29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

# SENSITIVITY ANALYSIS OF TURBINE ENGINE SUSTAINMENT

**Eliseo E. Iglesias*, Dr. Harry Millwater****
***Graduate Student, University of Texas at San Antonio, ****Chair, Dept. of Mechanical Engineering, University of Texas at San Antonio**

**Keywords**: *Sensitivity Analysis, Monte Carlo Sampling, Complex Variable, Minimum function*

## Abstract

*In current engine reliability models, the relationship between the MTTF (Mean Time to Failure) and the time to failure (TTF) of individual components is not well understood. Probabilistic sensitivity analysis provides a tool to identify the critical failure modes in an engine. These sensitivity measures reveal any effect of an individual component's uncertainty on the MTTF of the entire system. Using a conservative model, the minimum function, several different methods of identifying these critical failure modes were applied to both a 5-component mock engine model and a 22-component full turbine engine module (Rear Gear Box). Complex variable numerical differentiation methods were applied in order to calculate local probabilistic sensitivities. From this investigation two numerical differentiation methods were developed: direct complex variable Monte Carlo (direct CVMC) and complex variable score function (CVSF). Direct CVMC requires fewer samples but is non-robust. CVSF circumvents any discontinuities in the response of interest but inherits a larger variance in the derivatives of interest. While direct CVMC provided first order derivatives, this required an approximation of the minimum function. CVSF, with an additional order of magnitude sampling cost, circumvents discontinuities in the minimum function and provided higher order derivatives.*

## 1 Turbine Engine Sustainment

In military gas turbine engines there are a countless number of failure modes. Typical maintenance, repair, and operations (MRO) record the failure rates of components throughout the life of the engine. However, maintenance schedules often require inefficient and costly overhauls of aircraft. The aim of probabilistic sensitivity analysis is to use the data collected to build a probabilistic model for each failure mode in the engine or engine module and incorporate that information into an entire engine or engine module model. Sensitivity analysis of such an engine model would provide a decision making tool for MRO by identifying critical failure modes that contribute most to the detriment or increase in the MTTF of the engine and/or engine module.

A typical aircraft contains three major systems: Engine, Structural Support, and Propeller. Within the Engine system lays the 'turbine', 'compressor,' and 'rear gear box.' This paper focuses on the probabilistic sensitivity analysis of the rear-gearbox (RGB). StandardAero, an MRO company, provided this data.

### 1.1 Review of Sensitivity Methods

Sensitivity analysis has been a topic of interest in reliability for the past 50 years. Starting with Birnbaum's work on identifying important variables in multicomponent system [5], he defined sensitivity as a derivative taken of the joint probability density function (JPDF) (representing all the individual probability density functions or PDFs) with respect to a specific PDF of interest. This sensitivity measures the effect of a "perfect component" on the reliability of the system. Others like Barlow and Proschan [4] have developed similar methods building on Birnbaum's sensitivity work by considering fault tree analysis. Natvig

[15] delves deeper into fault tree sensitivity analysis in developing measures that identify critical 'branches' or cuts in a defined fault tree. These measures, however, become analytically difficult when working with large systems.

Satelli, et al, formulated a variance method, Global sensitivity analysis that uses Monte Carlo sampling to calculate probabilistic sensitivities [21]. The 'sensitivity index' as defined by Satelli, et al, is a measure of the contribution a variable's variance to the variance of the system.

This paper presents a method that calculates a local probabilistic derivative using Monte Carlo sampling, complex variable automatic differentiation, and the Score Function [13]. Such sensitivity could reveal a variable's effect on the MTTF of the system with few assumptions.

## 1.2 Objective

The main objective presented is the formulation of probabilistic sensitivities in the form of derivatives applied to a mock engine model (5 variables) and a full 22-component RGB model. Both models are composed of variables that represent an individual component's time to failure (TTF). The model considers the system TTF as the *minimum* TTF between all components (a series system).

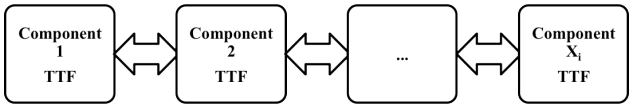


Fig 1.1. Series System

The response of interest is the *minimum function*, which as will be shown presents challenges when trying to take derivatives. If $g$ is the response function of interest then the probabilistic derivatives of interest are $\partial^n \mu_g / \partial \mu_{X_i}^n$, and $\partial^n \mu_g / \partial \sigma_{X_i}^n$. These derivatives may not only hold valuable sensitivity information but also could be used to construct a Taylor series expansion to estimate the mean of other wise analytical difficult responses.

## 2.2 Models

The Mock Engine model (shown in Table 2.1) comprises of 5 failure modes taken from the 22-component RGB Turbine Engine Module given by StandardAero. The Score Function method was applied to this model. Ten derivatives (with respect to each variables natural parameters) were calculated and compared with forward finite differencing.

| X | Distribution | Given Parameters | |
|---|---|---|---|
| **1** | Lognormal | λ (lambda) | Z (zeta) |
| **2** | Normal | μ | σ |
| **3** | Weibull | α3 (shape) | β3 (scale) |
| **4** | Weibull | α4 | β4 |
| **5** | Weibull | α5 | β5 |

Table 2.1: 5-Component Mock Engine Model

# 2 Score Function Method

## 2.1 Formulation

In probabilistic analysis the mean of any response or function is calculated using the following multidimensional integral [3]:

$$\mu_g = \int \cdots \int_0^\infty g(\underline{X}) f_{\underline{X}} d\underline{X} \qquad (2.1)$$

where $g$ is the response of interest, $\underline{X}$, is a vector of all the input variables for $g$, and , $f_{\underline{X}}$, is the corresponding set of probabilistic density distributions (PDFs) describing each input variable. Since the set of variables in the problem objective are independent and interact in a series manner, $f_{\underline{X}}$, becomes the product:

$$f_{\underline{X}} = \prod_{i=1}^{k} f_{X_i} \qquad (2.2)$$

In essence, the product shown in Eq. 4.2 is the joint probability distribution function (JPDF) of the system. Eq. 2.1 displays the bounds on the integration as zero and infinity. The integration must cover the entire space of the input variables [3]. However, since all the input variables describe time to failures the space spans only zero to infinity. The derivatives of interest for this problem are, $\partial^n \mu_g / \partial \mu_{X_i}^n$, and $\partial^n \mu_g / \partial \sigma_{X_i}^n$ ($n$ is the order of the derivative). In

order to obtain these derivatives the Leibniz integral rule must be applied to Eq. 2.1 where, $\theta_i$, is an arbitrary distribution specific parameter. The last two terms are considered the "flux terms" that are evaluated at the boundaries of the distribution [14]. Since the boundaries are zero and infinity the flux terms go to zero. As a result Eq. 2.1 becomes:

$$\frac{d^n \mu_z}{d\theta_i^n} = \int \cdots \int_0^\infty \frac{\partial^n}{\partial \theta_i^n} \left( g(\underline{X}) f_{\underline{X}} \right) d\underline{X} \qquad (2.3)$$

where $\theta_i$ represents the mean or standard deviation of the variable of interest. There are a multitude of methods to evaluate the derivative shown above. The Score Function method uses some algebraic manipulation to separate the integrand into a $\theta_i$ dependent portion and a non $\theta_i$ - dependent portion. The former can be reformulated into a kernel function.

## 2.2 Kernel Functions

The Score Function method assumes that the only portion of the integrand in Eq. 2.4 dependent on the distribution parameter, $\theta_i$, is the JPDF, $f_{\underline{X}}$. This allows the derivative to operate only on the JPDF:

$$\frac{d^n \mu_z}{d\theta_i^n} = \int \cdots \int_0^\infty g(\underline{X}) \frac{\partial^n f_{\underline{X}}}{\partial \theta_i^n} d\underline{X} \qquad (2.4)$$

A normalization factor, $f_{X_i}/f_{X_i}$, is then introduced to produce the following:

$$\frac{d^n \mu_z}{d\theta_i^n} = \int \cdots \int_0^\infty g(\underline{X}) \left( \frac{\partial^n f_{X_i}}{\partial \theta_i^n} \frac{1}{f_{X_i}} \right) f_{\underline{X}} d\underline{X} \qquad (2.5)$$

The middle term, $\left( \frac{\partial^n f_{X_i}}{\partial \theta_i^n} \frac{1}{f_{X_i}} \right)$, is defined as the *kernel function,* $\kappa_{\theta_i}^{(n)}$. Analytically, this reformulation poses a complication to an already complex integral. However, Monte Carlo sampling numerical integration simplifies the matter. By normalizing the integrand one set of realizations sampled from each variable's PDF is enough to evaluate any derivative of any order, *n* [13]. By using sampling the integral in Eq. 2.5 becomes:

$$\frac{d^n \mu_z}{d\theta_i^n} \approx E\left[ g(\underline{X}) \bullet \kappa_{\theta_i}^{(n)}(x_i^k, \theta_i) \right] \qquad (2.6)$$

The above numerical technique has a significant advantage over finite differencing of any order in that the response only needs to be evaluated once over all the realizations created. That same set of realizations can be recycled to determine any other derivatives of interest with the appropriate kernel function. While finite differencing is dependent of perturbation size and order, the score function method is only dependent on the variance of the integrand that can be mitigated with a higher amount of samples [14].

## 2.3 Jacobian Transformation Matrix

For normal distributions the first order kernel functions for derivatives with respect to mean and standard deviation are the following:

$$\kappa_\mu = \frac{x-\mu}{\sigma^2}, \kappa_\sigma = \frac{1}{\sigma}\left( \left( \frac{x-\mu}{\sigma^2} \right)^2 - 1 \right) \qquad (2.7\text{A-B})$$

This coupled with Monte Carlo sampling produces an approximation of the derivatives of interest.

$$\partial \mu_g / \partial \mu_i \approx E\left[ g(\underline{X}) \bullet \kappa_{\mu_i}(x_i^k, \mu_i) \right] \qquad (2.8)$$

$$\partial \mu_g / d\sigma_i \approx E\left[ g(\underline{X}) \bullet \kappa_{\sigma_i}(x_i^k, \sigma_i) \right] \qquad (2.9)$$

For random variables that follow a non-normal distribution of 2 parameters the derivatives of interest is not a direct calculation. As Millwater [13] shows, in order to obtain first order derivatives of interest, $\partial \mu_g / \partial \mu_{X_i}$, and $\partial \mu_g / \partial \sigma_{X_i}$, first the derivatives with respect to the distribution's *natural parameters* (*a* and *b)* must be determined: $\partial \mu_g / \partial a_i$ and $\partial \mu_g / \partial b_i$. Then from these derivatives a Jacobian transformation matrix is formulated using the distribution's relationship between *standard* ($\mu$ and $\sigma$) and *natural* (*a* and *b*) parameters. Using these relationships and their partial derivatives a Jacobian transformation matrix becomes:

$$J = \begin{bmatrix} \mu'_a & \mu'_b \\ \sigma'_a & \sigma'_b \end{bmatrix} \quad (2.10)$$

Taking the inverse and transpose of the Jacobian the transformation formulation follows as:

$$\begin{bmatrix} \partial\mu_g/\partial\mu_i \\ \partial\mu_g/\partial\sigma_i \end{bmatrix} = \left[\left[J\right]^{-1}\right]^T \begin{bmatrix} \partial\mu_g/\partial a_i \\ \partial\mu_g/\partial b_i \end{bmatrix} \quad (2.11)$$

Any 2-parameter distribution with functional relationships between *standard* and *natural* parameters may use Eq. 2.11 to transform the natural derivatives to standard ones [13].

## 2.4 Issues

The Score Function method provides an insight into which distribution parameters affect the mean time to failure of the model. However, in order to obtain the derivatives of interest, $\partial^n\mu_g/\partial\mu_{X_i}^n$ and $\partial^n\mu_g/\partial\sigma_{X_i}^n$, from non-normal distributions a Jacobian transformation matrix is needed. Also, computing higher order derivatives necessitates second order kernel functions alongside a second order Jacobian transformation matrix.

# 3 Complex Variable Differentiation Methods

The complex variable differentiation method is an infinitesimal perturbation gradient estimator introduced by Lantoine [11]. This analysis uses the complex realm of numbers to numerically compute derivatives accurately. In the Score Function approach every sensitivity of interest necessitates a unique kernel function derived from each variable's corresponding PDF. In larger systems of composed of multiple non-identically distributed random variables the score function method becomes a tedious and inefficient process. By moving to the multi-complex realm the mathematical framework allows for the numerically efficient determination of derivatives and probabilistic sensitivities.

### *3.0.1 Complex Numbers*

A complex number can be represented as the following:

$$\mathbb{C} := \left\{ x + yi \middle| x, y \in \mathbb{R} \right\} \quad (3.1)$$

Lantoine shows that a *multi-complex* number can be expanded in the following manner:

$$\mathbb{C}^n := \left\{ \begin{matrix} x_0 + x_1 i_1 + \dots \\ + x_n i_n + x_{12} i_1 i_2 + \dots \\ + x_{n-1} i_{n-1} i_n + \\ x_{1\dots n} i_1 \dots i_n \end{matrix} \middle| x_0, \dots, x_n \in \mathbb{R} \right\} \quad (3.2)$$

Any real number is representable by a multi-complex number. The only difference is the added dimensions. These extra dimensions allow for rapid derivative computations of holomorphic functions. Lantoine uses the following single random variable function to conduct complex Taylor series expansion:

$$f\left(x + hi_1 + \dots + hi_n\right) = f(x) + (i_1 + \dots + i_n) hf'(x) + \dots + (i_1 + \dots + i_n)^2 h^2 \frac{f''(x)}{2} + HOT \quad (3.3)$$

In a single function evaluation the derivative of any order is carried through, automatically by multi-complex portion of the result.

$$f^{(n)}(x) = \frac{Im_{1\dots n}\left(f\left(x + hi_1 + \dots + hi_n\right)\right)}{h^n} + O(h^2) \quad (3.4)$$

Unlike finite differencing, this method requires *one* function evaluation in order to compute the derivative of interest. The error, according to Eq. 3.4, depends on step size, *h*. By choosing a small step size (1E-10 for all subsequent analysis presented) the derivative is approximated:

$$f^{(n)}(x) \approx \frac{Im_{1\dots n}\left(f\left(x + hi_1 + \dots + hi_n\right)\right)}{h^n} \quad (3.5)$$

According to Lantoine, this method is restricted to functions that fit the following criteria:

1. The function must be holomorphic within the complex domain.
2. The function must satisfy the multi-complex Cauchy-Riemann equations
3. The function can be represented, near every point in the complex domain by a Taylor series

The complex variable differentiation method has the potential to break down when non-continuous functions are applied. As will be seen later in this chapter, the minimum function has such discontinuities in its derivatives.

### 3.1 Matrix Representation of Multi-Complex Variables

A 2-dimensional matrix provides a compact way of representing a multi-complex number. As seen in the previous section, Lantoine represents a complex number using a long chain of multi-complex terms. In order to conduct efficient computations, Lantoine uses a matrix representation. Below is an example of the bi-complex case.

$$\mathbb{C}^2 := \left\{ \begin{matrix} x_0 + x_1 i_1 + x_2 i_2 \\ +x_{12} i_1 i_2 \end{matrix} \middle| \, x_0, x_1, x_2, x_{12} \in \mathbb{R} \right\} \quad (3.6)$$

The classical multi-complex representation, shown above, can be written as a matrix:

$$\left\{ \begin{pmatrix} x_0 & -x_1 & -x_2 & -x_{12} \\ x_1 & x_0 & -x_{12} & -x_2 \\ x_2 & -x_{12} & x_0 & -x_1 \\ x_{12} & x_2 & x_1 & x_0 \end{pmatrix} \middle| \, x_0, x_1, x_2, x_{12} \in \mathbb{R} \right\} \quad (3.7)$$

A similar matrix can be constructed for a tri-complex number (4 complex components) and higher using the same pattern. The higher the multi-complex order the higher the derivative order in the output when a function is evaluated using said multi-complex number. Lastly, Lantoine asserts that multi-complex addition and multiplication maintain the associative and commutative properties of real numbers. This is crucial when taking real-valued functions into the multi-complex realm.

### 3.2 Direct Complex Variable Differentiation

In complex variable Monte Carlo (CVMC) the parameter of interest is perturbed by a small step size, $h$, along an imaginary axis. This perturbed parameter is used to create complex samples. Then the function is evaluated using these samples. The output of this function becomes a complex value where the real part is the real evaluation of the function and the imaginary part is the derivative with respect to the perturbed variable.

In practice the following is the procedure for incorporating Monte Carlo sampling into the Multi-complex variable method.

1. Convert given set of natural parameters to standard: $a, b \rightarrow \mu, \sigma$
2. Perturb standard parameters in the appropriate complex directions by step size $h$ and store in matrix form (below is a bi-complex example).

$$[\mu] = \begin{bmatrix} \mu & -h & -h & h \\ h & \ddots & -h & -h \\ h & -h & \ddots & -h \\ h & h & h & \mu \end{bmatrix} \quad (3.8)$$

3. Convert standard parameters into the appropriate, distribution specific, sampling (natural) parameters (now in multi-complex form). $[\mu],[\sigma] \rightarrow [a],[b]$
4. Generate complex or multi-complex random samples via the appropriate inverse CDF (distributions dependent).

$$[x] = 1 - CDF([a],[b],[c],U) \quad (3.9)$$

where $U$ is a random number between 0 and 1.

5. Evaluate the function using these samples and collect the imaginary terms from the output to determine the derivatives of interest.

### 3.2 Integral Forms of Functions

Direct CVMC requires converting functions of real numbers into functions of matrices (representing complex numbers). In Higham's book, *Functions of Matrices,* several methods are presented to solve the problem of taking the logarithm, exponential, square root and gamma function of a matrix [10].

Logarithms, square roots and the gamma function can be represented at the following integrals:

*Logarithm*

$$\log(A) = \int_0^1 (A-I)\left[t(A-I)+I\right]^{-1} dt \quad (3.10)$$

*Square Root*

$$A^{1/2} = \frac{2}{\pi} A \int_0^\infty \left(t^2 I + A\right)^{-1} dt \quad (3.11)$$

*Gamma Function*

$$\Gamma(A) = \int_0^\infty \text{expm}\left((A-I)\log(t)\right) e^{-t} dt \quad (3.12)$$

where $A$ is a $A \in \mathbb{C}^{n \times n}$ matrix, $I$ is the identity matrix, and *expm* is a built in function in MATLAB which uses a Pade approximation to

take the exponential of a matrix. MATLAB's built-in integral command was used to evaluate these functions. Higham [10] provides one stipulation to these integral forms: matrix $A$ must have no eigenvalues on the negative portion of the real numbers.

### *3.2.1 2P Weibull Transformations*

Every distribution except the Weibull distribution has an invertible mapping between standard and natural parameters. Non-Weibull transformations are easily converted into functions of multi-complex variables and parameters. The Weibull distribution does not have such an invertible relationship between standard and natural parameters. A Newton-Raphson iteration method for mapping from standard to natural (used for sampling) parameters was developed.

Ang and Tang [3] provide a formulation for the mean and standard deviation of the Weibull distribution.

$$\mu = \beta\Gamma\left(\frac{1}{\alpha}+1\right) \tag{3.13}$$

$$\sigma = \beta\sqrt{\Gamma\left(\frac{2}{\alpha}+1\right)-\Gamma\left(\frac{1}{\alpha}+1\right)^2} \tag{3.14}$$

where $\alpha$ and $\beta$ are the shape and scale parameters, respectively.

In order to use a Newton-Raphson iteration method to estimate the natural parameters of the Weibull distribution, the shape parameter is isolated using Eq. 3.13 and 3.14.

$$\frac{\mu^2+\sigma^2}{\mu^2} = \frac{\Gamma\left(\frac{1}{\alpha}+1\right)}{\Gamma\left(\frac{2}{\alpha}+1\right)^2} \tag{3.15}$$

Nielsen [16] and Cousineau [7] use a similar method of moments to isolate one parameter and use the sample mean and variance of data to calculate said parameter. Below is a reformulation of Eq. 3.16.

$$f(\alpha) = \frac{\Gamma\left(\frac{1}{\alpha}+1\right)}{\Gamma\left(\frac{2}{\alpha}+1\right)^2} - \frac{\mu^2+\sigma^2}{\mu^2} \tag{3.16}$$

Using an initial guess for the shape parameter to determine $f(\alpha)$ and $\partial f(\alpha)/\partial\alpha$

$$\frac{\partial f(\alpha)}{\partial\alpha} = 2\Gamma\left(\frac{2}{\alpha}+1\right)\frac{\psi\left(\frac{1}{\alpha}+1\right)-\psi\left(\frac{2}{\alpha}+1\right)}{\alpha^2\Gamma\left(\frac{1}{\alpha}+1\right)^2} \tag{3.17}$$

After evaluating Eq. 3.14 and 3.15 a new shape parameter is determined below.

$$\alpha_{New} = \alpha_{Old} - \frac{\partial f(\alpha)/\partial\alpha}{f(\alpha)} \tag{3.18}$$

This new shape parameter is fed into the beginning of the algorithm (Eq. 3.16) and the process is repeated until the difference between new and old shape parameters is less than a set tolerance (1E-12). After determining the shape parameter, Eq. 3.12 is used to compute the scale parameter.

### *3.3.1 The Minimum function*

The model for the system time to failure of both the mock engine and RGB module models remains as follows:

$$g = \min(\mathbf{X}^K)$$

where, $\mathbf{X}^K$, is a vector of realizations for each variable input. This section explores other formulations of the minimum function based on the work of V.L. Rvachev [20]. These new formulations assist in finding the minimum between complex numbers since in DCVMC all samples are now complex.

### *3.3.2 Rvachev's Representations*

Rvachev defines R-Functions as a real-valued function whose sign is determined by the sign of their arguments [22]. These functions were suggested and developed by Russian mathematician, V.L. Rvachev. Unfortunately, most of his and his colleagues' works on R-Functions have not been translated from the original Russian texts [22].

The max and min functions are a subset of the family of R-functions and while there are a wide variety of applications there are a few properties worth noting [22].

$$\min(x,y)\max(x,y) = xy \tag{3.19}$$

$$\min(x,y)+\max(x,y) = x+y \tag{3.20}$$

After some algebraic manipulation, Eqns. 3.19 – 3.20 can be rearranged in the following manner:

$$\min(x,y)^2-(x+y)\min(x,y)+(xy)=0 \quad (3.21)$$

Another result from these properties is:

$$\max(x,y)^2-(x+y)\max(x,y)+(xy)=0 \quad (3.22)$$

As a result the min and the max between x and y are the smallest and largest root, respectively, of the quadratic equation:

$$Z^2-(x+y)Z+xy=0 \quad (3.23)$$

Using the quadratic formula the roots of Eq. 3.23 are defined as

$$Z=\frac{1}{2}\left(x+y\pm\sqrt{x^2+y^2-2xy}\right) \quad (3.24)$$

The smaller and larger roots are

$$\min(x,y)=\frac{1}{2}\left(x+y-\sqrt{(x-y)^2}\right) \quad (3.25)$$

$$\max(x,y)=\frac{1}{2}\left(x+y+\sqrt{(x-y)^2}\right) \quad (3.26)$$

Rvachev generalizes the Minimum operator as a family of functions [22],

$$x\wedge_\alpha y=\frac{1}{1+\alpha}\left(x+y-\sqrt{x^2+y^2-2\alpha xy}\right) \quad (3.27)$$

When $\alpha=1$ this function becomes the original minimum function shown in Eq. 3.25. At different values of $\alpha$ (other than 1) the function becomes differentiable everywhere except at the origin. By setting $\alpha=0.999....$ with as many nines as computationally feasible the min function can be approximated.

Taking advantage of this property, Rvachev's approximation of the min and max function are reformulated into the following:

$$f(Z)=Z^2-(x+y)Z+\alpha xy \quad (3.28)$$

Instead of directly solving for roots using the quadratic formula a Newton-Raphson iterative method was used to solve for the now multi-complex roots (using the formulations below).

$$\frac{df(Z)}{dZ}=2Z-(x+y) \quad (3.29)$$

$$Z=Z-\frac{\partial f(Z)/\partial Z}{f(Z)} \quad (3.30)$$

In this algorithm the initial guess is the lower valued number. Then this guess is input into $f(Z)$ as well as $\frac{df(Z)}{dZ}$. Eq. 3.30 then computes a new value of Z ($x$ and $y$ remain constant). The algorithm is stopped when the absolute difference between the old and new Z value approaches a predetermined tolerance. Since a quadratic polynomial does not contain a square root operator this method holds the potential to run more efficiently (the square root of a matrix is computationally expensive). Using the following property:

$$\min(x,y,z)=\min(x,\min(y,z)) \quad (3.31)$$

the min function iterates through all variables.

## 3.4 Complex Variable Score Function Differentiation

The Score function method (shown below) uses kernels, complex perturbations, and Monte Carlo sampling to calculate the derivative. The sensitivity of interest:

$$\frac{\partial^n \mu_g}{\partial \theta_{X_i}^n}=\int\cdots\int g\left(X^K\right)\kappa_\theta^n\left[f_{X_i}\cdots f_{X_K}\right]dx_i \quad (3.32)$$

where the kernel is defined as

$$\kappa_\theta^n=\frac{\partial^n f_{X_i}}{\partial \theta_{X_i}^n}\frac{1}{f_{X_i}} \quad (3.33)$$

The problem lies in determining the first order, second order, and cross term kernels for all distribution types (in the 22-component system there are four types) analytically. However, using complex perturbations, the kernels are numerically determined.

$$\kappa_\theta^n=f_{X_i}\left(\theta_{X_i}+ih\right)\frac{1}{f_{X_i}\left(\theta_{X_i}\right)} \quad (3.34)$$

Since the variables are considered independent the joint probability density function (JPDF) becomes the product of each random variable's PDF.

$$JPDF=f_{X_i}\cdots f_{X_K} \quad (3.35)$$

Any needed kernel can be calculated by calling on the appropriate PDF and complex-perturbing the necessary parameters.

$$\kappa_{\theta_{X_i}\theta_{X_j}}=f_{X_i}\left(\theta_{X_i}+i_1h\right)f_{X_j}\left(\theta_{X_j}+i_2h\right) \quad (3.36)$$

The advantage of the Complex Variable Score Function over CVMC is the ability to use real samples (less computationally expensive than complex samples) and the built-in min function in MATLAB. Error from the two approximate minimum functions introduced in the previous section can be avoided as well. However, CVSF carries the SF's higher variability. As will be shown in the following

case studies, the Score Function method in general needs more samples in order to decrease variability and inaccuracy of the derivative.

## 3.5 Examples & Verification

In order to verify either complex differentiation method (CVMC or CVSF) several case studies were investigated. First, the kernels produced with CVSF were verified against the analytical solution (Case Study A). Case studies B and C use a three and five random variable case, respectively, to confirm CVMC based derivatives. Finally, case study D takes a two random variable case and uses CVMC to calculate up to second order terms. These were them verified with *Mathematica*©.

### *3.5.1 Case Study A: Kernel Verification*

Single Random Variable Example: 1 standard normally distributed random variable.

$$x \sim N(0,1) \tag{3.37}$$

The following response was chosen for this case study for its higher order derivative behavior.

$$g = x^4 \tag{3.38}$$

Below, the normal kernels up to third order are shown.

$$\kappa_{\mu}^{(1)} = \frac{x-\mu}{\sigma} \tag{3.39}$$

$$\kappa_{\mu}^{(2)} = \frac{\mu^2 - \sigma^2 - 2\mu x + x^2}{\sigma^4} \tag{3.40}$$

$$\kappa_{\mu}^{(3)} = \frac{-(\mu - x)(\mu^2 - 3\sigma^2 - 2\mu x + x^2)}{\sigma^4} \tag{3.41}$$

For this case study, the integral in Eq. 2.3 was evaluated analytically using the software package, *Mathematica*©. The results are shown below:

$$\partial \mu_g / \partial \mu_X = 4\mu\left(\mu^2 + 3\sigma^2\right) \tag{3.42}$$

$$\partial^2 \mu_g / \partial \mu^2 = 12\left(\mu^2 + \sigma^2\right) \tag{3.43}$$

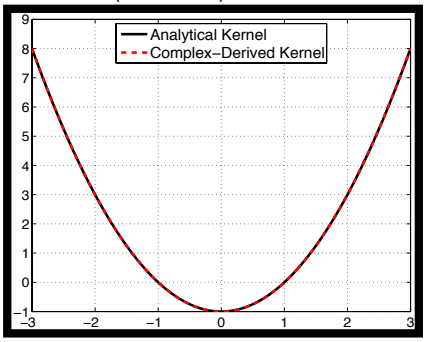


Fig.3.2. Second Order Normal Kernel

$$\partial^3 \mu_g / \partial \mu_X^3 = 24\mu \tag{3.44}$$

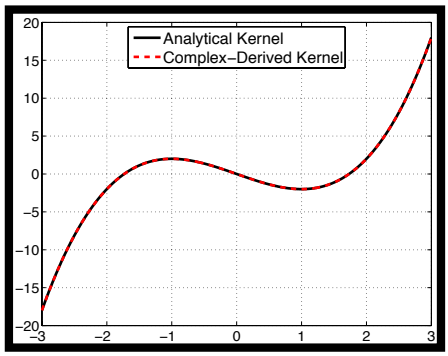


Fig.3.2. Second Order Normal Kernel

As demonstrated in the figures above, CVSF perfectly agrees with the analytical representation of these kernel functions. The figures represent each normal based kernel evaluated in the space between -3 and 3 (span of 6 standard deviations).

### *3.5.2 Case Study B: 3 Random Variables – Linear Response*

This linear case contains three random variables, which are normally distributed with a mean of 1 and standard deviation of 0.1.

$$X_i \sim N(1,0.1) \tag{3.45}$$

$$g = \sum_i^3 X_i \tag{3.46}$$

The function, g, (Eq. 3.46) relates the time-to-failure of the entire system as the sum of the time-to-failure of three components. Since the function is linear the analytical solution for the mean of g, $\mu_g$, can be determined as:

$$\mu_g = \sum_i^3 \mu_{X_i} \tag{3.47}$$

The derivatives of this function with respect to statistical moment parameter can be easily determined as:

$$\partial \mu_g / \partial \mu_{X_i} = 1, \partial \mu_g / \partial \sigma_{X_i} = 0 \tag{3.48-49}$$

| | SF | | CVMC | |
|---|---|---|---|---|
| | $\partial\mu_g/\partial\mu_{X_i}$ | $\partial\mu_g/\partial\sigma_{X_i}$ | $\partial\mu_g/\partial\mu_{X_i}$ | $\partial\mu_g/\partial\sigma_{X_i}$ |
| **$X_1$** | 0.82 | -0.37 | 1.00 | 0.01 |
| **$X_2$** | 0.97 | 0.23 | 1.00 | -0.0016 |
| **$X_3$** | 0.65 | 0.28 | 1.00 | 0.0063 |

Table 3.1. Comparison between SF and CVMC derived derivatives

Using 1E4 samples both the Score Function and CVMC methods computed similar derivatives. As suggested earlier, the Score Function requires more samples for the derivative to converge towards the correct value.

### *3.5.3 Case Study C: 2 Random Variables – Minimum function response*

Two normally distributed random variables were used with the parameters:

$$x, y \sim N(1, 0.1) \qquad (3.50)$$

The verification of the calculated derivatives was performed using *Mathematica*©. A script was written to numerically determine the derivative from the integral in Eq. 2.3. Below in are the results of the results.

| Derivative | Value |
|---|---|
| $\partial\mu_g/\partial\mu_X$ | 0.500 |
| $\partial\mu_g/\partial\sigma_X$ | -0.282 |
| $\partial^2\mu_g/\partial\mu_X^2$ | -2.82 |
| $\partial^2\mu_g/\partial\sigma_X^2$ | -1.410 |
| $\partial^2\mu_g/\partial\mu_X\partial\mu_Y$ | 2.82 |
| $\partial^2\mu_g/\partial\sigma_X\partial\sigma_Y$ | 1.405 |

Table 3.2. Derivatives Values for the Minimum between two variables

*note: There are total 14 derivatives. Table. 3.2 represents non-trivial solutions & non-repeated derivatives.

#### *3.5.3.1 Variance of Derivatives*

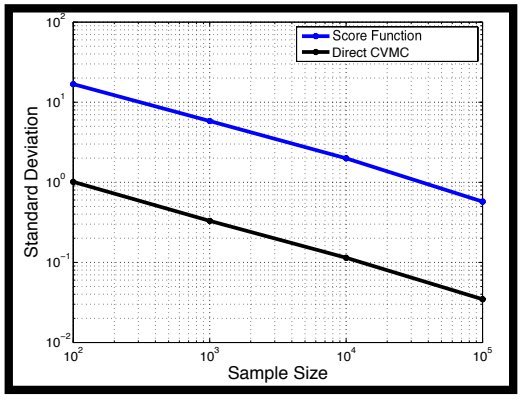


Fig. 3.3. Standard Deviation of $\partial^2\mu_g/\partial\sigma_X^2$

Figures 3.3 and 3.4 display (on a log-log plot) the standard deviation of the two higher order derivatives. The derivative was computed 100 times at each samples size (varied between 100 and 100,000 samples) to determine the variance.

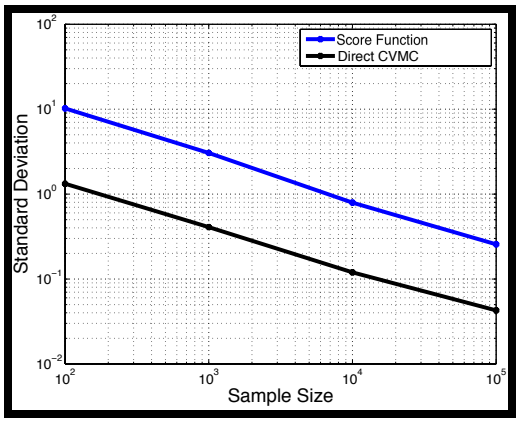


Fig. 3.4 Standard Deviation of $\partial^2\mu_g/\partial\mu_X\partial\mu_Y$

Direct CVMC provides an order of magnitude improvement in the derivative variance. However, the CVSF uses real samples which can be computationally more efficient.

#### *3.5.3.2 Derivative Accuracy*

The second metric in this head to head comparison gauges the accuracy of the derivative for each increase in sample size. The relative error determines the difference between the "true value" of the derivative and its approximation. In this analysis the "true" values used are the numerical derivatives obtained through *Mathematica*© (shown in Table. 2). The algorithm for this comparison is shown below in Eq. 3.51.

$$error = \left| \frac{\partial^n\mu_g/\partial\theta_{X_i}^n - \partial^n\mu_g\hat{/}\partial\theta_{X_i}^n}{\partial^n\mu_g/\partial\theta_{X_i}^n} \right| \times 100 \qquad (3.51)$$

where $\partial^n\mu_g\hat{/}\partial\theta_{X_i}^n$ is the approximate derivative.

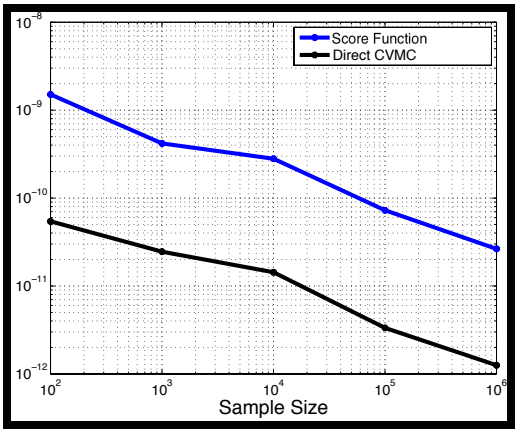


Fig. 3.5. Relative Error of $\partial^2\mu_g/\partial\sigma_X^2$

The plots above and below are on a log-log scale. A first order, second order full, and a mixed derivative are examined in this analysis. Working under the assumption that *Mathematica©* is computing accurate derivatives these figures support direct CVMC as the method of choice with regard to accuracy. Similar to the variance analysis, direct CVMC shows an order of magnitude decrease in relative error with respect to *Mathematica's* results. However, direct CVMC uses an approximation to the Minimum function. In the next chapter, this approximation, though convenient, loses accuracy when calculating higher order derivatives. As a result the Score function method becomes a viable method that circumvents analyticity issues of the minimum function while avoiding the need of an "approximate" function.

### *3.5.4 Case Study D: 2 Random Variables – Minimum function response: Higher Order Derivatives*

Two normally distributed random variables were used with the parameters, $\mu = 1$ and $\sigma = 0.1$.

| **% Difference** | $\frac{\partial\mu_g}{\partial\mu_X}$ | $\frac{\partial^2\mu_g}{\partial\mu_X^2}$ | $\frac{\partial^3\mu_g}{\partial\mu_X^3}$ | $\frac{\partial^4\mu_g}{\partial\mu_X^4}$ |
|---|---|---|---|---|
| **Solution** | 0.5 | -2.82 | 2E-14 | 141.05 |
| **N=1E4** | 0.97 | 6.89 | > 100 | > 100 |
| **N=1E5** | 0.77 | 2.67 | > 100 | > 100 |
| **N=1E6** | 0.02 | 0.21 | > 100 | > 100 |

Table 3.3. Solution vs. Direct CVMC of $\partial^n/\partial\mu_X^n$

| **% Difference** | $\frac{\partial\mu_g}{\partial\sigma_X}$ | $\frac{\partial^2\mu_g}{\partial\sigma_X^2}$ | $\frac{\partial^3\mu_g}{\partial\sigma_X^3}$ | $\frac{\partial^4\mu_g}{\partial\sigma_X^4}$ |
|---|---|---|---|---|
| **Solution** | -0.282 | -1.41 | 21.15 | -317.36 |
| **N=1E4** | 4.56 | 3.11 | 208.36 | > 300 |
| **N=1E5** | 0.97 | 1.37 | 202.89 | > 300 |
| **N=1E6** | 0.48 | 0.12 | > 300 | > 300 |

Table 3.4. Solution vs. Direct CVMC of $\partial^n/\partial\sigma_X^n$

Both Table 3.3 and 3.4 show a significant improvement as sample size is increased up to second order derivatives. Higher order derivatives did not improve accordingly. This lack of higher order derivative accuracy could be due to the minimum function being used. In the analyticity analysis of the minimum function it was shown that in order to extract higher order derivatives the exact function could not relied upon due to the its lack of analyticity. As a result, an approximation (99.999%) of the minimum function was used.

#### *3.5.4.1 Complex Variable Score Function: Higher Order Derivatives*

The same analysis was done using CVSF on this case study. In the figures below the derivative of interest was determined 100 times in order to produce the standard deviation. This process was replicated 3 more times; increasing the sample size by an order of magnitude after each step.

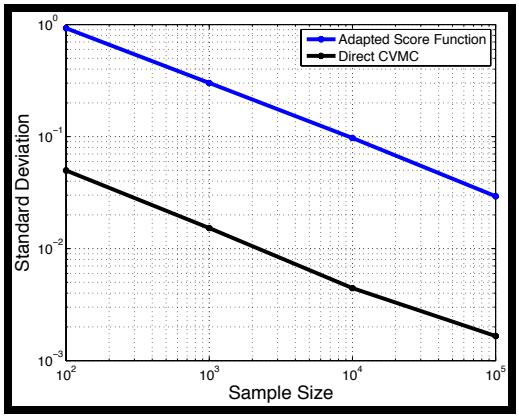


Fig. 3.5. CVSF: Standard Deviation of $\partial\mu_g/\partial\mu_X$

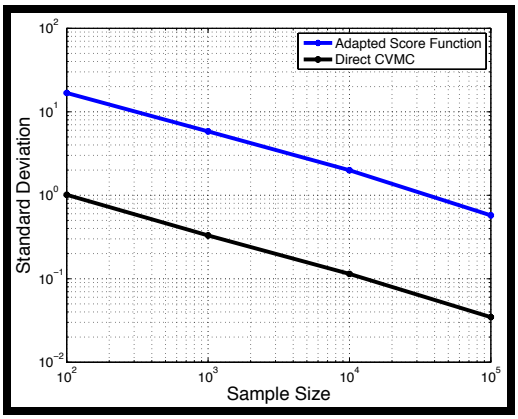


Fig. 3.5. CVSF: Standard Deviation of $\partial^2\mu_g/\partial\mu_X^2$

Direct CVMC, as seen above, has the advantage of producing numerical derivatives with a standard deviation an order of magnitude less than the adapted Score Function method. However, the accuracy of the derivative has a significant effect on the Taylor series approximation. While more samples do not improve higher order derivative accuracy in Direct CVMC, Score Function guarantees better results with more samples.

| % Difference | $\frac{\partial\mu_g}{\partial\mu_X}$ | $\frac{\partial^2\mu_g}{\partial\mu_X^2}$ | $\frac{\partial^4\mu_g}{\partial\mu_X^4}$ |
|---|---|---|---|
| **Solution** | 0.5 | -2.82 | 141.05 |
| **N=1E4** | 9.3 | 22.6 | 155.9 |
| **N=1E5** | 0.6 | 1.6 | 40.2 |
| **N=1E6** | 1.9 | 0.9 | 25.7 |
| **N=1E7** | 0.1 | 1.0 | 4.8 |
| **N=1E8** | 0.3 | 0.1 | 1.0 |

Table 3.5. Solution vs. CVSF of $\partial^n/\partial\mu_X^n$

| % Difference | $\frac{\partial\mu_g}{\partial\sigma_X}$ | $\frac{\partial^2\mu_g}{\partial\sigma_X^2}$ | $\frac{\partial^3\mu_g}{\partial\sigma_X^3}$ | $\frac{\partial^4\mu_g}{\partial\sigma_X^4}$ |
|---|---|---|---|---|
| **Solution** | -0.282 | -1.41 | 21.15 | -317.357 |
| **N=1E4** | 22.6 | 62.1 | 174.9 | 2107.2 |
| **N=1E5** | 3.3 | 43.4 | 179.9 | 1803.4 |
| **N=1E6** | 0.9 | 34.9 | 5.7 | 769.4 |
| **N=1E7** | 1.0 | 7.0 | 11.6 | 60.9 |
| **N=1E8** | 0.13 | 0.50 | 13.10 | 7.40 |

Table 3.6. Solution vs. CVSF of $\partial^n/\partial\sigma_X^n$

Tables 3.5 and 3.6 confirm that increasing the sampling decreases the relative error of the derivatives. However in some cases (Table 4) a minimum of 1E7 samples is needed to attain approximate derivatives, especially for third order and higher terms. Luckily in the adapted Score Function these samples are real and not complex, allowing the only expense being the amount of available memory. The computational cost comes into play when evaluating the PDFs with complex perturbations.

## 4 Sensitivity Results

### 4.1 Mock Engine Model

The Mock Engine model comprises of five different failure modes each fitted with a unique probability distribution. The system time to failure of the entire system is defined as the minimum time to failure between all five components.

In MATLAB© the realizations were created using built-in inverse sampling commands For each random variable in the model 1E4 realizations were created and evaluated through the system response (the minimum function). The built-in minimum function in MATLAB was used for efficiently. Also, without much additional expense, this function outputs which variable in the system contributed to the system minimum.

| $X_i$ | $\%X_{g_i}$ |
|---|---|
| 1 | 15% |
| 2 | 37% |
| 3 | 1% |
| 4 | 27% |
| 5 | 20% |

Table 4.1. % Contribution to Minimum by Variable

According to Table 4.1, the third variable contributes very little to the overall system minimum (least important variable). The second variable contributes 37% of the system minimum in Monte Carlo simulation.

Below in Table 6.2 the results from a CVMC second order derivative analysis are listed. The CVMC method uses a Rvachev approximation of the min function in order to avoid differentiability issues. The minimum function was “smoothed” by using a 99.999%

approximation. However, by using an approximate function higher order derivatives loss significant accuracy.

| $X_i$ | $\partial\mu_g/\partial\mu_{X_i}$ | $X_i$ | $\partial\mu_g/\partial\sigma_{X_i}$ |
|---|---|---|---|
| **2** | 0.362 | **1** | -0.001 |
| **4** | 0.275 | **3** | -0.005 |
| **5** | 0.156 | **5** | -0.139 |
| **1** | 0.008 | **2** | -0.192 |
| **3** | 0.003 | **4** | -0.319 |

Table 4.2. First Order Derivatives: Mock Engine Model

The table above contains the derivatives of interest sorted in descending order. The 2nd variable's mean has the greatest effect on the mean system minimum. However, the fourth variable's standard deviation has the most effect on the system minimum. According to these local derivatives an increase in the mean time to failure and decrease in the variance of the time to failure of the second and fourth variable respectively, would incur the most significant effect on the system mean time to failure.

Table. 4.3, below, is a summary of the important variable as identified by the different methods presented. First listed are the global sensitivity indices, where the first order indices, $S_i$, indicate a variable's effect on the system variance, and, $S_{T_i}$, the total effect (includes higher order indices and interactions) of a random variable. The third method presented, $\%X_{g_i}$, represents which variable in the system most often contributed to the system minimum. Lastly, the final two methods listed indicate the important variable based on its probabilistic effect on the mean minimum time to failure of the model.

| **Method** | $X_i$ |
|---|---|
| **GSA:** $S_i$ | 1 |
| **GSA:** $S_{T_i}$ | 1 |
| $\%X_{g_i}$ | 2 |
| $\partial\mu_g/\partial\mu_{X_i}$ | 2 |
| $\partial\mu_g/\partial\sigma_{X_i}$ | 4 |

Table 4.3. Important Variable: Mock Engine Model

According to Table 4.3 the greatest contributor to the system mean time to failure variance is the first variable. The variable that produces the system minimum is the second variable. The first probabilistic derivative agrees with the previous result: the second variable is most important. This variable's mean has the greatest effect on the mean time to failure of the system. Lastly, the fourth variable's variance enacts the greatest effect on the mean time to failure of the system.

## 4.2 22- Component System

This Gas-Turbine engine module, similar to the Mock engine model, contains 22 different failure modes each fitted with a unique probability distribution. The response that describes the behavior of the module's time to failure is the minimum time to failure between all 22 variables. The same methods are applied to this module.

The RGB module shown above contains 16 variables described by a Weibull distribution, 4 by a lognormal distribution, 1 by an Exponential distribution, and 1 by a Normal distribution.

The percent by variable contribution to the minimum of the model is shown below in Table 4.4. Again, 1E4 samples were used in the analysis, thus each percentage represents the fraction of times a particular variable caused the system minimum.

| $X_i$ | $\%X_{g_i}$ |
|---|---|
| 13 | 13% |
| 16 | 9% |
| 3 | 9% |
| 14 | 8% |
| 12 | 8% |

Table 4.4. Top Five Contributors to 22-CS System Minimum

According to Table 4.4 the 13th variable in the RGB module contributes the most to the system minimum. The 12th, 14th, and 16th variables contribute to 25% of the total minimums of the system in the Monte Carlo simulation. The 7th variable contributes less than 1% of the system time to failure minimums.

Table 4.5, below, displays a sorted list (in descending order) of the derivatives of interest as estimated using the same CVMC method described in the section 6.1.

| $X_i$ | $\partial\mu_g/\partial\mu_{X_i}$ | $X_i$ | $\partial\mu_g/\partial\sigma_{X_i}$ |
|---|---|---|---|
| **13** | 0.1315 | **22** | 7.9E-05 |
| **12** | 0.1044 | **18** | -1.9E-05 |
| **14** | 0.0888 | **3** | -4.3E-04 |
| **6** | 0.0564 | **7** | -1.7E-03 |
| **20** | 0.0497 | **19** | -2.7E-03 |

Table 4.5. First Order Derivatives

The top three derivatives with respect the individual variable mean are the 13, 12, and 14. The 13th variable's standard deviation and mean has the greatest effect on the system MTTF. Both derivatives indicate that an increase and decrease in the mean and variance, respectively, of the 13th variable will have the greatest positive impact on the MTTF of the module (increase in the MTTF).

| **Method** | $X_i$ |
|---|---|
| **GSA:** $S_i$ | 13 |
| **GSA:** $S_{T_i}$ | 13 |
| $\%X_{g_i}$ | 13 |
| $\partial\mu_g/\partial\mu_{X_i}$ | 13 |
| $\partial\mu_g/\partial\sigma_{X_i}$ | 13 |

Table 4.6 Important Variables: RGB Turbine Engine Module

Across different probabilistic analyses the 13th variable in the RGB module is identified at the most important. This variable contributes most to the variance of the module, the minimums in Monte Carlo simulations, and also to the module's MTTF. The probabilistic derivatives indicate that improving the 13th failure mode in the RGB module would have the greatest effect in improving the MTTF of the module.

## 4.3 Concluding Remarks

This paper presents two novel ways to determine probabilistic sensitivities in the form of derivatives: direct Complex Variable Monte Carlo and Complex Variable Score Function. In the case studies it was shown that both methods succeed in producing accurate derivatives. However, for direct CVMC, an approximation to the minimum must be used in order to avoid discontinuities where the method breaks down. As a result higher order derivatives using an Rvachev approximation fail accuracy tests. CVSF prevails in producing higher order derivatives of the minimum function. However, for accuracy, an order of magnitude increase in sampling size is needed. CVSF also gains an advantage in two ways. First, the function of interest is evaluated with real samples (complex sample creation can be avoided). Second, instead of perturbing every sample in a complex direction, the JPDF is perturbed in the appropriate complex direction, *once*. In general the complex differentiation method provides robust numerical derivatives flexible towards smooth functions and problematic functions with discontinuities.

## References


[1] Al-Fawzan, M. A. Methods for estimating the parameters of the Weibull distribution. *King Abdulaziz City for Science and Technology* (91), 2000.

[2] Ang, A. H.-S., & Tang, W. H. *Probability concepts in engineering planning and design* (Vols. 2 (Decision, risk, and reliability)), 1984.

[3] Ang, A. H.-S., & Tang, W. *Probability Concepts in Engineering Planning and Design: Emphasis on Applications in Civil & Environmental Engineering.* Wiley, 2007.

[4] Barlow, R., & Proschan, F. Importance of system components and fault tree events. *Stochastic Processes and their Applications , 3(2)*, 153-173, 1975.

[5] Birnbaum, Z. On the importance of different components in a multicomponent system. *Washington Univ Seattle Lab Of Statistical Research*, 1968.

[6] Cheok, M., Parry, G., & Sherry, R. Use of importance measures in risk-informed regulatory applications. *Reliability Engineering & System Safety , 60(3)*, 213-226, 1998.

[7] Cousineau, D. Fitting the three-parameter Weibull distribution: review and evaluation of existing and new methods. *Dielectrics and Electrical Insulation, IEEE Transactions on , 16(1)*, 281-288, 2009.

[8] Glasserman, P. *Gradient Estimation Via Perturbation Analysis.* Springer, 1991.

[9] Hüsler, J. Limit properties for multivariate extreme values in sequences of independent, non-identically

distributed random vectors. *Stochastic Processes and their Applications , 31(1)*, 105-116, 1989.

[10] Higham, N. J. *Functions of Matrices: Theory and Computation.* SIAM, 2008.

[11] Lantoine, G., Russell, R. P., & Dargent, T. Using Multicomplex Variables for Automatic Computation of High-Order Derivatives. *ACM Trans. Math. Softw. , 38* ( 3, Article 16), 2012.

[12] Louzada, F., Bereta, E., & Franco, M. On the Distribution of the Minimum or Maximum of a Random Number of iid Lifetime Random Variables. *Applied Mathematics , 3(4)*, 2012.

[13] Millwater, H. Universal properties of kernel functions for probabilistic sensitivity analysis. *Probabilistic Engineering Mechanics , 24(1)*, 89-99, 2009.

[14] Millwater, H., & Feng, Y. Probabilistic sensitivity analysis with respect to bounds of truncated distributions. *Journal of Mechanical Design , 133(6)*, 2011.

[15] Natvig, B. *Multistate systems reliability theory with applications.* John Wiley & Sons, 2010.

[16] Nielsen, M. A. Parameter estimation for the two-parameter Weibull distribution. Brigham Young University. Department of Statistics, 2011.

[17] Qiu, Z., Yang, D., & Elishakoff, I. Probabilistic interval reliability of structural systems. *International Journal of Solids and Structures , 45(10)*, 2850-2860, 2008.

[18] Quintana, C., Millwater, H., Singh, G., & Golden, P. Optimal Allocation of Testing Resources for Statistical Simulations. *Engineering Optimization*, 2014.

[19] Rausand, M., & Høyland, A. *System reliability theory: models, statistical methods, and applications* (Vol. Vol. 396). John Wiley & Sons, 2004.

[20] Rvachev, V. *Theory of R-functions and Some Applications.* Kiev: Naukova Dumka, 1974.

[21] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., et al. *Global Sensitivity Analysis: The Primer.* John Wiley & Sons, Ltd, 2008.

[22] Shapiro, V. Theory of R-functions and applications: A primer. Cornell University, 1991.

[23] Van der Borst, M., & Schoonakker, H. An overview of PSA importance measures. *Reliability Engineering & System Safety , 72(3)*, 241-245, 2001.

[24] Wirsching, P., & Millwater, H. Analysis methods for probabilistic life assessment. *ASM International* , 250-268, 2002.

## 8 Contact Author Email Address

Eliseo Iglesias
mailto:eli.iglesias.utsa@gmail.com
Dr. Harry Millwater
mailto: Harry.Millwater@utsa.edu

## Copyright Statement