For Proceedings of IMAC XX, Los Angeles, CA
Feb. 4-7, 2002
Approved for Unlimited Release, LA-UR-01-5870

A Metamodel-Based Approach to Model Validation for Nonlinear Finite Element Simulations

Scott W. Doebling[1], Francois M. Hemez[2], John F. Schultze[3], Amanda L. Cundy[4]

Engineering Sciences and Applications Division
Los Alamos National Laboratory
MS P946, Los Alamos, New Mexico, USA

Abstract

Metamodeling, also known as response surface analysis, is the *de facto* standard for mathematical representation of complex phenomena in many fields, especially when first principles physical relationships are not well-defined, e.g. economics, climatology, and government policy. Metamodels provide a computationally efficient, low-dimension relationship for studying the behavior of a physical system. They can be used for understanding the physical system, predicting its response, optimizing its design or the parameters in a physical model, and performing verification and validation. [1] Metamodels can be derived from simulation results or fit directly to observed test data. [2]

In structural dynamics, typical practice is to develop a first-principles-based model such as a finite element model to study the behavior of the system. However, it is common that the features of interest in a structural dynamics simulation are relatively low order (e.g. first few modal frequencies, peak acceleration at certain locations) and sensitive to relatively few model and simulation parameters. In these cases, metamodeling provides a convenient format to facilitate activities of model validation, including parameter screening, sensitivity analysis [3], uncertainty analysis, and test/analysis correlation.

This paper describes the creation of metamodels, and presents some examples of how metamodels can be employed to facilitate model validation for nonlinear structural dynamic response simulation.

Overview of Model Validation

The purpose of this paper is threefold:

a) To discuss some of the philosophical issues surrounding the validation of computational models for structural dynamics response simulation

b) To present a paradigm for model validation that goes beyond the realm of test/analysis correlation

c) To examine a supporting tool for simplified modeling and error metric definition known as "metamodeling"

Model validation is a topic that is beginning to receive significant attention in the structural dynamics literature. Owing mainly to its roots in the field of test/analysis correlation of modal vibration models, model validation has been approached mainly from the standpoint of comparisons of model predictions with real-world measurements to make a statement about the accuracy of the prediction, and hence, the underlying model. In the aerospace industry, there are even formal acceptance criteria for the required accuracy of modal parameter predictions. [4] The implication of such acceptance criteria is that once the predictions have demonstrated sufficient correlation with experimental data, then the model can be trusted to accurately predict the response of the structure.

But is agreement with some set of experimental measurements a sufficient criterion to deem a model "trustworthy?" To explore this issue, consider the following sample of definitions of model validation taking from various sources:

a) "The process of determining the degree to which a computer simulation is an accurate representation of the real world, from the perspective of the intended uses of the model" [5]

b) "Solving the right equations" [6]

c) "The substantiation that a model within its domain of applicability possesses a satisfactory range of accuracy consistent with the intended applications of the model" [7]

All three of these definitions cover the general idea of model validation, and each of them emphasizes a particular aspect. However, the third definition above (from the simulation

1. Technical Staff Member, doebling@lanl.gov
2. Technical Staff Member, hemez@lanl.gov
3. Technical Staff Member, schultze@lanl.gov
4. Graduate Research Assistant, cundy@lanl.gov.

sciences literature) contains some keywords that give significant insight into the true objectives of model validation:

a) **Accuracy:** The agreement of the simulation prediction with some reference data set considered to represent "reality" (the referent is generally derived from an experimental result)

b) **Satisfactory:** Recognition that the accuracy referred to above is not absolute -- the objective is to demonstrate an *adequate* level of accuracy. This recognizes that there is always uncertainty present in the simulation model, the simulation input parameters, and the experimental measurements.

c) **Domain:** The validation of a model can only be defined over a prescribed domain of the simulation input parameters. This domain needs to be specified whenever a statement about the model's validity is made.

d) **Applications:** What the model will be used to analyze. The application drives the predictive requirements of the model.

So a more complete view of model validation incorporates not only the ideas of accuracy for a particular application, but recognizes that this accuracy is satisfactory, not absolute, and that there exists a *region of validity* for the prediction over some domain of the simulation parameters.

As an aside regarding predictive accuracy, consider what is meant by the term "predictive." Can an finite element (FE) model accurately simulate the results of an experiment purely by knowledge of the input parameter values, without prior knowledge of the values of the measured response? The term "predictive accuracy" is used quite frequently in discussions about model validation, but in attempting to quantify predictive accuracy, it is not clear that there is any theoretical construct to define what it is or how it is measured. Consider the following extracts from a discussion in Ref. [8] regarding Ref. [5]:

> "Prediction: Use of a computational model to foretell the state of a physical system under conditions for which the computational model has not been validated" Prediction refers to a simulation result for a *specific case* of interest that is *different from* cases that have been validated. It is important to define whether a particular prediction is "interpolative" or "extrapolative" with respect to the parameter values of the experimental cases used to validate the model. The tendency would be to trust interpolative predictions more than extrapolative predictions.

A distinction should also be drawn between "model validation" as discussed here and the "Verification, Validation, and Accreditation (VV&A)" of computational codes, as discussed in Ref. [6]. VV&A generally refers to the determination and certification that a computational code performs its functions in a computationally proper way and that the results are mathematically correct to some acceptable level of numerical precision. Model validation (as used here) refers to the code being used correctly to model a particular instance of physical phenomena. For example, the FE code could represent Coulomb friction "correctly" (as determined by VV&A), but if the user selects the wrong friction coefficients to model a given assembly, or if Coulomb friction is not an adequate model form for the phenomenon, the resulting model will not be valid.

Another way to look at this issue is the proper operation of a tool (e.g. the FE code) vs. the proper use of that tool for a particular application (e.g. a model of a particular structure under loading). The tool can work exactly the way that it is supposed to, and yet the tool can be used incorrectly for a specific situation. For example, a circular saw may be in perfect working order and have an excellent design, but if one uses it to drive nails one will be very disappointed (and possible severely injured!) It is the correct *usage* of the tool (as opposed to the correct *functioning* of the tool) that is of interest here. The tool has been *verified* against operational standards, but is it a *valid* instance of the proper tool for the job?

This introduction has presented many issues regarding the definitions and issues surrounding model validation. In the remainder of this paper, the intent is to present a broader view of the tasks supporting model validation, as well as explore metamodeling as a useful tool for performing these model validation tasks.

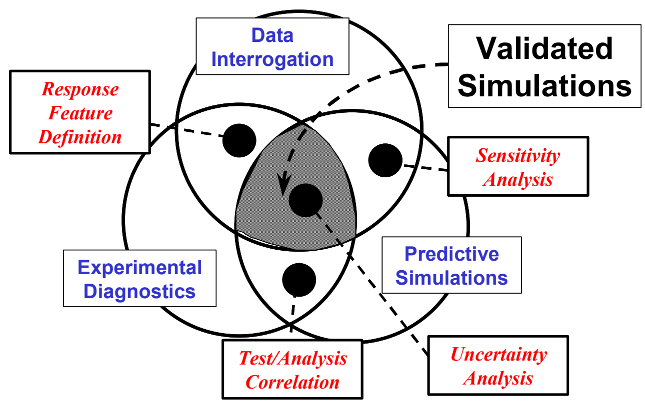

Figure 1: The Paradigm of Validated Simulations Shown with Model Validation Tasks

Model Validation – Beyond Test/Analysis Correlation

If the objective of computational structural mechanics is the development of validated simulations, consider the diagram of the technologies involved in such an endeavor, as shown in Figure 1: Predictive Simulations, Experimental Diagnostics, and Data Interrogation. The intersection of these three technologies is validated simulations, the goal of the computational structural mechanics modeling process.

In order to declare a model to be valid (that is, to predict the quantities of interest to satisfactory accuracy for the application of interest over a specified parameter domain), several tasks must be performed. These tasks are categorized here under the general "process of model validation." They exist in the regions of intersection as shown in Figure 1.

a) Response Feature Definition

b) Sensitivity Analysis

c) Test/Analysis Correlation

d) Uncertainty Analysis.

Response feature definition involves the selection of what particular numerical aspects of the simulation and experimental output are of interest. A feature can be any numerical value that is extracted from the signals. Examples of common features are modal frequency, peak acceleration, peak stress, and temporal moment. Features have certain desirable properties such as low dimensionality and computational efficiency. **Features should always be dictated by the application of interest for the model, as well as what is measurable from a practical standpoint.** For example, modal frequency may not be the best feature of interest when the model will be used to predict peak acceleration levels under a high-frequency shock. Definition of appropriate features is crucial to the model validation process because which features are selected will drive all of the other model validation tasks.

Sensitivity analysis is simply the analysis of the influence of the model input parameters on the response features of interest. The goals of sensitivity analysis include identifying which model input parameters (e.g. material properties, geometry, loading, contact) or combinations of input parameters exhibit the most influence on the response features. This process is known as parameter effects analysis, and it facilitates *parameter screening*, or reducing the number of input parameters by eliminating those that have little effect on the features of interest. Sensitivity analysis is performed on the computational model; generally it is not necessary to have experimental data to perform this task. However, sensitivity analysis can help to prioritize which parameters should be controlled and/or measured during the validation experiments. Thus, it is generally advisable to perform sensitivity analysis prior to planning the validation experiments. Basic sensitivity analysis techniques include the determination of local gradients at points in the parameter space. Advanced sensitivity analysis techniques utilize the power of *design of experiments* (DoE) to perform model evaluations at key combinations of parameters throughout the parameter space. [3] Thus, the influence of input parameters on the response features of interest can be diagnosed with a minimal number of simulation runs. Formulation of a metamodel for the simulation response features can be a useful aid when performing SA.

Test/analysis correlation (TAC) is the "meat" of the model validation process. TAC is the comparison of features from the simulation model prediction with corresponding quantities from experimental measurements in order to assess the "accuracy" of the prediction. Generally a *metric* is defined that assesses the error between the measured and simulated features. This metric can be a simple mathematical norm, such as a Euclidian distance, or it can be a statistical test, such as the Kullback-Leibler maximum entropy. [12] Generally speaking, TAC is the process of assessing the *fidelity* of the simulated feature values with respect to the measured feature values, whereas *validity* refers to the suitability or trustworthiness of the model for a particular application [9].

While much technology exists to perform comparisons between response features in this way, there are many issues not sufficiently addressed in the modern literature. For example, we know that the validity of the model must be defined over a certain region in the parameter space. But at what points in the parameter space should these experiments be conducted? How do we define an appropriate distance metric for a multidimensional input space? How do assessments of model fidelity at discrete locations in the input parameter space yield confidence in the model over the entire space? Metamodeling will provide a construct where we can begin to answer some of these questions.

Uncertainty analysis (UA) is the process of: (a) Propagating uncertainties from the input parameters through the simulation to the output features (i.e. assessing how much variability is expected in the output as a function of input variability); and (b) Assessment of uncertainties in the experimental measurements and attributing these observed uncertainties to the appropriate sources. While mathematical techniques exist to perform both of these tasks, most are sampling-based and require repeated evaluations of the simulation, which can become computationally expensive. Metamodeling can help ease this burden by providing a fast-running surrogate for the full simulation model.

In the remainder of this paper, we will focus on the mathematical technique of metamodeling and explore its usefulness for completing these primary tasks of model validation.

Metamodeling – What is it?

A metamodel is a relatively simple mathematical relationship that provides an approximation of the input/output relationship created by the FE simulation. The term "metamodel" is often used interchangeably with the terms "response surface model," "black-box model," "surrogate model," or "reduced-order model." It can take the form of a polynomial, a sinusoid, a neural network, a set of differential equations, etc.

Think of the FE simulation as a mathematical engine that processes inputs and yields outputs. The response features (or simply "features") are the outputs of the simulation that are of interest. A feature can be a time history, a peak stress, a modal frequency, etc. The input parameters (or simply "parameters", or in DoE language, "factors") are the inputs to the simulation that are of interest. These could be boundary or initial conditions, material properties, friction coefficients, modal damping ratios, etc.

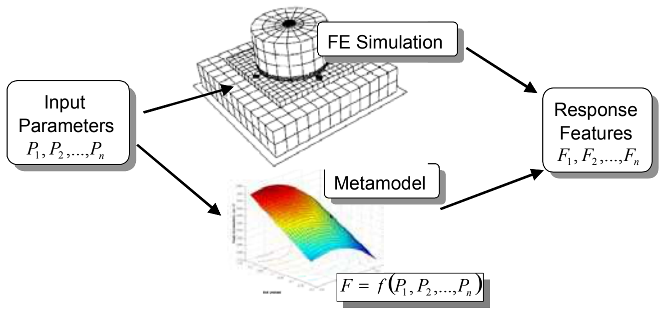

Figure 2: Metamodel as a Surrogate for the FE Simulation

As shown in Figure 2, the metamodel takes the place of the FE simulation in the computational process, becoming a surrogate that is typically of much smaller size and computational cost than the FE simulation.

The metamodel can be visualized as surface plot for 2 input parameters at a time, as shown in Figure 3. The metamodel is a scalar function of multi-dimensional inputs. (Thus for multi-dimensional features, one metamodel is required for each dimension). As shown in Figure 3, the dependent variable (ordinate) is a response feature. The independent variables (abscissas) are simulation input parameters. There can be an unlimited number of input parameters in the metamodel, but the required data and number of simulation runs increases dramatically with the dimension of the input space.

The relationship of the problem entity, the simulation model, and the metamodel is shown in Figure 4 (adapted from Ref. [1]). The three "models" represent increasing levels of abstraction, each to be "validated" with respect to another. While the simulation model might represent the response of the aircraft to a higher level of fidelity, the metamodel might provide a more convenient representation to perform an uncertainty analysis (because of lower computational cost per run). From an engineering point of view, the key is to eliminate from the model those physical phenomena and parameters that have no effect on the response features of interest, while still preserving the correct relationship between the key input parameters and the key response features of interest.

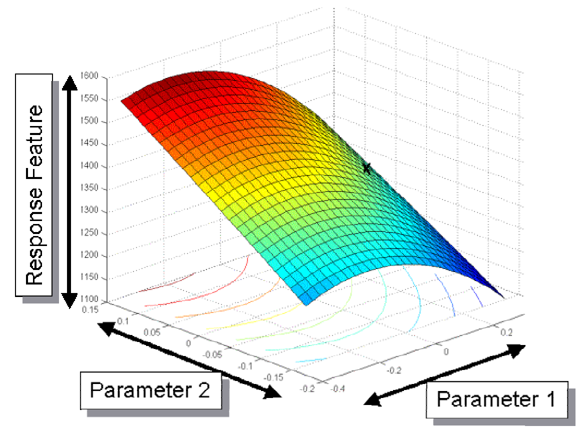

Figure 3: A Response Surface Visualization of a Multi-Dimensional Metamodel

In many engineering applications of interest, both the response feature space and the space of influential parameters are of low dimension, facilitating the use of metamodels for validation tasks.

Metamodeling – How is it done?

To create a metamodel that will serve as a surrogate for the FE simulation model, the basic process is one of calculating predicted values of the features at various sample points in the parameter space by performing a simulation at each of those points. Then regression techniques are used to fit the appropriate metamodel form to the sampled data. The general idea is shown in Figure 5: A number of feature values from simulation runs across the parameter domain are fit with a metamodel. The key is to select the parameters carefully, to minimize the number of dimensions in the parameter space, and then to select the combinations of parameter values where the simulation is performed.

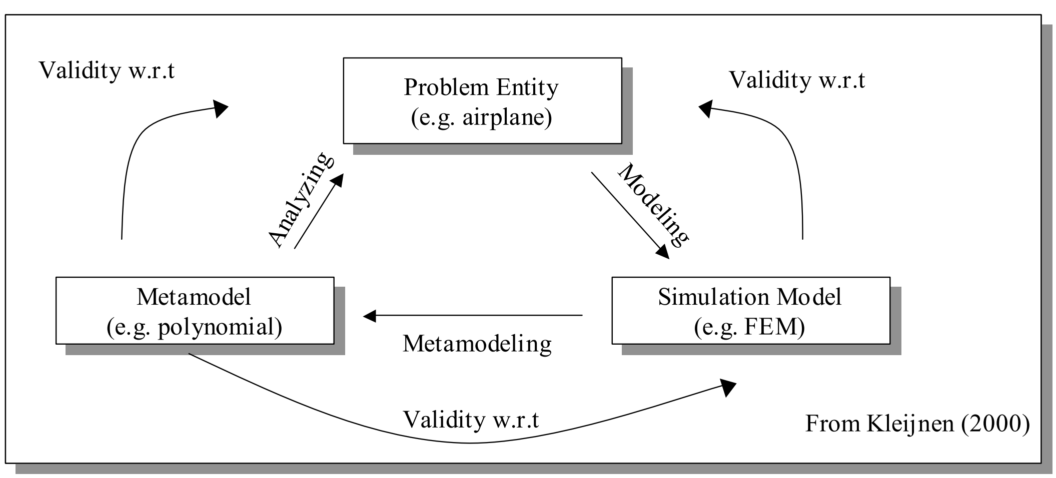

Figure 4: Relationship between Problem Entity, Simulation Model, and Metamodel

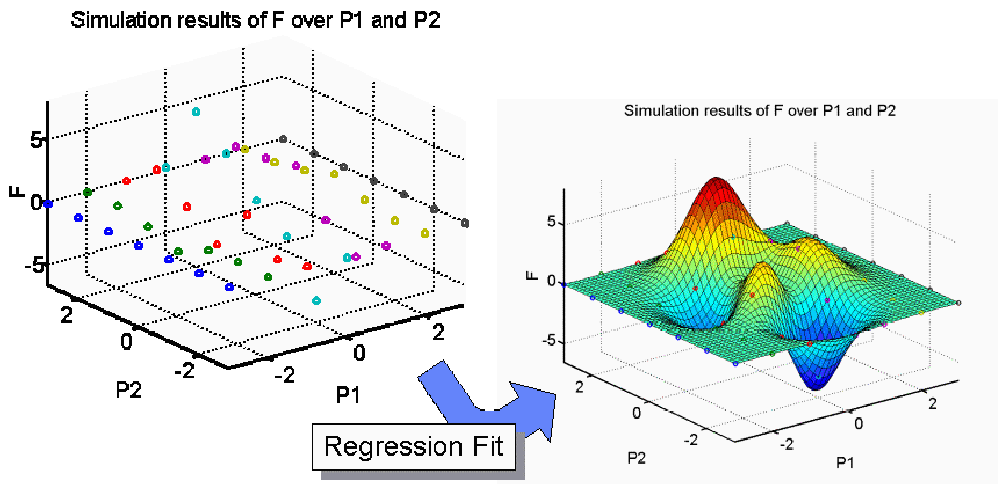

Figure 5: Metamodel is Fit to Feature Values from Simulations at Samples in the Parameter Space

The first step is the selection of the parameters over which the metamodel will be generated. Generally, results of preliminary sensitivity runs or expert judgment/experience are helpful in selecting a candidate set of parameters. Also included in this step is the selection of value ranges for these parameters. The range selected for each parameter should reflect the range that one expects to observe for the domain of the prediction of interest. Generally it is desirable to select the range of the parameters so that there is a slight overshoot of the expected parameter values. This overshoot will ensure that predictions made with the metamodel over the domain of interest will be interpolative rather than extrapolative.

The second step of metamodeling is the selection of samples of the parameter values at which to perform the simulation runs. A number of "vectors" are selected, where each entry in the vector is one of the simulation input parameters. The use of DoE techniques is helpful for this step. [2] DoE techniques define a number of "levels," or discrete values, for each parameter. For example, a three-level design uses the high and low value, plus a mean value, for each parameter. The DoE techniques are distinguished from each other in both how many levels are used per parameter, as well as what combinations of the parameter levels are selected for the simulation runs. It is important to know the intended form of the metamodel at this point, because the form will dictate what type of design is appropriate. For example, if a quadratic metamodel is selected, then at least three levels will be required for each parameter, preferably more.

A popular experiment design for exploration of the parameter space is the central composite design (CCD). [2] The CCD specifies a "ring" of points around the mean value to define the shape of the simulation response in all directions in the parameter space. It can be visualized for a 2-parameter space as shown in Figure 6. The design includes "factorial" points to separate the linear effects of parameters from each other and "axial" points to help define the curvature of the metamodel in each direction. It features an overshoot of the domain of interest as shown in the figure, to ensure that predictions will be interpolative. (For design of physical, rather than computational, experiments, the CCD also features repetitions at the center point to estimate the magnitude of the purely random error.) *The CCD is intended primarily for use with quadratic metamodels with interactions (cross-effects terms).*

The third step in metamodel generation is the evaluation of the FE simulation at each of the sampled parameter vectors from the DoE. This step is where the selection of an appropriate parameter sample design becomes crucial, especially if the FE simulations are computationally expensive. The feature values computed from each simulation run are recorded to form a vector of simulated feature values. (For the sake of simplicity, let us assume for now that there is only one feature of interest.)

The fourth step in generating a metamodel is to perform a parameter effects analysis. The parameter effects analysis is a statistical process whereby the regression is performed, the coefficients of the regression ("effects") are analyzed, and the

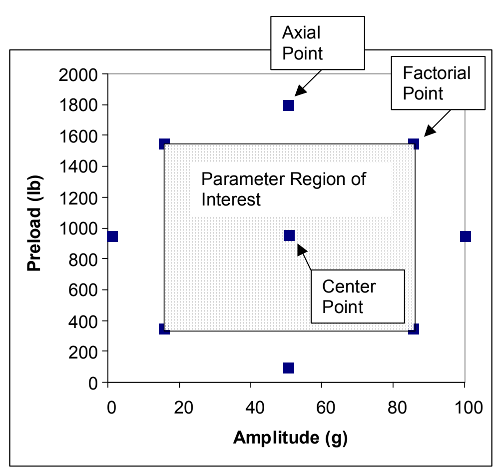

Figure 6: Example of a Central Composite Design for 2 Parameters

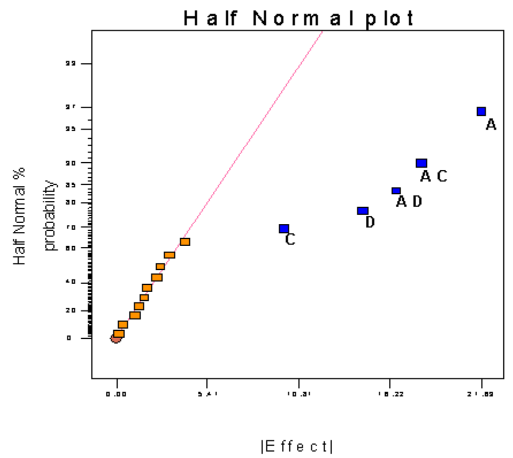

Figure 7: Half-Normal Plot of Factor Interactions for Parameter Effects Analysis

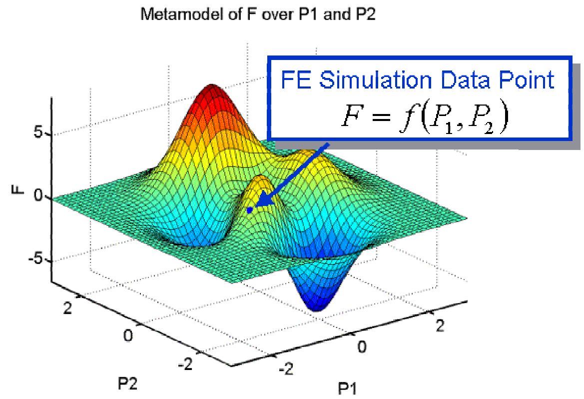

Figure 8: Illustration of Metamodeling for Visualization of FE Model Response over the Parameter Domain

importance of each of the parameters is computed. The parameters that are less important can then be removed from the metamodel to reduce the overall dimension of the parameter space. (This is a good thing, as explained below.) This reduction of the number of parameters via effects analysis is known as *parameter screening*.

There are several ways of examining the parameter effects to perform parameter screening. One common way is via the use of the Half-Normal Probability plot, an example of which is shown in Figure 7. The line represents a fit of a normal distribution. Effects that lie off of the line to the right are non-normal (i.e. the effects are not random, so they should be kept in the analysis). The parameters that have been selected are removed from the fit set, and the normal fit is repeated. When the remaining terms fall mostly on the line, then all of the significant terms have been selected. In this case, the main effects parameters are A, C, and D, along with the interactions AC and AD.

After parameter screening, it is typically desirable to perform another set of simulations using a new set of parameter values computed via DoE. If a significant number of parameters have been removed from the domain, then it is possible to define more levels per parameter at the same total number of computational runs. Thus, there is more information provided for the same computational cost.

The final step in the metamodel generation process is to perform the final regression followed by a regression error analysis. There are standard techniques for regression error analysis, including the examination of residual error for systematic effects, statistical tests of fit significance, calculation of outlier effects, etc. [10] If the regression passes the error analysis tests adequately, then it can be used as a surrogate model for the FE simulation. A commercial software tool that is useful for this type of analysis is Design Expert 6 (DX6) [11]. This software package enables the performance of all the tasks involved with metamodeling, including DoE, parameter effects analysis, parameter screening, and regression error analysis. The half-normal plot shown in Figure 7 was generated using DX6.

Metamodeling – How is it useful?

Presuming that we can formulate a metamodel as a surrogate for our FE simulation of interest, how can the attributes of the metamodel be exploited to facilitate the main tasks of model validation? The first task, response feature definition, is a precursor to metamodel construction and thus is not facilitated by the existence of a metamodel. However, the other three tasks are facilitated by metamodeling as described here:

a) Sensitivity Analysis

Metamodeling facilitates sensitivity analysis in several key ways. First of all, the ability to visualize the metamodel as a response surface over two of the parameters at a time allows a quick, intuitive method for visualizing the relationships between the model input parameters and the response features, as shown in Figure 8. Also, the interactions between multiple parameters can be seen. Areas in the parameter space that contain extrema or regions of high gradient can be seen, and these will be of significant interest in the model validation study. Dimensions in the space that exhibit extremely small gradients (flat surface) indicate a parameter that has little effect on the feature. As mentioned in the previous section, parameter screening is facilitated by the regression step of metamodeling, as well as providing a reduced dimensional basis to enhance the quality of the metamodel.

b) Test/Analysis Correlation

As described in the introduction, the validity of a model should always be described over a domain of parameters. The metamodel that is defined as a surrogate for the FE simulation exists over the domain of parameters, and thus is well suited to compare simulation predictions to experimentally observed feature values.

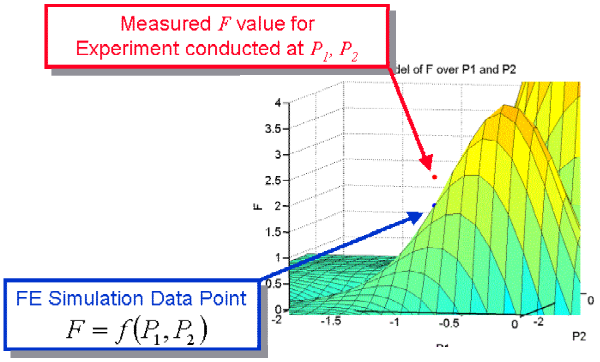

Figure 9: Illustration of Metamodeling for Visual Comparison of Experimentally Measured Feature to Simulated Feature

Consider the example shown in Figure 9. The measured feature value at (P1, P2) is shown as it compares to the surface of the metamodel over those two parameters. An error metric can then be defined and visualized as well. In this case, the error metric could be defined as the distance in the z-direction from the simulation at (P1,P2) to the measured value. Alternatively, the error metric could be defined as the minimum distance from the measured data point to the surface of the metamodel. This metric would account for some inherent uncertainty in the values of (P1,P2) where the experiment was actually conducted.

The metamodel also facilitates the selection of points in the parameter space where further validation experiments should be conducted. For example, areas where the metamodel has extrema or high gradients may be desirable areas of the parameter space to explore further. Also, large areas of the parameter space where there has not been significant experimentation may be of interest. Areas in the parameter space that are already well populated by experimental data, or where the surface is relatively flat, may not be of interest for further validation experiments. Metamodeling enhances the ability to visualize how the "interesting" regions of the parameter space (e.g. due to the shape of the surface) compare to the locations of available experimental data or planned experiments.

c) Uncertainty Analysis

The two main aspects of uncertainty analysis introduced earlier in this paper are certainly enhanced via the use of a metamodel. First of all, the propagation of the uncertainties on the input parameters through the FE simulation model usually employs some sort of sampling-based technique, whereby samples are drawn from the input distributions and run through the simulation. The sample statistics of the resulting features are then computed. Clearly such approaches are computationally costly, especially for expensive FE models. However, with a metamodel as a fast-running surrogate for the FE simulation, such approaches become much more attractive. When the computational cost is reduced from the solution of a million-degree of freedom differential equation to the evaluation of a 20-term polynomial, one can suddenly make lots and lots of simulation evaluations inexpensively.

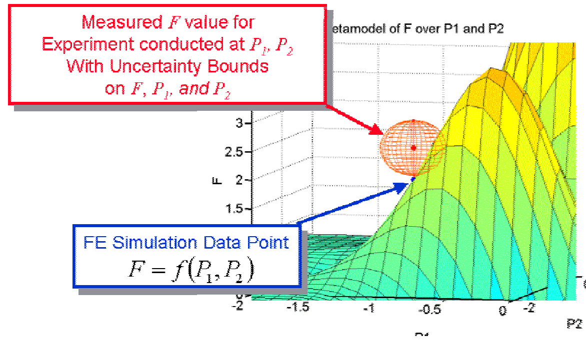

Figure 10: Illustration of Metamodeling for Visualization of Feature and Parameter Uncertainty

Likewise, when attempting to use inverse modeling techniques to attribute feature uncertainty to source input parameters, a smooth, differentiable metamodel can make the optimization process much more palatable. Of course, there are numerous simulation runs of the full FE model required to define the metamodel. However, if a DoE approach is used to estimate the metamodel form with a minimum number of simulation runs, the metamodel-based approach to the optimization can be less expensive than performing the optimization using gradients computed from the FE model at each iteration. Also, the optimization algorithm does not have to be wrapped around the computational code if an accurate metamodel is available.

Visualization of uncertainty and its effects is greatly enhanced by using the metamodel surface. For the example shown in Figure 10, an uncertainty region is added around the experimental data point. The uncertainty distance in the vertical direction indicates the variability observed in the experimental measurement set, while the uncertainty in the horizontal plane indicates the uncertainty in the values of P1 and P2 where the experiments were conducted (i.e. there is uncertainty and/or variability in the actual settings, loadings, etc. of the experiments).

Conclusions and Future Research

The purpose of this paper is to discuss some of the philosophical issues surrounding the validation of computational models for structural dynamics response simulation, to present a paradigm for model validation that goes beyond the realm of test/analysis correlation, and to examine a supporting tool for simplified modeling and error metric definition known as "metamodeling." Following from the success shown in the field of simulation science, it is recognized that model validation is more than the comparison of experimental and simulated data. Metamodeling is introduced as a tool that can facilitate the tasks of model validation, as well as provide interesting visualization options in support of model validation studies.

References

1. Kleijnen, J.P.C. and Sargent, R.G. (2000) "A methodology for fitting and validating metamodels in simulation," *European Journal of Operational Research*, 120 14-29.
2. Myers, R.H. and Montgomery, D.C. (1995) *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, J. Wiley & Sons.
3. Saltelli, A., Chan, K., and Scott, E.M., Ed., (2000) *Sensitivity Analysis*, Wiley Series in Probability and Statistics.
4. Hasselman, T. K., Coppolino, R. N., Zimmerman, D. C. (2000) "Criteria for Modeling Accuracy: A State-of-the-Practice Survey," *Proc. IMAC-XVIII*, San Antonio, Texas.
5. American Institute of Aeronautics and Astronautics (1998) "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," AIAA-G-077-1998.
6. Roach, P.J. (1998) "Verification and Validation in Computational Science and Engineering," Hermosa Publishers.
7. Schlesinger, S., et al. (1979) "Terminology for model credibility," Simulation, 32 (3), 103-104.
8. Oberkampf, W. L. (2001) "Verification and Validation in Computational Mechanics," Short Course, Sixth US National Congress on Computational Mechanics, Aug 1-3, 2001.
9. Gross, D.C., et al. (1999), "Report from the Fidelity Implementation Study Group," *Spring Simulation Interoperability Workshop*, Paper 167.
10. Johnson, R.A. and Wichern, D.W. (1998) *Applied Multivariate Statistical Analysis*, Prentice Hall.
11. Design Expert Version 6.0.3, (2000) Stat-Ease, Inc., Minneapolis, MN, www.statease.com.
12. Kullback, S. and Leibler, R.A. (1951) "On Information and Sufficiency," *Annals of Mathematical Statistics*, Vol. 22, 79-86.