MULTIVARIATE MODELLING OF COMPLEX SIMULATION-BASED SYSTEMS

Wouter Hendrickx, Tom Dhaene

COMS Research Group
University of Antwerp, Belgium
wouter.hendrickx@ua.ac.be

ABSTRACT

Complex computer simulations can take a long time to complete. Often one can vary several input parameters for the simulation. In such case, one wants to know the output of the simulation as a function of the input parameters in order to characterize and explore the whole design space. Traditionally, simulations are performed on a rectangular grid in the input space [1]. This approach quickly becomes infeasable as too many simulations are needed. This paper presents an adaptive methodology to capture the complex input-output behaviour of the simulator in a multivariate surrogate model.

1. INTRODUCTION

Compact and efficient meta-models or surrogate models of simulation data are vital in many engineering applications [2–4]. They are preferable to huge tabulated data sheets and provide the user an opportunity to quickly inspect global output behaviour of the simulation. A methodology for intelligent sample selection and iterative model building is proposed. We first discuss the rational and polynomial models we fit through our samples in section 2. Section 3 covers the iterative sample selection algorithm. The last section illustrates this approach with 2 examples.

2. MULTIVARIATE MODELS

This section describes the multivariate polynomial and rational models used in our adaptive modelling scheme. The properties, construction and evaluation of the models are discussed.

2.1. Model definition

Given multiple sample locations $x_i \in \mathbb{R}^d$ (input) and corresponding function values f_i (output), we try to find a function $H : \mathbb{R}^d \to \mathbb{R}$ which approximates the values f_i at x_i, i.e. $H(x_i) \approx f_i$ in some sense. Polynomial and rational models are used.

Polynomial models are of the form

$$H(x) = \sum_{I \in \mathcal{I}} \alpha_I x^I \tag{1}$$

where $I \subset \mathbb{N}^d$, I finite and x^I is a shorthand notation for $x_1^{I_1} \dots x_d^{I_d}$. Section 2.2 elaborates on the selection of suitable $\mathcal{I}$ sets.

Rational models are of the form

$$H(x) = \frac{\sum_{I \in \mathcal{I}_N} \alpha_I x^I}{\sum_{I \in \mathcal{I}_D} \beta_I x^I} \tag{2}$$

Note that such function H goes to infinity as x approaches the zero set of the denominator.

The multi-indices in $\mathcal{I}$ can be given a fixed order $I_1, \dots, I_N$. We can now construct the matrix

$$M_{\mathcal{I}}(x_1, \dots, x_n) \tag{3}$$

with

$$(M_{\mathcal{I}}(x_1, \dots, x_n))_{ij} = x_i^{I_j} \tag{4}$$

and the diagonal matrix $F = \mathrm{diag}(f_1, \dots, f_n)$.

Finding a polynomial with degrees $\mathcal{I}$ that approximates the values f_i at x_i in a least squares sense can be done by solving the least squares system

$$M \begin{pmatrix} \alpha_{I_1} \\ \vdots \\ \alpha_{I_N} \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \tag{5}$$

In case $n = N$ the system is square and can be inverted using Gaussian elimination, otherwise least squares techniques such as QR-factorization can be used.

Finding a rational function approximating f_i at x_i can be done by solving

$$(M \mid -FM) \left(\alpha_{I_1^n} \dots \alpha_{I_N^n} \beta_{I_1^d} \dots \beta_{I_D^d}\right)^{\mathrm{T}} = 0 \tag{6}$$

To avoid the all-zero solution, one of the unknowns can be fixed at one. Frequently, one choses $\beta_{(0,\ldots,0)} = 1$.

Note there is no guarantee an interpolant exists, even if the number of samples matches the degrees of freedom. Consider for example interpolating the outputs 0,1 and 4 at input points $(0,0)$, $(0,1)$ and $(0,2)$ respectively (corresponding to the function $f(x) = x_2^2$) using $\mathcal{I} = \{(0,0),(1,0),(0,1)\}$.

2.2. Suitable degree sets

Two choices of degree sets $\mathcal{I}$ are quite natural. We define

$$\mathcal{I}_m^c = \{(i_1,\ldots,i_d) \mid \max_{j=1}^{d} i_j \leq m\} \tag{7}$$

and the homogenuous degree set

$$\mathcal{I}_m^h = \{(i_1,\ldots,i_d) \mid \sum_{j=1}^{d} i_j \leq m\} \tag{8}$$

The first set can be seen as all degrees within a (hyper)cube, the second as those inside a simplex with vertices at the origin and at the points with all but one coordinates zero and one coordinate equal to m.

We argue that degree sets of the second type are more natural, as an interpolant or approximant stays of the same form when we apply a linear transformation on our coordinate system. For example, consider $\mathcal{I}_3^h$ in two dimensions and apply the transform $x = ax' + by' + c$ and $y = dx' + ey' + f$ to the monomial $x^2\,y$, then

$$x^2y = (ax' + by' + c)^2(dx' + ey' + f) = a^2dx^3 + \ldots \tag{9}$$

in which all terms are still monomials with degrees in $\mathcal{I}_3^h$. This is not the case for $\mathcal{I}_m^c$.

We now present a weighted degree scheme. Although the homogenuous set is the best choice when all input variables are of equal importance, in cases where we want some coordinate direction to have more degrees, a more sophisticated scheme is required. Suppose one has a weight vector $(w_1,\ldots,w_d)$ attaching a degree of importance to each coordinate axis. Define $W_i = \prod_{j\neq i} w_j$. We now define

$$\mathcal{I}_k^w = \{(i_1,\ldots,i_d) \mid \sum_{j=1}^{d} W_j\, i_j \leq k\} \tag{10}$$

For example, in 2 dimensions with weights $(w_1,w_2) = (1,2)$ (thus $(W_1,W_2) = (2,1)$ we have

$$\mathcal{I}_0^w = \{(0,0)\} \qquad \mathcal{I}_1^w = \{(0,0),(0,1)\} \tag{11}$$

$$\mathcal{I}_2^w = \{(0,0),(0,1),(0,2),(1,0)\} \tag{12}$$

Our toolbox contains a utility called a degree manager. Given certain weights and a number of degrees requested, the degree manager returns a set $\mathcal{I}_k^w$. This set is chosen such that the number of elements in $\mathcal{I}_k^w$ is greater than or equal to the number of degrees requested.

In the modelling toolbox the *degree manager* just solves the diophantine equations (equations with integer coefficients and solutions) $\sum_{j=1}^{d} W_j\, i_j = k$ for increasing k until a suitable number of solutions is found.

2.3. Grid evaluation

An advantage of polynomial and rational models is that they can be easily evaluated on a regular grid when the dimension is reasonably small. Just arrange all coefficients in a d-dimensional tensor/matrix α with α_I the coefficient of x^I in the polynomial. We need to evaluate

$$\sum_{i_1\ldots i_d} \alpha_{(i_1\ldots i_d)} x_{1,k_1}^{i_1} \cdots x_{d,k_d}^{i_d} \tag{13}$$

for each $(k_1,\ldots,k_d)$ where $x_{i,k}$ is the kth grid step of the ith coordinate.

The values of the polynomial can be calculated iteratively by generating $\alpha^{(1)}$ with

$$\alpha^{(1)}_{(i_1\ldots i_{d-1}k_d)} = \sum_{i_d} \alpha_{(i_1\ldots i_d)} x_{d,k_d}^{i_d} \tag{14}$$

and continuing this process until all i's have been replaced by k's.

3. SAMPLING AND MODELLING

We build multivariate models in an iterative way. Each modelling iteration comprises several steps, which will be described in this section.

3.1. The initial sample set

In order to begin the modelling cycle, an initial sample set is needed. The sample set should fill the input space reasonably well. Two types of initial sample sets are used in the toolbox: a gridded sample set and a latin hypercube sample set.

The gridded sample set selects points on a n^d grid (e.g. $n = 2$ or $n = 3$).

A latin hypercube sample set of size n can be seen as putting n rooks on a d-dimensional chess board with sides of size n, such that no rook attacks another. Formally, make a table with d column, and fill each column with a different random permutation of $(1,\ldots,n)$.

Each row represents a sample point. Of course the range $(1, \ldots, n)$ is scaled to fit the sampling domain.

3.2. Building different models

Based on a sample set, we build several models. By pairwise comparison of these models some information on their accuracy can be extracted. To create a variety of models, we use different settings for three model parameters: *variable weighting*, *degrees of freedom* and a *polynomial/rational flag*.

Different *weighting* of the variables (as described in section 2.2) makes one variable more important than another. The *degrees of freedom* are given as a percentage of the number samples, 100% means an interpolant, any smaller value indicates an approximant. The *polynomial/rational flag* indicates which of the input variables appear in the denominator.

3.3. Grid evaluation

To make a decent comparison between the models, grid evaluation is a suitable choice (when $d < 5$). Using the procedure described in section 2.3 this can be done efficiently.

Note that once a model is evaluated on a grid, it can easily be cross-checked with the reference dataset at all grid locations to assess how well the model fits the data. Of course in reality the full dataset will not be at hand, so the cross-check results can be used for validation purposes, but not to select new sample locations.

3.4. Model comparison

All created models are compared pairwise, and an error matrix E is generated, with E_{kl} the root mean square error

$$\left(\sum_{i=1}^{n_g} |H_k(x_i) - H_l(x_i)|^2 \right)^{1/2} \qquad (15)$$

between model k and model l. Of course E has zeros on the diagonal, E is symmetric, and if E_{kl} is small this indicates the models k and l are quite similar.

All models receive a quotation according to following formula:

$$\frac{1}{Q_l} = \sum_{k \neq l} \frac{1}{E_{kl}} \qquad (16)$$

which identifies models which are most similar to other models. All models are ordered with respect to their quotations Q_l. Experiments show that models with a small Q_l are better than those with large Q_l.

3.5. New sample selection

The *best* models are used to select new samples in the input or design space. Each pair of models contributes a certain number of new sample locations. These are selected based on maximum absolute error between the two models. All points where a pair of models differ the most are grouped in a new sample set N.

The set of samples used to build the models is removed from N to avoid double samples. Then some measures are taken to avoid clustering of points in N. Points are removed so that the minimal distance between 2 sample points in N is greater than some threshold. Finally, if the resulting N is too small, some random points are added to speed up the modelling process.

3.6. The iteration

All these steps are performed iteratively, as shown in figure 1. An initial sample set and set of model parameters is selected. Multiple models are built approximating the samples. These models are evaluated at grid points and cross-checked. An error matrix is constructed and the models are ordered according to quotations. The accuracy of the models are estimated and new sample locations are chosen. If the accuracy is insufficient, new samples are added and we repeat the process of model building, evaluating and cross-checking.

4. EXAMPLES

4.1. Academical 2D example

The first example is an analytical function taken from [5]. Its definition is

$$H(x, y) : \Omega \to \mathbb{R} : (x, y) \mapsto \exp^{\frac{x}{3}+2} / \Gamma(y) \qquad (17)$$

with $\Omega = [-3, 3] \times [-3, 3]$. H varies heavily in y dimension due to the Gamma function and quite smoothly in the x direction. Therefore this example can be used to show that suitable variable weighting (as described in 2.2) can be benificial for the modelling process. For testing purposes we sampled this function on a 50×50 grid (2500 datapoints). Figure 2 gives a plot of this function.

Each iteration the best model was evaluated on a 50^2 grid and the results are shown figure 3. The error

Figure 1: Modelling flow-chart

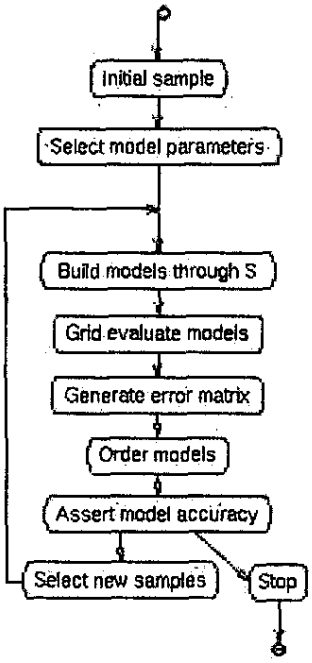


Figure 2: Academical 2D example

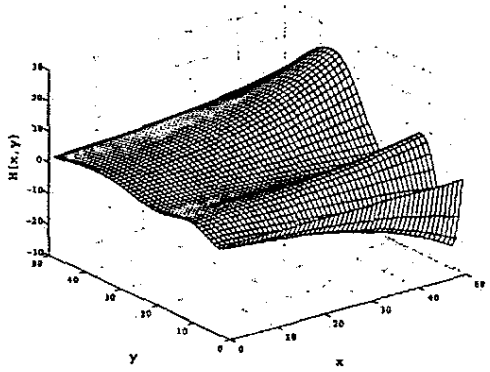
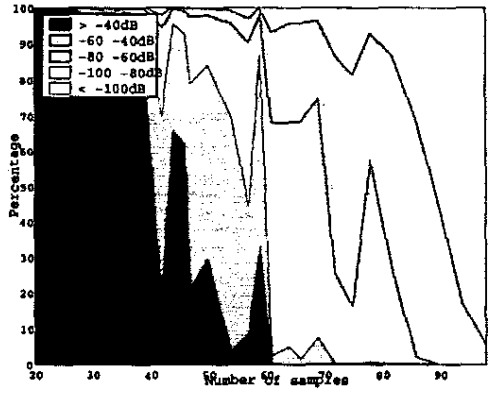


Figure 3: Error percentages plot for the academical example, see 4.1 for more information



is defined as the absolute error between the analytic function H and the surrogate model. This figure illustrates how our algorithm converges. The horizontal axis shows the number of sample points used to build the model, while the vertical axis contains percentages of gridpoints having certain accuracy. The darkest region shows the percentage of gridpoints with error greater than -40dB. The other regions show percentages of gridpoints with error in $[-60\text{dB}, -40\text{dB}]$, $[-80\text{dB}, -60\text{dB}]$, $[-100\text{dB}, -80\text{dB}]$ and smaller than -100dB respectively.

4.2. The Step Discontinuity

We use a full-wave electromagnetic MATLAB toolbox for simulating the S-parameters of a step discontinuity in a rectangular waveguide. The three parameters represent frequency (7 to 13GHz), gap height (2 to 8mm) and step length (0.5 to 5mm). We sampled the code on a 50^3 grid (125,000 points) for verification purposes (this took us about 1 hour of computation). Figure 4 shows plots of the reflection coefficients $|S_{11}|$ for frequency fixed at 7GHz, 10GHz and 13GHz.

Figure 5 gives an error percentage plot similar to 3. Only the scale of the horizontal axis differs. We see that if 100 samples are chosen, the accuracy of the surrogate model is better than -60dB. Typically, this is more than sufficient for high-accuracy electronical simulation purposes.

5. CONCLUSION

We presented an adaptive modelling and sampling scheme for automatic design space exploration and modelling of complex systems. An application of this scheme to an academical and real-life example demonstrates our modelling and sample scheme works.

Figure 4: Step discontinuity, plots of $|S_{11}|$ in function of height and length for frequency fixed at 7, 10 and 13GHz

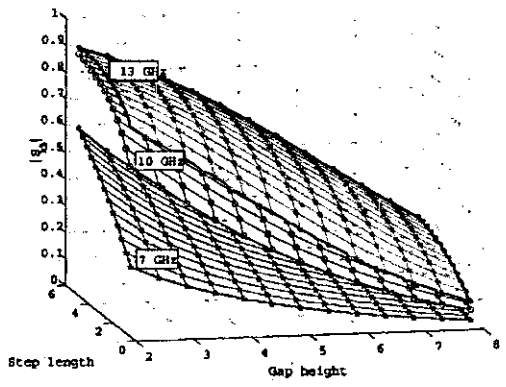
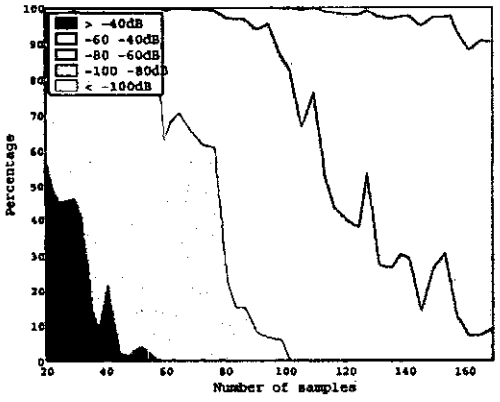


Figure 5: Error percentages plot for the step discontinuity



6. ACKNOWLEDGEMENT

We thank Robert Lehmensiek for the MATLAB code used to simulate the step discontinuity in section 4.2.

7. REFERENCES

[1] "Nimrod: Tools for Distributed Parametric Modelling," http://www.csse.monash.edu.au/~davida/nimrod/.

[2] Tom Dhaene, "Automated fitting and rational modeling algorithm for em-based s-parameter data," *Lecture Notes in Computer Science*, vol. 2367, pp. 99–105, 2002, 6th International Conference on Applied Parallel Computing (PARA 2002).

[3] Jan De Geest, Tom Dhaene, Niels Faché, and Daniël De Zutter, "Adaptive cad-model building algorithm for general planar microwave structures," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 9, pp. 1801–1809, 1999.

[4] Jan De Geest, *Adaptieve techniek voor de modellering van passieve planaire microgolfcircuits*, Ph.D. thesis, University of Ghent, 2000.

[5] Robert Lehmensiek, *Efficient adaptive sampling applied to multivariate, multiple output rational interpolation models, with applications in electromagnetics-based device modelling*, Ph.D. thesis, University of Stellenbosch, South Africa, 2001.