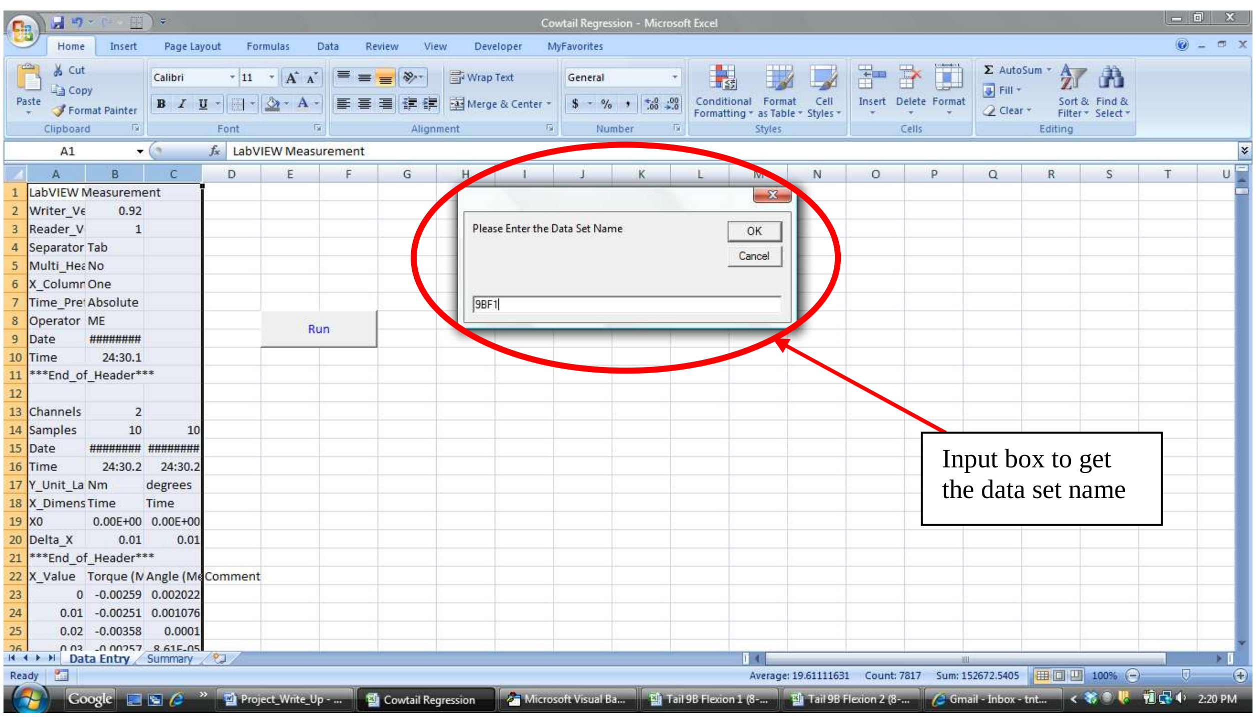
A user will open a delimited text output file in a separate workbook of Excel, then the first three columns is copied and pasted into the worksheet “Data Entry” as seen below.



When the user presses the “Run” button, the Analyze subroutine is called which is a hub in which all of the other routines are organized and called.

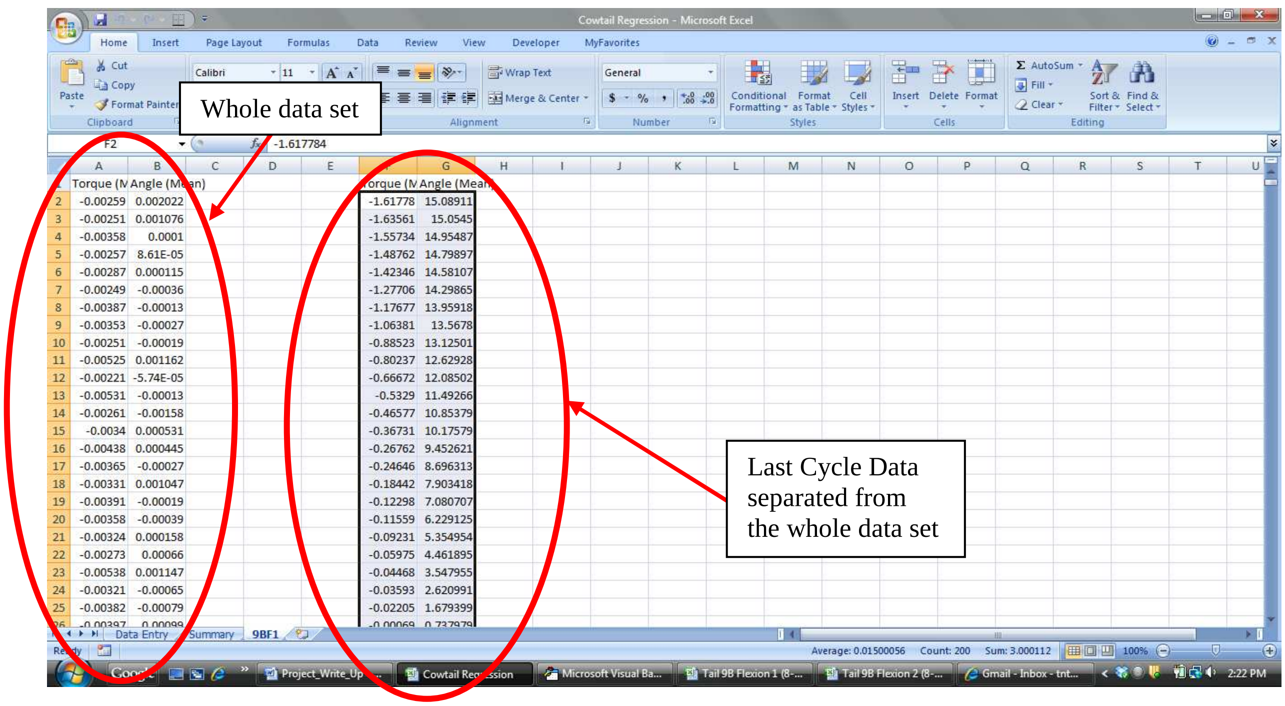


The user is immediately prompted to enter a name for the data set. This name will go onto a new worksheet whereon will be kept all of the calculations and it will be used to organize the "Summary" worksheet.



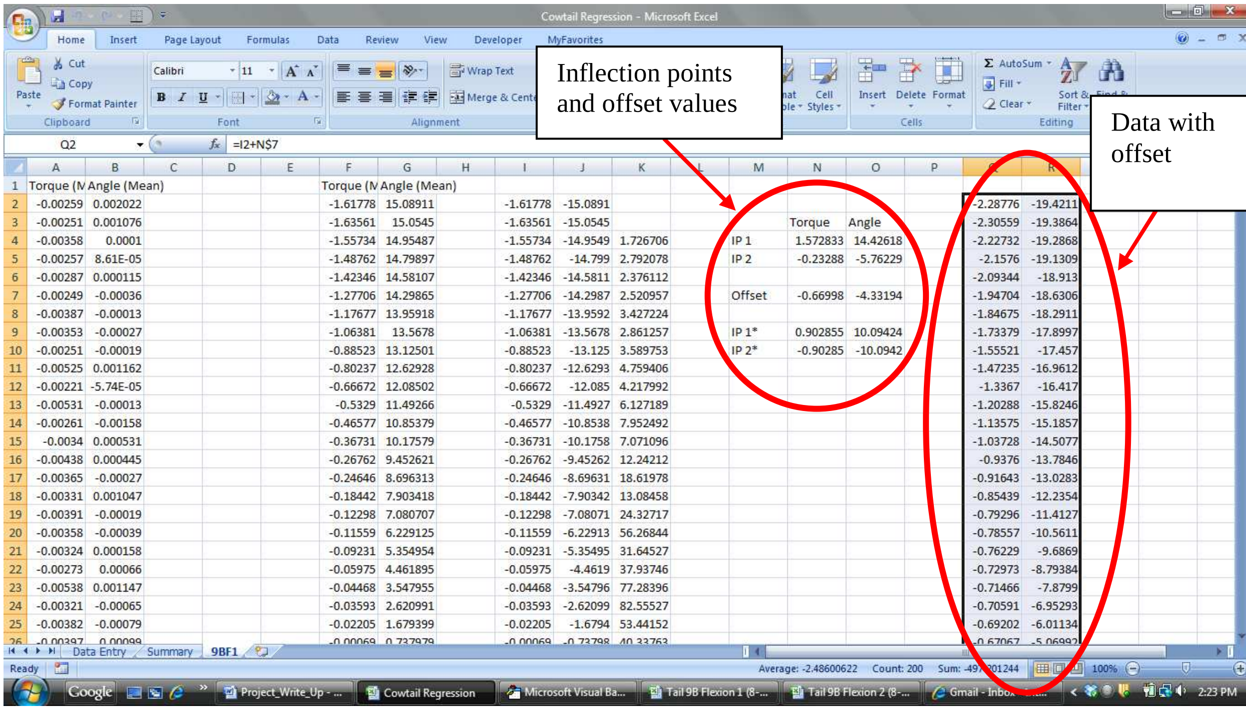
FindCurve

With the name inputted, the data is moved to the newly created worksheet and the last cycle of data is taken and copied into its own columns. This is done by marking all of the major maxima and minima and comparing them to each other to see where the last of the cycles is located.



CenterData

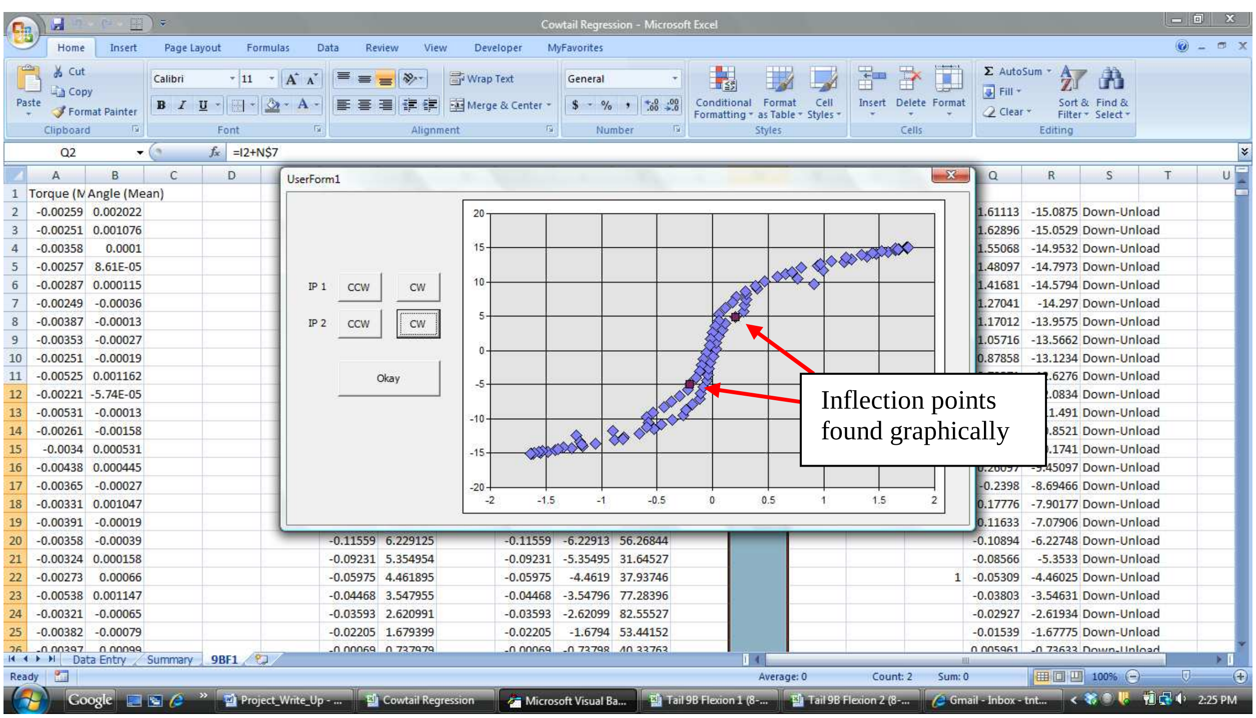
The column of angular position data is multiplied by -1 to give the plot the “S” shape that is traditional for such biological materials. An initial guess at where the inflection points might be is hardcoded into the routine. The user will have a chance to rectify poorly designated inflection points later. The data is offset so that, when graphed, all of the data is centered using the inflection points.



Segment

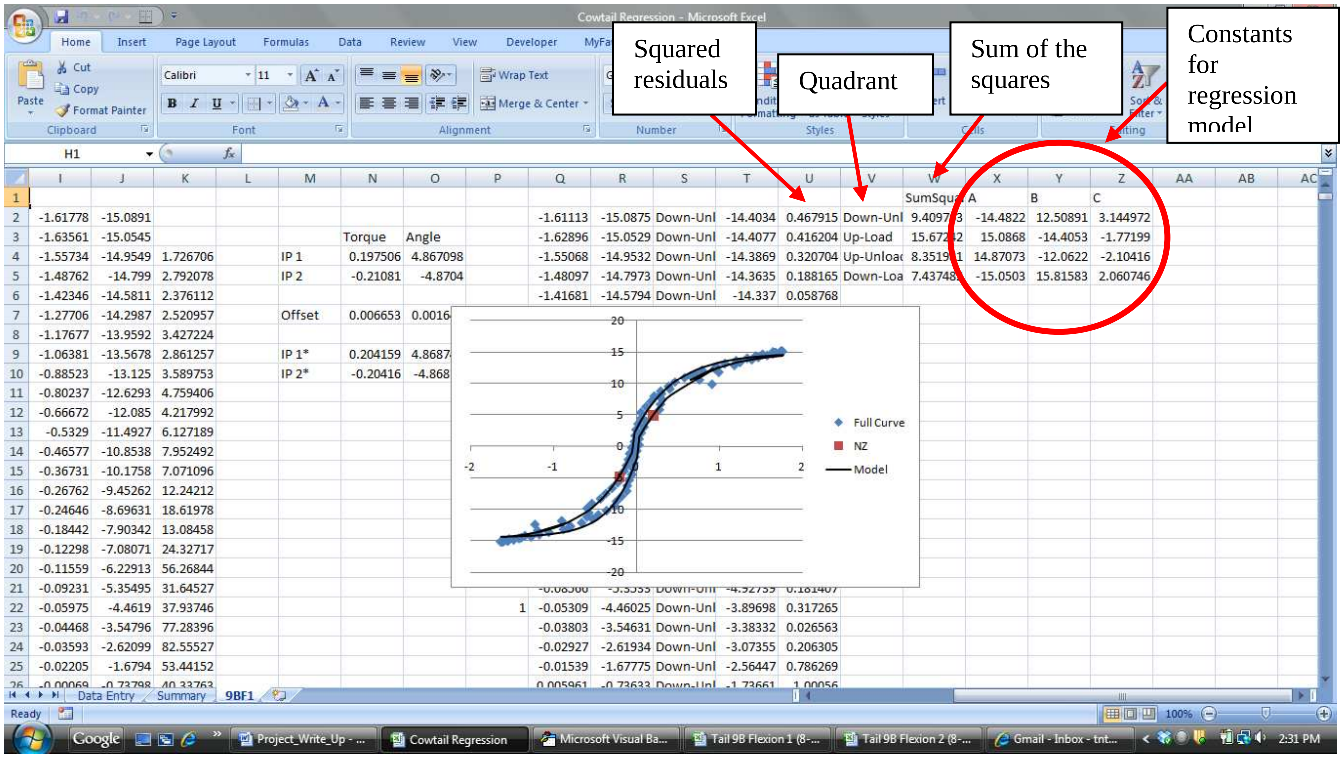
A userform, seen below, appears and allows a user to determine where the actual inflection points should be. The buttons move the highlighted inflection points clockwise or counterclockwise around the curve.

Another column segmenting the quadrants of the curve appears to the side of the offset data. This tells a user where the curve is manifesting loading or unloading and the curve is above or below the angle = 0 line.



SolverABC

Setting up the data for least-squares regression, two more columns are added onto the right of the offset data column. In column T is the value that is calculated from the equation $Angle = A + B * e^{C*Torque}$, where A, B, and C are constants that must be found with a non-linear solver algorithm. Because the algorithm is sensitive to the starting position of the inputs, I set up a loop to randomize the starting points 10 times and save the constants that provide the lowest sum of the squared residuals. The squared residuals are found in a column and the sum of the squares for each quadrant are stored in nearby cells. The data is then plotted as points with a line representing the regression model, allowing the user to visually confirm a good fit.



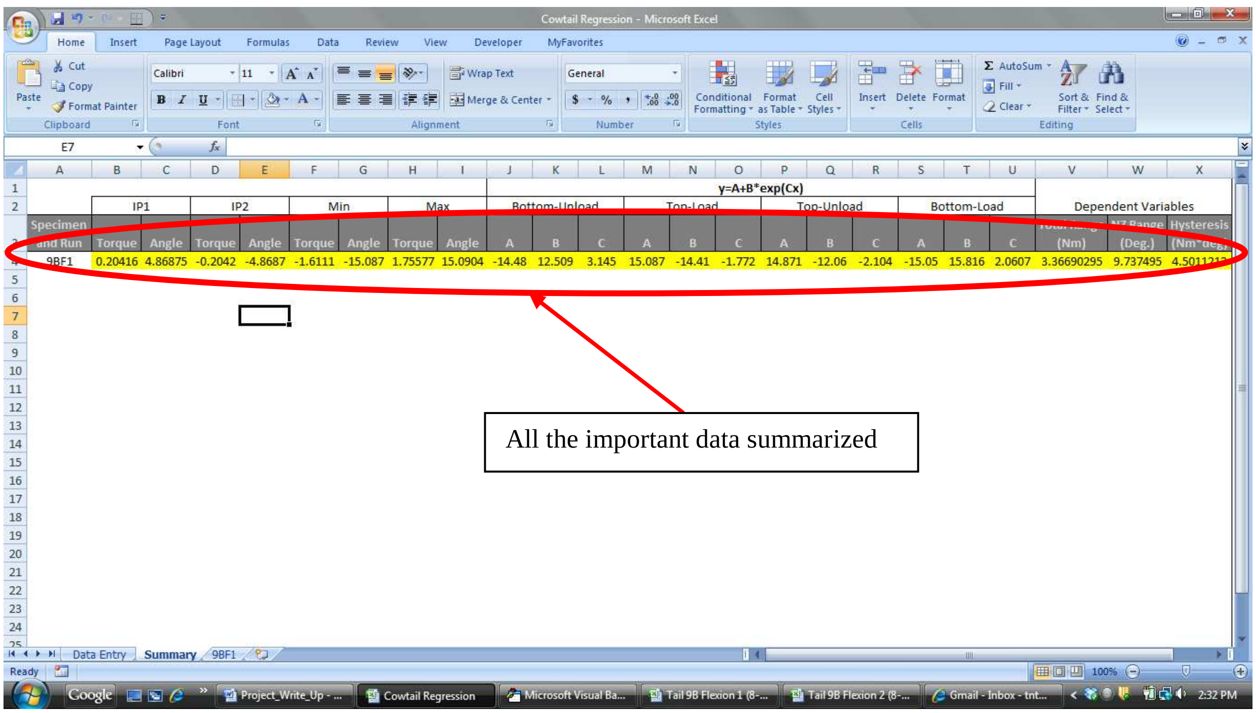
ComputeDependentVariables

In this sub routine, most of the calculations are simple subtraction to find the distance between two points. However the hysteresis is defined as the area between the loading and unloading curves.

A function routine was created to perform the integral of the regression models so that the differences of the integrals on the top and the bottom could be added together to come up with the total hysteresis for the specimen.

SummaryDisplay

Finally, when all of the useful and interesting data has been manipulated and pulled from the data set, it is outputted to the worksheet "Summary" where it can wait organizedly to be further analyzed statistically. The worksheet is also useful for seeing some of the data, but it is incomplete as some of the data was held only in the VBA variables until the summary page and it was not meant to be an aesthetically pleasing view of the data. The summary page is the preferred and clean portal to view the data results.



Difficulties

Many difficulties arose throughout the completion of this project. Time went into properly getting formulas into the worksheet cells so that it is dynamic based on the positioning of the inflection points. More time was spent finding ways to verify that the routines were coming up with correct results and to adjust for variations in data sets. The most time was spent in the userform chart object.

To link a Microsoft Chart object to a worksheet requires an intermediate Spreadsheet object to also be included in the userform. In my case it is hidden, but is there out of necessity. The chart can be linked to the Spreadsheet object which in turn can receive cell values from the underlying worksheets.

Conclusion

This has been a fun project and one that will significantly speed up my data analysis to finish my master's thesis. The project works as it was intended, and I feel empowered to have created it.