Trent Hinkle

Trenth12@gmail.com

Abstract

This document is a user guide for the Generalized Pareto Statistical Model. The document covers everything the user should need to know including but not limited to: proper usage of the program, threshold selection, proper parameter estimation, interpretation of validation plots/confidence intervals, explanation of quantiles/return periods/return levels, model prediction, incorrect scenarios, and special cases.

Python Generalized Pareto INLS Flexor Force Model User-Guide

NAVFAC EXWC

Table of Contents

[Data Import 2](#_Toc460583307)

[Potential Data Import Issues: 2](#_Toc460583308)

[Threshold Selection: Mean Residual Life Plot 3](#_Toc460583309)

[Threshold Selection Help 4](#_Toc460583310)

[Theoretical Example: Memory error from Mean Residual Life Plot 5](#_Toc460583311)

[Parameter estimation 5](#_Toc460583312)

[Theoretical Example: Bad Confidence Intervals 8](#_Toc460583313)

[Example 1: No proper Maximum in Shape vs Log-Likelihood plot 8](#_Toc460583314)

[Example 2: Shape Parameter Accuracy 9](#_Toc460583315)

[Example 3: Shape Parameter Equal to Zero 12](#_Toc460583316)

[Model Validation 13](#_Toc460583317)

[Quantile Plot: 13](#_Toc460583318)

[Probability Plot: 14](#_Toc460583319)

[Density/Histogram Plot: 14](#_Toc460583320)

[Return Level Plot: 15](#_Toc460583321)

[Interpretation of Quantiles and Return Periods/Return Levels 16](#_Toc460583322)

[Quantile: 16](#_Toc460583323)

[Return Period/Level: 16](#_Toc460583324)

[Extending number of observations to time: 16](#_Toc460583325)

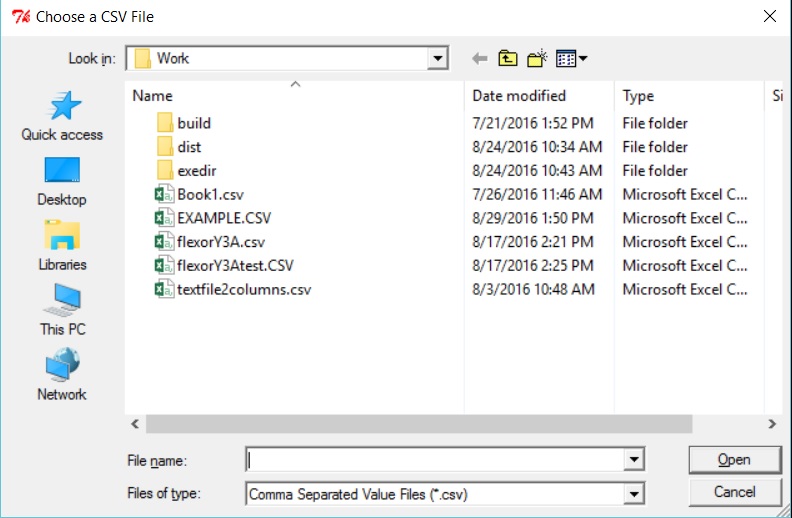
[Model Prediction 16](#_Toc460583326)

# Data Import

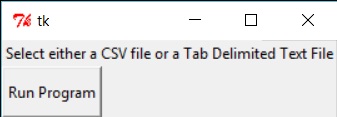
Upon opening ‘ParatoModel.exe’ you will be prompted to select a .csv file which should include the Flexor Force data from an ANSYS AQWA Simulation.

The Image below will pop up upon starting ‘ParatoModel.exe’

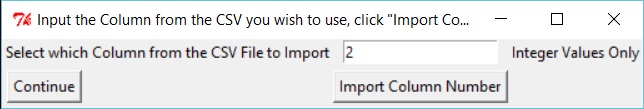
Simply select the .csv file you wish to import, in this case we are importing ‘EXAMPLE.CSV’. Then click ‘open’.



After importing the .csv file you wish you will be prompted by another pop up and all you need to do is click ‘Run Program’



Then you will be prompted to select which Column of Data from the .csv File you wish to import as follows:



Simply enter in the Column you wish to import (Column number 2 in this case). Then click ‘Import Column Number’ followed by ‘Continue’.

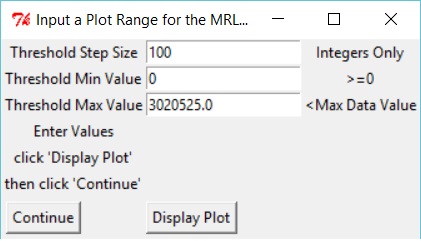
## Potential Data Import Issues:

The program should be able to read in .csv files straight from ANSYS AQWA. If problems occur it could mean than there are blank entries somewhere in the .csv file (Usually in the beginning where explanatory information is printed). If this is the case and the program is not allowing data to be imported, you could simply delete the explanatory text leaving simply the time data and force data, or you could simply press enter in the blank boxes (usually above the force data) so the program recognizes that it is an empty cell as opposed to a nonexistent cell. However, these problems should not occur as the program has been written to bypass these problems. If problems persist please do not hesitate to contact the creator of the program listed at the beginning of this user-guide.

# Threshold Selection: Mean Residual Life Plot

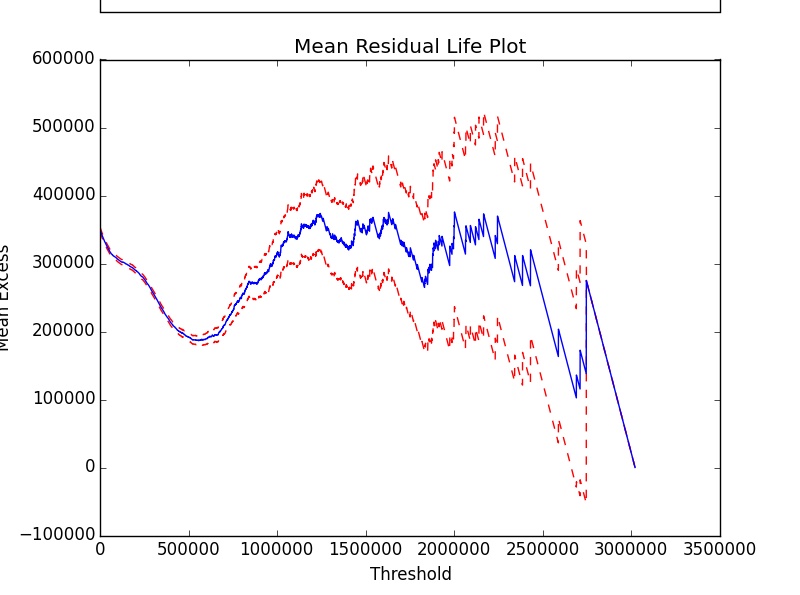
Next you will be prompted to enter in the following information: Threshold Step Size, Threshold Min Value, and Threshold Max Value. The initial values are set at 100 for Threshold Step Size, 0 for Threshold Min Value and the maximum data value for Threshold Max Value.

When Satisfied with your selected values, please click ‘Display Plot’



Upon clicking ‘Display Plot’ please allow the program time to execute the plot.

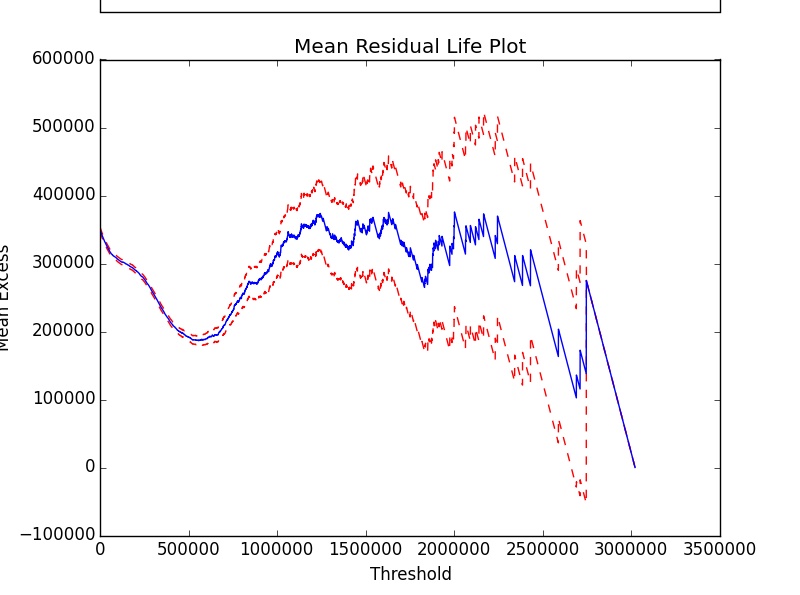
Having a lower threshold step size and a higher range for min/max Threshold values will make the program take longer to execute the plot. The plot should look similar to the following image:



Now, with this plot we must select a ‘System’ Threshold Value. This Threshold Value determines the Data used to train the model (Only Data points higher than the Threshold will be used to train the Model).

## Threshold Selection Help

To select a System Threshold Value, you look for areas on the plot where the blue line appears stable and linear. The Threshold Value you select will be the beginning of the stability and linearity. Tight Confidence Intervals (Red Dashed Line) are also better than Wide Confidence Intervals. See the plot below for help on selecting a System Threshold:



Linear/Stable

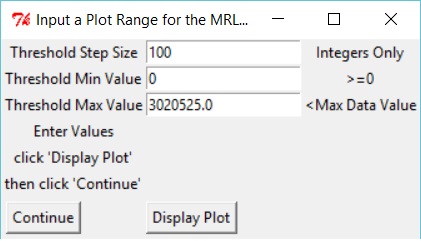
As you can see by the three green arrows and stars in the picture, one could select approximately 200,000 as a threshold value and one could also select approximately 650,000 as another potential threshold value. 1,200,000 could also be a potential Threshold Value, it is relatively linear from that value but not particularly stable.

**\*Something important to note is that the system threshold you select must be within the last plotted Threshold Minimum and Maximum values, and it must be a multiple of the Threshold Step Size.\***

As stated previously, a potential threshold selection for this data could be 650,000 as well as 650,100, 650,200, etc. Just make sure the value is a multiple of the Threshold step size (in this case the step size is 100).

**\*In the real plots the program will display, there is an interactive coordinate system based upon mouse position so you will be able to accurately see potential threshold selections.\***

When Satisfied with the plot, please click ‘Continue’ on the Mean Residual Life Plot User Input Screen

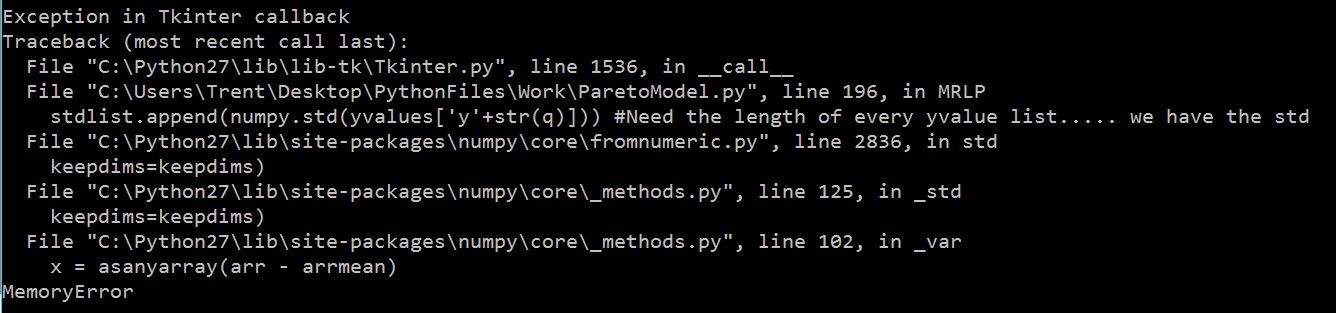


## Theoretical Example: Memory error from Mean Residual Life Plot

Sometimes if there is too much data to be imported a small enough threshold step size will generate a memory error and the plot will not be displayed. This happens because there are a lot of calculations happening behind the scenes when ‘Display Plot’ is clicked.

If this happens, simply decrease the range for Threshold min/max value or increase the Threshold Step Size

Below is an image of the output to the command line when a Memory Error occurs from the Mean Residual Life Plot

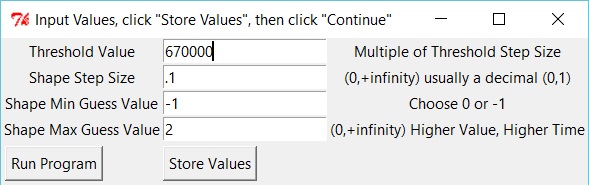


If this happens please close out of command line and restart the program as this usually makes the program crash if you try to re-plot. Remember to decrease the range for Threshold min/max value or increase the Threshold Step Size. It is advised to Increase the threshold step size first before decreasing the threshold range.

# Parameter estimation

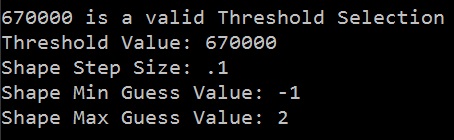
Next, you will be prompted with a pop up to input your Threshold Selection, Shape Step Size, Shape Min Guess Value, and Shape Max Guess Value. Initial values will be blank for Threshold Value, .1 for Shape Step Size, -1 for Shape Min Guess Value, and 2 for Shape Max Guess Value. It is advised to leave the initial values alone for the first run through. If doing multiple run through be sure to remember the Threshold Value if you are keeping it the same.

Simply input the Threshold Value you chose from the Mean Residual Life Plot. Then input the other values (Default Values are good to try at first).

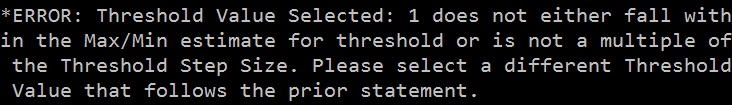


As you can see in this run we have selected 670,000 as the Threshold Value

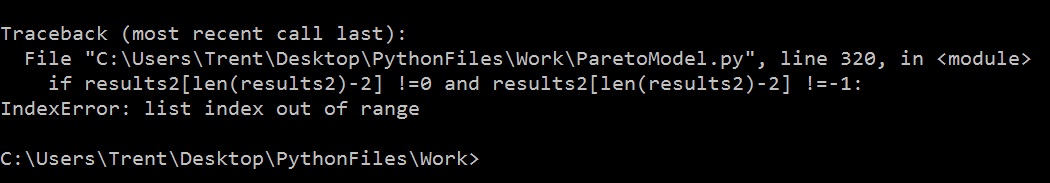
When ready click ‘Store Values’ Followed by ‘Run Program’ and the following will output into the command line indicating that the program is ready to run:



If an improper selection of Threshold is made the following error will be displayed after clicking ‘Store Values’:

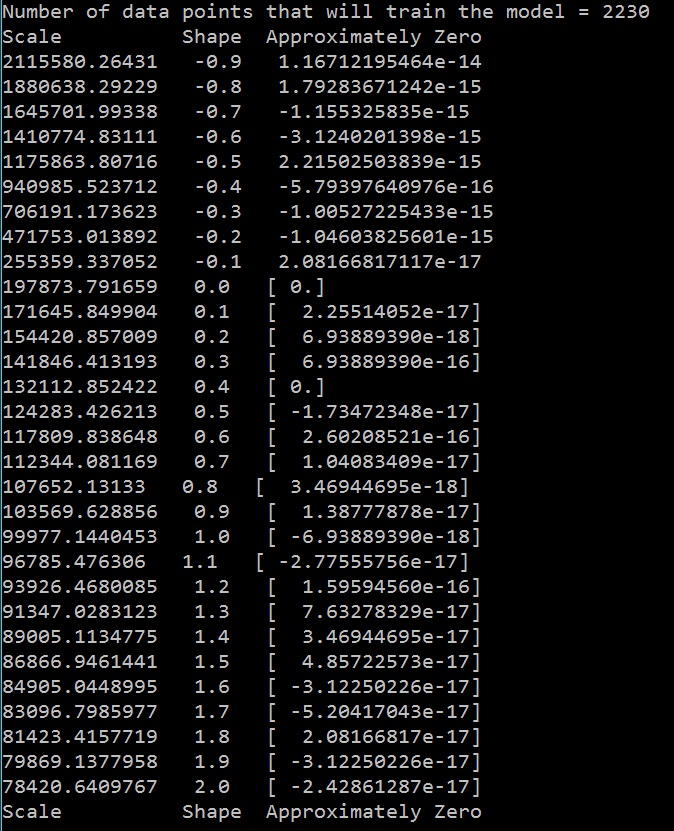


If this is ignored and one clicks ‘Run Program’ the following Error will be displayed and the program will crash:



This happens because no values will be stored when ‘Store Values’ is clicked when an improper Threshold Value Selection is entered under ‘Threshold Value’. If this happens one will have to restart the program, ensuring to enter a valid threshold selection.

Upon successfully entering in a proper Threshold Value the following will output into the command line:



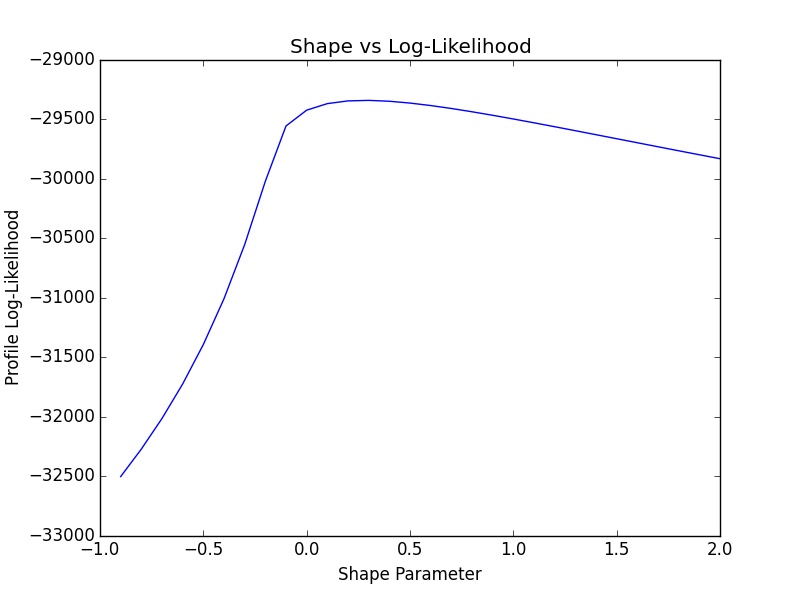
This tells the user the number of data points that are training the model as well as the estimated scale parameters based on guess shape parameter values.

It also tells the user what the derivative of the log-likelihood function with respect to scale is with the estimated Scale and Shape values plugged in. This number should be zero because we are maximizing the Log-Likelihood function (Derivative is equal to zero). Therefore, be sure to go through and check to see whether or not the terms under ‘Approximately Zero’ are close to zero. 1e-9 should be good enough.

If you are not getting values close to zero under ‘Approximately Zero’ then the algorithm is not learning the Scale parameter properly based on the corresponding Shape parameter.

If this problem is encountered start over with a different threshold selection or change the guess Shape min/max range so that it does not include shape values that learn improper Scale values (Where the ‘Approximately Zero’ column is not close to zero).

The next figure that will pop up is the Shape vs Log-Likelihood plot:



Example of Shape vs Log-Likelihood Plot with a Proper Maximum

Command Line Output for Parameter Estimation

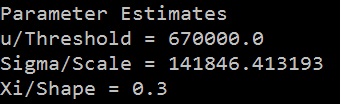
0.3

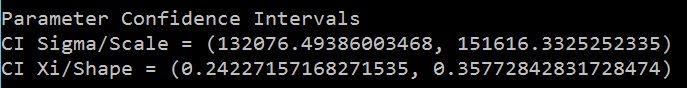
Maximum

The Plot should look like the above image (for the most part), should resemble a negative parabola and display a maximum (indicated on the plot). This maximum in the plot (Here it seems to be around .3), is the Shape parameter that maximizes the log-likelihood function, and therefore is the specific shape parameter that fits the force data the best (with its associated scale parameter).

If there is not a definitive maximum on the plot please restart the program with a wider selection range on minimum and maximum shape values under step 3. However please do not select a minimum shape value of less than -1.

The following is the command line output for the Parameter Estimates as well as the associated confidence intervals (This will print out after the Shape vs Log likelihood plot is closed). Do be sure to check the confidence intervals to make sure they are reasonably tight for both parameters.





**\*Please make sure for the Shape Parameter that the confidence intervals do not cross over either 0 or -1. Unless the shape value learned is equal to 0 (then it will definitely be above and below 0)\***

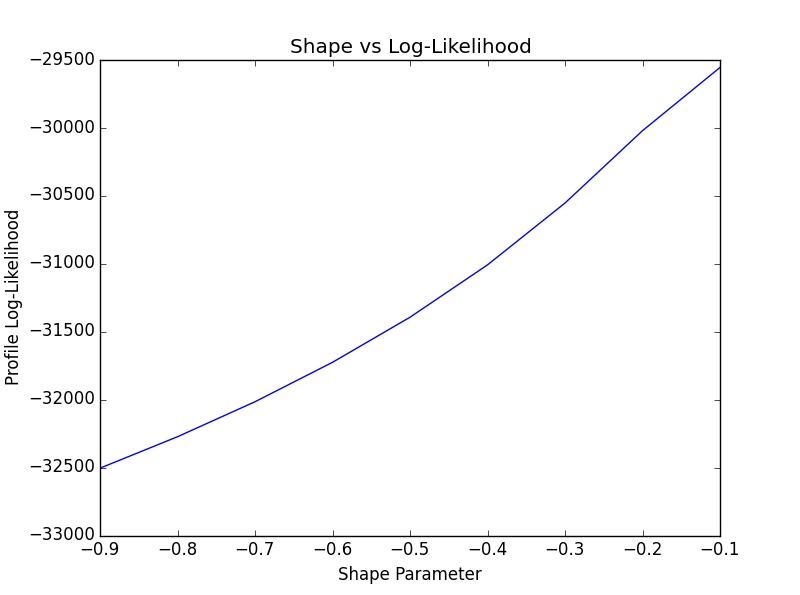
## Theoretical Example: Bad Confidence Intervals

Ex: If Shape is learned to be -.1 and the confidence interval for the shape parameter is (-.2137, .2538) then this is not good because the confidence interval is crossing 0. The same can be said for the confidence interval crossing -1 as well. If this happens try a larger testing dataset (lower threshold). Disregard this for shapes equal to 0.

## Example 1: No proper Maximum in Shape vs Log-Likelihood plot

Pictured below is an example of a maximum not being displayed in the plot. As stated before, this means that the range on the guesses for the shape value was not wide enough. As you can see, the range we selected was -1 to -0.1.

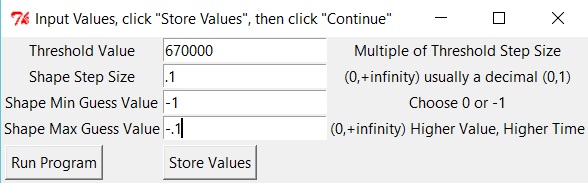
Example of Shape vs Log-Likelihood Plot without a Proper Maximum



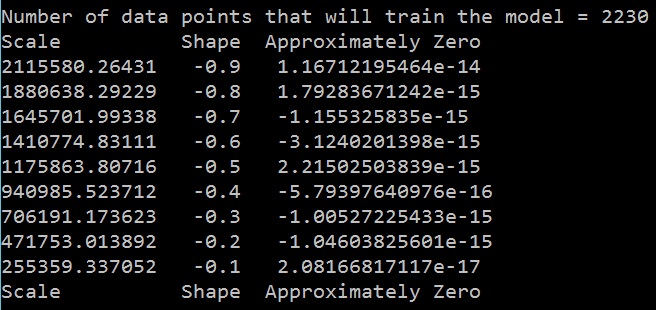
As you can see, the line seems to continue with an upward trend toward a maximum somewhere to the right towards a shape value greater than -.1 (max value picked)

Judging from the graph, we can tell that the maximum will lie somewhere to the right of -0.1. Therefore we need to increase the range on the shape parameter until we see the maximum (Or the Program will not learn proper parameter values corresponding to the given data set). The default guesses for Shape parameter are set from -1 to 2 with a step size of 0.1. It is recommended to keep these at default for the first trial and if a maximum is not found, then increase the maximum guess until the proper shape value is learned by the program.

The following images show how the above plot was generated:



As stated, Smaller Range for Guess Shape (-1,-0.1)



Smaller Range for Guess Shape

As we can see from the plot above given the small range for Shape, the true maximum on the Shape vs Log-Likelihood plot was not found. When this occurs it is advised to restart the program and increase the range on the shape parameter guesses.

## Example 2: Shape Parameter Accuracy

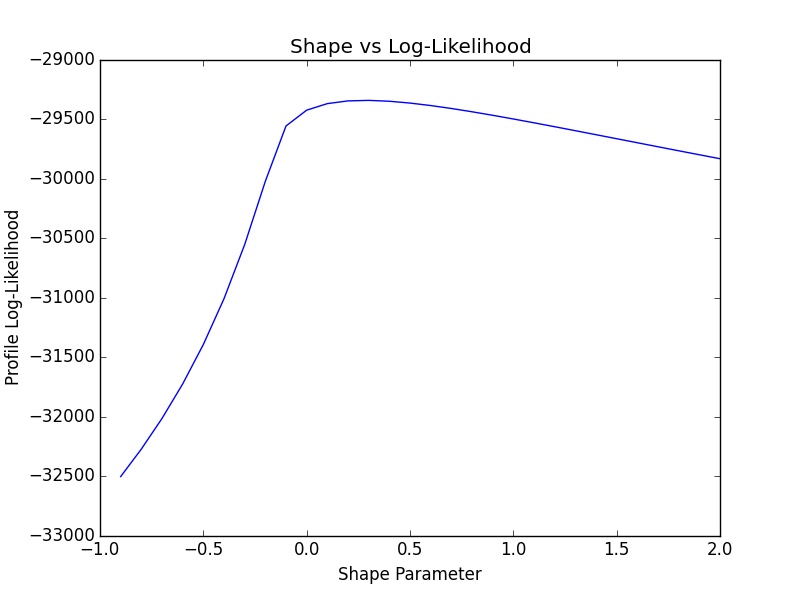
In this example we cover ways to increase the accuracy of the shape parameter.

A more accurate shape parameter estimation is completed as follows:

Restart the program with a more narrow selection range on the minimum and maximum shape values that surround the maximum. Also **lower the shape step size** in this interval and make sure to select the same threshold value.

**\*One could simply keep the same guess shape range and while making the shape step size smaller as well. However, given large data sets this will potentially make the code take a long time to execute which is not efficient or desirable.\***

Below we display the Shape vs Log-Likelihood plot from section 3 which displays a true known maximum:



Maximum

Represents a new theoretical range around the maximum

Upper Range

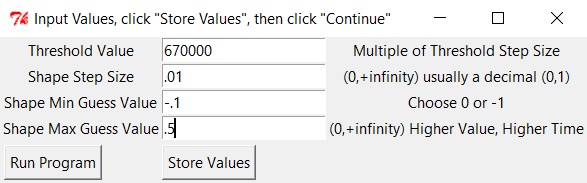
Lower Range

As we can see from before, the plot displays a maximum around a shape value of 0.3.

The Red dashed lines represent a potential reduced range for guess shape values (In this case between -0.1 and 0.5)

To generate a more accurate solution, simply restart the program with **the same threshold value selections** and input a smaller range for the guess shape values while also selecting a **smaller shape step size**. This is the most important aspect or a more accurate solution will not be achieved.

Below are pictures that show an in-depth of example of generating a more accurate shape value:

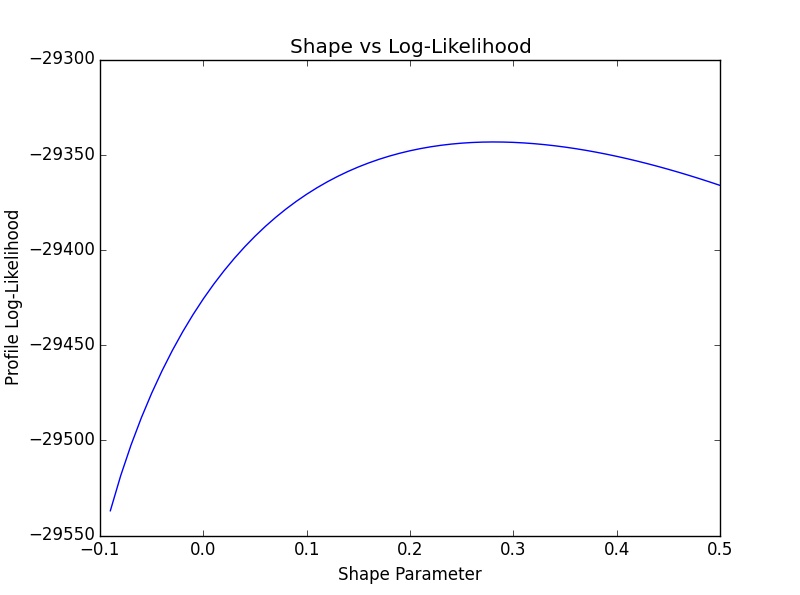


Smaller Shape Step Size, Closer Min/Max Guess Values

This input will allow the program to generate a more accurate Shape parameter estimation. However, the smaller step size will also make the program take longer to execute.

**\*The most important part of this is the smaller shape step size.\***

Below we show the updated plot of Shape vs Log-Likelihood that has been ‘Zoomed in’ around the maximum.

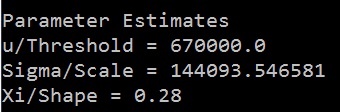


Improved Accuracy Shape vs Log-Likelihood Plot

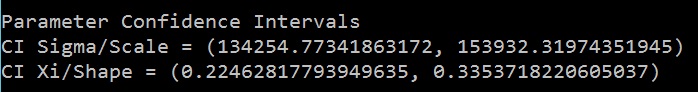
Max seems to lie between 0.2 and 0.3

As we can see, given the smaller step size a more accurate Shape Parameter can be learned

The Output from this parameter estimation was as follows:



As shown, a shape value of 0.28 was learned as opposed to a shape value of 0.3 from the less accurate parameter estimation.



Confidence Intervals seem to be the same, simply shifted by .02 exactly like the Shape parameter itself.

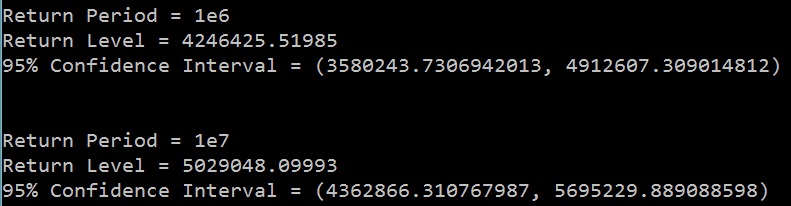
We could run this technique multiple times decreasing min/max guess shape range and decreasing the shape step size every time to get a new more accurate result every run through (to an extent).

## Example 3: Shape Parameter Equal to Zero

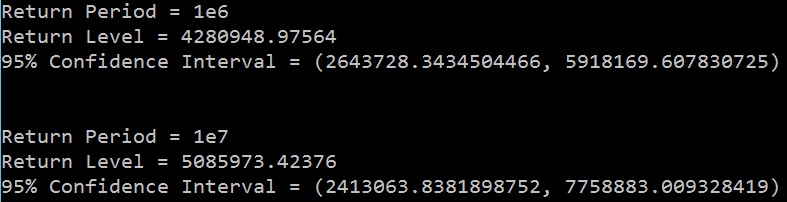
If after parameter estimation the shape parameter is learned to be 0 it is recommended to run through and perform a shape parameter accuracy (Example 2 under parameter estimation) test to make sure that the shape parameter learned is definitely equal to 0. When shape is equal to zero Confidence Intervals will be tighter than usual when high return periods are used and confidence intervals will be wider than usual when small return periods are selected.

The following images are return levels compared when the shape parameter is learned as 0 vs when the shape parameter is learned as .003 (this was optimized using shape parameter accuracy test around 0 multiple times). Threshold value selected was 1,100,000. The following images compare return levels between the two learned shape values:

Shape = 0:



Shape = .0003:

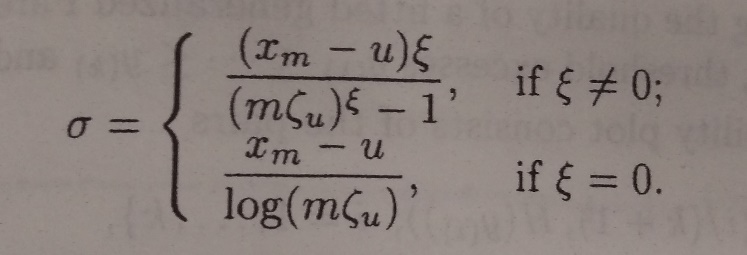


As we can see, the confidence intervals on return level when shape is equal to 0 are tighter than when we optimize shape value to be .0003, the confidence intervals are wider. Therefore, when interpreting the confidence interval on return level when shape is equal to zero there is some uncertainty when it comes to the accuracy.

The reason a shape parameter accuracy test is warranted is that the confidence intervals on Return Levels may not be 100% accurate when shape is equal to 0. When shape is equal to zero the Generalized Pareto Distribution becomes an Exponential Distribution. When calculating the confidence interval for Return Level we use the following variance:



When shape or Xis is equal to zero it simply reduces down to sigma^2. For return level confidence intervals we substitute sigma for the following:



Once we substitute for sigma we take the square root of the variance and multiply by 1.96 for 95% confidence intervals.

**\*It is unsure at this time if the confidence interval method for shape = 0 is correct.\***

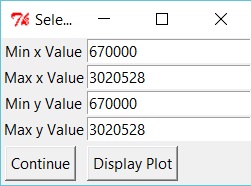
**\*It is highly recommended to do the shape parameter accuracy test to potentially learn a shape value that is not zero to have a more accurate and trustworthy solution.\***

# Model Validation

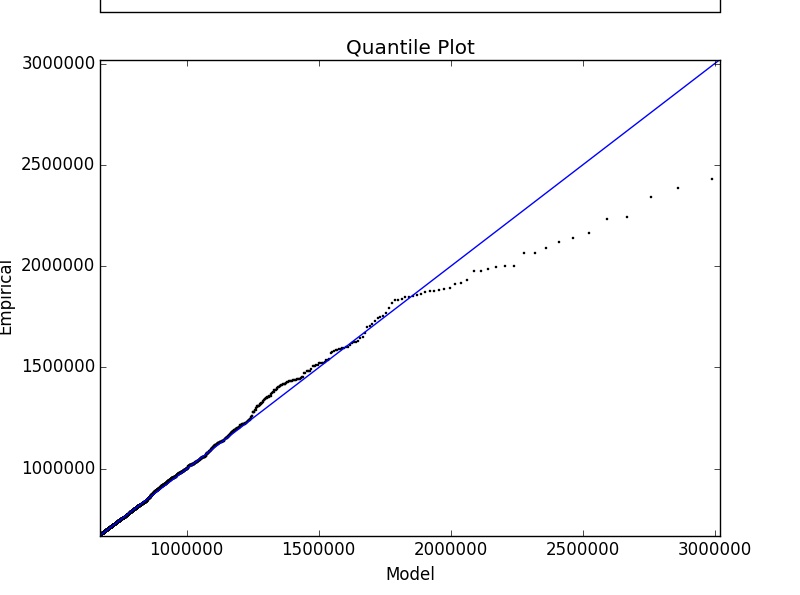
The following section will display Quantile plots, Probability plots, Histogram plots, and Return Level Plots. These plots are used to validate the model (see how well the model fits the simulation data).

## Quantile Plot:

First select the plot range for the quantile plot. It will automatically set a proper plot range from the minimum data value to the maximum data value.



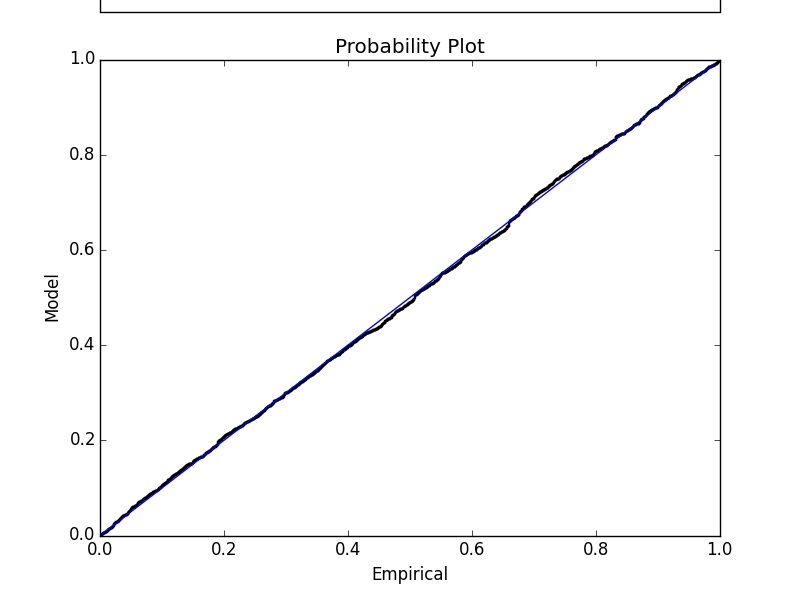
Example Quantile Plot



The solid blue line represents the best model fit while the black scatter plot line represents the real data. The closer the black scatter plot is to the blue line the better the fit is. Essentially we want the black scatter plot data to be linear with a slope of one. Simply click ‘Display Plot’, when satisfied click the ‘x’ on the graph and then click ‘Continue’.

The above image displays decent fit for the Quantile plot.

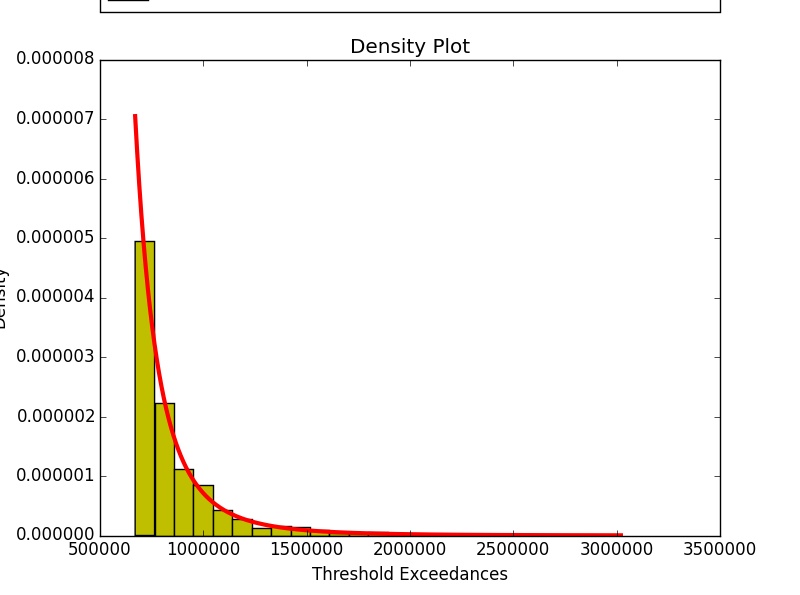
## Probability Plot:



Same as with the quantile plot the blue line is the best model fit and the real data (Empirical) is the black scatter plot. We want the black scatter plot to be linear with a slope of 1 just as the blue line is. When satisfied click the ‘x’ on the graph and the program will continue.

As you can see, the fit here for the probability plot is great.

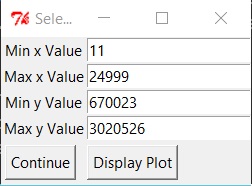
## Density/Histogram Plot:



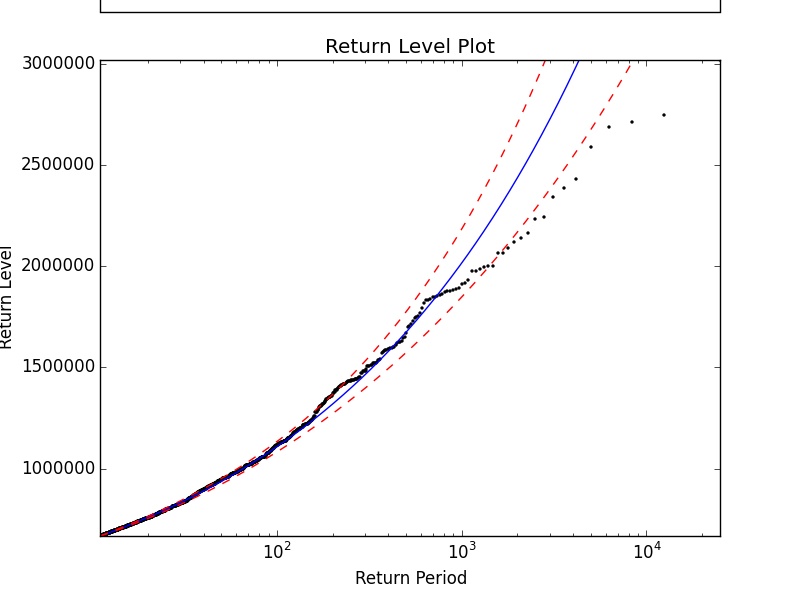
The density plot is not as important as other plots but in this plot we want the red line to line up with the yellow histogram plot. Essentially the yellow histogram plot is a percent wise histogram plot of the simulation data and the red line is the best model fit. When satisfied click the ‘x’ on the graph and the program will continue.

## Return Level Plot:

First select the plot range for the Return Level Plot. It will automatically set a proper plot range from the minimum data value to the maximum data value. Generally for the Return Level Plot you will not have to change the plot range. Simply click ‘Display Plot’, when satisfied click the ‘x’ on the graph and then click ‘Continue’.



Example Return Level Plot



The Black Dotted line (Real Data), should reasonably agree with the model and follow the blue curve (Best Model Fit). The Real Data should also reasonably fall within the red dashed lines (95% Confidence Intervals).

# Interpretation of Quantiles and Return Periods/Return Levels

## Quantile:

To get a quantile (which is essentially a theoretical data point) you need the quantile function and the parameter estimates (shape, scale, and threshold) and you import a probability let’s say .95 or 95% (as an example). What the quantile function tells you then is given a probability what is the force in the flexor less than or equal to. So we can say 95% of the time the force in the flexor should be less than the calculated value. This calculated value is the quantile.

## Return Period/Level:

A Return Period in this context is simply seen as a ‘number of observations’ and the return level is the predicted force value that is exceeded on average once every ‘m’ number of observations (essentially the maximum force value after ‘m’ observations). Therefore, given a number of observations (basically a random number of potential force readings, remember this is a return period) one can calculate the maximal force value expected to be exceeded (Return Level) based on these number of observations.

For example let’s say we wanted to see the potential maximum force value in the flexor after 1,000,000 observations. Then we simply plug in 1,000,000 into the return level function and it will give us a return level or maximum force value for that ‘m’ number of observations. So we can say after 1,000,000 observations the force should not exceed the calculated force value (return level).

### Extending number of observations to time:

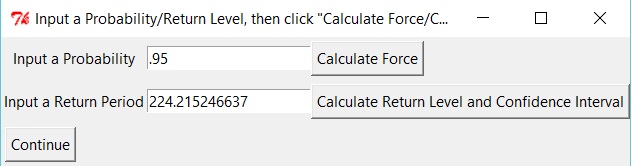
It is also possible to calculate Return Levels (Upper force values) from times as opposed to number of observations. To do this you just need to calculate the time step per observation first (From ANSYS AQWA simulations). Then select a specific time you want to measure a return level on. Once you have this time (example 24 hours) you simply calculate how many observations from the simulations it would take to reach your time limit and then plug in this ‘m’ number of observations into the return level function to get a return level based on a time.

# Model Prediction

**\*Please see ‘Interpretation of Quantiles and Return Periods/Return Levels’ above before calculating values to get an explanation on what Probabilities, Quantiles, Return Periods, and Return Levels mean and how to interpret the resulting Force values (Essentially Maximums).\***

The final step in the program is model prediction.

Once all plots have been displayed you will be prompted with the following pop up:



Can calculate either as many times as desired

This allows one to calculate a Force from a Probability (Quantile explained above in section 4) and also allows one to calculate a Return Level (Also explained above in section 4).

The initial values will be .95 probability and the return period that corresponds with .95 probability.

To calculate either simply input a Probability or a Return Period (number of observations) and click ‘Calculate Force’ or ‘Calculate Return Level and Confidence Interval’

This can be repeated as many times as desired.

When finished please click ‘Continue’ and the program will terminate.