Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

SEQUENTIAL DESIGN AND RATIONAL METAMODELLING

Wouter Hendrickx
Tom Dhaene

Department of Mathematics and Computer Science
University of Antwerp
Antwerp, BELGIUM

ABSTRACT

Metamodelling techniques are used in many engineering applications for efficient exploration of the design space of complex deterministic simulation systems, and for optimisation purposes and sensitivity analysis. This paper presents a new sequential design and rational metamodelling technique which combines adaptive modelling and sampling algorithms.

1 INTRODUCTION

Metamodelling techniques for simulation and optimisation purposes are becoming increasingly popular. Quite often the evaluation of a complex simulator on a dense grid in the design space (for example using the Nimrod or AppLeS software packages) is not an option for finding optima or gaining global knowledge of a system's behaviour. Adaptive or sequential metamodelling frameworks, as presented in Dhaene et al. (1995), De Geest (2000), Van Beers and Kleijnen (2004) and Hendrickx and Dhaene (2005), drastically reduce the number of evaluation points needed to build a high quality metamodel.

A *simulator model* or *system* can be seen as a black-box that maps inputs onto outputs. Most of the time, inputs and outputs can be represented as vectors of numbers. Then, the system can be seen as a function

$$S : \mathbb{R}^d \to \mathbb{R}^s : x \mapsto S(x),$$

where x are the input variables, and $S(x)$ are the outputs.

An example of a simulator model can be the numerical implementation of Maxwell's equations to calculate the behaviour of an antenna. Input parameters to the simulator can be length, angle, etc., of the antenna. Output parameters can describe the reflection and radiation behaviour (scattering parameters).

A *metamodel* or *surrogate model* (see for example Barton 1998) is a *model of a model* in which the latter model itself is already an approximation of reality (a simulation using discretisations and approximations).

In Chung and Alonso (2000) and also in Søndergaard (2003) metamodels are used for locating an optimum in the input space. De Geest et al. (1999), De Geest (2000) and Hendrickx and Dhaene (2005) use metamodels for gaining a global view of the system's behaviour over the input space or design space . This can be advantageous if evaluating the system itself takes a very long time (which frequently happens when simulating complex systems). Sometimes these metamodels can be used as building blocks for larger scale simulations.

In the last decade, several researchers proposed iterative sampling schemes for metamodelling. Van Beers and Kleijnen (2004) build Kriging metamodels and use cross-validation and jackknifing techniques to iteratively select new input locations at which to evaluate the system. Kriging techniques are further discussed in Martin and Simpson (2005) and Santner et al. (2003). Dhaene et al. (1995) describe a rational modelling technique for one input variable, called AFS (Adaptive Frequency Sampling). This approach was generalised to the case of more inputs by De Geest (2000) and De Geest et al. (1999) and is called MAPS (Multidimensional Adaptive Parameter Sampling). Polynomial and rational models are used to model the system. The AFS and MAPS algorithms are tailored towards electromagnetic simulators and are commercially available in ADS Momentum and ADS Advanced Model Composer software respectively.

Different techniques are used to build metamodels. Each metamodel is based on a limited set of input points (the *sample set* S) at which the system has been evaluated. Based on these locations $(x_1, \ldots, x_n)$ and their corresponding function values $(f_1, \ldots, f_n)$, an interpolant or approximant H can be built such that $H(x_i) \approx f_i = S(x_i)$.

Several techniques are at hand to create interpolants. First of all there are polynomial and rational models, used by De Geest et al. (1999), De Geest (2000), Lehmensiek and Meyer (2001), Dhaene (2002) and Hendrickx and Dhaene

(2005). They are used in this work and therefore we defer their description to section 2.

Kriging metamodels are inspired by the theory of stochastic processes. Kriging is named after its first practitioner Krige (1951) and a wide range of work on this technique has been published since then. Reference publications on the subject are Sacks et al. (1989), Cressie (1993) and Santner et al. (2003).

Radial basis functions only recently became popular for scattered data interpolation. Their form is very straightforward and easy to implement. When the number of sample points gets large, several techniques are available to speed up the modelling process. A lot of theory is available in the literature. Only recently Buhmann (2003) and Wendland (2005) have gathered the different aspects into two books on the subject.

At last, we mention the space mapping methodology, conceived by Bandler et al. (1994) and further discussed in several PhD theses (e.g., Søndergaard 2003). It tries to map a low-accuracy metamodel onto the system by a linear space transformation. This technique has mainly been used for optimisation and not for global approximation.

This paper will discuss a polynomial and rational based adaptive sampling methodology for sequential design with rational metamodels and apply it to several academical and microwave simulation examples.

Our approach is also relevant to random simulations, as the models constructed by the toolbox approximate rather than interpolate the data. In fact each model is a least-squares fit of the rational model to the data; therefore, our toolbox is also suitable for modelling random simulation data. We conducted our tests on deterministic data in order to observe the systematic reduction of the modelling error. Monitoring the modelling error would have been difficult with random simulation data. In the toolbox, we provide an option to add noise to the test sets. This can be useful to inspect the behaviour of the toolbox in a random simulation setting.

2 POLYNOMIAL AND RATIONAL METAMODELS

We construct polynomial and rational metamodels. This section will discuss several aspects of such models.

2.1 Metamodel Definition

Given multiple sample locations $x_i \in \mathbb{R}^d$ (input) and corresponding function values f_i (output), we try to construct a function $H : \mathbb{R}^d \to \mathbb{R}$ which approximates the values f_i at x_i, i.e., $H(x_i) \approx f_i$. 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 is finite and x^I is a shorthand notation for $x_1^{I_1} \ldots x_d^{I_d}$. Section 2.2 elaborates on the selection of suitable $\mathcal{I}$ sets.

Rational models are of the form

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

where $\mathcal{I}_N$ and $\mathcal{I}_D$ are the degree sets for numerator and denominator respectively. Note that such a 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 $I_1, \ldots, I_N$. We can now construct the matrix

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

with

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

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

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-factorisation can be used.

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

$$\left(M_{\mathcal{I}_N} \mid -F M_{\mathcal{I}_D} \right) \left(\alpha_{I_1^n} \ldots \alpha_{I_N^n} \beta_{I_1^d} \ldots \beta_{I_D^d} \right)^{\mathrm{T}} = 0 \tag{6}$$

(with $\{I_1^n, \ldots, I_N^n\} = \mathcal{I}_N$ and $\{I_1^d, \ldots, I_D^d\} = \mathcal{I}_D$). To avoid the all-zero solution, one of the unknowns can be fixed at one. Frequently, one chooses $\beta_{(0,\ldots,0)} = 1$.

Note that 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 polynomial interpolation with degree set $\mathcal{I} = \{(0, 0), (1, 0), (0, 1)\}$.

Furthermore, calculating the coefficients of the interpolant or approximant gets more and more time-consuming as the number of unknowns increases. Gohberg et al. (1995) present an advanced algorithm for interpolation which has the potential of reducing the computational costs. Unfortunately, we have been unable to create an efficient MATLAB implementation to use with our code.

2.2 Suitable Degree Sets

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

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

and the homogeneous degree set

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

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 transformation $x = ax' + by' + c$ and $y = dx' + ey' + f$ to the monomial $x^2 y$, then

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

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

The homogeneous set is the best choice when all input variables are of equal importance. In cases where we want some coordinate directions to have more degrees than others, 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 $W_i = \prod_{j \ne i} w_j$. We now define

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

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

$$\mathcal{I}_0^w = \{(0,0)\}, \qquad \mathcal{I}_1^w = \{(0,0),(0,1)\}, \\ \text{and } \mathcal{I}_2^w = \{(0,0),(0,1),(0,2),(1,0)\}. \tag{11}$$

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. Finding the order of the degrees is not the main part of the calculations, so no alternatives for generating the degree sets have been checked.

2.3 Grid Evaluation

In the tests described in section 4, metamodels are often evaluated on a rectangular grid. Grid evaluation of rational metamodels can be done in a reasonably efficient way using some basic algebra. Therefore, we claim grid evaluation to be a feasible method for model comparison if the dimensionality of the input space stays low.

3 SAMPLING AND MODELLING

The algorithm described in this paper uses a sequential metamodelling and sampling technique. Each iteration selects new samples in the design space and builds new metamodels, until enough accuracy has been achieved.

3.1 The Initial Sample Set

To create an initial sample set, the complexity of each variable is explored. To explore the ith input space dimension, we fix all but the ith coordinate to the middle of the domain and try to fit a one-dimensional model to the system along the free dimension. The number of samples needed to create such a one-dimensional model is a measure for that dimension's complexity.

The initial sample set is then just a full-factorial design, where the number of steps along each dimension differs in accordance with the complexities assigned earlier.

3.2 Polynomial and Rational Metamodel Parameters

Based on a sample set, we build several models. Three model parameters can be tuned to create a variety of models: *variable weighting*, *degrees of freedom* and *polynomial/rational flags*.

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 of samples, 100% means an interpolant, any smaller value indicates an approximant. For example when the degrees of freedom parameter is set to 60% and the sample set contains 100 locations, the degree sets $\mathcal{I}_D$ and $\mathcal{I}_N$ will be chosen such that $\#\mathcal{I}_D + \#\mathcal{I}_N = 60$ and the sets are conform to the variable weighting. The *polynomial/rational flags* indicates which of the input variables appear in the denominator. For example, a trivariate metamodel with flags (true, false, true) will contain monomials in all variables in the numerator, and monomials in variables 1 and 3 in the denominator.

Initially, a set number of metamodel parameters is fixed. Degrees of freedom are assigned randomly (within certain bounds). Weighting is assigned randomly, but the complexities derived in 3.1 are taken into account. Polynomial/rational flags are just set at random. Each iteration the correctness of these metamodel parameters is assessed and bad parameter sets are replaced, as described in section 3.6.

3.3 Model Building for One Output

For each set of metamodel, parameters selected in the previous section, a metamodel is built through the samples. At this stage, only the first of the output is modelled in order to save time, since each output would require its own separate metamodel. The first output is used for metamodel ranking and new sample selection (this will be described in the next sections). All other outputs are modelled after the best metamodel parameters have been selected.

3.4 Grid Evaluation and Model Quality Assessment

To make a decent comparison between the models, evaluation on a large full-factorial design is a suitable choice. Once a model is evaluated on a grid, it can easily be cross-checked with the reference data set at all grid locations. When a test-data set of the system is available, this error information can be used to verify the accuracy of the models. Of course in reality the full data set will not be at hand, so the cross-check results can be used for validation purposes, but not to select new sample locations.

To assert each metamodel's accuracy, all metamodels are compared pairwise. An error matrix E is generated, with E_{kl} the root mean square error

$$\left(\frac{1}{n_g} \sum_{i=1}^{n_g} |H_k(x_i) - H_l(x_i)|^2 \right)^{1/2} \tag{12}$$

between metamodels k and l (n_g is the number of grid points in the full-factorial design and the x_is are the grid points in some order). Of course E has zeros on the diagonal, E is symmetric, and if E_{kl} is small this indicates the metamodels k and l are quite similar.

All models receive a score according to following formula:

$$\frac{1}{Q_l} = \sum_{k \neq l} \frac{1}{E_{kl}} \tag{13}$$

which identifies metamodels which are most similar to other metamodels. All metamodels are ordered with respect to their scores Q_l. Experiments show that models with a small Q_l are better than those with large Q_l. An exception to this rule occurs when two metamodels have very similar metamodel parameters. In such cases, the Qs for both of these metamodels will be extremely small. Therefore, we ignore metamodels with very small Q values.

3.5 Expanding the Sample Set

Using the three best metamodels (by best we mean metamodels with lowest score), the algorithm selects possible new sample locations. Locations where two of these metamodels differ the most are nominated as possible new sample points. We gather all qualified sample locations in the new sample set N.

If the error is large at a point x, other points nearby x are likely to have large error values too. Just taking the set of points where the error is largest will probably just select all points around such point x. Therefore we take some measures to avoid clustering of points in N. Points are removed so that the minimal distance between two sample points in N is greater than some threshold. Furthermore, the number of sample points added in each iteration is restricted by two bounds. If, on the one hand, the number of points in N is smaller than the lower bound, some random points are added. If, on the other hand, N is too large (according to an upper bound) some random points of N are removed until N is small enough. N is then added to the sample set.

3.6 Selecting New Metamodel Parameters

We want to improve our model's quality by adjusting the variable weighting, polynomial/rational flags and degrees of freedom parameters. The parameters corresponding to the worst metamodels (according to the scores) are removed and new parameters are added. In order to select new parameters in an intelligent way, we choose new parameters close to those of the best metamodels. In practise we just average the three best metamodel's parameters and then add a small perturbation to avoid duplicate parameters. The worst metamodel parameters used in the current iteration are replaced by these new metamodel parameters and will be used in the next iteration.

3.7 The Stopping Criterion

A stopping criterion is needed to decide when to stop iterating. We want the iteration to stop, once we are convinced the best metamodel has a certain accuracy or when it is clear the desired accuracy cannot be reached.

At this point, only the first output parameter has been modelled. In order to ensure all outputs achieve a certain accuracy, we now model all outputs, but only using the metamodel parameters of the best model. These metamodels are then cross-checked with those of the previous iteration, and their maximum absolute deviation is used as an accuracy measure. When three consecutive iterations produce metamodels which resemble each other up to a desired accuracy (e.g., 2 significant digits), the metamodelling scheme stops and the current metamodels are returned.

If after a large number of iterations no progress is made towards the desired accuracy, the algorithm also exits. In that case it failed to converge, so the best model found until then is returned and an error message is shown to the user.

3.8 The Iteration

Figure 1 depicts how all the steps discussed in this section relate. First, an initial sample set and a set of metamodel parameters are selected. Then metamodels are built for the first output. These metamodels are evaluated on a grid and compared in pairs. Based on pairwise differences, scores are assigned to the different models. The best metamodel parameters are then used to model all other outputs. These metamodels are compared to those of the previous iteration. If the stopping criterion is satisfied, the algorithm exits. Otherwise, the worst metamodel parameters are replaced by new parameters, and new samples are added to the sample set at locations where different metamodels differ the most. Then the loop of model building, assigning scores, …starts all over again.

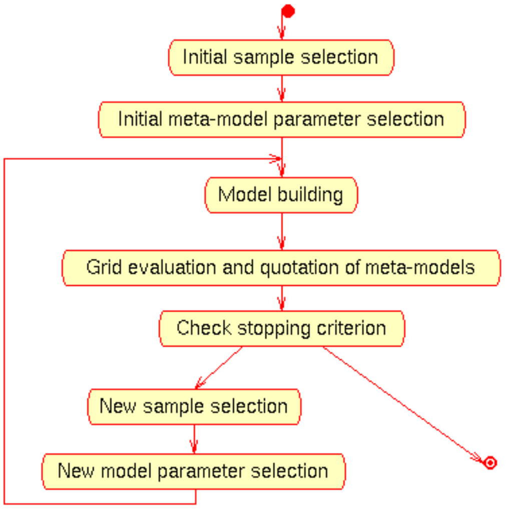
Initial sample selection
Initial meta-model parameter selection
Model building
Grid evaluation and quotation of meta-models
Check stopping criterion
New sample selection
New model parameter selection

Figure 1: Modelling Flowchart

4 TESTS

This section describes some tests we conducted using our toolbox. First, one academic 2-dimensional example is described (the system is given explicitly by a function definition). Afterwards, both an academic example and two real-life examples with 3 inputs are discussed, and several aspects of the algorithm are illustrated.

4.1 Academic 2D Example

First, an analytical function taken from Lehmensiek (2001) is discussed. Its definition is

$$H(x,y) : \Omega \to \mathbb{R} : (x,y) \mapsto C \exp^{\frac{x}{3}+2} / \Gamma(y) \qquad (14)$$

with $\Omega = [-3,3] \times [-3,3]$. H varies heavily in the y dimension due to the gamma function and quite smoothly in the x dimension. A plot of this function is given in Figure 2. We chose the parameter C such that the maximal absolute value of H on the domain is exactly one.

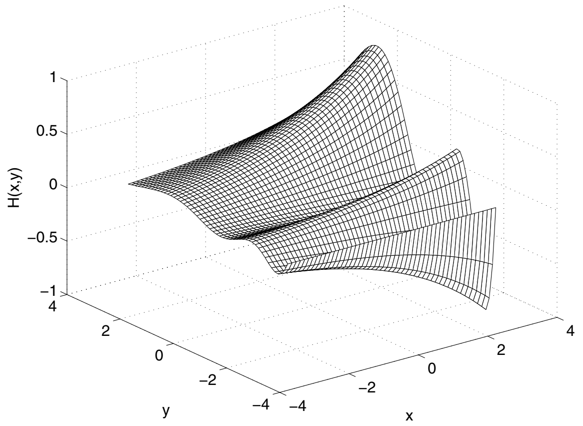

Figure 2: The Academic 2D Example

For this simple example, we wanted to achieve an absolute accuracy of 10^{-3} in 50×50 grid points. Figure 3 depicts error percentages. The x-axis contains the number of samples, while the height of each grey-shaded region depicts the percentage of grid points having an accuracy within that region. For example, we see the darkest region disappears after 53 samples where taken. This indicates that using only 53 samples in the design space, we were able to build a metamodel with an error smaller than 0.1 in all grid points. After 63 samples, the error was reduced below 0.01. When 90 samples where selected, the algorithm decided the desired 10^{-4} accuracy was achieved over the whole design space. Figure 3 shows that this conclusion was indeed correct.

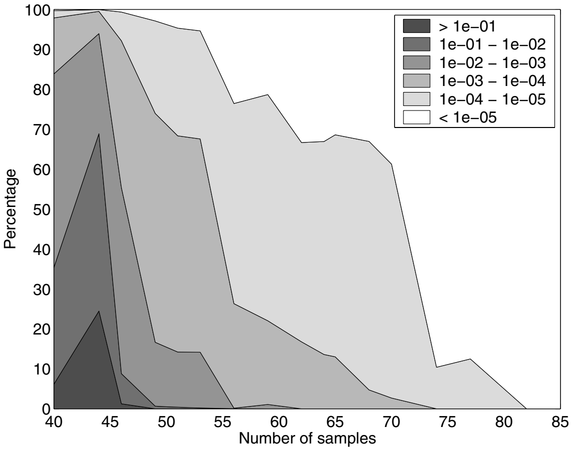

Figure 3: Modelling Error for the Academic 2D Example

4.2 Academic 3D Example

In this section, we discuss an example which is an extension of the one presented in 4.1. We consider the function

$$H : \tilde{\Omega} \to \mathbb{R} : (x, y, z) \mapsto C \frac{\exp(2z+1)}{\Gamma(6y-3)\ \Gamma(5x)}$$

with $\tilde{\Omega} = [0, 1]^3$, the unit cube. C was selected such that H had a maximum near 1 on the domain (in fact all function values then fall within [-1.1,1.1]). That way we can safely use the absolute error as a measure for accuracy. Figure 4 shows a plot of the function for input parameter z fixed at value 1.

Figure 5 shows an error plot similar to the one of last section. Note that the final size of the sample set is much larger than the one for the 2D example. Again, we set the desired accuracy to 10^{-4}. As the plot indicates, 10^{-3} was reached after 250 samples and 10^{-4} was achieved after 360 samples. The algorithm terminated shortly after that accuracy was reached.

It is interesting to see which weights the algorithm has used for the different models. Figure 6 shows how the parameter weighting evolves as more and more sample points are selected. In the beginning (small sample set) variable 1 (x) gets the highest weighting, while variable 2 (y) gets a lower weight. At about 150 samples the metamodelling scheme decided y was the variable that varied most, and consequently gave it a bigger weight. z provides the lowest variation, therefore it gets the lowest weight possible. Note that during the process all weights were bounded between 2 and 10.

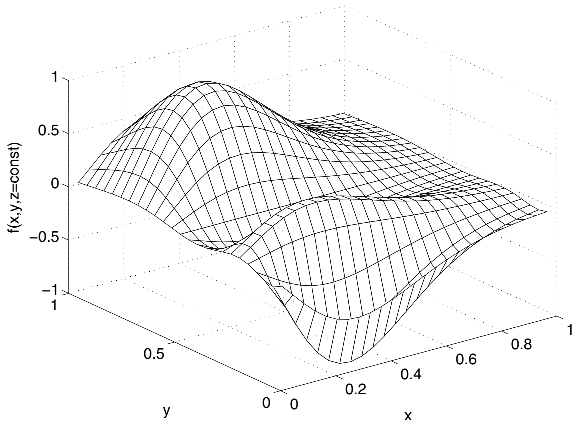

Figure 4: A Plot of the Academic 3D Example, with the Third Variable Fixed at 1

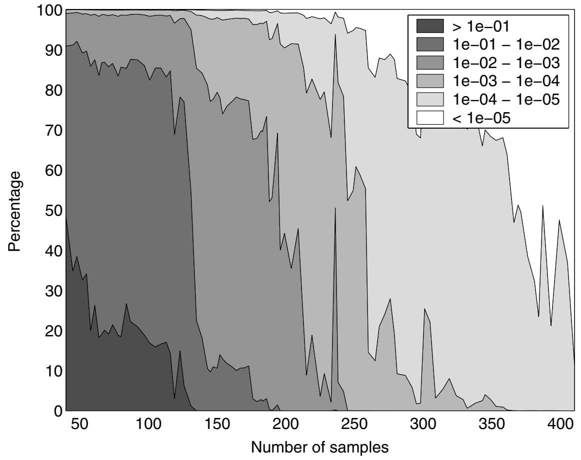

Figure 5: Modelling Error for the Academic 3D Example

The modelling process took some time to decide which inputs to include in the denominator of the rational models. The final models include both x and z in numerator and denominator, while y only occurs in the numerator.

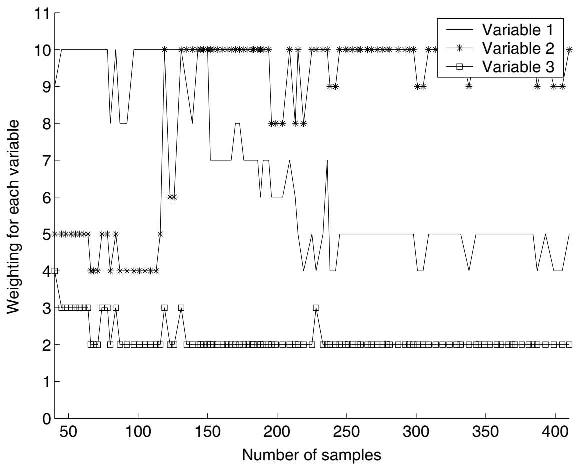

Figure 6: Weights Assigned to Each Variable in the Academic 3D Example

4.3 Real-Life Example: The Step Discontinuity

We use a full-wave electromagnetic MATLAB toolbox (Lehmensiek 2001) for simulating the scattering parameters of a step discontinuity in a rectangular waveguide. The three input parameters represent frequency (7 to 13GHz), gap height (2 to 8mm) and step length (0.5 to 5mm). The two outputs parameters are S_{11} and S_{12}, which are the complex valued scattering parameters of the structure. We sampled the code on a 50^3 grid (125,000 points) for verification purposes (this took us about 1 hour of computation on a P4 2.4GHz desktop computer). Figure 7 shows plots of the magnitude of the complex S_{11} for frequencies fixed at 7 GHz, 10 GHz and 13 GHz.

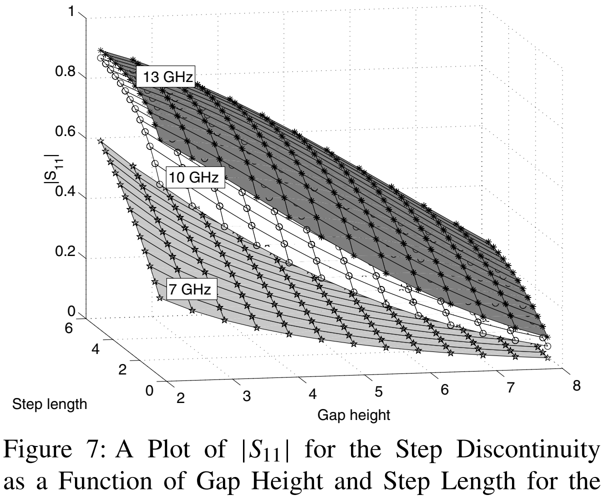

Figure 7: A Plot of $|S_{11}|$ for the Step Discontinuity as a Function of Gap Height and Step Length for the Frequencies 7, 10 and 13 GHz

In line with the accuracy level of the full-wave simulator, it was decided to set the target accuracy to 10^{-3} (or in logarithmic scale, -60dB).

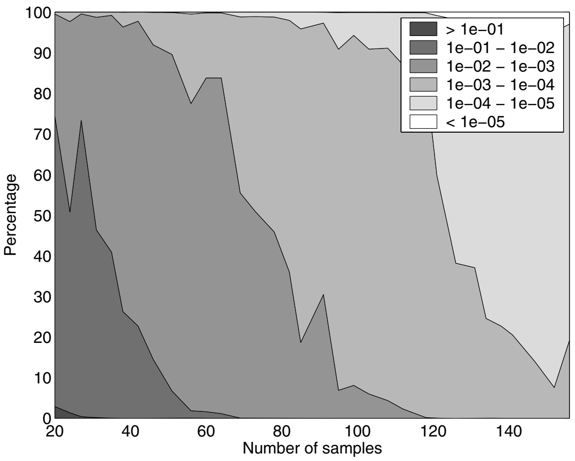

Figure 8: Metamodel Error Plot for the Step Discontinuity

4.4 Real-Life Example: The Inductive Posts

A full-wave electromagnetic MATLAB toolbox (Lehmensiek 2001) was used to simulate the scattering parameters of two centred inductive post discontinuities in a rectangular waveguide. The three input parameters we varied were frequency (7 to 13GHz), post diameter (1 to 5mm) and distance between posts (4 to 18mm). Figure 9 shows a plot of the magnitude of the first scattering parameter S_{11} at three fixed frequencies. We sampled the function on a 40^3 grid for metamodel verification purposes.

Figure 10 shows the error plot for the inductive posts example. Even after more than 300 iterations, we failed to achieve the desired accuracy. We conjecture inaccurate outputs or noise on the outputs of the simulator might be the cause of this result. After 100 samples an accuracy of 0.01 is achieved. After that, we quickly reach 10^{-3} accuracy in 90% of the domain, but fail to model the whole domain up to that accuracy.

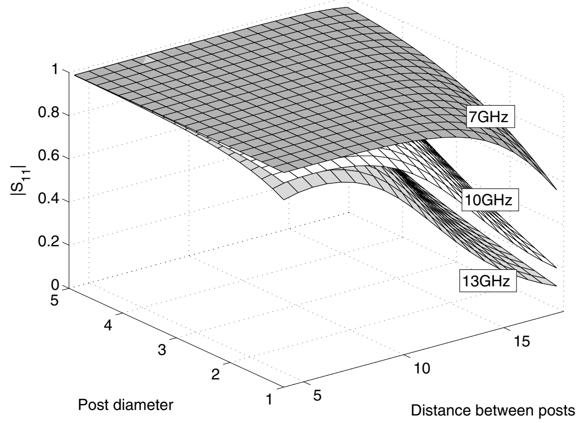

Figure 9: A Plot of $|S_{11}|$ for the Inductive Posts as a Function of Post Diameter and Distance Between Posts for the Frequencies 7, 10 and 13 GHz

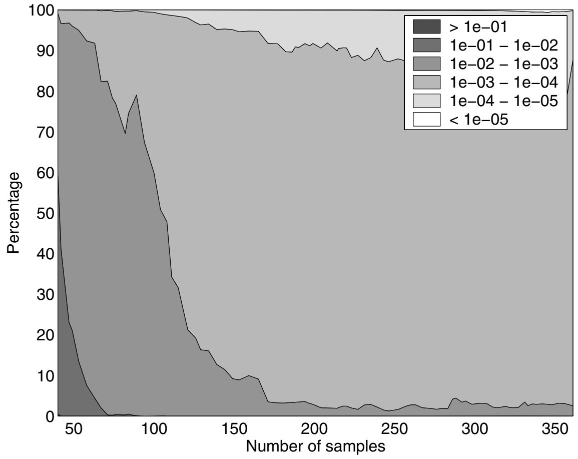

Figure 10: Metamodel Error Plot for the Inductive Posts

5 CONCLUSION

After conducting several tests on our toolbox, we conclude that using rational and polynomial models with 2 or 3 inputs, we were able to model most of our examples to the desired accuracy without encountering any problems. The inductive posts example fails to reach the desired accuracy. In section 4.2 it was illustrated that the adaptive weight selection works as it should. The error plots clarify how the metamodelling error is reduced in each iteration, in order to achieve a global accuracy of 10^{-3} or 10^{-4}, depending on the example.

6 SOFTWARE

The MATLAB code and datasets used in this paper is grouped into the Multivariable MetaModelling toolbox (M^3-toolbox, Hendrickx 2005), which is publicly available on our research group's website, <www.coms.ua.ac.be>. Instructions are provided to reproduce the results generated in this paper.

ACKNOWLEDGMENTS

We thank Robert Lehmensiek for the MATLAB code used to simulate the step discontinuity in section 4.3 and the inductive posts used in section 4.4.

REFERENCES

ADS Advanced Model Composer, Agilent Technologies, EEsof division, Santa Rosa, CA.

ADS Momentum, Agilent Technologies, EEsof division, Santa Rosa, CA.

AppLeS parameter sweep template. <http://grail.sdsc.edu/projects/apst/>.

Bandler, J., R. Biernacki, S. H. Chen, P. Grobelny, and R. Hemmers, 1994. Space mapping technique for electromagnetic optimization. In *IEEE Transactions Microwave Theory and Techniques*, 42:2536–2544.

Barton, R. R., 1998. Simulation metamodels. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson, and M. S. Manivannan, 167–174. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Buhmann, M. D., 2003. *Radial basis functions*. Cambridge University Press.

Chung, H.-S., and J. J. Alonso, 2000. Comparison of approximation models with merit functions for design optimization. Report AIAA-2000-4754.

Cressie, N. A. C., 1993. *Statistics for spatial data*. John Wiley & Sons.

De Geest, J., 2000. Adaptieve techniek voor de modellering van passieve planaire microgolfcircuits. Doctoral dissertation, University of Ghent, Belgium.

De Geest, J., T. Dhaene, N. Faché, and D. De Zutter, 1999. Adaptive CAD-model building algorithm for general planar microwave structures. In *IEEE Transactions on Microwave Theory and Techniques*, 47(9):1801–1809.

Dhaene, T., 2002. Automated fitting and rational modelling algorithm for EM-based S-parameter data. In *Lecture Notes in Computer Science*, 2367:99–105.

Dhaene, T., J. Ureel, N. Faché, and D. De Zutter, 1995. Adaptive frequency sampling algorithm for fast and accurate S-parameter modeling of general planar structures. In *Microwave Symposium Digest, 1995, IEEE MTT-S International*, 3:1427–1430.

Gohberg, I., T. Kailath, and V. Olshevsky, 1995. Fast Gaussian elimination with partial pivoting for matrices with displacement structure. In *Mathematics of computation*, 64(212):1557–1576.

Hendrickx, W., and T. Dhaene, 2005. Multivariate modelling of complex simulation-based systems. In *Proceedings of the 4th international workshop on multidimensional (nD) systems (NDS 2005)*.

Hendrickx, W., 2005. M^3 Toolbox. Available on <www.coms.ua.ac.be> in the software section.

Krige, D., 1951. A statistical approach to some basic mine valuation problems on the Witwatersrand. In *Journal of the chemical, metallurgical and mining society of South Africa*, 52:119–139.

Lehmensiek, R., 2001. Efficient adaptive sampling applied to multivariate, multiple output rational interpolation models, with applications in electromagnetics-based device modelling. Doctoral dissertation, University of Stellenbosch, South Africa.

Lehmensiek, R., and P. Meyer, 2001. Creating accurate multivariate rational interpolation models for microwave circuits by using efficient adaptive sampling to minimize the number of computational electromagnetic analyses. In *IEEE Transactions Microwave Theory and Technology*, 49(8):1419–1430.

Martin, J. D., and T. W. Simpson, 2005. Use of Kriging models to approximate deterministic computer models. In *AIAA Journal*, 43(4):853–863.

Nimrod: Tools for distributed parametric modelling. <http://www.csse.monash.edu.au/~davida/nimrod/>.

Sacks, J., W. J. Welch, T. Mitchell, and H. P. Wynn, 1989. Design and analysis of computer experiments. In *Statistical science*, 4(4):409–435.

Santner, T. J., B. J. Williams, and W. I. Notz, 2003. *The design and analysis of computer experiments*. Springer series in statistics.

Søndergaard, J., 2003. Optimization using surrogate models by the space mapping technique. Doctoral dissertation, Technical University of Denmark, Lyngby, Denmark.

Van Beers, W. C. M., and J. P.C. Kleijnen, 2004. Kriging Interpolation in Simulation: A Survey. In *Proceedings of the 2004 Winter Simulation Conference*, ed. R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peter, 113–121. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Wendland, H., 2005. *Scattered data approximation*. Cambridge University Press.

AUTHOR BIOGRAPHIES

WOUTER HENDRICKX is a PhD student with the research group Computer Simulation and Modelling of complex systems of the University of Antwerp, Belgium. He is currently supported by the FWO - Flemish Fund for Scientific Research. His e-mail address is <wouter.hendrickx@ua.ac.be> and his web page can be found on the site of the research group <www.coms.ua.ac.be>.

TOM DHAENE received his PhD degree in electrotechnical engineering from the University of Gent, Belgium, in 1993. His PhD research focused on different aspects of full-wave electro-magnetic circuit modelling, transient simulation, and time-domain characterisation of high-frequency and high-speed interconnections. Since September 2000, he is a Professor at the University of Antwerp, Belgium, in the Department of Mathematics and Computer Science. He is head of the COMS research group (Computer Modelling and Simulation of Complex Systems). His research interests are in the field of metamodelling (or surrogate modelling) of complex systems, circuit and electromagnetic (EM) modelling of high-speed interconnections, and adaptive system identification of linear time-invariant systems. His e-mail address is <tom.dhaene@ua.ac.be> and his web page can be found on the site of the research group <www.coms.ua.ac.be>.