

## GAUSSIAN PROCESS RESPONSE SURFACE MODELING AND GLOBAL SENSITIVITY ANALYSIS USING NESSUS

John M. McFarland<sup>1</sup>, John A. Dimeo<sup>2</sup>, and Barron J. Bichon<sup>1</sup>

<sup>1</sup> Southwest Research Institute  
6220 Culebra Road  
San Antonio, TX 78238, US  
john.mcfarland@swri.org  
barron.bichon@swri.org

<sup>2</sup> Elder Research, Inc.  
300 W Main St., Suite 301  
Charlottesville, VA 22903, US  
dimeo@elderresearch.com

**Keywords:** Software, Uncertainty Quantification, Response Surface Modeling, Gaussian Process, Reliability Analysis, Global Sensitivity Analysis, Variance Decomposition.

**Abstract.** *NESSUS<sup>®</sup> is a general-purpose software program for probabilistic analysis that includes state-of-the-art algorithms, flexible methods for interfacing with external numerical models, and a mature graphical user interface. NESSUS was originally developed for NASA under a long-term research and development program to develop methods and tools for reliability analysis of space shuttle main engine components. In the past few years, recent NESSUS development has focused on the incorporation of advanced response surface modeling and global sensitivity analysis methods. NESSUS now includes a variety of tools for building and analyzing Gaussian Process (GP) models. This includes general-purpose GP response surface models as well as the Efficient Global Reliability Analysis (EGRA) method, which uses adaptive sampling to target surrogate model accuracy in the vicinity of the limit state. In addition, several methods have been implemented into NESSUS for the calculation of variance-based sensitivity indices, including sampling-based methods and analytical solutions based on response surface models. This paper gives an overview of these recent enhancements. In particular, we introduce the NESSUS Response Surface Toolkit (RST), which is a recently released standalone software application included with NESSUS for building, visualizing, and assessing response surface models.*

## 1 INTRODUCTION

NESSUS (Numerical Evaluation of Stochastic Structures Under Stress) is a general-purpose software program for probabilistic analysis. It was originally created by a team led by Southwest Research Institute (SwRI) as part of a 10-year NASA project started in 1984 to develop a probabilistic design tool for the space shuttle main engine with a focus on probabilistic finite element analysis. The methods and capabilities in NESSUS were designed to support predicting the probabilistic response and/or probability of failure for computational intensive models. The input variations were modeled using probability density functions and propagated using traditional and newly developed probabilistic algorithms. In 1999, SwRI was contracted by Los Alamos National Laboratory to adapt NESSUS for application to extremely large and complex weapon reliability and uncertainty problems in support of its Stockpile Stewardship program. In 2002, SwRI was contracted by the NASA Glenn Research Center to further enhance NESSUS for application to large-scale aero-propulsion system problems. The end result of these large research programs was a completely redesigned software tool that includes a sophisticated graphical user interface (GUI), capabilities for performing design of experiments and sensitivity analysis, a probabilistic input database, a geometric uncertainty modeling tool for perturbing geometry in existing finite element models, and state-of-the-art interfaces to many third-party codes such as Abaqus, ANSYS, LS-DYNA, MSC.NASTRAN, and NASGRO.

NESSUS has seen continuous improvement and application since its beginnings in the late 1980s. The most recent enhancements include extensive capabilities for Gaussian process response surface modeling and variance-based global sensitivity analysis. In particular, the NESSUS Response Surface Toolkit was released in 2015 as a standalone application specifically created to address response surface model fitting, visualization, goodness of fit, and sensitivity analysis.

This paper describes the recent NESSUS enhancements for response surface modeling and sensitivity analysis. Section 2 gives a description of the Gaussian process model formulation used by NESSUS and the NESSUS Response Surface Toolkit. Section 3 gives an overview of the Response Surface Toolkit itself. Section 4 describes three different approaches for reliability analysis with Gaussian process models in NESSUS, including the Efficient Global Reliability Analysis (EGRA) method, which uses adaptive sampling. Section 5 gives an overview of global sensitivity analysis in NESSUS.

## 2 NESSUS GP MODEL FORMULATION

Gaussian Process (GP) regression (also known as kriging) is a powerful technique for developing fast-running response surface or “surrogate” models for computationally expensive computer simulations. The response surface model enables detailed analysis such as Monte Carlo sampling, numerical optimization, model calibration, and sensitivity analysis, that would be intractable if working directly with original simulation. This section gives a summary of the GP model formulation used by the NESSUS software.

NESSUS uses a Gaussian process response surface model formulation employing a low-order trend function and a squared-exponential correlation function. NESSUS supports both constant and linear trend functions. A separable correlation function is employed with unique correlation length parameters for each input variable:

$$c(\mathbf{a}, \mathbf{b}) = \exp \left[ - \sum_{i=1}^d \left( \frac{a_i - b_i}{\theta_i} \right)^2 \right], \quad (1)$$

where  $\mathbf{a}$  and  $\mathbf{b}$  are vectors of dimensionality  $d$ , and  $\theta_i$  denotes a correlation length parameter associated with dimension  $i$ . The covariance matrix associated with the training data is given by:

$$\mathbf{\Sigma} = \sigma^2 \mathbf{R} + \mathbf{\Sigma}_{noise}, \quad (2)$$

where  $\sigma^2$  is the process variance,  $\mathbf{R}$  is the correlation matrix based on Eq. (1), and  $\mathbf{\Sigma}_{noise}$  is a diagonal noise covariance matrix with all diagonal entries given by  $\sigma_{noise}^2$ . When modeling training data from a deterministic computer simulation, the noise variance is commonly assumed to be zero, resulting in a response surface that directly interpolates the training data.

In NESSUS, the GP parameters (the trend coefficients, process variance, correlation lengths, and optionally noise variance) are estimated using maximum likelihood. The profile log likelihood function is maximized by searching over the correlation parameter space, and the trend coefficients and process variance are set to their optimal values conditional on the correlation parameters. The noise variance can also be estimated as part of the procedure, in which case NESSUS maximizes the restricted likelihood function. Further details can be found in ref. [1]. NESSUS supports both gradient-based and global search methods.

NESSUS automatically performs goodness of fit assessment using leave-one-out cross-validation. This is determined by holding all of the GP model parameters (trend and covariance parameters) constant, deleting one observation from the GP model, predicting its value, and repeating for each observation in the training set. Instead of explicitly creating  $n$  GP models, the cross-validation errors can be computed efficiently as described by ref. [2]. NESSUS summarizes the model accuracy by a “cross-validation  $R^2$ ” (also known as “prediction  $R^2$ ” [2]):

$$R_{CV}^2 = 1 - SSE_{CV} / SS_T, \quad (3)$$

where  $SSE_{CV}$  is the sum of squares of the leave-one-out cross-validation residuals and  $SS_T$  is the usual total sum of squares of the training data response values.

### 3 NESSUS RESPONSE SURFACE TOOLKIT

The NESSUS Response Surface Toolkit (RST), released in 2015, is a standalone application included with NESSUS that provides a suite of tools for designing, fitting, assessing, visualizing, and making predictions with response surface models. Some of the key capabilities include:

- Design of experiments using Latin Hypercube sampling, with options for correlation reduction and point spacing optimization
- Fit polynomial regression and Gaussian process regression response surface models
- Evaluate goodness of fit using leave-one-out cross-validation
- Plot 2D and 3D slices of the model
- Evaluate importance of variables using variance-based (global) sensitivity analysis
- Make model predictions at user-specified input values
- Export models for use in the NESSUS software
- Import models created by the NESSUS software

The RST provides streamlined capability for quickly generating and visualizing a design of computer experiments based on Latin Hypercube sampling (Figure 1). The interface allows the user to declare, name, and assign bounds to the variables. Then the user simply specifies the number of samples and clicks on the “Generate Design” button. By default, 100 random designs are generated and the best is selected as the one that maximizes the minimum distance between points. The number of designs can be configured by the user from the “Iterations” field. An option is also provided for correlation reduction using the Iman-Conover method [3]. Random number seed control is also provided for reproducibility. Once the samples have been generated, pairwise or three-way plots among the variables can quickly be generated using a menu driven system, as shown in Figure 2.

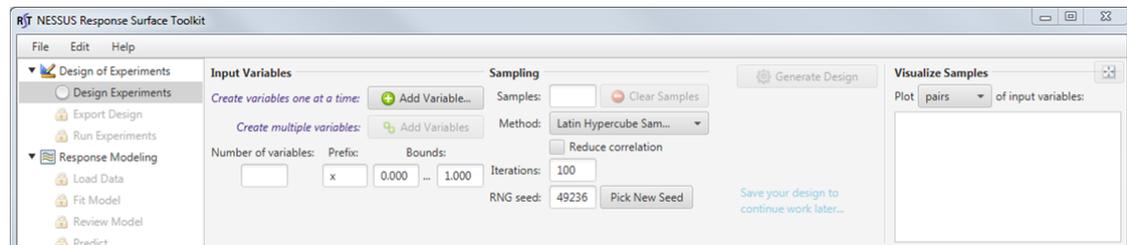


Figure 1. NESSUS RST design of experiments settings

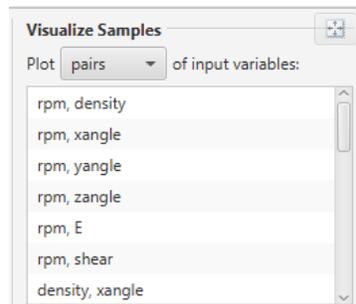


Figure 2. NESSUS RST menu for visualization of design of experiments sample points

Generated designs can be saved with metadata (such as the variable bounds, random number seed, and other settings) and also exported to a delimited text file. The RST also includes the capability to “augment” an existing design [4]. For an initial design with  $n$  points, augmenting adds  $n$  new points to the design such that the overall  $2n$ -point design maintains a Latin Hypercube structure.

The RST provides a fast and flexible interface for loading existing training data to create a response surface, as shown in Figure 3. Training data can be read from plain text files using whitespace, comma, and other delimiters. Headers, when available, are used to initially define variable names, and can be overridden by the user. Each column detected in the file can be assigned as an input variable, a response variable, or ignored. The default settings treat the last column as the response variable and all others as input variables.

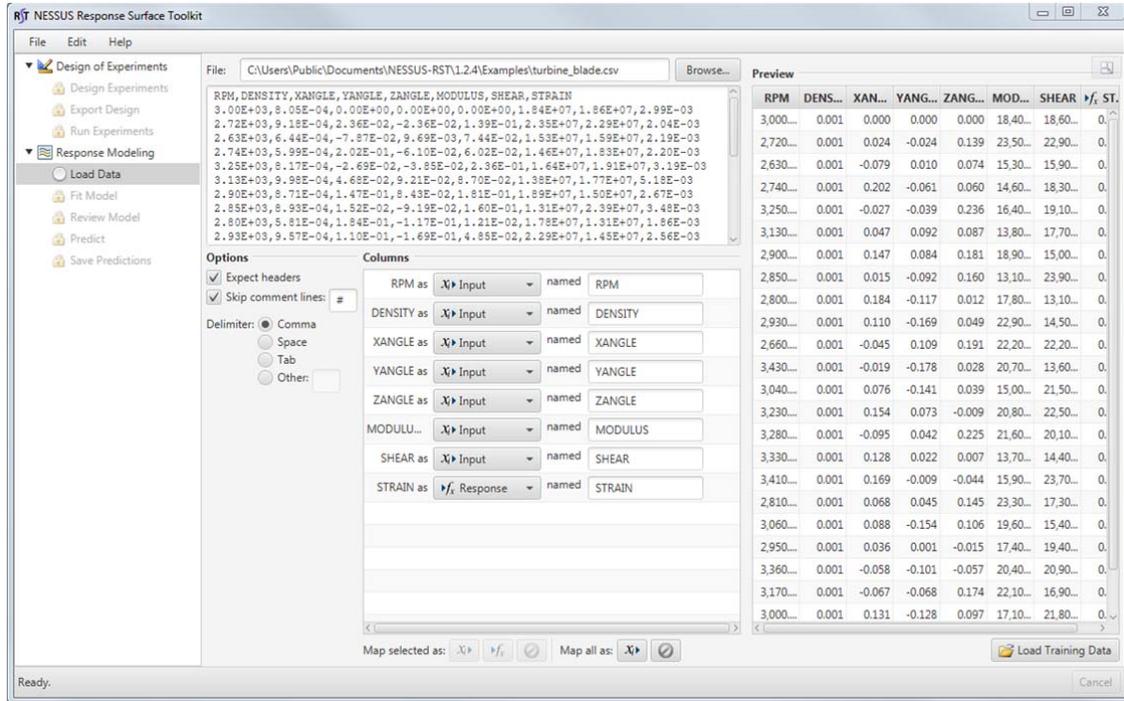


Figure 3. Loading training data using NESSUS RST

Once the training data have been loaded, the RST displays a summary of correlations between the inputs and the response variable, and provides the ability to show a scatter plot between any two or three variables (Figure 4).

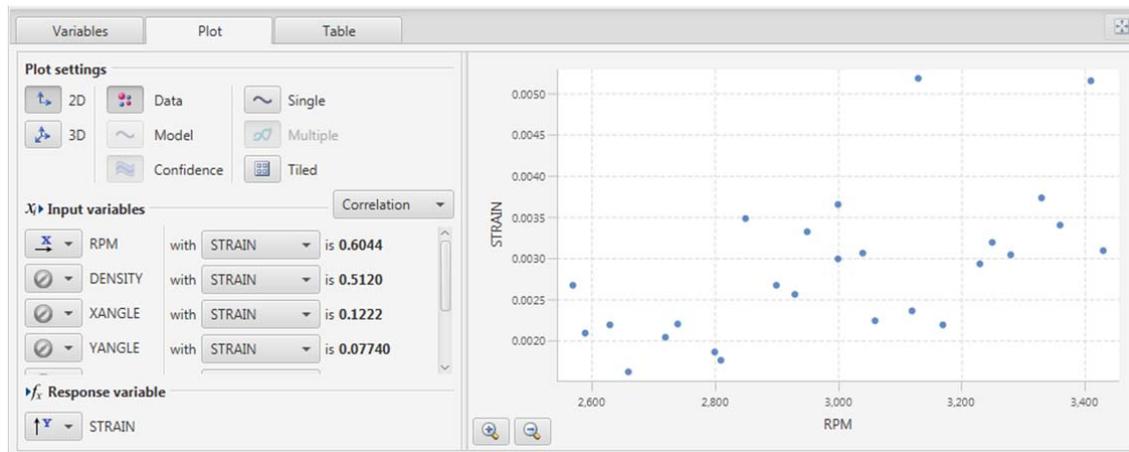


Figure 4. NESSUS RST training data visualization prior to response surface model fitting

Next, the user is provided a simple set of options for fitting the response surface model, as shown in Figure 5. Two model types are available: Gaussian Process and polynomial regression. The Gaussian Process model provides options for a constant or linear trend, three settings for observation noise, and several options for the optimization method used during parameter estimation. The options for observation noise include noise-free ( $\sigma_{noise}^2 = 0$ ), estimation of  $\sigma_{noise}^2$  by restricted likelihood maximization, and user-specification of the noise

standard deviation,  $\sigma_{noise}$ . The option to fit the noise variance is especially useful when the training data are the results of physical experiments, as opposed to computer experiments.

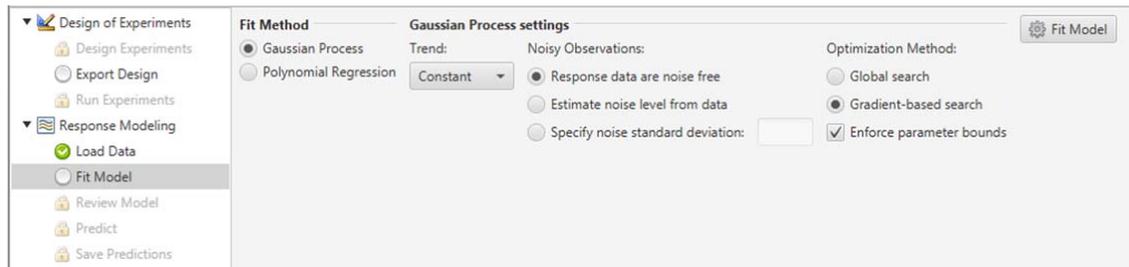


Figure 5. NESSUS RST response surface fitting options

Upon fitting the response surface model, the RST displays a summary of information that includes cross-validation results, input variable sensitivities, and surface visualization (Figure 6). This includes fit and cross-validation  $R^2$  values, as well as the noise standard deviation, reciprocal condition number, and log likelihood for GP models. The reciprocal condition number can be used to identify ill-conditioning of the GP correlation matrix, which can adversely impact performance when the value begins to approach roughly  $10^{-13}$  or smaller. This can be remedied by removing training points or re-fitting the model with a small amount of noise variance.

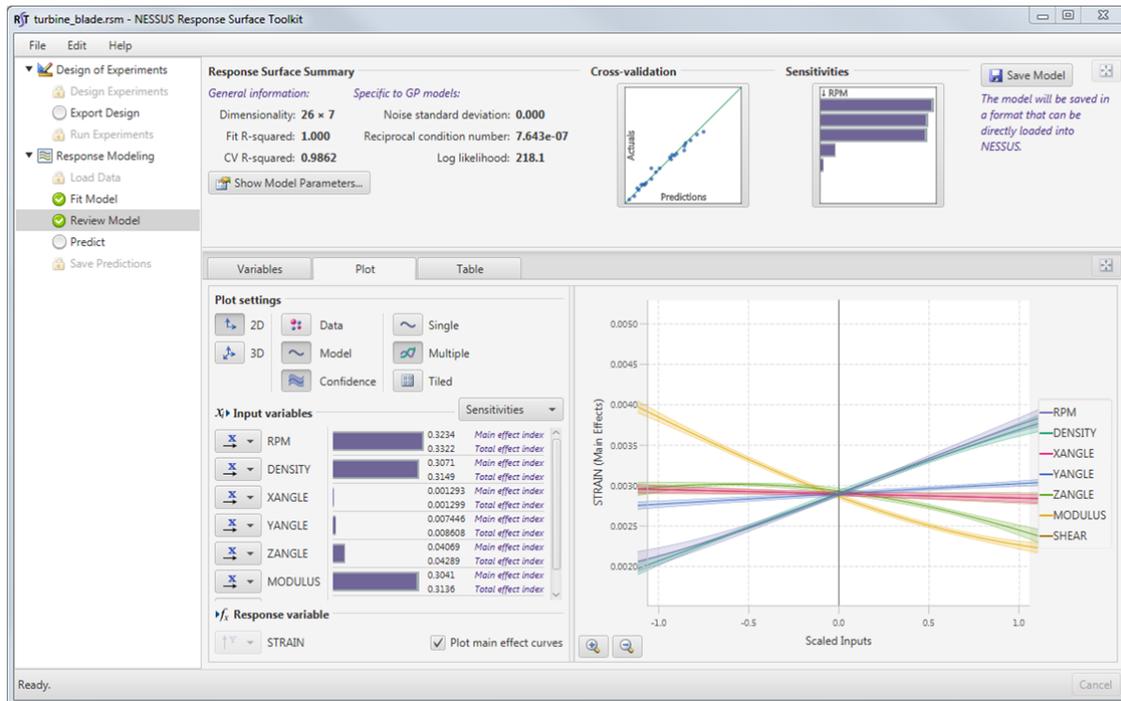


Figure 6. NESSUS RST screen for reviewing and visualizing the response surface model

The miniature “Sensitivities” plot in the Review Model screen can be expanded to a full-size plot, as shown in Figure 7. On this plot, the x-axis shows the main and total effect indices obtained using variance-based global sensitivity analysis. In order to do this, each response surface input variable is modeled using a normal distribution with mean and standard

deviation estimated from the training data (the range of the training data is treated as five standard deviations). The sensitivity indices are then computed efficiently using the analytical solution given in [5]. For complete control of how the probability distributions for the input variables are defined, the user can load the response surface model into NESSUS and use the global sensitivity analysis type (see Section 5).

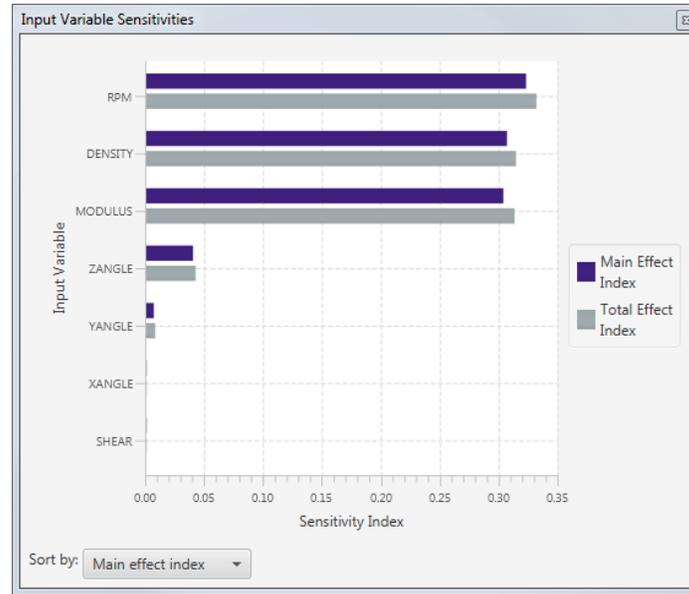


Figure 7. Variance-based sensitivity indices in NESSUS RST

The controls in the bottom of the Review Model screen are geared towards response surface visualization when the Plot tab is selected. By default, a 2D line plot of normalized main effect curves for all inputs, with confidence bounds, is shown, as seen in Figure 6. In this plot, the x-axis represents a scaled version of each input variable, in which the training data values are used to scale each input to the range  $[-1,1]$ . In other words, a value of  $-1$  on this scale indicates that the variable is equal to the smallest value of that variable occurring in the training set. Similarly, the largest value in the training set corresponds to  $+1$ .

For each curve, the y-axis plots the main effect of the response, which is the expectation of the response conditional on the value of the given input variable. The same probability distributions used for variance decomposition are employed for determination of the main effect curves. By default, the plot also shows a central 95% confidence interval for each main effect curve as a shaded region. The confidence bounds are derived from the GP model fit. Both the main effect curves and their confidence bounds are computed efficiently using a combination of derivations from refs. [5,6].

To the left of the plot, a variety of settings are available to control the behavior of the plot. The “Input variables” section provides control of which variables are shown in the plot. When the plot type is “Multiple”, individual variables can be removed from the plot using these controls. For convenience, the global sensitivity indices are shown next to each variable, making it easy to quickly find and plot the most important variables.

The type of plot can be configured under the “Plot settings” area. The “Single” plot type limits the plot to a single curve. In this mode, the input variable selections control which curve is plotted, as opposed to adding or removing curves, like they do in “Multiple” plot mode.

The “Tiled” plot type presents an array of plots, each of which plots the main effect curve for a single input variable. An example of the tiled plot is shown in Figure 8, where the plotting area has been maximized so that all nine plots can be seen at once. Unlike the “Multiple” plot type, which uses scaling in order to display all variables on a single plot, the tiled plot uses the original units for each input variable, which can make the interpretation more intuitive. Also note that the tiled plot uses the same y-axis range for each individual plot, making it possible to quickly assess the relative influence of each variable.

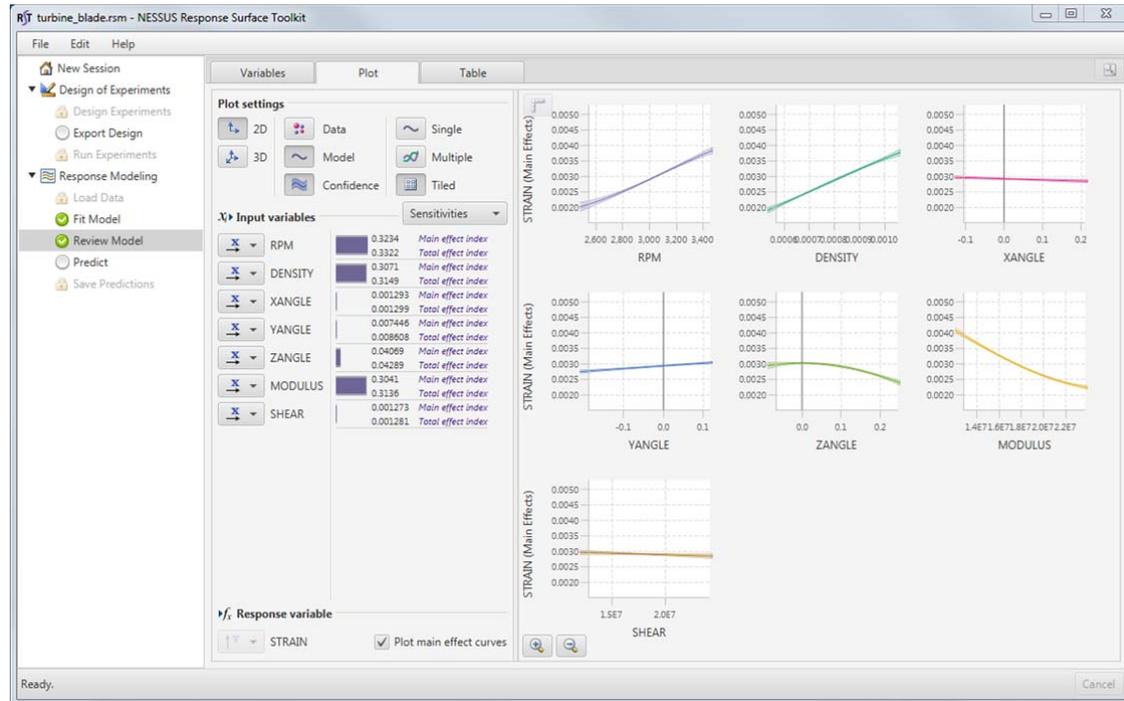


Figure 8. NESSUS RST “Tiled” plot mode, showing the main effect curve for each response surface variable.

Under the plot settings area, display of the training data can also be enabled for either the single or tiled plot types. When this is done, the training data are projected onto the two-dimensional plot axes. Some caution is needed when interpreting this kind of display of the data, since not all dimensions of the training data input space are represented in the plot.

Three-dimensional visualization of the response surface can also be enabled using the plot settings. In 3D plotting mode, the user selects two input variables for the horizontal axes, and the response is plotted on the vertical axis. In 3D mode, the surface is conditioned on fixed values of the input variables not being plotted, unlike 2D mode, which averages over the remaining variables to obtain main effect curves.

The RST allows for reading and writing both polynomial and GP response surface models to a platform-independent file. When writing a GP model to a file, the following data are included:

- The training data input and output values
- The Cholesky factor of the GP correlation matrix,  $\mathbf{R}$
- Solution of the linear system used for prediction,  $\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})$ , where  $\mathbf{Y}$  are the training data response values and  $\mathbf{F}\boldsymbol{\beta}$  is the trend function evaluated at the training data locations

- The correlation parameters determined via maximum likelihood estimation
- The trend coefficients, process variance, and noise variance

The response surface model file can then be read back into either NESSUS or NESSUS RST. In addition to capturing the options used to fit the model such as the trend order and noise variance, this has the advantage that the GP mean and covariance parameters do not need to be estimated again, nor does the correlation matrix need to be reconstructed or factored. Thus, for large data sets, reloading the GP from file provides significant time and computation savings.

## 4 NESSUS RELIABILITY ANALYSIS

Unlike the Response Surface Toolkit, the NESSUS software is designed for general purpose probabilistic analysis. In particular, NESSUS has several probabilistic methods for reliability analysis, including Monte Carlo sampling, Latin Hypercube sampling, importance sampling, Advanced Mean Value, First Order Reliability Method (FORM), and more. Any of these reliability analysis methods can be used in conjunction with response surface modeling.

There are three approaches for using response surface models for reliability analysis in NESSUS: declaring an equation in the problem statement as a regression model, using a response surface-based reliability method, and using the Efficient Global Reliability Analysis (EGRA) method. Further information about each of these approaches is given in the following sections.

### 4.1 Regression model type

The first approach is to declare one or more equations in the problem statement as a numerical model, and then define the numerical model as a regression model type. NESSUS provides options for both polynomial and Gaussian Process regression models. The GP model can be defined based on user-specified training data or based on a previously saved GP model file. Thus, this is the way to use in NESSUS a GP model created by RST.

If the GP regression model is specified based on training data, then upon running the analysis, NESSUS will first fit the GP model using maximum likelihood estimation, as described in Section 2. NESSUS will also save the resulting GP model to a file, so that it can be analyzed using the RST or re-used from NESSUS without having to re-fit the model.

Once the regression model is defined, the NESSUS analysis behaves the same as with any other equations or numerical models. Any probabilistic methods can be used, including global sensitivity analysis as described in Section 5. Since the response surfaces are computationally very efficient, it is common to use them in conjunction with exhaustive sampling methods such as Monte Carlo or Latin Hypercube sampling.

This approach also allows one to build up the problem statement hierarchically. For example, a finite element model for predicting stress might be replaced by a response surface, whose output is fed into an analytical equation for crack growth prediction. A sequence of multiple independent response surface models could also be used.

### 4.2 Response surface method

In the second approach, the three steps of design of computer experiments, response surface fitting, and response surface sampling are considered collectively as a reliability analysis method. With this approach, the user defines the problem statement so that NESSUS interfaces directly with the original (computationally expensive) model or models. For example,

the problem statement may include calls to a finite element model or other external numerical code.

NESSUS provides three probabilistic analysis methods that fit into this category: the “Response Surface Method (RSM)”, “Gaussian Process Response Surface Method (RSM\_GP)”, and “Efficient Global Reliability Analysis (EGRA)”. EGRA uses adaptive sampling and will be discussed in the next section.

The NESSUS Response Surface Method uses low-order polynomial fitting and is based on traditional Design of Experiments techniques such as the central composite design. The Gaussian Process Response Surface Method was added in NESSUS version 9 and is preferred when interfacing NESSUS with external models such as finite element simulations. The user has three options when working with this method. The first is the number of response surface training points that NESSUS will use. This determines how many evaluations of the problem statement NESSUS will perform.

The second option is specification of the bounds used for the design of experiments. The bounds are specified in terms of an equivalent standard normal variable (“u-space” in NESSUS terminology). The default bounds for each variable are -3 to +3 in u-space, which will correspond to +/-3 standard deviations for variables that are assigned normal distributions. NESSUS uses the bounds to generate a Latin Hypercube design. Unlike the Latin Hypercube Sampling reliability analysis method, this Latin Hypercube design is based on uniform distributions, regardless of the user-specified distribution types.

Finally, the user can also specify the number of times to sample the response surface when performing the probabilistic analysis. This sampling is performed using basic Monte Carlo simulation. The default is to use 100,000 samples.

#### **4.3 Efficient Global Reliability Analysis (EGRA) method**

The Efficient Global Reliability Analysis (EGRA) [7] method is similar to the Gaussian Process Response Surface Method in NESSUS, in that it incorporates all steps of the analysis including design of computer experiments, response surface fitting, and sampling. The difference is that EGRA uses an adaptive sampling approach to iteratively add new training points to the GP model, specifically targeting accuracy of the GP model in the vicinity of the limit state. By ensuring accuracy of the GP model in this critical region, EGRA can efficiently provide an accurate estimate of the reliability. EGRA is an extension of the Efficient Global Optimization (EGO) [8] method for numerical optimization.

EGRA uses the so-called expected feasibility function (EFF) to select the location at which a new training point should be added to the Gaussian process model by maximizing the expectation that the point lies on the limit state contour. A point could be expected to lie on the limit state contour if its predicted value is near the limit state value, or if the uncertainty in its prediction is such that there is a significant probability of its true value being near the limit state value. Because the uncertainty is higher in regions of the space with fewer observations, this provides a balance between exploiting areas that are predicted to be near the limit state, and exploring areas where more information is needed. The general procedure of EGRA is:

1. Build an initial Gaussian process model of the response function.
2. Iteratively add training data until a stopping criterion is met:
  - a. Find the point that maximizes the EFF.
  - b. Evaluate the true response function at the new point and add it to the training set.
3. Use Monte Carlo sampling or another appropriate probabilistic method to compute the probability of failure by sampling the converged Gaussian process response surface.

To define the expected feasibility function, let the limit state be denoted by the equality  $g(\mathbf{x}) = z_0$ . The prediction from the GP model at any point in the design space follows a Gaussian distribution, and can be written:

$$\hat{g}(\mathbf{x}) \sim N[\mu_g(\mathbf{x}), \sigma_g^2(\mathbf{x})] \quad (4)$$

where the mean  $\mu_g(\mathbf{x})$  and variance  $\sigma_g^2(\mathbf{x})$  are the GP mean and variance conditioned on the training data (see, e.g. ref. [1]). Inspired by the contour estimation work in ref. [9], the expected feasibility function is defined as:

$$EF(\mathbf{x}) \equiv E[\varepsilon(\mathbf{x}) - \min(|\hat{g}(\mathbf{x}) - z_0|, \varepsilon(\mathbf{x}))] \quad (5)$$

where the  $\varepsilon$  term is used to focus the search in the immediate vicinity of the response threshold. This expectation can be calculated by integrating over  $z_0 \pm \varepsilon$ :

$$EF(\mathbf{x}) = \int_{z_0 - \varepsilon}^{z_0 + \varepsilon} (\varepsilon - |z_0 - g|) f_{\hat{g}} dg \quad (6)$$

This integral can be expressed analytically in terms of the standard normal probability density function and cumulative density function, as given in ref. [7].

In NESSUS, the initial training set is generated with  $[(n_{\text{var}} + 1)(n_{\text{var}} + 2)]/2$  samples, where  $n_{\text{var}}$  is the number of random variables. These samples are generated following a Latin Hypercube design that is uniformly spaced with bounds based on critical values from an equivalent standard normal distribution. By default, the design space is defined based on equivalent standard normal values of  $\pm 5$ , but this value can be changed by the user. Alternatively, NESSUS allows the user to prescribe the search bounds for each variable in the original space (x-space).

To specify the expected feasibility, NESSUS uses  $\varepsilon = 2\sigma_g(\mathbf{x})$ . The expected feasibility function is often multimodal, so NESSUS uses the DIRECT optimization algorithm [10] to ensure that the global optimum is found. The default stopping criterion for the expected feasibility is 0.00001, but this can also be changed by the user.

Finally, NESSUS uses multimodal adaptive importance sampling [11] by default to sample the final GP model for probability of failure calculation. However, the user can also select basic Monte Carlo sampling or Latin Hypercube sampling. The sample size can also be set by the user.

## 5 NESSUS GLOBAL SENSITIVITY ANALYSIS

Introduced in version 9.6, NESSUS includes variance-based global sensitivity analysis as an alternative analysis type to reliability analysis. When global sensitivity analysis is used, NESSUS decomposes the variance of the top-level response variable from the problem statement with respect to the user-defined input random variables. The main and total effect sensitivity indices [12] are computed for each variable.

Three numerical methods are provided for variance decomposition. The first is referred to in NESSUS as ‘‘Structured Monte Carlo’’ sampling, which implements the efficient sampling scheme outlined by [12]. The default is to use Sobol sequences, but NESSUS also supports Latin Hypercube sampling and basic Monte Carlo sampling. Sample estimates for the main and total effect indices are adapted from the equations proposed in refs. [13,14]. The second method available in NESSUS is the Fourier Amplitude Sensitivity Test (FAST) [15].

The third method available in NESSUS is “GP with analytical sensitivities”, which is similar to the GP Response Surface Method for reliability analysis, in that it performs both the design of computer experiments and response surface model fitting. The training points are generated based on a uniform Latin Hypercube design with bounds set at -3 and +3 in equivalent standard normal space. Once the GP model is constructed, the sensitivity indices are computed exactly, as described in ref. [5]. This method is only supported with random variables that have either normal or uniform distributions.

It is also possible to use sampling-based estimation of the sensitivity indices in conjunction with response surface modeling. This enables one to perform global sensitivity analysis for a GP model for random variable types that are not supported by the analytical calculations. It also allows the user to perform sensitivity analysis on response surface models that have already been built, or to achieve greater control of the sampling scheme used for design of computer experiments. This is done by using the regression model type described in Section 4.1 to define response surface models for one or more of the equations in the problem statement. The response surface model can then be specified based on either existing training data or a previously saved response surface model file.

## 6 CONCLUSIONS

NESSUS is a general-purpose software program for probabilistic analysis and was originally developed in the 1980’s as part of a long-term research and development program with NASA. This paper describes the most recent enhancements to the NESSUS software, which have focused on Gaussian process response surface modeling and global sensitivity analysis. NESSUS includes several options for employing GP models in reliability analysis, including adaptive sampling via the Efficient Global Reliability Analysis (EGRA) method. NESSUS also includes capabilities for variance-based global sensitivity analysis using sampling-based estimates or response surface-based solutions.

The NESSUS Response Surface Toolkit (RST) was introduced in 2015 as standalone companion application included with NESSUS. The RST provides a suite of tools for designing, fitting, assessing, visualizing, and making predictions with response surface models. The RST includes options for fitting both Gaussian process and polynomial regression models, and it can read and write to an open “.rsm” file format that is compatible with NESSUS.

## REFERENCES

- [1] J. McFarland, Uncertainty analysis for computer simulations through validation and calibration. Ph.D. thesis, Vanderbilt University, 2008.
- [2] J. Martin, T. Simpson, Use of kriging models to approximate deterministic computer models. *AIAA Journal*, **43**(4), 853-863, 2005.
- [3] R.L. Iman, W.J. Conover, A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics Part B. Simulation and Computation*, **11**(3), 311-334, 1982.
- [4] C.J. Sallaberry, J.C. Helton, S.C. Hora, Extension of Latin hypercube samples with correlated variables. Technical Report SAND2006-6135, Sandia National Laboratories, 2006.

- [5] W. Chen, R. Jin, A. Sudjianto, Analytical variance-based global sensitivity analysis in simulation-based design under uncertainty. *Journal of Mechanical Design*, **127**(5), 875-886, 2005.
- [6] J. Oakley, A. O'Hagan, Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society B*, **66**(3), 751-769, 2004.
- [7] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan, J.M. McFarland, Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, **46**(10), 2459-2468, 2008.
- [8] D. Jones, M. Schonlau, W. Welch, Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, **13**(4), 455-492, 1998.
- [9] P. Ranjan, D. Bingham, G. Michailidis, Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, **50**(4), 527-541, 2008.
- [10] D. Jones, M. Schonlau, W. Welch, Efficient global optimization of expensive black-box functions. *INFORMS Journal on Computing* **12**, 272-283, 1998.
- [11] T. Zou, Z. Mourelatos, S. Mahadevan, P. Meernik, Reliability analysis of automotive body-door subsystem. *Reliability Engineering and System Safety*, **78**(3), 315-324, 2002.
- [12] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, *Sensitivity analysis in practice: a guide to assessing scientific models*, Wiley, Hoboken, NJ, 2004.
- [13] I.M. Sobol', E.E. Myshetskaya, Monte Carlo estimators for small sensitivity indices. *Monte Carlo Methods and Applications*, **13**(5-6), 455-465, 2007.
- [14] S. Kucherenko, S. Tarantola, P. Annoni, Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, **183**(4), 937-946, 2012.
- [15] A. Saltelli, S. Tarantola, K. Chan, A quantitative, model independent method for global sensitivity analysis of model output. *Technometrics*, **41**(1), 39-56, 1999.