Theory Manual

Open-Source pyMAPOD Framework – v.beta

Developed by Computational Design Laboratory (CODE Lab)

Department of Aerospace Engineering, Iowa State University

Leading Developers:

Professor: Leifur Leifsson

Ph.D Student: Xiaosong Du

We gratefully acknowledge the support of the Center for Nondestructive Evaluation Industry/University Cooperative Research Program at Iowa State University

Code and updates are available at the link: https://github.com/Fightingwolf2105/MAPOD

Content

1. Introduction

Model-assisted probability of detection (MAPOD) [1, 2] is an important terminology in nondestructive testing (NDT) area [3], not only because it describes the reliability of detecting system, but also it can greatly reduce the required experimental information.

MAPOD calculation has been widely applied to varieties of NDT areas, such as ultrasonic testing, eddy current testing, and x-ray based testing. Researchers, such as J. Aldrin [4, 5], J. Knopp [6], and R. Miorreli [7], have made great progress and improvements. However, as far as the authors know, most of the MAPOD calculations have to rely on commercial software, such as CIVA. Due to these reasons, the authors decide to develop this open-source MAPOD framework, aiming at providing convenient tools to the NDT researchers.

This open-source MAPOD framework is developed using python, which makes it cross-platform, although some necessary python modules are still needed. More details on prerequisites are shown in Chapter 1. Basic mathematical theories, such as linear regression, "ahat vs. a" plots, and probability of detection (POD) calculation, are given in Chapter 2. Chapter 3 has some test cases to demonstrate the process, and validate the results with MIL-HDBK-1823 [8, 9], officially verified software by Department of Defense, United States. Researchers, who have sufficient background in POD calculation or only want to make practical application to their research, can feel free to go through the associated document, User Guide, directly.

The Computational Design (CODE) lab would like to thank the center of nondestructive evaluation (CNDE) for funding this program. In addition, we also want to say thanks to all the colleagues, Dr. Jiming Song, Dr. William Meeker, Dr. Ronald Roberts, Dr. Leonard Bond, etc. for providing physics-based NDT simulation models, valuable ideas and suggestions.

2. Operation environment

2.1 Python on Linux / Windows systems

This open-source framework is constructed using python, which is well known for its cross-platform capability. Therefore, the users can run it on any systems. The authors wrote and tested the code on python v2.7, Window x64 local desktop and laptop machine.

2.2 Suggested compiler and necessary modules

The code is written within Enthought Canopy compiler, which is very powerful and convenient for integrating python toolboxes. Users simply select Package Manager under the toolbar Tools, to add any necessary modules.

In this work, authors are trying to avoid using additional modules as much as possible. The necessary modules and corresponding utilizations are:

collections: the module 'OrderedDict' is used to specify random inputs with statistical distributions
prob_distrs: randomly generate sample points, using latin hypercube sampling (LHS) sheme
numpy: various numerical operations, such as numpy.array
pandas: read data frame from excel file
matplotlib.pyplot: view imported data, generate "ahat vs. a" plots, and POD curves
sys: sys.exit() used when incorrect data format is provided
mlab: link python with Matlab, making Matlab a callable module for python
sklearn: linear_model of sklearn is implemented, in particular, the linear_model.LinearRegression and linear_model.LassoLarsCV are utilized for linear regression
math: used for calculating factorial of integral values

scipy.stats: used for generating normally distributed sample points

3. Mathematical theories

This chapter will talk about least squares method and maximum likelihood method for linear regression in Section 3.1. In Section 3.2 linear regression will be applied to "ahat vs. *a*" regression, followed by POD calculation. Section 3.3 has the details on the calculation of confidence interval.

3.1 Linear regression

3.1.1 Introduction

Linear regression [10] is a statistical method that allows us to summarize and study relationships between two continuous or quantitative variables: one variable, denoted x, is named as independent variable or predictor, while the other one, denoted y, is named as dependent variable or response. The linear relationship between x and y is usually given as:

$$y = \beta_0 + \beta_1 \cdot x + \varepsilon, \tag{1}$$

where β_0 is the intercept of the regression line, β_1 is the slope of the regression line, ε is random error following a zero-mean and constant-standard deviation Normal distribution:

$$\varepsilon \sim Normal(0, \sigma_\varepsilon^2). \tag{2}$$

Due to this random error, which may come from operation system or other noises, the observed value varies slightly each time even at the same x value. And the prediction using estimated parameters usually represents the expected value of y at that x location (Fig. 1).

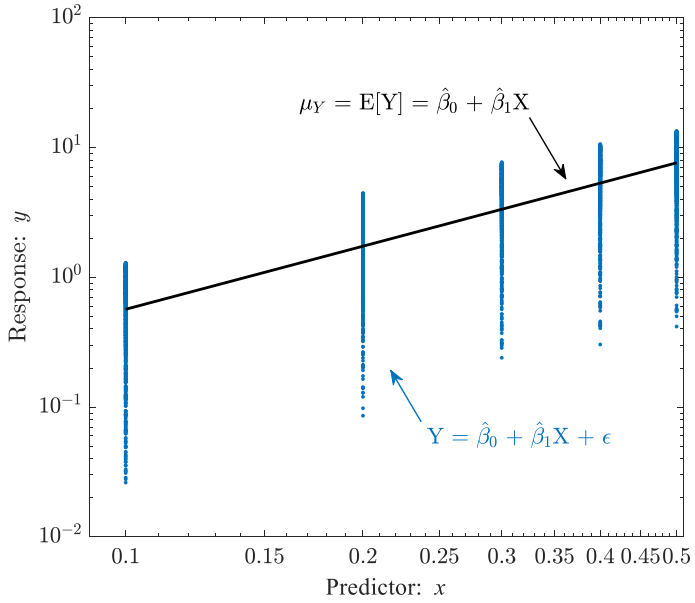

Figure 1. Linear regression line on random observations.

As shown above, the only unknowns in linear regression problems are β_0, β_1, and σ_ε, which can be solved by various methods. Among all these methods, least squares (LS) method and maximum likelihood (ML) method are the most commonly used. We will discuss both of LS and ML methods, and also their relationship, in the following sections.

3.1.2 Least squares method

As shown in Eqn. 1, the linear relation is constructed using β_0, β_1, and σ_ε. It can be converted as:

$$\varepsilon = y - (\beta_0 + \beta_1 \cdot x), \tag{3}$$

which is the residual between observed values and estimation from linear regression line. A widely used method for estimating the coefficients is least squares method [11], which minimizes the sum of the squared of residual at each observed value:

$$\min(S) = \min \sum_{i=1}^{n} (\varepsilon_i)^2 = \min \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2. \tag{4}$$

Taking the first-order derivative of Eqn. 4 about the coefficients β_0 and β_1 as 0, it is straightforward to obtain

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \tag{5}$$

and

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x}, \tag{6}$$

where $\bar{x}$ and $\bar{y}$ are mean values of x and y, respectively, and $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimates of β_0 and β_1, respectively.

Since the expected value of ε

$$E[\varepsilon] = E[Y - (\beta_0 + \beta_1 X)] = 0. \tag{7}$$

From Eqn. 4, it is straightforward to obtain

$$E[\varepsilon^2] = \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2 / n. \tag{8}$$

Therefore, the standard deviation of ε is

$$\sigma_\varepsilon = \sqrt{E[\varepsilon^2] - (E[\varepsilon])^2} = \sqrt{\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 \cdot x_i))^2 / n}, \tag{9}$$

which is actually of the format of the root mean squared error (RMSE).

The Eqn. 4 can be solved by many optimization method, such as Newton method, pattern search or OLS method within python. However, in this code we aim at avoiding prerequisite modules as much as possible, so Eqn. 5, 6, and 9 are packed. And please remember the expression in these three equations, because we will compare them with those estimated from maximum likelihood method, to prove that they are the same under this situation.

3.1.3 Maximum likelihood method

Now let's review the assumption made in linear regression:
1. The distribution of X is arbitrary.
2. If $X = x$, the $Y = \beta_0 + \beta_1 x + \varepsilon$, for some coefficients, β_0 and β_1 and some random noise variable ε.
3. $\varepsilon \sim N(0, \sigma_\varepsilon^2)$, and is independent of X.
4. ε is independent across observations.

Based on these assumptions, the response Y is independent across observations, conditional on the predictor X. Besides, the noise variable ε has zero mean and constant variance, and follows the Normal distribution. Therefore, the conditional probability density function of Y for each x, given arbitrary number of data sets, (x_1, y_1), (x_2, y_2), …, (x_n, y_n), can be written as

$$\prod_{i=1}^{n} p\left(y_i \mid x_i; \beta_0, \beta_1, \sigma_\varepsilon^2\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} e^{\frac{(y_i-(\beta_0+\beta_1 x_i))^2}{2\sigma_\varepsilon^2}}. \tag{10}$$

For any estimates on unknown parameters, β_0, β_1, and σ_ε, the pdf becomes

$$\prod_{i=1}^{n} p\left(y_i \mid x_i; \hat{\beta}_0, \beta_1, \sigma_\varepsilon^2\right) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\hat{\sigma}_\varepsilon^2}} e^{\frac{(y_i-(\hat{\beta}_0+\beta_1 x_i))^2}{2\hat{\sigma}_\varepsilon^2}}, \tag{11}$$

which is called likelihood, a function of the parameter values. For the convenience of calculation, usually it is taken as log-likelihood,

$$L\left(\hat{\beta}_0, \beta_1, \hat{\sigma}_\varepsilon^2\right) = \log \prod_{i=1}^{n} p\left(y_i \mid x_i; \hat{\beta}_0, \beta_1, \sigma_\varepsilon^2\right), \tag{12}$$

and so,

$$L\left(\hat{\beta}_0, \beta_1, \hat{\sigma}_\varepsilon^2\right) = -\frac{n}{2}\log 2\pi - n\log\sigma_\varepsilon - \frac{1}{2\hat{\sigma}_\varepsilon^2}\sum_{i=1}^{n}\left(y_i - (\hat{\beta}_0 + \beta_1 x_i)\right)^2. \tag{13}$$

We can maximize Eqn. 13 to get the best estimates on unknowns. This method is called maximum likelihood [12]. Any optimization methods can be used to maximize Eqn. 13, like what was mentioned above for least squares method. We can still use the same method, taking first-order derivative of Eqn. 13 and setting it as 0, to obtain

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{14}$$

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x}, \tag{15}$$

$$\sigma_\varepsilon = \sqrt{\sum_{i=1}^{n} \left(y_i - (\beta_0 + \beta_1 \cdot x_i) \right)^2 / n}. \tag{16}$$

It is obvious that the estimation equations above are the same as Eqn. 5, 6, and 9. Therefore, in the framework, we code up the numerical expression of unknown parameters, directly, without having to claim which method it comes from. The authors prefer to maximum likelihood method, because it is more convenient and straightforward to compute Fisher information matrix and apply Wald method for confidence intervals on "ahat vs. a" plots and POD curves. More details on this type of confidence interval calculation are given in Section 2.3.2.

3.2 POD calculation

3.2.1 Background

The concept of POD was initially developed to quantitatively describe the detection capabilities of NDT systems, starting from pioneering work since the late 1960's for the aerospace industry. POD curves have been widely generated through various NDT equipment, such as ultrasound, eddy currents, magnetic particle inspection and radiography, focusing on different quantities of interests. A commonly used term is "90% POD" and "90% POD with 95% confidence interval", which are written as a_{90} and $a_{90/95}$, respectively. POD curves were initially only based on experiments, however, to save computational budgets it can be enhanced by utilizing physics-based computational models, which is known as the MAPOD methodology. Here in current work, POD is performed using linear regression method.

3.2.2 "ahat vs. *a*" regression based POD

For signal response data, much more information is supplied in the signal for analysis than is in hit/miss data. Here, POD function is generated from the correlation of "ahat vs. *a*" data [8, 9]. And through reviews on experiments data, it shows a log-log scale between ahat and *a*:

$$\ln \hat{a} = \beta_0 + \beta_1 \ln a + \varepsilon, \tag{17}$$

where the coefficients β_0 and β_1 can be determined by the maximum likelihood method, and the ε has a Normal distribution with zero mean and standard deviation σ_ε, $N(0, \sigma_\varepsilon)$. This standard deviation can be determined by the residuals of the observed data, as shown in Section 2.1.

The POD can be obtained as the probability that the obtained signal lies above arbitrary user-defined threshold:

$$POD(a) = 1 - \Phi\left[\frac{\ln \hat{a}_{threshold} - (\beta_0 + \beta_1 \ln a)}{\sigma_\varepsilon} \right], \tag{18}$$

where Φ is the standard normal distribution function.

From Eqn. 18, it is straightforward to obtain:

$$POD(a) = \Phi\left[\frac{\ln a - \dfrac{\ln \hat{a}_{threshold} - \beta_0}{\beta_1}}{\dfrac{\sigma_\delta}{\beta_1}} \right], \tag{19}$$

which is a cumulative log-normal distribution function with mean μ and standard deviation σ given by:

$$\mu = \frac{\ln \hat{a}_{threshold} - \beta_0}{\beta_1}, \tag{20}$$

$$\sigma = \frac{\sigma_\varepsilon}{\beta_1}, \tag{21}$$

where the parameters β_0, β_1, and σ_ε can be obtained by least squares method, maximum likelihood method or the numerical expression discussed in Section 2.1.

3.3 Confidence interval

In statistics, especially when uncertainty exists in NDT system, it is impossible to specify a value to a variable with 100% certainty. POD requires two numerical values, a_{lower} and a_{upper}, depending on the sample set, and varying for each random set. The interval within a_{lower} and a_{upper} is called a "confidence interval" [8, 9], which is usually expressed as:

$$P(a_{lower} \le a \le a_{upper}) = const, \tag{22}$$

where the *const* is the "confidence level", a_{lower} is called the "lower confidence limit" and a_{upper} is called the "upper confidence limit". In POD calculation, only lower confidence limit is used. If confidence level is set as 95%, the interpretation is based on repeated sampling, meaning if samples of the same size are drawn repeatedly from a population and a confidence interval is calculated from each sample, then we can expect 95% of these different intervals to contain the true value.

In POD calculation, the defect size a90 meaning the 90% probability to be detected is always considered within application of POD or framework of design. To take the uncertainty into account, the upper bounds of 95% confidence interval is considered, written as $a_{90/95}$. Note that these two values are not characteristic properties of an NDT system, but rather are calculated from the particular random results.

There are various methods, such as bootstrap, Wald method, and likelihood ratio method, existing in the area of confidence interval calculation. In this work, we will talk about bootstrap due to its simplicity, and Wald method due to efficiency. In current version of the framework, we select to pack up the Wald method. Bootstrap method is introduced here as a comparison on the POD results in Section 3.

3.3.1 Bootstrap

Brad Efron invented a revolutionary new statistical procedure called the bootstrap [13, 14], in 1979. This is a computer-intensive procedure that substitutes fast computation for theoretical mathematics. The main benefit of the bootstrap is the confidence intervals on parameters without having to make unreasonable assumptions.

The idea of bootstrap is simple:

(1) Gather the sample data set (x_1, y_1), (x_2, y_2), … (x_n, y_n), and use it to estimate the unknown parameters, $\boldsymbol{\theta}$.

(2) Draw a new random sample of size n, with replacement, from the sample data set, and estimate the unknowns.

(3) Repeat step (2) as many times as necessary or user-defined arbitrary number of times, e.g. 1,000.

(4) Put the 1000 additional estimates on unknown parameters into an ascending order, separately.

(5) Confidence intervals can be obtained, based on the ordered sets of estimates. For example, 95% lower confidence interval is the 975th value, $\hat{\theta}^*_{975}$ while 95% upper confidence interval is the 25th value, $\hat{\theta}^*_{25}$.
(6) This step is optional. We can calculate pivot confidence interval, following the formula

$$CI_{lower} = 2\hat{\theta} - \theta^*_{975}, \tag{23}$$

$$CI_{upper} = 2\hat{\theta} - \theta^*_{25}, \tag{24}$$

where $\hat{\theta}$ is the estimate on θ from the original set, CI_{lower} and CI_{upper} are lower and upper confidence interval, respectively.

The samples obtained from step 2 and 3, are called bootstrap samples. And each of newly generated bootstrap samples approximates the original set of data, making the new estimates on **θ** approximate the results from original set. This approximating distribution is used to set confidence interval.

3.3.2 Wald method

Wald method [15, 16] is a well-known likelihood-based procedure for calculating confidence interval, and usually performs well in large samples. For a location-scale distribution or for a distribution which can be transformed to a location-scale distribution, the Wald confidence interval is easy to compute for quantiles. Therefore, it is suitable for exponential, Weibull, and lognormal distributions. The MIL-HDBK-1823, the officially used POD software from the department of defense (DOD), United State, also utilizes this method for calculation of confidence interval.

A. Fisher information

In mathematical statistics, Fisher information [17] is used for measuring the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X. Formally, it is the expected value of the observed information. Observed information is the negative of the second derivative, the Hessian matrix, of the "log-likelihood" (the logarithm of the likelihood function).

Suppose that we observe random variables, $X_1, X_2, \ldots, X_n$, independently and identically distributed with density $f(X; \boldsymbol{\theta})$, where **θ** is assumed to be a n-dimensional vector. The log-likelihood of the parameters **θ** given the data $X_1, X_2, \ldots, X_n$ is

$$l(\boldsymbol{\theta} \mid X_1, X_2, ..., X_n) = \sum_{i=1}^{n} \log f(X_i \mid \boldsymbol{\theta}). \tag{25}$$

Then the observed information matrix can be obtained as

$$J(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial^2}{\partial \theta_1^2} & \frac{\partial^2}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2}{\partial \theta_1 \partial \theta_n} \\ \frac{\partial^2}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2}{\partial \theta_1^2} & \cdots & \frac{\partial^2}{\partial \theta_2 \partial \theta_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2}{\partial \theta_p \partial \theta_1} & \frac{\partial^2}{\partial \theta_p \partial \theta_2} & \cdots & \frac{\partial^2}{\partial \theta_n^2} \end{pmatrix} l(\boldsymbol{\theta}). \tag{26}$$

With observed information matrix ready, the Fisher information can be obtained

$$I(\boldsymbol{\theta}) = E\left(J\left(\boldsymbol{\theta}\right)\right). \tag{27}$$

B. Wald method

As mentioned above, Wald method is widely used for the calculation of confidence interval, and is simple to apply. Based on the Lawless' general procedure for a location-scale distribution, computational details for confidence interval using Wald method is given as follows.

Let x_p be the quantile of a location-scale distribution with parameter u and b respectively, while w_p is the quantile of the same distribution with $u = 0$, and $b = 1$. Then, $x_p = u + w_p b$ which we can estimate by

$$\hat{x}_p = u + w_p \hat{b}, \tag{28}$$

using maximum likelihood (ML) estimates $\hat{u}$ and $\hat{b}$. The pivotal quantity is

$$Z_p = \frac{\hat{x}_p - x_p}{se(\hat{x}_p)}, \tag{29}$$

where

$$se(\hat{x}_p) = \left(\mathrm{var}(u) + w_p^2\,\mathrm{var}(\hat{b}) + 2w_p\,\mathrm{cov}(\hat{u}, b)\right)^{1/2}, \tag{30}$$

where the variance and covariance terms come from the asymptotic covariance matrix for $(\hat{u}, \hat{b})$, which is the inverse of Fisher's observed information matrix, $I(\hat{u}, \hat{b})$, evaluated at $(\hat{u}, \hat{b})$. The diagonal elements of $(I(\hat{u}, \hat{b}))^{-1}$ give the variances and the off-diagonal elements give the covariance.

Due to assumption of the asymptotic normality of maximum likelihood estimates, Z_p is approximately $N(0, 1)$. Let Z be a $N(0, 1)$ random variable and z_α be the value such that

$$P(Z < z_\alpha) = \alpha. \tag{31}$$

A Wald 100(1 - α)% CI for x_p is given by

$$\left(\hat{x}_p + z_{\frac{\alpha}{2}} se(x_p), \hat{x}_p - z_{\frac{\alpha}{2}} se(x_p)\right). \tag{32}$$

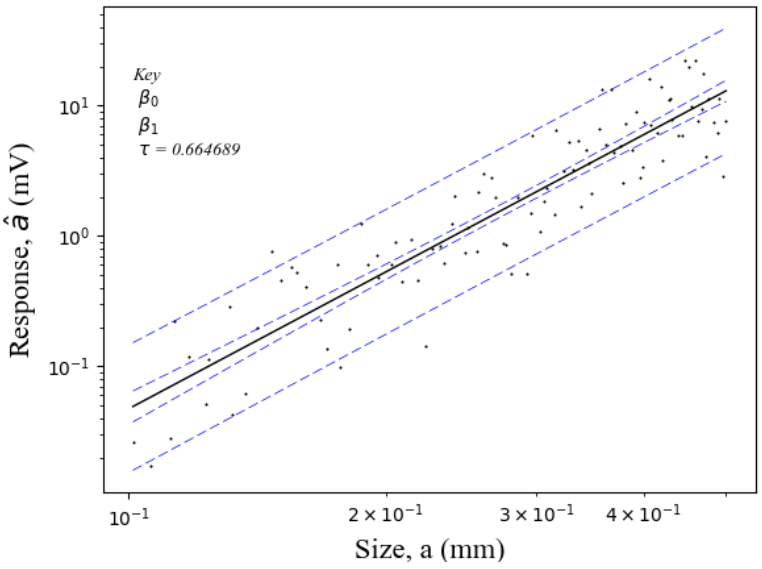

Figure 2. "ahat vs. *a*" regression, within bounded lines.

C. Application on "ahat vs. *a*" regression and POD curves

As mentioned in [8, 9], "ahat vs. *a*" regression is constructed with a regression line (the solid), surrounded by two sets of nearly parallel bounds (dotted lines), as shown in Fig. 2.

The innermost set is the 95% confidence bounds on the line itself. The outer set of dotted lines is called the 95% prediction bounds. A new response value is expected to be contained by these bounds in 95 of 100 similar situations. Usually these sets of lines go further at both ends, meaning less confidence in the solid line as we get further from the centroid of the data. For linear regression problem, we obtain the estimates on the intercept and slope of the solid line, also with uncertainties on those estimates. Near the centroid the uncertainty in the slope has little influence, but becomes increasingly influential away from the centroid, resulting in this "dog-bone" confidence bounds.

The estimated response, $\hat{y}$, is given by the regression equation

$$\hat{y} = \hat{\beta}_0 + \beta_1 x. \tag{33}$$

Based on the variance of a sum, the variance on $\hat{y}$ can be expressed as

$$\text{var}(\hat{y}) = \text{var}(\hat{\beta}_0 + \beta_1 x) = \text{var}(\hat{\beta}_0) + 2x\,\text{cov}(\beta_0, \hat{\beta}_1) + x^2\,\text{var}(\beta_1), \tag{34}$$

from which the 95% Wald confidence bounds on $\hat{y}$ can be constructed as

$$\hat{y}_{\alpha=0.95} = y \pm 1.645 se_{\hat{y}} = \hat{\beta}_0 + \beta_1 x \pm 1.645\sqrt{\text{var}(\hat{y})}. \tag{35}$$

The 95% prediction bounds can be constructed following the same process, except that the variance of the random residual also needs to be included

$$\mathrm{var}_{total}(\text{资料}) = \mathrm{var}(y) + \sigma_\varepsilon^2. \tag{36}$$

The variances and covariance terms in Eqn. 34 can be obtained from the inverse matrix of Fisher information. The distribution of model response, y, has the same format of Eqn. 10, and the corresponding log-likelihood follows Eqn. 13. Thus, the resulted Fisher information matrix is

$$I\left(\hat{\beta}_0, \beta_1, \hat{\sigma}_\varepsilon\right) = \begin{pmatrix} \frac{n}{\hat{\sigma}_\varepsilon^2} & \frac{\sum_{i=1}^n X_i}{\sigma_\varepsilon^2} & 0 \\ \frac{\sum_{i=1}^n X_i}{\hat{\sigma}_\varepsilon^2} & \frac{\sum_{i=1}^n X_i^2}{\sigma_\varepsilon^2} & 0 \\ 0 & 0 & \frac{2n}{\hat{\sigma}_\varepsilon} \end{pmatrix}, \tag{37}$$

then the covariance matrix has the format of

$$Var\left(\hat{\beta}_0, \beta_1, \hat{\sigma}_\varepsilon\right) = \begin{pmatrix} V_{00} & V_{01} & V_{02} \\ V_{10} & V_{11} & V_{12} \\ V_{20} & V_{21} & V_{22} \end{pmatrix}. \tag{38}$$

When applying the Wald method to POD curves, the covariance matrix on $\hat{\mu}$ and $\hat{\sigma}$ is can be calculated from Eqn. 38

$$Var\left(\hat{\mu}, \hat{\sigma}\right) = \begin{pmatrix} \frac{1}{\hat{\beta}_1}\left[V_{00} + 2\hat{\mu}V_{01} + \mu^2 V_{11}\right] & \frac{1}{\hat{\beta}_1^2}\left[\hat{\sigma}V_{01} - V_{20} - \mu V_{12} + \hat{\mu}\hat{\sigma}V_{11}\right] \\ \frac{1}{\hat{\beta}_1}\left[\hat{\sigma}V_{01} - V_{20} - \mu V_{12} + \hat{\mu}\hat{\sigma}V_{11}\right] & \frac{1}{\beta_1^2}\left[V_{22} - 2\hat{\sigma}V_{21} + \sigma^2 V_{11}\right] \end{pmatrix}. \tag{39}$$

With these information ready, it is still not that straightforward because POD curve is actually a curriculum density function of a log-normal distribution, which is not location-scale. However, the log format of the random variable follows the normal distribution. Therefore, we can generate the lower confidence interval on this corresponding normal distribution, $N(\hat{\mu}, \hat{\sigma})$, then take the exponential value of the results, due to the monotone characteristics of log-normal distribution [18, 19].

4. Polynomial Chaos Expansions

4.1 Generalized format

The polynomial chaos expansions (PCE) [20] method has the generalized format of:

$$Y = M(\mathbf{X}) = \sum_{i=1}^{\infty} \alpha_i \mathbf{\Psi}_i(\mathbf{X}), \tag{40}$$

where, $\mathbf{X} \in \mathbb{R}^M$ is a vector with random independent components, described by a probability density function $f_\mathbf{X}$, $Y \equiv M(\mathbf{X})$ is a map of $\mathbf{X}$, i is the index of ith polynomial term, $\mathbf{\Psi}$ is multivariate polynomial basis, and α is corresponding coefficient of basis function. In practice, the total number of sample points needed does not have to be infinite, instead, a truncated form of the PCE is used

$$M(\mathbf{X}) \approx M^{PC}(\mathbf{X}) = \sum_{i=1}^{P} \alpha_i \mathbf{\Psi}_i(\mathbf{X}), \tag{41}$$

where, $M^{PC}(\mathbf{X})$ is the approximate truncated PCE model, P is the total number of sample points needed, which can be calculated as

$$P = \frac{(p+n)!}{p!n!}, \tag{42}$$

where, p is the required order of PCE, and n is the total number of random variables.

4.2 Solving for coefficients

Since a polynomial basis has the characteristics of orthonormality, the equation can be solved by taking the expectation of Eqn. 40 multiplied by Ψ_j,

$$\alpha_i = E[\Psi_i(\mathbf{X}) \cdot M(\mathbf{X})], \tag{43}$$

which is called quadrature method [21]. This method works well for low-dimensional problems, but suffers the "curse of dimensionality".

Another method is to treat the model response as a summation of PCE prediction and corresponding residual

$$M(\mathbf{X}) = M^{PC}(\mathbf{X}) + \varepsilon_P = \sum_{i=1}^{P} \alpha_i \mathbf{\Psi}_i(\mathbf{X}) + \varepsilon_P \equiv \boldsymbol{\alpha}^T \mathbf{\Psi}(\mathbf{X}) + \varepsilon_P, \tag{44}$$

where, ε_p is the residual between $M(\mathbf{X})$ and $M^{PC}(\mathbf{X})$, which is to be minimized in least-squares methods.

Then the initial problem can be converted to a least-squares minimization problem

$$\hat{\boldsymbol{\alpha}} = \arg\min E[\boldsymbol{\alpha}^T \Psi(\mathbf{X}) - M(\mathbf{X})]. \tag{45}$$

The first method, used for solving this problem above and applied in this work, is called ordinary least-squares (OLS) [22], with the coefficients obtained by solving

$$\hat{\boldsymbol{\alpha}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{Y}, \tag{46}$$

where $\mathbf{Y}$ is vector of model response, $A_{ji} = \Psi_i(x^j)$, $j = 1, \ldots, n$, $i = 1, \ldots, P$.

The second method used for solving Eqn. 45, is the least-angle regression sparse (LARS) [23, 24], adding one more regularization term to favor low-rank solution [25]

$$\hat{\boldsymbol{\alpha}} = \arg\min E[\boldsymbol{\alpha}^T \boldsymbol{\psi}(\mathbf{x}) - M(\mathbf{x})] + \lambda \| \boldsymbol{\alpha} \|_1, \tag{47}$$

where λ is a penalty factor, $\|\boldsymbol{\alpha}\|_1$ is L1 norm of the coefficients of PCE.

The LARS [26] algorithm is based on the sparsity-of-effects principle, meaning that only low-order relationship among inputs are important. These two types of methods solving for least-squares minimization problem are very efficient in calculation, and can accept an arbitrary number of sample points.

The mean value of PCE is

$$\mu^{PC} = E[M^{PC}(\mathbf{X})] = \alpha_1, \tag{48}$$

where α_1 is the coefficient of the constant basis term $\Psi_1 = 1$. The standard deviation of PCE is

$$\sigma^{PC} = E[(M^{PC}(\mathbf{X}) - \mu^{PC})^2] = \sum_{i=2}^{P} \alpha_i^2, \tag{49}$$

where it is the summation on coefficients of non-constant basis terms only.

References

[1] Thompson, R., Brasche, L., Forsyth, D., Lindgren, E. and Swindell, P., "Recent Advances in Model-Assisted Probability of Detection", 4th European-American Workshop on Reliability of NDE, Berlin, Germany, June 24-26, 2009.

[2] Aldrin, J., Knopp, J., Lindgren, E., and Jata, K., "Model-Assisted Probability of Detection Evaluation for Eddy Current Inspection of Fastener Sites," Review of Quantitative Nondestructive Evaluation, Vol. 28, 2009, pp. 1784-1791.

[3] Blitz, J., Simpson, G., "Ultrasonic Methods of Non-destructive Testing," London Chapman & Hall, 1996.

[4] Aldrin, J., Medina, E., Lindgren, E., Buynak, C., and Knopp, J., "Case Studies for Model-Assisted Probabilistic Reliability Assessment for Structural Health Monitoring Systems," Review of Progress in Nondestructive Evaluation, Vol. 30, 2011, pp. 1589-1596.

[5] Aldrin, J., Medina, E., Lindgren, E., Buynak, C., Steffes, G., and Derriso, M., "Model-Assisted Probabilistic Reliability Assessment for Structure Health Monitoring Systems," Review of Quantitative Nondestructive Evaluation, Vol. 29, 2010, pp. 1965-1972.

[6] Knopp, J., Blodgett, M., Aldrin, J., "Efficient propagation of uncertainty simulations via the probabilistic collocation method", Studies in Applied Electromagnetic and Mechanics; Electromagnetic Nondestructive Evaluation Proceedings, Vol. 35, 2011.

[7] Miorelli, R., Artusi, X., Abdessalem, A., and Reboud, C., "Database Generation and Exploitation for Efficient and Intensive Simulation Studies," 42nd Annual Review of Progress in Quantitative Nondestructive Evaluation, 2016, pp. 180002-1 – 180002-8.

[8] "Nondestructive Evaluation System Reliability Assessment," MIL-HDBK-1823, Department of Defense Handbook, April 2009.

[9] "Nondestructive Evaluation System Reliability Assessment," MIL-HDBK-1823, Department of Defense Handbook, April 1999.

[10] Kutner, M., Nachtsheim, C., Neter, J., and Li, W., "Applied Linear Statistical Models," McGraw-Hill Irwin, ISBN 0-07-238688-6.

[11] Abdi, H., "The method of least squares," Encyclopedia of Measurement and Statistics, 2007.

[12] Shalizi, C., "The Method of Maximum Likelihood for Simple Linear Regression," Online course, Carnegie Mellon University.

[13] Efron, B., "Bootstrap Methods: Another Look at the Jackknife," The Annals of Statistics, Vol. 7, 1979, pp. 1-26.

[14] "Bootstrap Confidence Intervals," Online course, Duke University.

[15] Dean, N., and Pagano, M., "Evaluating Confidence Interval Methods for Binomial Proportions in Clustered Surveys," Journal of Survey Statistics and Methodology, Vol. 3, No. 4, 2015, pp. 484-503.

[16] Aho, K., and Bowyer, R., "Confidence Intervals for Ratios of Proportions: Implications for Selection Ratios," Methods in Ecology and Evolution, Vol. 6, 2015, pp. 121-132.

[17] Godo, B., and Nagy, A., "Fisher Information and Topological Pressure," Journal of Mathematical Physics, Vol. 58, 2017.

[18] Cheng, R., and Iles, T., "Confidence Bands for Cumulative Distribution Functions of Continuous Random Variables," Technometrics, Vol. 25, No. 1, 1983.

[19] Cheng, R., and Iles, T, "One-Sided Confidence Bands for Cumulative Distribution Functions," Technometrics, Vol. 30, No. 2, 1988.

[20] Wiener, N., "The Homogeneous Chaos," American Journal of Mathematics, Vol. 60, 1938, pp. 897-936.

[21] Zhang, Z., El-Moselhy, T., Elfadel, I, and Daniel, L., "Calculation of Generalized Polynomial-Chaos Basis Functions and Gauss Quadrature Rules in Hierarchical Uncertainty Quantification," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 33, No. 5, May 2014, pp. 728 – 740.
[22] Blatman, G., "Adaptive sparse polynomial chaos expansion for uncertainty propagation and sensitivity analysis". Ph.D. thesis, Blaise Pascal University - Clermont II. 3, 8, 9, 2009.
[23] Blatman, G., and Sudret, B., "An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis," Probabilistic Engineering Mechanics, Vol. 25, No. 2, 2010, pp. 183–197.
[24] Blatman, G., and Sudret, B., "Adaptive sparse polynomial chaos expansion based on Least Angle Regression," Journal of Computational Physics, Vol. 230, 2011, pp. 2345–2367.
[25] Udell, M., Horn, C., Zadeh, R., and Boyd, S., "Generalized Low Rank Models," Foundations and Trends in Machine Learning, Vol. 9, No. 1, 2016, pp. 1-118.
[26] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R., "Least Angle Regression," The Annals of Statistics, Vol. 32, No. 2, 2004, pp. 407-499.