

Primary Standards Laboratory Uncertainty Calculator User's Guide

Sandia National Laboratories¹ uncertainty@sandia.gov

January 15, 2019

¹Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

Contents

1	Introduction	3
1.1	Features	3
1.2	Implementation	4
1.3	Installation	4
2	Calculations	5
2.1	Uncertainty Propagation	5
2.1.1	Inputs	6
2.1.2	Outputs	9
2.2	Reverse Propagation	13
2.3	Sweeps	13
2.4	Data Sets and Analysis of Variance	14
2.5	Curve Fitting	15
2.5.1	Curve Fit Algorithms	16
2.5.2	Uncertainty in Curve Fitting	16
2.5.3	Use of dates in curve fitting	16
2.5.4	Fit Curve and Uncertainty	17
2.5.5	Prediction	18
2.5.6	Residual Plots	18
2.5.7	Correlations	18
2.6	Risk Analysis	18
2.7	Distribution Explorer	19
2.8	T-Table Calculator	19
3	Projects	21
3.1	Saving and Loading	21
3.2	Data sharing	21
3.3	Reports	22
3.4	Settings/Preferences	23
4	Python usage	24
5	Command line usage	25
5.1	Command Line Examples	26
5.1.1	Uncertainty Propagation	26
5.1.2	Risk Analysis	26
5.1.3	Line Fit	26
6	Example Calculations	28
6.1	RC Circuit Uncertainty	28
6.1.1	Risk Analysis	30
6.2	Contact Resistance	32

6.3	Selection of calibration provider for density measurement	33
6.4	Risk analysis of drifting asset	36
7	Copyright and License	39
	References	40

Chapter 1

Introduction

The PSL Uncertainty Calculator software was developed by the Primary Standards Laboratory (PSL) at Sandia National Laboratories. After repeatedly writing individual bits of Python code to perform uncertainty propagation calculations for different systems, the decision was made to generalize and make it available to the PSL's customers as a tool for performing GUM and Monte Carlo uncertainty propagation on any measurement system. The initial software versions were limited to GUM and Monte Carlo uncertainty propagation, and were primarily used primarily at the PSL and Sandia. The calculations had to be written in Python Code. Eventually, a simple user interface was added to remove the coding requirement. Additional features, such as risk analysis and evaluating uncertainty in curve fitting, were later added as more code was developed.

1.1 Features

The PSL Uncertainty Calculator has a number of features that differentiate it from other uncertainty software solutions:

- The GUM method is solved symbolically, providing the user with the actual formulas used, not just numerical results.
- Expanded uncertainties can be calculated automatically based on degrees of freedom and the t-distribution.
- The number of input variables is unlimited, and input variables can have multiple (e.g. Type A and Type B) uncertainty components.
- Over 90 probability distributions are available. If used as a Python module, custom distributions can also be defined.
- Multiple output functions can be computed in parallel, and any output functions may be used as the input to another measurement function.
- Uncertainty propagation can be calculated in reverse to determine how much uncertainty an individual measurement must have to meet a target uncertainty requirement.
- The ability to propagate uncertainty through arbitrary functions, including iteratively-solved functions.
- The software is based on Python, quickly becoming the most popular language for data analysis as well as communication with hardware instruments.
- Command line interface for embedding the calculator in other programming languages.
- Calculation of uncertainty in curve fitting, including arbitrary curve models.
- Calculation results and plots can be saved in HTML, PDF, Word, or Open Office formats.
- Desktop usage, no internet connection is required and all data stays local.
- Open source software so algorithms can be analyzed at the code level.

1.2 Implementation

Many of the statistics and calculations used in this software leverage the well-established computing packages available in Python, including NumPy, SciPy, and SymPy. The SymPy package allows this software to provide and solve symbolic equations for uncertainty propagation rather than relying on numeric methods. SciPy contains a vast collection of statistical methods and probability distribution definitions that are used throughout this software. The Monte Carlo simulations make use of the Mersenne Twister random number generator as implemented in SciPy, generally regarded as a quality PRNG for Monte Carlo methods.

The available calculation types are:

- Uncertainty propagation by the GUM and Monte Carlo methods
- Reverse uncertainty propagation to determine measurement uncertainty required to meet a target combined uncertainty
- Uncertainty sweep to calculate multiple uncertainty propagations over a range of inputs
- Reverse uncertainty sweep to calculate multiple reverse uncertainty propagations
- Analysis of variance on a two-dimensional data set
- Curve fitting, accounting for uncertainty in x and y measurements, and providing uncertainty in the output curve
- Risk analysis for determining probability of false accept and false reject
- T-Table calculator for determining values based on a Student's-t distribution

1.3 Installation

The calculator user interface is available on Windows and Mac operating systems. For Linux systems, install and run as a Python package. On Windows, two options are available. A standalone EXE file can be run without administrator privileges. If admin rights are available, an installer EXE can be run to install the program for faster operation. On Mac operating systems, the APP can be copied to the Applications folder and run directly from there.

Chapter 2

Calculations

When the calculator user interface is launched, the Select Calculation Type page displays (see Figure 2.1). Select one of the calculation options, or alternatively load a setup file from disk using the **Project** menu. The **Project** menu also contains options for adding additional calculations, saving the current setup, or saving a report of all output results. The **Window** menu provides options for calculating t-table values and showing a list of all calculations in the project.

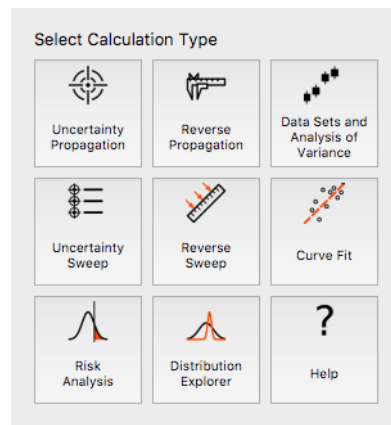


Figure 2.1: Select Calculation Type page

Upon selecting a calculation type, an additional menu displays. This menu changes based on the calculation type and provides options specific to that calculation, such as loading in specific data from a CSV file or saving a report of that calculation's output.

The following sections list the specific details of each calculation type.

2.1 Uncertainty Propagation

The uncertainty propagation function uses the method described in the GUM [1] and GUM Supplement 1 [2] to approximate the combined uncertainty u_c of a measurement function of multiple input variables $f = f(x_1, x_2, \dots, x_n)$. The GUM method is an approximation based on a first-order Taylor series expansion of the measurement function, solving for the uncertainty using:

$$u_c = \sqrt{\sum_{i=1}^N \left(\frac{\partial f}{\partial x_i} \right)^2 u(x_i)^2 + 2 \sum_{i=1}^N \sum_{j=i+1}^N \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} u(x_i, x_j)}$$

where $u(x_i)$ is the standard uncertainty in the measured variable x_i and $u(x_i, x_j)$ is the covariance between variables x_i and x_j . When possible, the partial derivatives are solved symbolically and a full symbolic expression for uncertainty can be obtained along with numeric values. In the rare case a symbolic derivative is not possible, the calculator will fall back on numerical methods for finding the derivatives.

At the same time, a Monte-Carlo uncertainty propagation is performed in accordance with GUM Supplement 1. Each of the input variables are sampled from their respective probability distributions, and passed through the measurement function to obtain a probability distribution for the output variable. Mean, median, and standard deviation statistics can be pulled from this distribution to summarize the results.

Both calculation methods also provide a method for computing expanded uncertainties. The GUM method uses the Welch-Satterthwaite (W-S) approximation to determine an effective degree of freedom and determine the resulting k-factor for a given confidence interval. It should be noted that the W-S formula is technically only valid for uncorrelated input variables, although the calculator will use it even for problems involving correlations. The Monte Carlo method determines expanded uncertainty intervals by taking either the shortest or symmetric percentiles of the output probability distribution.

2.1.1 Inputs

PSL Uncertainty Calculator - v1.2.0

Measurement Model: + -

Name	Expression	Description	Report?
1 f	X + Y		<input checked="" type="checkbox"/>

Measured Quantities:

Name	Mean	Standard Uncertainty	Degrees Freedom	Description
► Y	0	1.0	inf	
► X	0	1.0	inf	

Notes Correlations Settings

Calculate

Figure 2.2: Uncertainty Propagation Page

To enter an uncertainty propagation problem, start by defining the measurement model. Double-click on the name and the expression in the Measurement Model table to edit the values. Functions can be entered using standard format, similar to that used by Excel, using common mathematical symbols (+, -, *, /). Powers are entered using either the caret (^) symbol or a double asterisk (**) (the syntax used by the Python programming language). Along with constants “e” and “pi”, several common functions are also recognized:

- sin, asin, sinh, asinh
- cos, acos, cosh, acosh
- tan, atan, atan2 (y, x), tanh, atanh
- log (base e), ln (base e), log10 (base 10)
- sqrt
- root (e.g. root(x, 3) for cube root)
- exp
- radian, degree

Most Greek symbols are recognized by typing their names in English: “theta,” “delta,” etc. and will be displayed as symbols. After entering an expression, the field should render the expression as a math formula. If the field turns red, there was an error parsing the expression, which could mean a missing parenthesis, unrecognized function, or other error. Additional functions can be entered and calculated in parallel using the plus sign (+) and minus sign (-) buttons. These functions may be chained together, for example “f = a + b”, and “g = 2*f”.

Once a function is entered, the variables in that function are extracted to fill in the names in the Measured Quantities table. In this table, a mean, standard uncertainty, and degrees of freedom can be entered for each variable. If a variable has more than one uncertainty component (e.g. a type A and type B component), use the plus sign (+) button to add additional components. When multiple components are present, the standard uncertainty of each component is root-sum-squared together to obtain a standard uncertainty for the variable used by the GUM equation, and the degrees of freedom are combined using the W-S formula. The Monte Carlo method will separately sample each component before combining.

Measured Quantities:

Name	Mean	Standard Uncertainty	Degrees Freedom	Description	
► C ₃	0.22	0.005	inf	Capacitor 3	+ ...
▼ R ₁	32.2014	0.0036055...	15.6680835...		+
typeA		0.003	9	Type A uncertainty from resistance...	- ...
typeB		0.002	inf		- ...
► C ₂	0.1	0.05	inf	Capacitor 2	+ ...

Figure 2.3: Measured quantities with multiple uncertainty components for R1

The standard uncertainty field can also process simple mathematical expressions, including percentages based on the entered mean value. For example, with a mean of 10, entering an uncertainty of “.5 + 10%” will automatically reduce to 1.5, and be updated appropriately if the mean value is changed later.



Figure 2.4: Entering an uncertainty as a percent

When an input variable uncertainty is known as a non-standard uncertainty, such as a uniform distribution or a $k = 2$ uncertainty, the component can be customized using the ellipses (...) button. When this button is pressed, a new tab appears, showing additional parameters for defining a distribution type, such as normal, uniform, triangular, etc., along with a preview plot of the probability density function for that distribution. Normal and t distributions can be entered as uncertainty with either a k value or confidence percentage. Uniform distributions can be entered by using the half-width “a” parameter. For more information on what the parameters for each distribution are, press the question mark (?) button. The most common distributions are shown in the drop-down list box, but others can be enabled in the **Preferences** menu. The custom distribution is converted to a standard uncertainty (shown in the Measured Quantities table) for use in the GUM formula, while the Monte Carlo method directly samples the custom distribution. The Distribution tab also has a button to import distribution settings from a CSV file of sampled data or from another calculation in the program.

If the input variables are correlated, these can be entered on the **Correlations** tab (see Figure 2.5). Correlation coefficients are entered one pair at a time, and must be a value between -1 and +1.

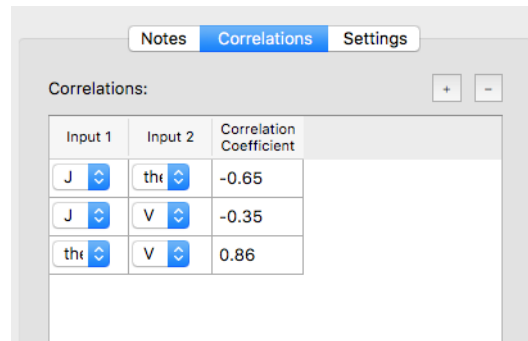


Figure 2.5: Correlation coefficient entry table

If the measurements consist of multiple measurement values stored in a CSV file, both the uncertainties and correlations can be loaded by selecting **Load Uncertainties from File** from the **Uncertainty** menu (see Figure 2.6). This will open a dialog box where each input variable can be assigned to a column of data. The mean and standard deviation of the mean, along with correlations between columns, will be automatically loaded.

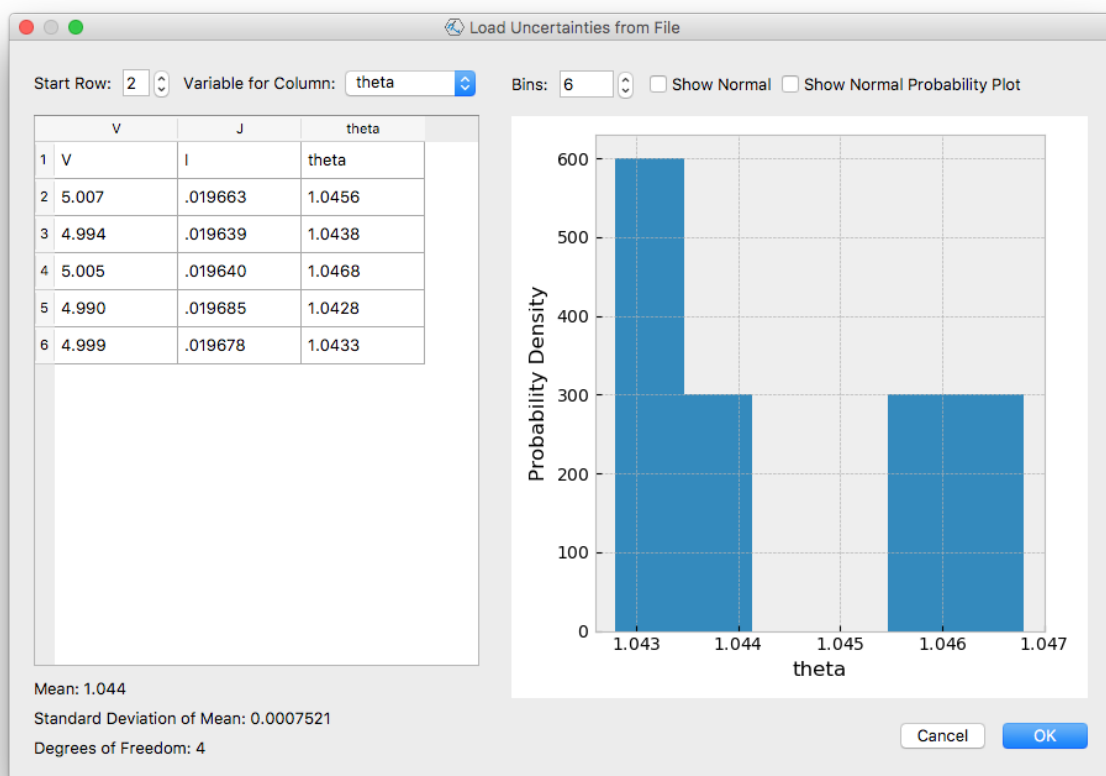


Figure 2.6: Importing measurement data into uncertainty propagation

Finally, the **Notes** tab contains a field for entering your own information to save with the calculation, and the **Settings** tab allows entry of the number of Monte Carlo samples and the random number generator seed. A seed of "None" will be randomized on every run.

2.1.2 Outputs

After entering all the details of the uncertainty propagation problem, click the **Calculate** button to run the calculation and show the results. Use the drop-down list box to select the output to view. Some of the output views are plots, which can be panned, zoomed, and saved using the plot toolbar below the figure. Others are text reports, which can be saved by right-clicking on the report.

2.1.2.1 Summary

The Summary view summarizes the results of the calculation, including expected mean values, standard uncertainty, expanded 95% uncertainty, and k factor for both the GUM and Monte Carlo methods. Additionally, a plot of the GUM method's probability distribution, along with a histogram of the Monte Carlo output distribution, displays (see Figure 2.7). If both methods produce a similar distribution, the GUM method is a valid approximation.

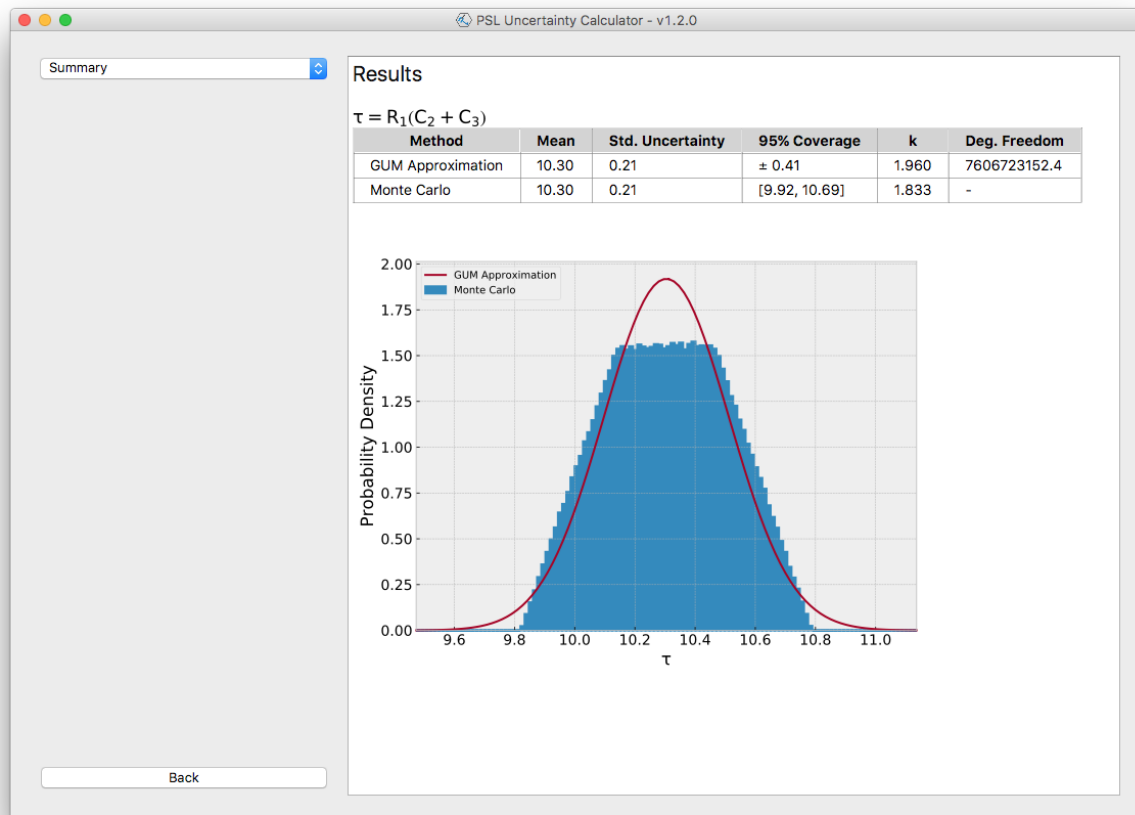


Figure 2.7: Output summary

2.1.2.2 Comparison Plots

The Comparison Plots view (see Figure 2.8) shows the same probability distribution plots in an interactive format. There are options for switching between single and joint probability (if multiple measurement functions are defined), displaying the Monte Carlo result as a histogram or an approximate PDF, and displaying coverage intervals.

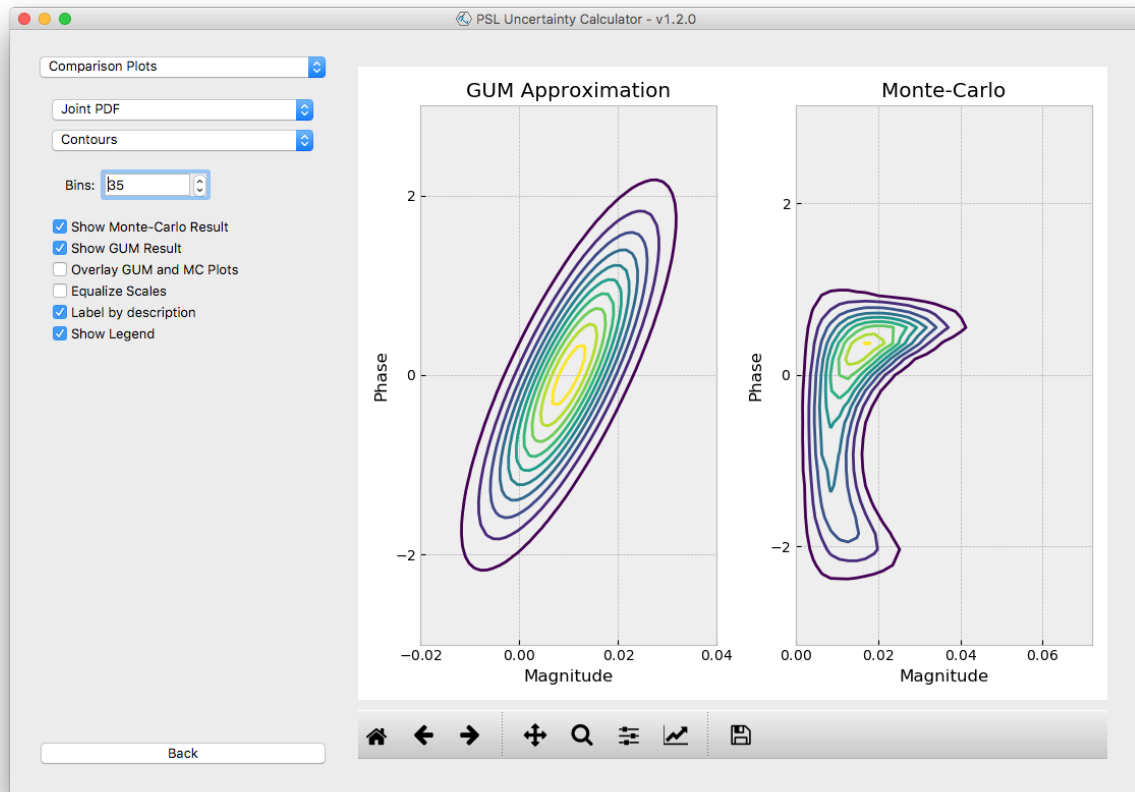


Figure 2.8: Comparison plots view in joint PDF mode (magnitude/phase example with correlation)

2.1.2.3 Expanded Uncertainties

The Expanded Uncertainties view lists a table of expanded uncertainties at predefined confidence intervals for both calculation methods. The GUM expanded uncertainty can show either a student-t expanded uncertainty, where the k-factor is determined using the effective degrees of freedom, or a Normal/k expanded uncertainty, where k is exactly 1, 2, or 3. The Monte Carlo method can show either the symmetric expanded uncertainty (e.g. the 2.5% and 97.5% percentiles for a 95% coverage), or the shortest range that covers the desired percentage of samples [3].

2.1.2.4 Uncertainty Components

The Uncertainty Components view breaks down the total uncertainty into components displayed in three tables.. It can be useful to determine which input is the largest contributor to the total uncertainty. The first table lists all the input variables and standardized uncertainties. The second table lists the uncertainty components as entered, and as combined for each variable. The third table lists the sensitivity coefficients. In the GUM method, these are the solved and evaluated partial derivatives for each input. The proportions are calculated for each variable from $(c_p^2 + u(x)^2)/u_c^2$ where c_p is the sensitivity coefficient, $u(x)$ is the variable's standard uncertainty, and u_c is the combined uncertainty of the function. The Monte Carlo method evaluates sensitivity coefficients and proportions by fixing all but one input constant at the mean values, and sampling the variable of interest. The resulting distribution is compared to the distribution of uncertainty including all inputs [3].

2.1.2.5 GUM Derivation

The GUM Derivation view displays the mathematical derivation used in calculating the GUM uncertainty (see Figure 2.9). Here, the sensitivity coefficients (partial derivatives) are solved and then combined in the uncertainty formula. Right-click on the window and select **Show markup** to obtain these formulas in LaTeX-compatible math format if desired.

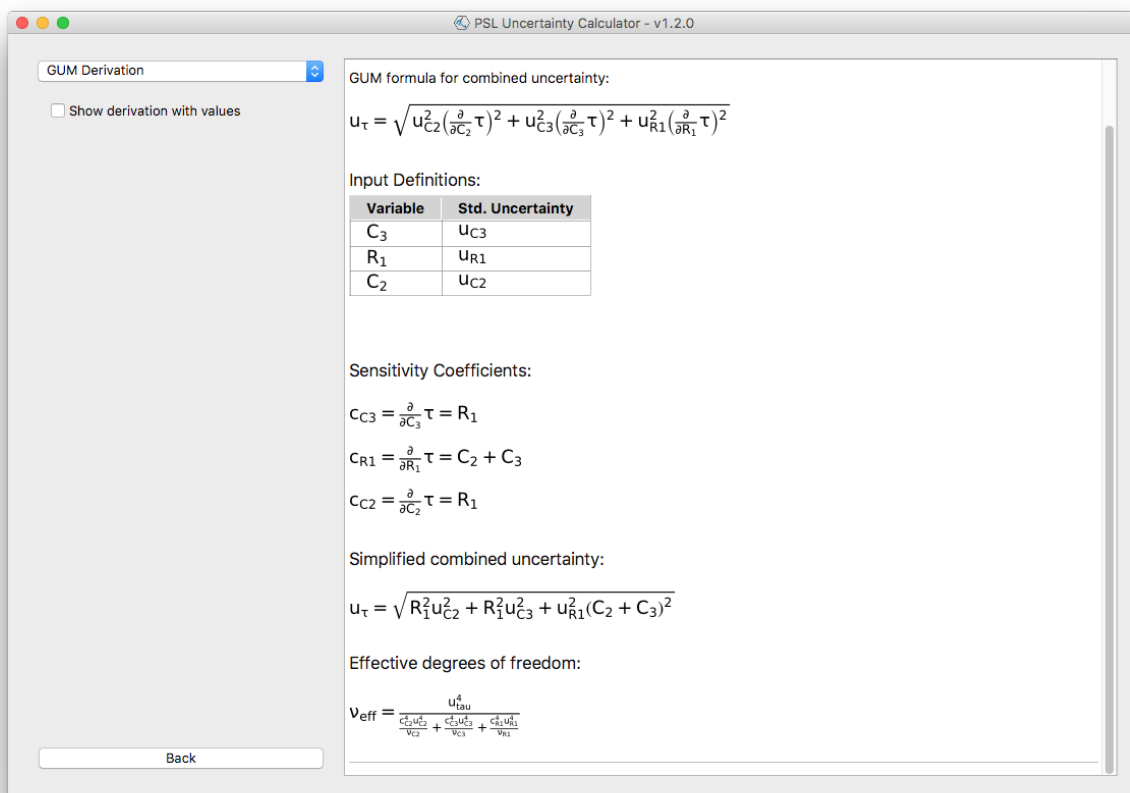


Figure 2.9: GUM derivation report

2.1.2.6 GUM Validity

The GUM Validity view follows the approach in Section 8 of GUM-S1 [2]. The endpoints of 95% confidence intervals for the GUM and Monte Carlo methods are compared to be within a specific tolerance based on the number of significant figures (usually 1 or 2). If the endpoints agree within this tolerance, the validity test passes and the GUM is a reasonable approximation to the true uncertainty.

2.1.2.7 Monte Carlo Distribution

The Monte Carlo Distribution view displays the output distribution from the Monte Carlo method. It can be compared and fit to a normal or any other available distribution. The probability plot of the distribution fit is also given. When the sample points fall on the straight line, the data fit the selected distribution well.

2.1.2.8 Monte Carlo Inputs

The Monte Carlo Inputs view displays histograms of all the sampled input variables. They can be plotted as individual histograms or as joint scatter plots to look for the effects of correlation.

2.1.2.9 Monte Carlo Convergence

The calculator runs the same number of samples regardless of the Monte Carlo simulation convergence. This plot should be checked to ensure that enough samples were run to adequately converge on a mean and uncertainty solution.

2.1.2.10 Full Report

The Full Report view combines all the previous output data into a single page.

2.2 Reverse Propagation

The Reverse Uncertainty Propagation calculator runs the same calculation as the regular Uncertainty Propagation, but in the other direction. Given a required uncertainty in the output of a measurement model, it will solve for the uncertainty in one of the input variables. This can be useful, for example, when selecting measurement equipment to use in order to meet a particular uncertainty requirement.

Reverse calculations are solved using both the GUM method and Monte Carlo method. The GUM method finds the symbolic GUM uncertainty equation and solves it for the variable of interest. The Monte Carlo method reverses the measurement function and treats the function output as another input variable. To do this, the Reverse Uncertainty calculator must account for the correlation between the model output and the original input variables. These correlation coefficients can be estimated using the sensitivity coefficients as described in the GUM [1] C.3.6 note 3.

The Reverse Uncertainty Propagation interface is similar to the standard uncertainty propagation interface, with the addition of a **Target** tab. This tab allows entry of the target (mean) value for the function and desired uncertainty for the function. The input variable to solve for is entered from the drop-down list. Any information about the solve-for variable in the Measured Quantities table is ignored.

After clicking **Calculate**, the results will display. The GUM method shows the solved-for variable's uncertainty in terms of the function uncertainty. The Monte Carlo method shows the results and a histogram of the reverse Monte Carlo simulation.

2.3 Sweeps

Uncertainty sweeps perform multiple forward or reverse uncertainty calculations over a range of inputs. The mean value or any parameter defining the uncertainty of any component may be swept over any range of values.

The Sweep interface adds a **Sweep** tab where the sweep parameters can be defined. Press the Plus Sign (+) button to add a sweep parameter. A dialog box will appear allowing you to select the sweep parameter (mean, uncertainty, degrees of freedom, or correlation), and the variable to sweep. If sweeping uncertainty, the uncertainty component and distribution parameter (such as a for uniform distributions) can be selected. A new column will appear in the Sweep table. The table values can be entered manually from the keyboard, or added as a range defined by a start, stop, and step using the Ellipses (...) button. Alternatively, the **Load Sweep Data** button can be pressed to add sweep data from a CSV file or the output of another calculation in the program. Multiple variables can be swept simultaneously.

When the calculation is run, a summary output window displays. For reverse sweeps, this window includes a table and plot of required input uncertainty for each row of the Sweep table. For forward uncertainty sweeps, a summary table displays, and the other output views available in forward uncertainty calculations are also available. Each of

these outputs present one calculation of the sweep at a time, and have a slider for selecting a specific sweep index to display (Figure 2.10).

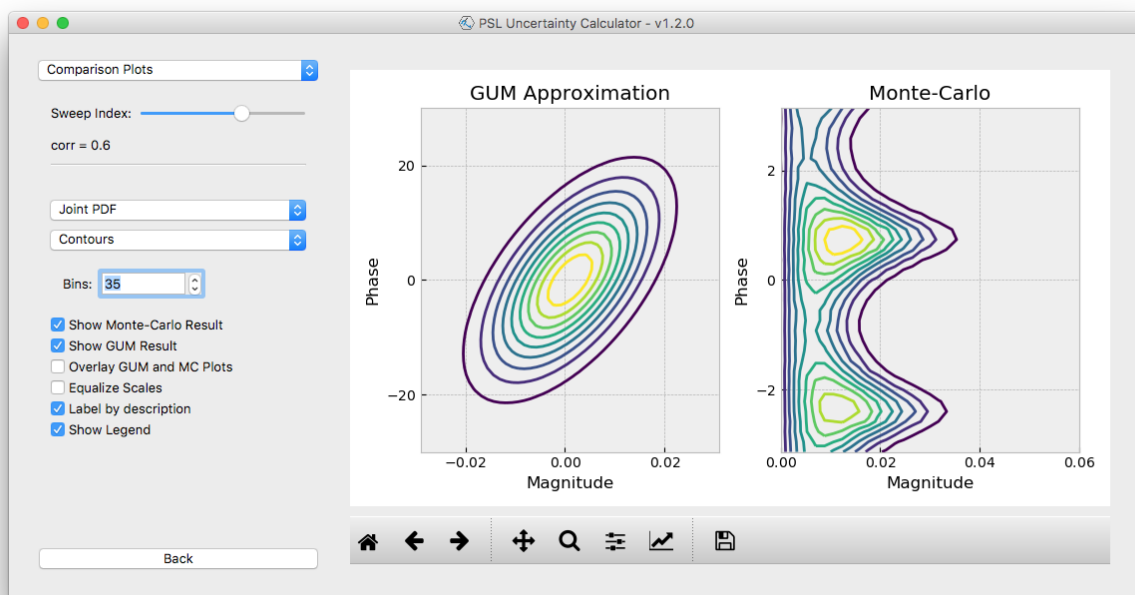


Figure 2.10: Sweep index slider, changing correlation coefficient of magnitude/phase example

2.4 Data Sets and Analysis of Variance

The Data Sets and Analysis of Variance function allows you to enter a table of data and perform Analysis of Variance (ANOVA) calculations. The data can be entered in two ways. In the first and default method, accessed by selecting **Enter all measured values**, data can be entered as a 2-dimensional data set representing multiple measurements in a group. For example, each column (group) in the table could be a different measurement temperature, while each row would list one measurement of multiples made at that temperature, or each column could be a day in which multiple measurements (rows) were taken under identical conditions. The resulting two-dimensional data will be reduced to a mean and type A uncertainty for each column/group. The second method of entering data, accessed by selecting **Enter group means and standard deviations**, is used when each group has already been reduced to a mean, standard deviation, and number of measurements. In either case, statistics on the data set, including pooled standard deviation (repeatability), reproducibility, and ANOVA information are calculated.

The data table can be entered manually, copied/pasted from Excel, or the data can be loaded from a CSV file. The CSV loader dialog does its best to determine how to interpret the data, and gives the option of grouping measurements by rows or by columns. The first row or column can be specified as the Group Value; for example, if the first row provides the temperature value at which resistance measurements were made, select this check box. A preview plot is shown (Figure 2.11), with the group number on the x axis and the individual measurements on the y axis. Measurements in the same group are plotted with the same color. This plot allows you to check that the CSV data is interpreted correctly. Note that missing data is acceptable, and will be disregarded in the ANOVA calculation; the groups do not have to be the same length.

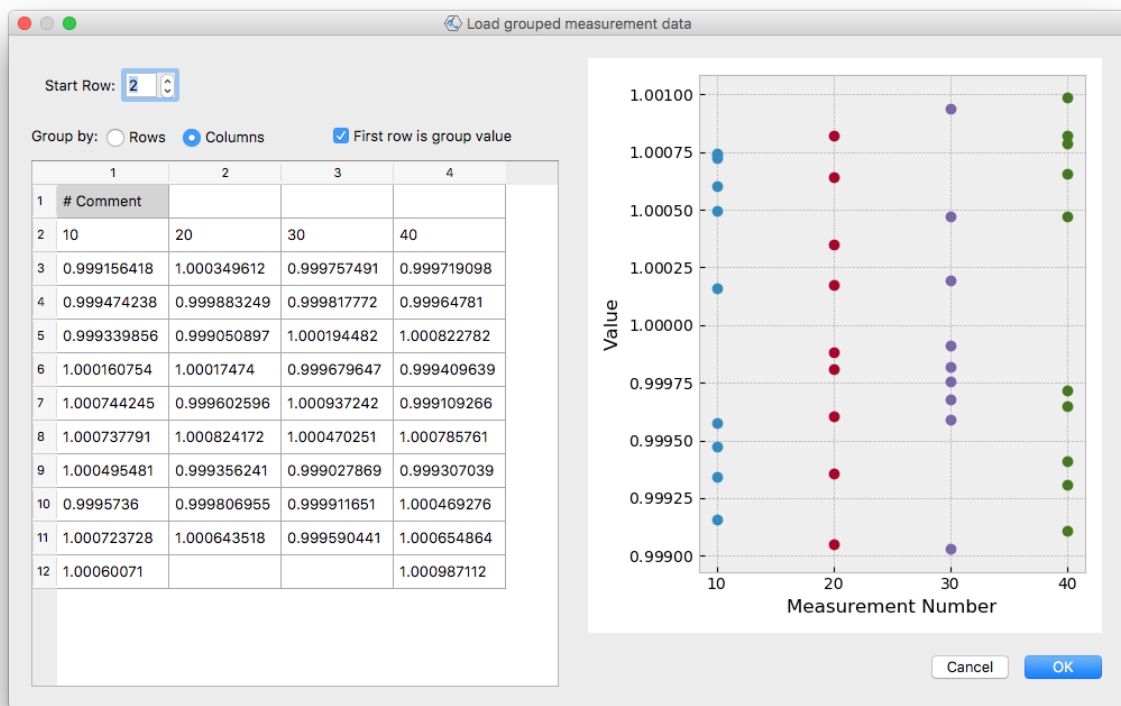


Figure 2.11: ANOVA data set import from CSV

Once the data is entered, statistics for each group (column) are provided in the first table, with statistics for the overall data set given in the second table. Results of one-way analysis of variance are displayed in the third table. Both the F-statistic and P-value are computed. The groups are statistically equivalent if the F-statistic is less than the critical F value (with 95% confidence), and the P-value is greater than 0.05.

The grouped mean and standard deviation can be used in other calculations, such as curve fitting.

2.5 Curve Fitting

The curve fitting calculation takes x and y data, with optional uncertainty in x and y, and computes the best fitting line, polynomial, or exponential function. Data can be entered manually or loaded from a file or an uncertainty sweep calculation. In contrast to other curve fitting calculators (such as in Excel), this one allows for uncertainty in both x and in y to be entered and accounted for. The model to fit can be:

- Line: $y = a + bx$
- Polynomial, up to 12th degree: $y = a + bx + cx^2 + \dots + kx^n$
- Exponential: $y = ae^{x/b} + c$
- Exponential Decay: $y = ae^{-x/b}$
- Logarithmic Growth: $y = a + b \log(x - c)$
- Logistic Growth: $y = \frac{a}{1 + \exp[(x - c)/b]} + d$
- Custom Function: Enter any one-line expression with an x variable and at least one other parameter to fit.

When run as a Python package, any arbitrary function can be fit to measured data, including iterative or other complicated functions that cannot be written in one line.

Two regression minimization options are available: vertical distances or orthogonal distances. Vertical distance regression is the typical approach to curve fitting, providing a fast and simple algorithm. Orthogonal distance regression is useful for curves with nearly vertical slopes, and also results in identical fit lines when the x and y axes are swapped. In some modes, an initial guess of the parameters can be used to speed up the minimization calculation.

2.5.1 Curve Fit Algorithms

The curve fit algorithms are:

- Line with no x uncertainty: `linefit` function described in [4]
- Line with uncertainty in x: York's algorithm, see [5] and [6]
- Arbitrary curve, no x uncertainty: Levenberg-Marquadt Algorithm, as implemented in `scipy.optimize.curve_fit`
- Arbitrary curve, with x uncertainty: Orthogonal Distance Regression, as implemented in `scipy.odr`

2.5.2 Uncertainty in Curve Fitting

There are four methods in the **Settings** tab for propagating uncertainties through the curve fitting algorithm.

- The Least Squares method is an analytical method for propagating uncertainties as defined in the above algorithms.
- The Monte Carlo method will sample every data point over its uncertainty multiple times and compute multiple possible curve fits. The parameters for the set of fits are averaged to determine the mean and uncertainty of each fitting parameter.
- The Markov-Chain Monte Carlo method uses a Bayesian approach with the Metropolis-in-Gibbs algorithm. See [7].
- The GUM method here uses the GUM uncertainty propagation formula, where the measurement function is the curve fitting algorithm and every measured x value and y value are inputs into the function. The partial derivatives in this method must be calculated numerically, as no symbolic solutions can be computed for the iterative curve-fitting algorithms.

In general, the Least Squares method is preferred, as it is analytical, fast, and the uncertainties result from the curve-fitting algorithms themselves. However, multiple methods can be run and compared.

The option **Treat uncertainties as relative values** allows entering $u(y)$ and $u(x)$ in a relative, weighted, sense. When this is checked, absolute magnitudes of the entered uncertainties are unimportant; they will be scaled to match the sample variance of the residuals after the fit. For more information, see documentation of `scipy.optimize.curve_fit`.

2.5.3 Use of dates in curve fitting

Frequently, measurements are made over the course of days or years to look for drift in a device's value. This requires using date and time values on the x axis. One easy approach is to convert every date into a decimal year, or perhaps the number of days since the first measurement was made. If traditional dates are desired, select the **X Values are Dates** check box. Data in the x column can then be entered in standard date formats, such as month-day-year. Valid entries include "9/25/2019," "9-25-19," and "25-Sept-2019." Internally, the dates are converted to ordinal time before being applied to the curve fitting algorithm, then converted back to date strings for use in the output plots and tables.

2.5.4 Fit Curve and Uncertainty

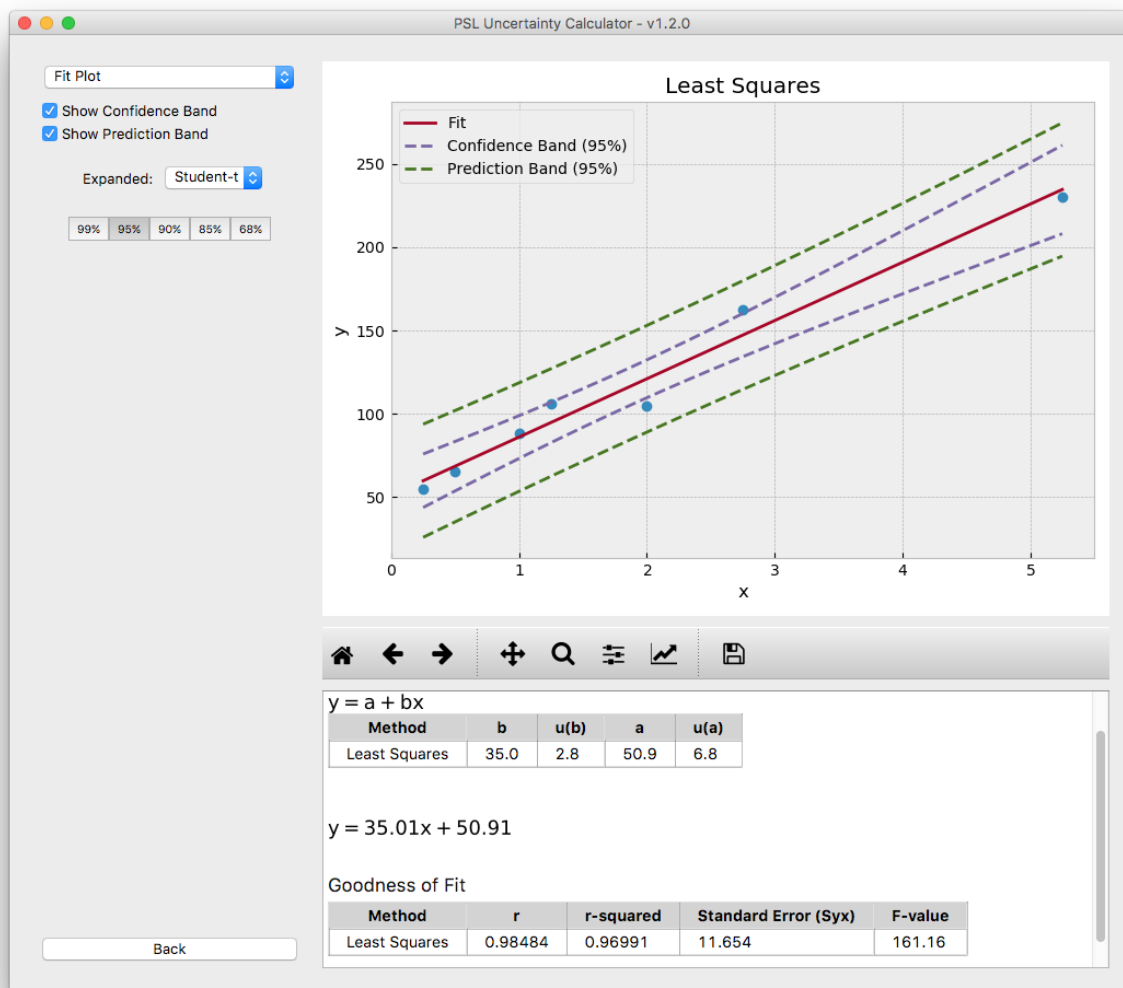


Figure 2.12: Output of curve fit calculation

After you enter the data click **Calculate**, several output views are available. The Fit Plot view shows the fit line and optionally the confidence and prediction bands, expanded to the desired confidence level. For straight line fits, the confidence band determines the confidence that the data is actually following this fit line. It is calculated from

$$u_{conf}(x) = \sigma_y \sqrt{\frac{1}{N} + \frac{(x - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

where σ_y is the uncertainty in an individual measurement, σ_B the uncertainty in the slope parameter, N the total number of measurement points, and \bar{x} the average of the measured x values. Similarly, the prediction band should be used to predict the uncertainty in a new measurement at any x value on the line. It is calculated from

$$u_{pred}(x) = \sigma_y \sqrt{1 + \frac{1}{N} + \frac{(x - \bar{x})^2}{\sum (x_i - \bar{x})^2}}$$

The σ_y value to use in the prediction band will depend on the particular problem. Three options are available from the user interface:

1. **Syx (Residuals)**. This is the default option that uses the standard error of the fit Syx and applies this value to every x-value at which a prediction is made. This is the only option when u(y) is not entered.
2. **Interpolate u(y)**. When u(y) is entered, these values can be used to determine σ_y . But since u(y) is only specified at specific x-values, σ_y must be extrapolated (linearly in this case) to the desired x-values at which predictions are made.
3. **Last u(y)**. Applies the last entered u(y) value to all x values. This option is useful, for example, when the x-axis is time and any new measurements in the future will use the same equipment uncertainty as the last measurement. The resulting prediction band should not be used to evaluate past measurements made under different measurement uncertainty conditions.

The confidence band and prediction band can be generalized to nonlinear curve fits using the gradient, where C is the covariance matrix of the fit parameters:

$$u_{conf}(x) = \sqrt{\nabla f^T \cdot C \cdot \nabla f}$$

$$u_{pred}(x) = \sqrt{u_{conf}^2 + \sigma_y^2}$$

2.5.5 Prediction

The Prediction output view allows the selection of an arbitrary x value and calculates the expected y value and associated prediction uncertainty.

2.5.6 Residual Plots

The Residual plots provide diagnostic tools on the regression. A histogram, a plot of the raw residuals, and normal probability plot can help determine whether the selected model (line, polynomial, etc.) is adequate to describe the data. If the model is adequate, the residuals are expected to fall in a roughly normal distribution centered at 0.

2.5.7 Correlations

The Correlations view shows the correlation coefficient between the fit parameters. In some problems, the slope and intercept may be highly correlated, for example.

2.6 Risk Analysis

The Risk Analysis calculation computes the risk associated with process and/or test distributions falling outside of specification limits. With only a process distribution and specification limits, the probability of a value on the distribution falling outside the limit is calculated by integrating the probability distribution function outside the limits to give the “process risk”. When a Test Measurement is added, the total probability of false accept and total probability of false reject is computed by the double integral of the product of the two distributions falling outside the limits. See [8] for the integrals when the distributions are normal. Non-normal distributions can also be entered and are integrated numerically using the same approach.

An optional guard band can be entered, as a relative offset to the specification limit; for example, with an upper specification limit of 2 and guard band of 0.1, the product will be rejected with a measurement over 1.9. The **Risk** menu provides a function for computing a guard band to achieve a desired probability of false accept. It uses

a numerical minimization technique to solve for the limits on the double integral, and so will not always converge or hit the target exactly.

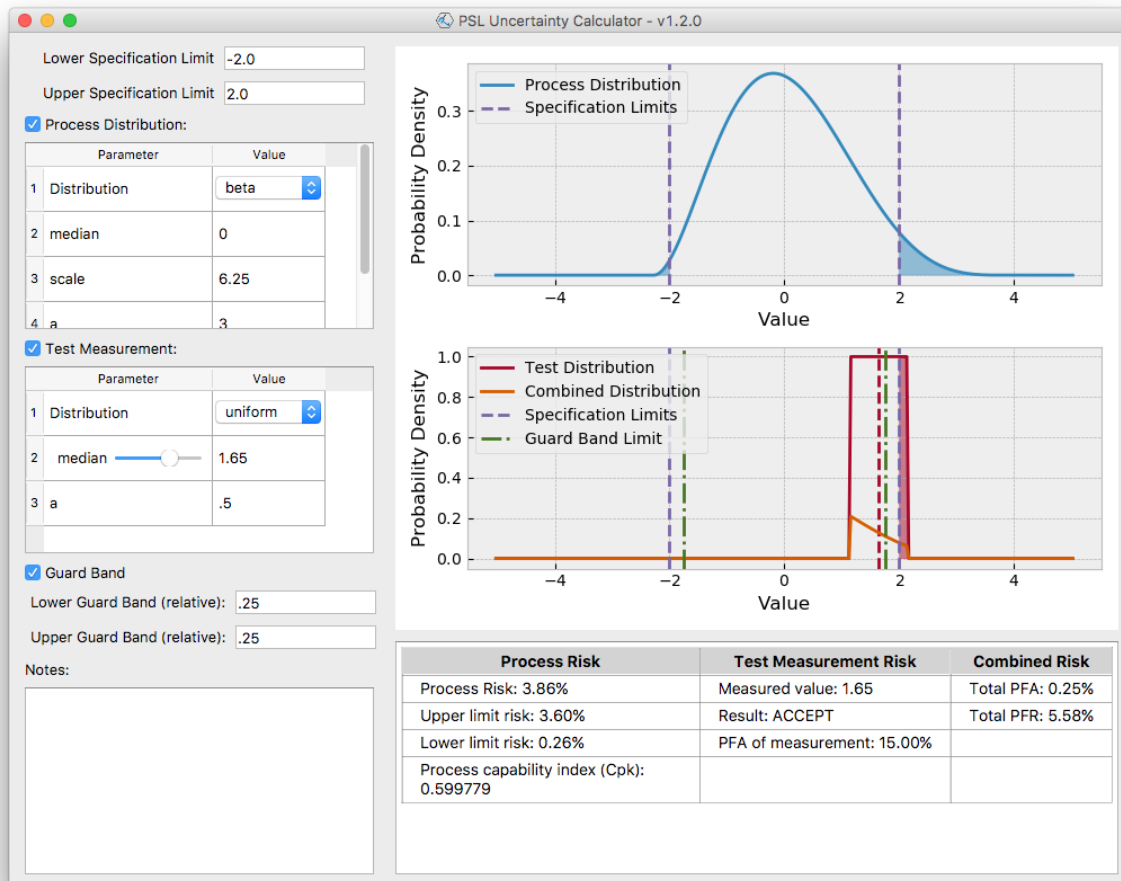


Figure 2.13: Risk analysis with beta process distribution and uniform test distribution

2.7 Distribution Explorer

You can observe different distribution parameters, draw random samples from distributions, and perform manual Monte Carlo simulations from the Distribution Explorer window. It is included in the calculator software mainly for educational purposes. To run a Monte Carlo simulation, set the expression of a distribution to a formula using previously defined expressions as inputs. Expressions should be entered using the same syntax as the Uncertainty Propagation function input. Click **Sample** to see the results. For each histogram, a distribution can be fit and normal probability plot shown to validate the fit to the data.

2.8 T-Table Calculator

From the **Window** menu, select **T-Table** for a dialog box that calculates values based on the Student-t distribution (see Figure 2.14). The calculator can solve for the coverage factor k , the confidence level, or the degrees of

freedom when given the other two parameters. A comparison plot of a normal and a t-distribution with the given degrees of freedom is also shown as a reference.

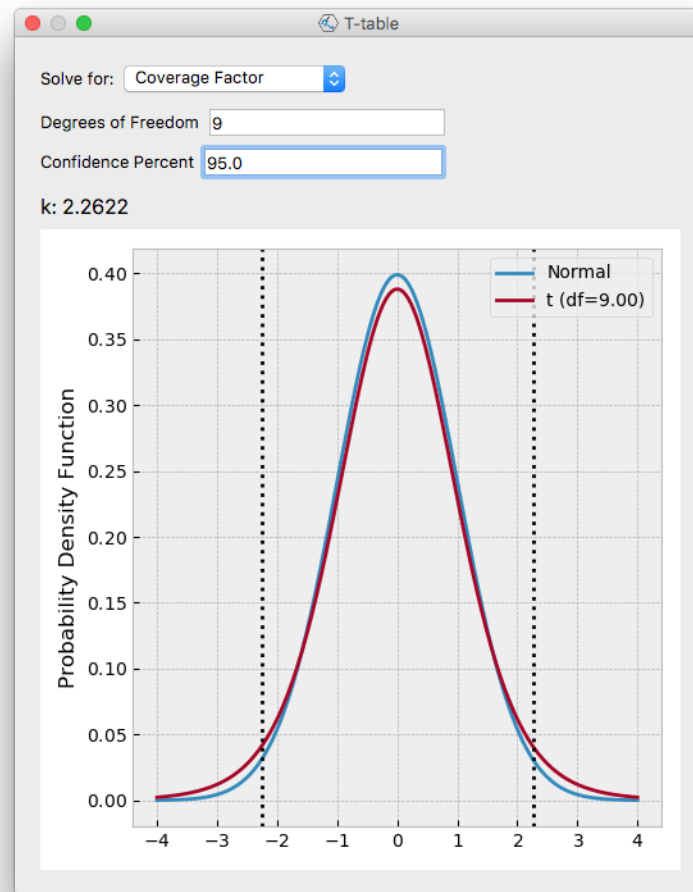


Figure 2.14: T-Table Calculator

Chapter 3

Projects

Multiple calculations can be run and the results shared between calculations using a Project. The calculations are added to the Project by selecting **Add new calculation** from the **Project** menu. All calculations are shown in the project tree (accessible from the **Window** menu). Here, calculations can be added, removed, and renamed.

3.1 Saving and Loading

The setup configuration for every calculation in a Project can be saved to a file and loaded back from a file using the **Project** menu. These files are saved in yaml (plain text) format. They can have a “.yaml” or “.txt” file extension. The **Examples** folder in the source repository contains several saved projects for reference.

3.2 Data sharing

In many calculation functions, data from one calculation result can be loaded as the input to another calculation. For example, the prediction band at a specific x date of a curve fit calculation (see Figure 3.1) can be imported into a risk analysis calculation (select **Import Process Distribution** in the menu) to determine the probability of an out-of-tolerance condition at a given date. Results of an uncertainty sweep or ANOVA data set can be loaded into a curve fit calculation (by selecting **Insert Data From** from the **Curve Fit** menu) or even another sweep calculation. Importing data is a one-time event. If the original calculation data changes, the data will need to be re-imported into the second calculation.

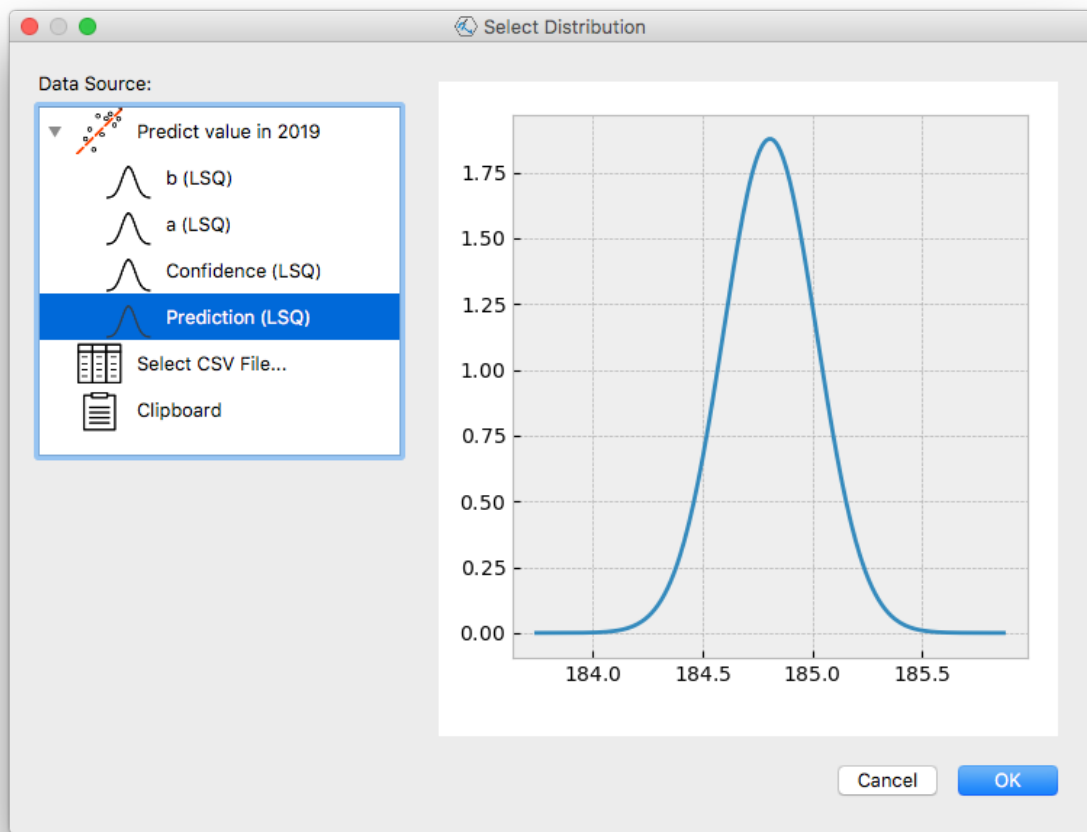


Figure 3.1: Loading the prediction band distribution of a curve fit into a risk analysis

3.3 Reports

The calculation-specific menu always contains an option to save the currently displayed report. Reports from all calculations in the Project can be saved to the same file from the **Project** menu. Right-clicking on any text-based output report displays a Save option in the context menu for saving that particular output.

Reports can be saved in several formats, including HTML, Markdown, PDF, Open Office ODT, and Microsoft Word. HTML and Markdown reports are always available. Open Office, Word, and PDF all require Pandoc to be installed to run the document conversion. The PDF format also requires LaTeX. Under Windows or Linux, the `pandoc` and `pdflatex` programs must be available on the system PATH for these formats to be available. On Mac OSX, Pandoc will be found if it was installed using the standard installer, and `pdflatex` will be found if installed from MacTeX. Alternatively, the **Preferences** window allows entry of customized installation locations.

Figures in HTML and Markdown can be drawn in either SVG (vector) or PNG (raster) format. SVG will provide the best results but may not work on older browsers. Math can be rendered using Matplotlib, which does not require any internet connection, or using the JavaScript MathJax script, which by default is downloaded from the public MathJax server. In either case, the resulting HTML file is stand alone, and all graphics and equations are embedded in the HTML/Markdown file itself.

3.4 Settings/Preferences

A **Preferences** window allows you to change some common settings, including:

- Plot style, colors, and other plotting parameters
- Significant figures and number formats to show in reports
- Probability distributions to show in drop down lists

These settings will be in place the next time the program is opened.

Chapter 4

Python usage

The back end of the calculator can be installed and run as a Python package to allow greater flexibility in defining the calculations, loading input data, and processing the results. Running as a package allows for uncertainty propagation through arbitrary functions, curve fitting to any curve, and loading/saving data to any file format available to Python. To install the Python package, open a terminal window in the uncertainty calculator source folder and run

```
python setup.py install
```

The calculator can be imported into your code with

```
import psluncert
```

Refer to the docs folder of the source code repository for examples in Jupyter Notebook format.

Chapter 5

Command line usage

Once the Python package is installed, a command-line interface to the calculator is available. This allows for the calculator's algorithms to be used within other programming languages and scripts such as LabView, Metcal, or R. The following command line programs are installed with the Python package:

- `psluncertui`: Launches the user interface
- `psluncert`: Calculates an uncertainty propagation
- `psluncertf`: Calculates all items in an uncertainty project file
- `psluncertrev`: Reverse uncertainty propagation
- `psluncertrisk`: Risk analysis calculation
- `psluncertfit`: Curve fit uncertainty calculation

Each command can be run with the '-h' flag to see all the arguments for that command. For example, running `psluncert -h` prints:

```
usage: psluncert [-h] [--variables VARIABLES [VARIABLES ...]]
               [--uncerts UNCERTS [UNCERTS ...]]
               [--correlate CORRELATE [CORRELATE ...]] [-o O]
               [-f {html,txt,md}] [--samples SAMPLES] [-s] [--verbose]
               funcs [funcs ...]
```

Compute combined uncertainty of a system.

positional arguments:

funcs Measurement model functions (e.g. "f = x + y")

optional arguments:

```
-h, --help                      show this help message and exit
--variables VARIABLES [VARIABLES ...]
                               List of variable measured values (e.g. "x=10")
--uncerts UNCERTS [UNCERTS ...]
                               List of uncertainty components, parameters separated
                               by semicolons. First parameter must be variable name.
                               (e.g. "x; unc=2; k=2")
--correlate CORRELATE [CORRELATE ...]
                               List of correlation coefficients between pairs of
                               variables. (e.g. "x; y; .8")
-o O                            Output filename. Extension determines file format.
-f {html,txt,md}                Output format for when output filename not provided
                               ['txt', 'html', 'md']
```

```

--samples SAMPLES    Number of Monte Carlo samples
-s                  Short output format, prints values only. Prints (GUM
                    mean, GUM std. uncert, GUM expanded, GUM k, MC mean,
                    MC std. uncert, MC expanded min, MC expanded max, MC
                    k) for each function
--verbose, -v        Verbose mode. Include plots with one v, full report
                    with two.

```

All command-line programs (except the user interface) take arguments **-o**, **-s**, and **-v** to control the output. Use **-s** to provide short, abbreviated output, printing the numbers only to stdout, for easy parsing by external programs. Use the **-v** or **-vv** option to report additional information in the output. The **-o** option, followed by a filename, will save the output report to the file in the format specified by the file extension, which can be html, pdf, docx, or odt.

5.1 Command Line Examples

5.1.1 Uncertainty Propagation

Calculate uncertainty of model $f = ab + c$ with a and c as normal distributions and b as a uniform distribution. Additionally, add correlation coefficients between a and b and between b and c:

```

psluncert "f=a*b+c" --variables "a=10" "b=5" "c=3" --uncerts "a; std=1"
"b; dist=uniform; a=.5" "c; unc=3; k=2" --correlate "a; b; .6" "c; b; -.3"

```

Output:

```

f = a b + c

Method          |Mean |Standard Uncertainty
-----|-----|-----
GUM Approximation |53.0 |7.1
Monte Carlo      |53.2 |7.1

```

Adding the '-s' option results in:

```

53, 7.09265572, 13.9013666, 1.95996636, 53.1622797, 7.07130974, 40.0791191, 67.2999898, 1.92474037

```

5.1.2 Risk Analysis

Compute risk given specification limits of -8 and +8, with a process distribution standard deviation of 4 and test distribution standard deviation of 1.

```

psluncertrisk --procdist "loc=0; scale=4" --testdist "loc=0; scale=1" -LL -8 -UL 8

```

Output:

```

Process Risk          |Test Measurement Risk    |Combined Risk
-----|-----|-----
Process Risk: 4.55%   |Measured value: 0         |Total PFA: 0.78%
Upper limit risk: 2.28%|Result: ACCEPT            |Total PFR: 1.47%
Lower limit risk: 2.28%|PFA of measurement: 0.00%|
Process capability index (Cpk): 0.666667 |

```

5.1.3 Line Fit

Fit a line through the points (1, 0.5), (2, 1.2), (3, 1.8), (4, 2.4), (5, 2.9), (6, 3.6) using Least-Squares fit:

```

psluncertfit -x 1 2 3 4 5 6 -y .5 1.2 1.8 2.4 2.9 3.6

```

Output:

Method	b	u(b)	a	u(a)
Least Squares	0.606	0.014	-0.053	0.053

Adding the '-s' option results in:

0.605714286, -0.0533333333

0.0135023304, 0.052584022

Chapter 6

Example Calculations

6.1 RC Circuit Uncertainty

In this example, a Resistor-Capacitor (RC) circuit is constructed from off-the-shelf components. The time constant of the circuit is calculated with

$$\tau = R_1(C_1 + C_2).$$

The components are given with tolerances, and thus are interpreted as uniform distributions.

Component	Value	Uncertainty
R_1	5 k Ω	1%
C_1	0.22 μ F	5%
C_2	0.1 μ F	1%

To determine the time constant and its uncertainty using the calculator, set up a new Uncertainty Propagation calculation. Enter the measurement model name (τ) and expression ($R*(C1+C2)$) as shown in Figure 6.1. The Measured Quantities will fill in with the variable names.

The screenshot shows a table with two columns: 'Name' and 'Expression'. The first row has the index '1' in a grey box, followed by the symbol τ in the 'Name' column, and the expression 'R*(C1+C2)' in the 'Expression' column. The 'Expression' cell is highlighted with a blue border.

	Name	Expression
1	τ	R*(C1+C2)

Figure 6.1: Entering the time constant measurement model

Next, enter the nominal value for each component in the Mean column. Note that because resistance was entered in kilo-ohms and capacitance in microfarads, the resulting time constant will be in milliseconds. Since the components are all uniform distributions, press the Ellipses (...) button in each row and change the distribution to **uniform**. The “a” parameter is the half-width of the distribution. In this case, “a” can be entered as a percent (Figure 6.2). Optionally, add a description for each variable and the measurement model.

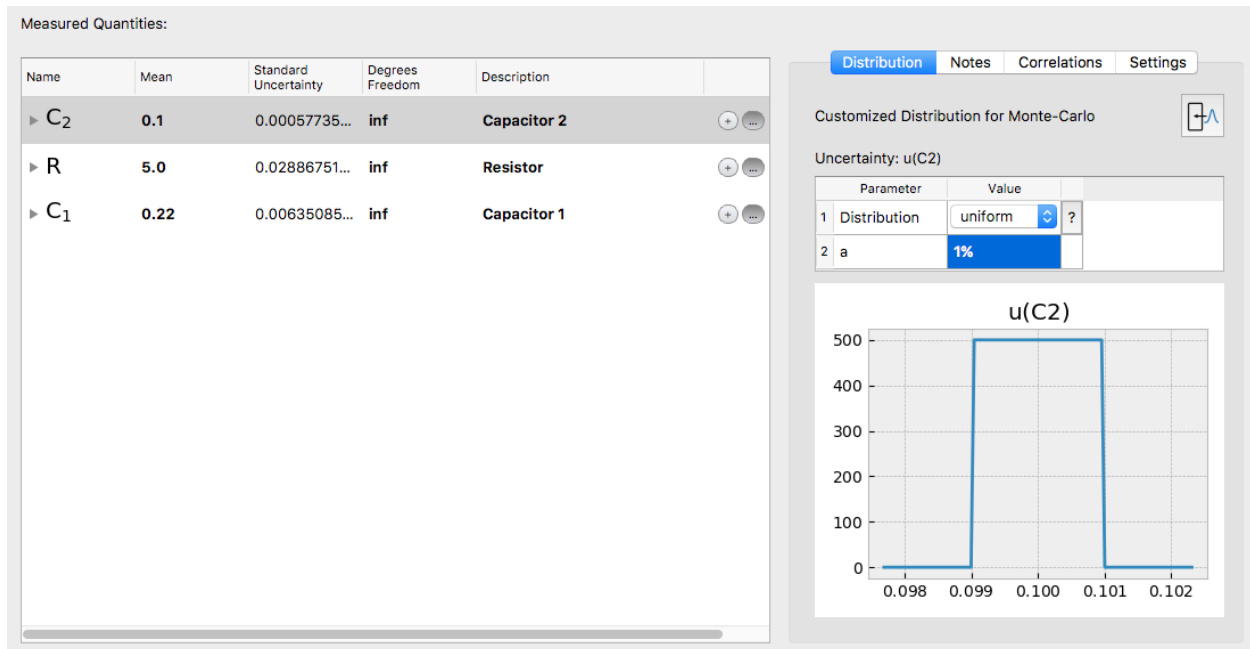


Figure 6.2: Setting input variable quantities

Now that the model is fully entered, click **Calculate** to find the total uncertainty. One can immediately see the GUM method and Monte Carlo method produce similar standard uncertainty values (0.033 ms), yet give very different probability distribution shapes. This is because the GUM method is normalizing all the uniform distributions before calculating its solution.

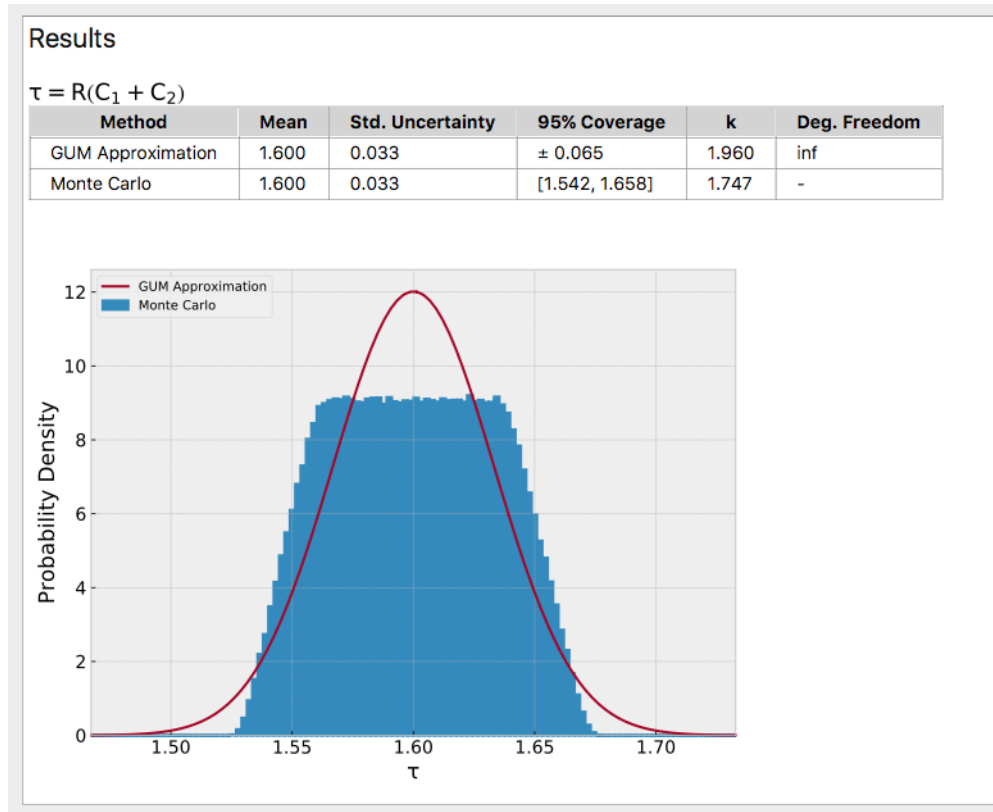


Figure 6.3: Time constant uncertainty results

6.1.1 Risk Analysis

Many circuits using these components will be manufactured with the specification that the time constant must be between 1.55 and 1.65 ms. What is the probability that any given circuit will fall outside the specification?

To answer this question, add a Risk calculation. From the **Project** menu, select **Add Calculation** and choose **Risk Analysis**. The previous uncertainty propagation problem provides the Process Distribution needed for the risk calculation. This distribution can be automatically copied into the Process Distribution using the **Risk** menu and selecting **Import process distribution**. From the dialog box, different distributions that have already been calculated are available for loading into the Risk calculation. Double click on **tau (MC)** to select the time constant Monte Carlo results (see Figure 6.4). With sampled data results from a Monte Carlo, different distributions can be fit to the histogram by selecting different options from the **Distribution to Fit** drop-down list box. To approximate the distribution using a histogram, select **histogram** from the list of distributions to fit, then click **OK**. The histogram probability distribution will be loaded into the process distribution in the risk calculator.

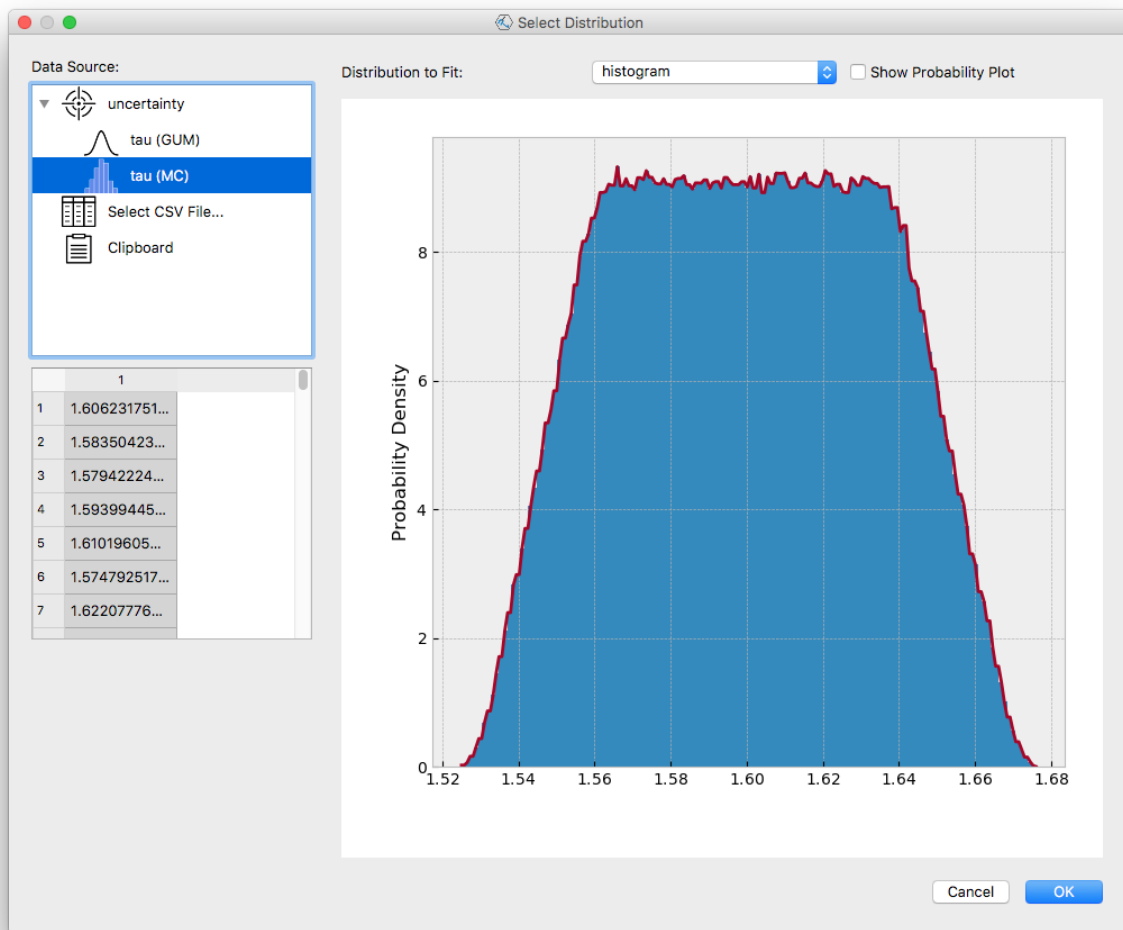


Figure 6.4: Using uncertainty propagation result in risk calculation

Enter the lower and upper specification limits of **1.55** and **1.65**. In the Process Risk column of the table, the total process risk is calculated as 12.73% (your results may vary slightly due to the random nature of the Monte Carlo simulation). This is the probability that any given part made from these components will fall outside the acceptable time constant limits.

Next, the RC circuits are each measured to determine whether they fall within the acceptance limits. The measurements are made with a measurement uncertainty of 0.01 ms. This corresponds to the Test Measurement distribution. Enter a test measurement standard deviation of **0.01** and median of **1.6** (see Figure 6.5). Now the Combined Risk column in the table is calculated to give the total risk of measuring, yet accepting, a bad part (probability of false acceptance [PFA]), and the risk of rejecting a good part (probability of false reject [PFR]).

The slider control next to Median in the test measurement can be used to explore results of a specific test measurement. The combined risk calculation integrates the combined distribution over all possible median values.

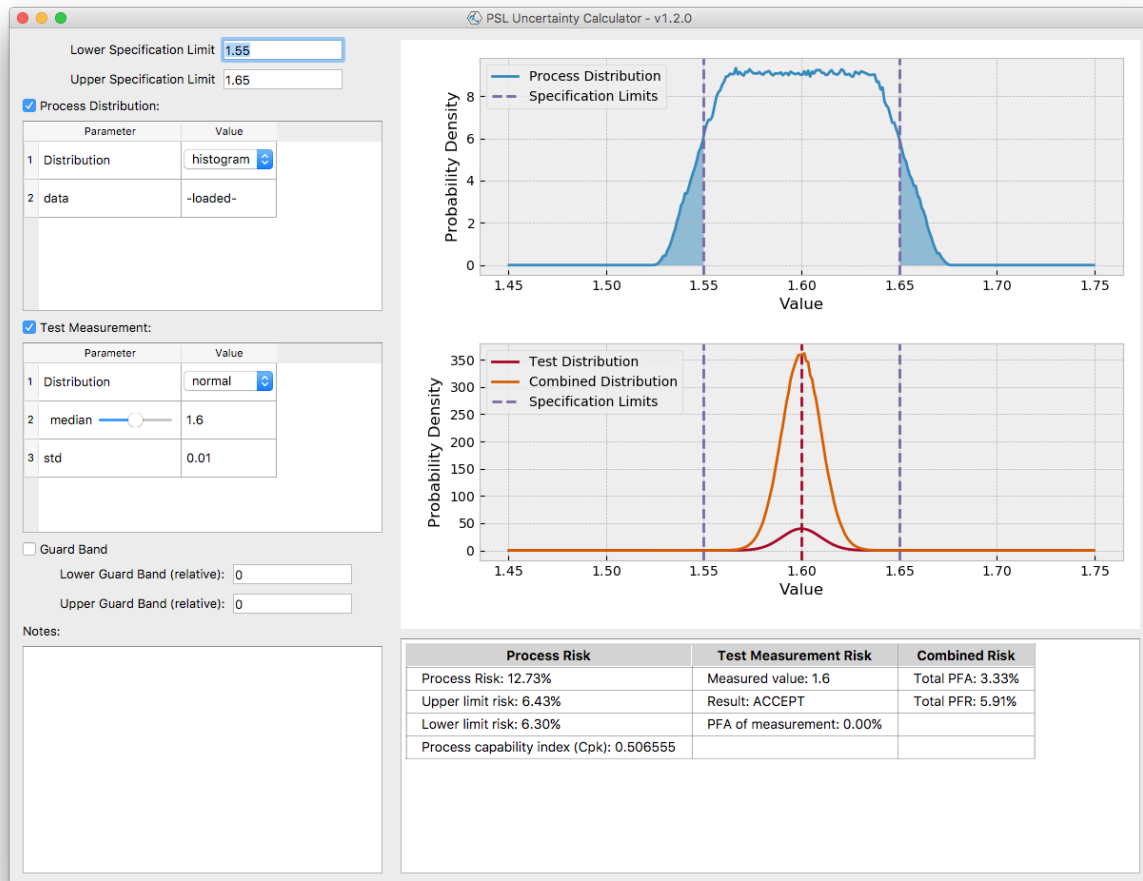


Figure 6.5: Risk of accepting a bad RC circuit part

6.2 Contact Resistance

When new nanoscale materials, such as semiconductor nanowires, are developed, one of the key properties that needs to be characterized is the resistance between the nanowire and the electrical contacts made to it. One common technique to measure the contact resistance is to place multiple contacts along the nanowire with different spacings between them. The resistance between each pair of contacts is measured and plotted as a function of length. A line is fit to the data points and extrapolated back to zero length. The y-intercept of this point should include only resistance from the two contacts, and not the nanowire itself.

The Uncertainty Calculator can be used to both fit the line and determine the uncertainty in extracting the y-intercept. Start with a Curve Fit calculation. In the **Curve Fit** window, enter x values of the different lengths between contacts, and y values of the resistance measurement. If uncertainty in the resistance is known, those values can be entered in the u(y) column. Note that the x and y column labels can be changed to Length and Resistance for properly labeled plots. In this example, measured length and resistance values are saved in the "contactresistance.txt" file in the Examples folder. Import the data in this file by selecting **Insert Data From** from the **Curve Fit** menu, double-clicking **Select CSV File**, and selecting the contactresistance.txt file.

After clicking **OK**, the data will display in the preview plot. The curve fit function and options can be selected. Here, the default of fitting a line using vertical distances is sufficient.

Click **Calculate** to display the fit line and its prediction band. The slope and intercept, their uncertainties, and goodness-of-fit parameters are shown. The parameter of interest, however, is a prediction of the line at zero length. To obtain this parameter, select **Prediction** from the drop-down list box, and enter an x-value of **0** in the X-Values table. The predicted value and its uncertainty are plotted as the orange errorbar and shown in the table (see Figure 6.6). For these measurements, the contact resistance ($\times 2$ for the 2 contacts) is 50.91 kOhm with 95% uncertainty of 34.36 kOhm. Dividing by two to obtain the resistance of a single contact gives

$$R_c = 25.5 \pm 17.3 \text{ k}\Omega.$$

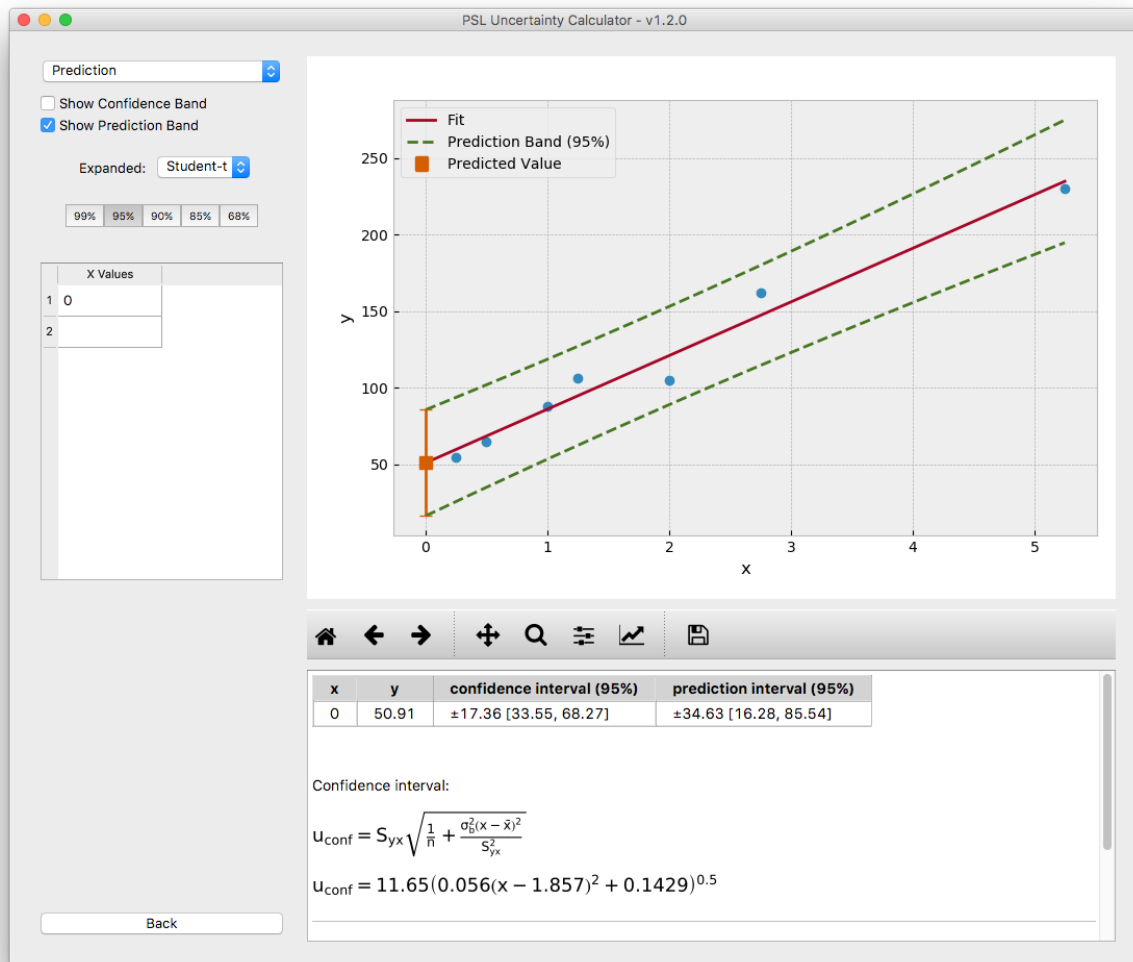


Figure 6.6: Determination of contact resistance and uncertainty

6.3 Selection of calibration provider for density measurement

The density of a gage ball must be calibrated by measuring the diameter and mass. Density is found using the formula:

$$\rho = \frac{6 \times 10^9 m}{\pi d^3}$$

where mass m is given in milligrams and diameter d in microns. The 10^9 factor converts units into grams and centimeters, resulting in a density in g/cm^3 . The nominal values of the gage ball are 86.03 g and 22.225 mm, leading to a nominal density of $14.967 g/cm^3$. A calibration provider must be found who can meet a combined density uncertainty of $0.04 g/cm^3$ ($k = 2$).

This problem involves using a reverse uncertainty propagation in order to find the input uncertainty to meet an output uncertainty requirement. Because there are two free variables (density and mass), there are actually infinite solutions to the problem. However, one variable can be swept while solving the reverse uncertainty requirement for the other variable, and any solution falling under the resulting curve will be acceptable to meet the maximum density uncertainty.

To enter this setup into the Calculator, add a **Reverse Sweep** to a new Project. Enter the measurement model and nominal quantities for mass and diameter. Next, enter the target uncertainty under the **Target** tab. The function to solve is **rho**, the target value is **14.967**, and target uncertainty is **0.02**. Note the $k = 1$ value should be entered here. Solve for the mass variable (Figure 6.7), and then use the **Sweep** tab to set up a sweep of uncertainties on the diameter variable.

The screenshot shows the PSL Uncertainty Calculator v1.2.0 interface. The 'Measurement Model' section contains a table with one entry:

Name	Expression	Description	Report?
1 ρ	$\frac{6000000000.0m}{\pi d^3}$	Gage Ball Density, diameter in um, mass in mg, density in g/cm^3	<input checked="" type="checkbox"/>

Below this is the 'Measured Quantities' section with a table:

Name	Mean	Standard Uncertainty	Degrees Freedom	Description
▶ d	22225.0	5.0	inf	Diameter measurement in micron
▶ m	86030.0	10.0	inf	Mass measurement in mg

On the right, the 'Target' tab is active, showing the following settings:

- Function: rho
- Target Value: 14.967
- Target Uncertainty: 0.02
- Solve For: m

A 'Calculate' button is located at the bottom right of the interface.

Figure 6.7: Configuring the density calculation

Under the **Sweep** tab, press the Plus Sign (+) button to add a new sweep column. The Sweep Parameter is **Uncertainty**, variable is **d**. Because this diameter is a single standard normal uncertainty, there is only one option for component and parameter. After clicking **OK**, a new column is added to the Sweep table. Press the Ellipses (...) button to enter a range of values for diameter uncertainty to sweep. Set the range from 1 to 9 with count of 9.

Unc(d, u(d), std)	
1	1.0
2	2.0
3	3.0
4	4.0
5	5.0
6	6.0
7	7.0
8	8.0
9	9.0
10	

Figure 6.8: Entering sweep values

After clicking **Calculate**, nine reverse uncertainty propagations are run for each of the nine values in the Sweep table. The results are shown (Figure 6.9) with a plot of mass uncertainty vs. diameter uncertainty. Any calibration provider falling below this line is acceptable to meet the density uncertainty requirement. Note you may want to use the **Preferences** menu to increase the number of significant digits shown. By default, uncertainties are displayed with 2 significant figures (by recommendation of the GUM), so the first few lines of this table may appear identical.

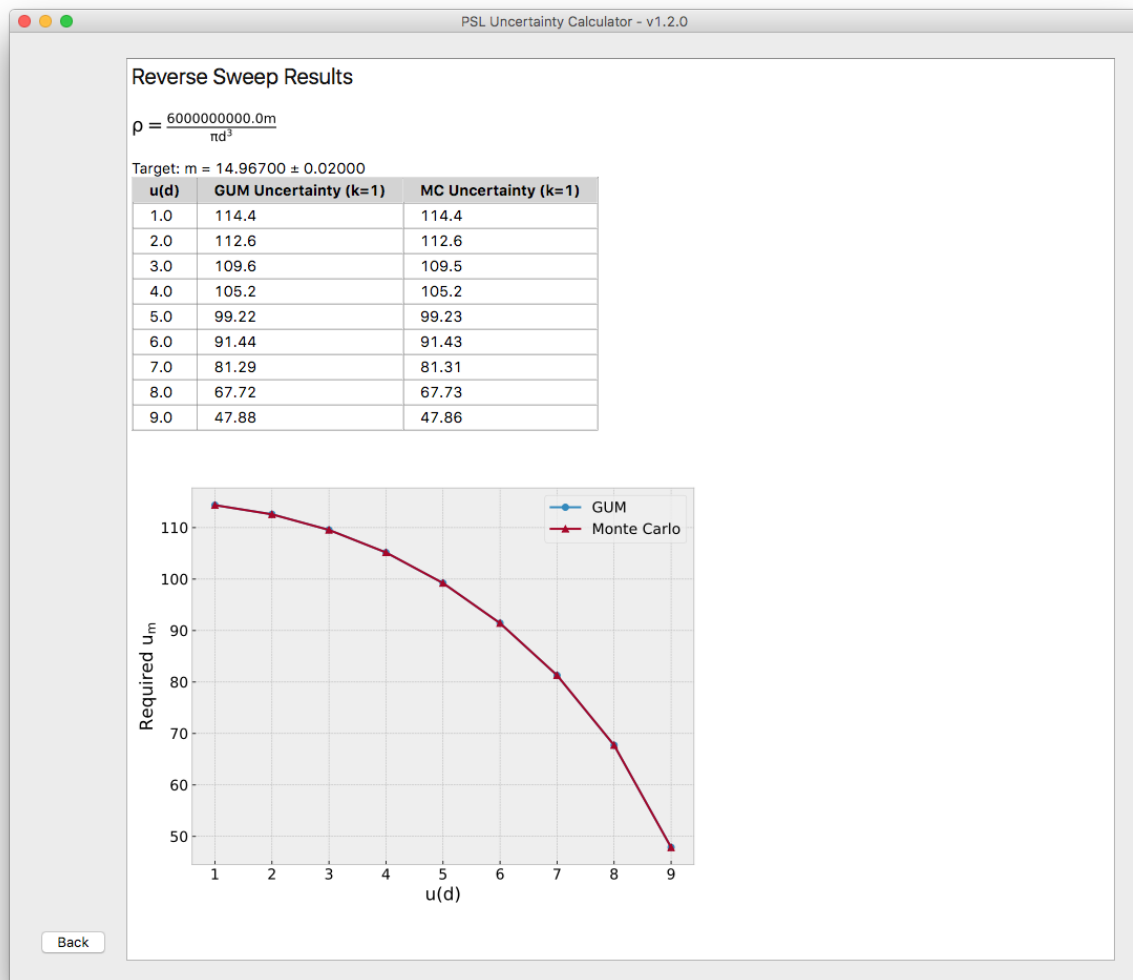


Figure 6.9: Reverse sweep calculation output

6.4 Risk analysis of drifting asset

A pressure gauge was calibrated five times and is showing predictable drift. A value between 175 and 185 PSI is considered in tolerance. As the gauge is drifting closer to the upper limit, what is the probability that it will drift out of tolerance by the next calibration due date?

The gauge was calibrated five times, each with five measurements. This data is stored in the "pressure.txt" file in the Examples folder. Because this is two-dimensional data, start with a Data Sets and Analysis of Variance calculation. From the **ANOVA** menu, choose **Load Table from CSV** and select the "pressure.txt" file. The data is laid out in rows, where the first column is the calibration date and the remaining columns are repeated measurements on that date. For the Calculator to interpret the data, select the **Rows** Group By option and check **First column is group value**. The plot should update to show dates along the x axis and values from 180 to 184 on the y axis (see Figure 6.10). Click **OK** to load the data. The plot on the Data sets and ANOVA window now display a single mean value with uncertainty bars at each date, effectively finding the mean and uncertainty of the five measurements on each date. Select the **Group names are dates** check box to indicate that each column can

be interpreted as a date and press **Calculate**.

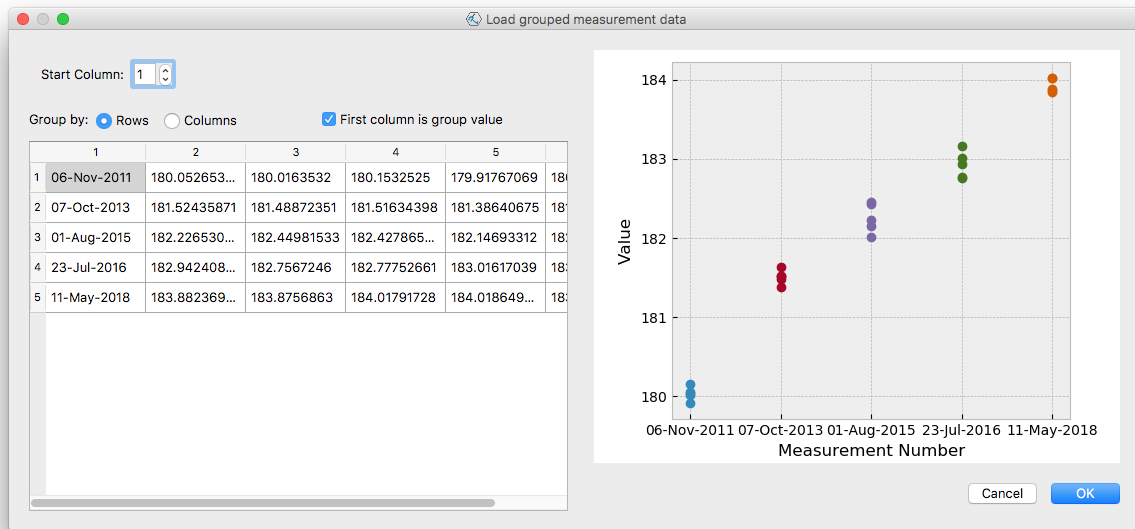


Figure 6.10: Loading grouped data

To predict the drift over the next year to the calibration due date of July 1, 2019, select **Add Calculation** from the **Project Menu** and select **Curve Fit**. Import the data we just loaded into the curve fit by selecting **Insert Data From** from the **Curve Fit** menu. Double-click the **Grouped Statistics** under the ANOVA data source. The Calculator will predefine the first columns as x, y, and u(y). Click **OK** to load the data, then click **Calculate** to find the line fit.

Next, use the fit line and its uncertainty to calculate the probability of the pressure gage falling outside the 185 PSI limit on the next calibration date. Add a new Risk Analysis to the calculator project by selecting **Add Calculation** from the **Project** menu and clicking **Risk Analysis**. There is no test measurement in this risk case, so deselect **Test Measurement**. From the **Risk** menu, select **Import Process Distribution From** and double-click the **Prediction (LSQ)** under **curvefit** in the data source. A window will appear prompting you to enter the prediction date. Enter the next calibration due date of **07-01-2019**. A normal distribution should be displayed, determined from the prediction band on that date. Click **OK** to accept this distribution. Next, enter the specification limits of **175** and **185**. The Process Risk will calculate a 15.7% chance of the gauge being above 185 on this date based on historical drift (Figure 6.11). This risk may be high enough to warrant an adjustment or repair to the gauge to prevent it from drifting out of specification during the next interval.

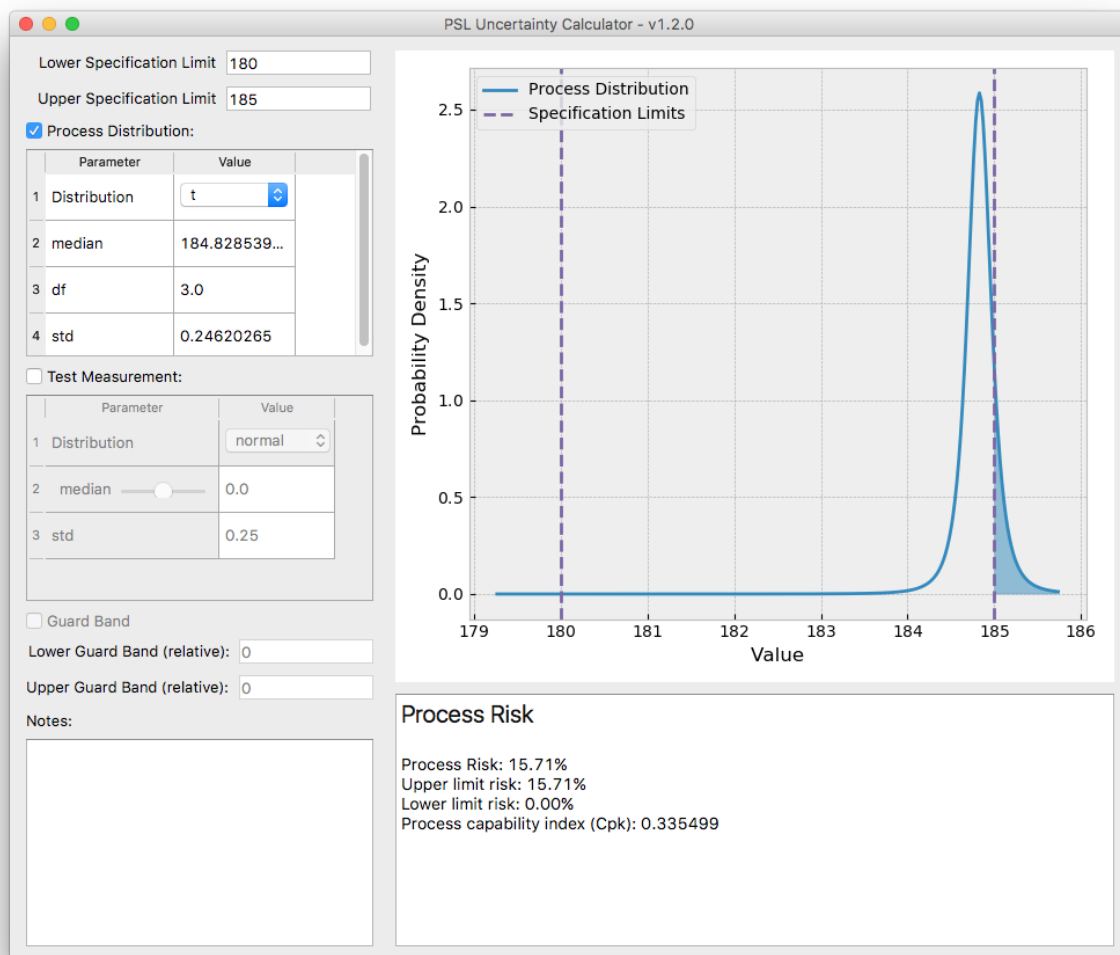


Figure 6.11: Risk of drifting out of spec

Chapter 7

Copyright and License

Copyright 2019 National Technology & Engineering Solutions of Sandia, LLC (NTESS). Under the terms of Contract DE-NA0003525 with NTESS, the U.S. Government retains certain rights in this software.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

References

- [1] BIPM, "Evaluation of Measurement Data - Guide to the Expression of Uncertainty in Measurement," Joint Committee for Guides in Metrology, 2008.
- [2] BIPM, "Evaluation of Measurement Data - Supplement 1 to the Guide to the Expression of Uncertainty in Measurement, Propagation of Distributions using a Monte Carlo Method," Joint Committee for Guides in Metrology, 2008.
- [3] H. Cox, "Software specifications for uncertainty evaluation," National Physical Laboratory, 2006.
- [4] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*. Cambridge University Press, 1988.
- [5] D. York, N. M. Evensen, M. L. Martinez, and J. De Basabe Delgado, "Unified equations for the slope, intercept, and standard errors of the best straight line," *American Journal of Physics*, vol. 72, no. 3, pp. 367–375, 2004.
- [6] R. Wehr and S. R. Saleska, "The long-solved problem of the best-fit straight line: Application to isotopic mixing lines," *Biogeosciences*, 2017.
- [7] C. Elster and B. Toman, "Bayesian uncertainty analysis for a regression model versus application of GUM Supplement 1 to the least-squares estimate," *Metrologia*, vol. 48, no. 5, p. 233, 2011.
- [8] D. Deaver and W. Everett, "How to maintain your confidence," in *Proc. 1993 NCSL Workshop and Symposium*, 1993.