Proceedings Paper:
Polynkin, Andrey, Toropov, Vassili and Shahpar, Shahrokh (2008) Adaptive and parallel capabilities in the Multipoint Approximation Method. In: 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Proceedings. AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 10-12 Sep 2008, Victoria, British Columbia, Canada. Conference Proceeding Series (AIAA-2). AIAA .

https://doi.org/10.2514/6.2008-5803

12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
10 - 12 September 2008, Victoria, British Columbia Canada

AIAA 2008-5803

Adaptive and parallel capabilities in the Multipoint Approximation Method

Andrey Polynkin*, Vassili Toropov†
University of Leeds, Leeds, LS2 9JT, UK

Shahrokh Shahpar‡
Rolls-Royce plc, Derby, DE24 8BJ, UK

I. Introduction

In the present work the Multipoint Approximation Method (MAM) has been enhanced with new capabilities that allow to solve large scale design optimization problems more efficiently. The first feature is adaptive building of approximate models during the optimization search. And the second feature is a parallel implementation of MAM.

A traditional approach to adaptive building of metamodels is to check several types for their quality on a set of design points and select the best type. The technique presented in this paper is based on the assembly of multiple metamodels into one model using linear regression. The obtained coefficients of the model assembly are not weights of the individual models but regression coefficients determined by the least squares minimization method.

The enhancements were implemented within Multipoint Approximation Method (MAM) method related to mid-range approximation framework. The developed technique has been tested on several benchmark problems.

II. Outline of Multipoint Approximation Method (MAM)

This technique (Toropov et al., 1993) replaces the original optimization problem by a succession of simpler mathematical programming problems. The functions in each iteration present mid-range approximations to the corresponding original functions. These functions are noise-free. The solution of an individual sub-problem becomes the starting point for the next step, the move limits are changed and the optimization is repeated iteratively until the optimum is reached. Each approximation function is defined as a function of design variables as well as a number of tuning parameters. The latter are determined by the weighted least squares surface fitting using the original function values (and their derivatives, when available) at several sampling points of the design variable space. Some of the sampling points are generated in the trust region, and the rest is taken from the extended trust region, i.e. the pool of points considered in the previous iterations (van Keulen et al., 1997).

A general optimization problem can be formulated as

$$F_0(\boldsymbol{x}) \to \min, \quad F_j(\boldsymbol{x}) \le 1 \ \ (j = 1,...,M), \quad A_i \le x_i \le B_i \ \ (i = 1,...,N) \tag{1}$$

* Research Fellow
† Professor of Aerospace and Structural Engineering, AIAA Associate Fellow
‡ RR Engineering Associate Fellow - Aerothermal Design Systems, AIAA Associate Fellow, FRAeS

where $\boldsymbol{x}$ refers to the vector of design variables. In order to reduce the number of calls for the response function evaluations and to lessen the influence of noise, the MAM replaces the optimization problem by a sequence of approximate optimization problems:

$$\tilde{F}_0^k(\boldsymbol{x}) \to \min,\ \tilde{F}_j^k(\boldsymbol{x}) \le 1\ (j=1,...,M),\ A_i^k \le x_i \le B_i^k, A_i^k \ge A_i,\ B_i^k \le B_i\ (i=1,...,N) \qquad (2)$$

where k is the iteration number.

The selection of the noise-free approximate response functions $\tilde{F}_j^k(\boldsymbol{x})\ (j=0,...,M)$ is such that their evaluation is inexpensive as compared to the evaluation of the response functions F_j, although they are not necessarily explicit functions of the design variables. The approximate response functions are intended to be adequate in a current search sub-domain. This is achieved by appropriate planning of numerical experiments and use of the trust region defined by the side constraints A_i^k and B_i^k.

III. Building of approximations

In the present work an approach is investigated based on the assembly of different approximate models into one metamodel in the form

$$\tilde{F}(\boldsymbol{x}) = \sum_{l=1}^{NF} b_l \varphi_l(\boldsymbol{x}) \qquad (3)$$

The use of multiple metamodels has recently been studied for example by F.Viana et al. (2008) and E.Acar et al. (2008) where coefficients b_l in (3) were treated as weights reflecting the accuracy of the individual surrogates on a set of validation points. Thus, more accurate components φ_l obtain larger values of the multipliers and vice versa provided that

$$\sum_{l=1}^{NF} b_l = 1 \qquad (4)$$

Individual surrogates such as Polynomial Response Surface (PRS), Kriging (KRG) and Radial Basis Functions (RBF), Gaussian Process (GP), and Support Vector Regression (SVR) were considered in the above studies.

This work considers an alternative approach for building the expression (3). The idea to use the regression analysis for combining different metamodels instead calculating the weights for each component was motivated by our early work (Toropov, 1989) where the regressors were intended to describe the behavior of separate structural (mechanical) sub-systems. In the present work, under "sub-systems" we imply individual metamodels obtained by any analytical or numerical method.

The procedure is based on the minimization of the weighted least squares problem

$$\sum_{p=1}^{P} w_{pj}\left[F_j(\boldsymbol{x}_p) - \tilde{F}_j(\boldsymbol{x}_p, \boldsymbol{b}_j)\right]^2 \to \min \qquad (5)$$

that leads to solving the linear system of NF equations with NF unknowns b_l where NF is the number of regressors in the model bank $\{\varphi_l\}$. Here the coefficients b_l are regression coefficients that should not be considered as weight factors, e.g. could be positive or negative. The parameters w_{pj} refer to the weights that reflect the inequality of data obtained in different sampling points P, see Toropov et al. (1993).

The functions φ_l in (3) are determined in the similar manner

$$\sum_{p=1}^{P} w_{pj}\left[F_j(\boldsymbol{x}_p) - \varphi_j\left(\boldsymbol{x}_p, \boldsymbol{a}_j\right)\right]^2 \rightarrow \min \tag{6}$$

where minimization is carried out with respect to the tuning parameters $\boldsymbol{a}_j$. This is done prior to the procedure (5) has been applied.

The selection of the regressors φ_l is based on the number of sampling points currently located in the trust region. In the mid-range approximation framework, inexpensive approximate models for objective and constraint functions are built using minimum required number of sampling points. The simplest case is a linear function of the tuning parameters $\boldsymbol{a}$:

$$\varphi(\boldsymbol{x}) = a_0 + \sum_{i=1}^{N} a_i x_i \tag{7}$$

This structure can be extended to an *intrinsically linear* function (Box, Draper 1987). Such functions are nonlinear, but they can be led to linear ones by simple transformations. The most useful function among them is the multiplicative function

$$\varphi(\boldsymbol{x}) = a_0 \prod_{i=1}^{N} x_i^{a_i} \tag{8}$$

Intrinsically linear functions have been successfully used for a variety of design optimization problems. The advantage of these approximation functions is that a relatively small number N+1 (N is number of design variables) of tuning parameters a_i is to be determined, and the corresponding least squares problem is solved easily. This is the most important feature of such approximations as it allows applying them to large scale optimization problems.

Other intrinsically linear functions may be considered in the model bank, e.g.

$$\varphi(\boldsymbol{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i \tag{9}$$

$$\varphi(\boldsymbol{x}) = a_0 + \sum_{i=1}^{N} a_i x_i^2 \tag{10}$$

$$\varphi(\boldsymbol{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i^2 \tag{11}$$

As more points are added to the trust region the approximations may be switched to higher quality models, e.g. rational model

$$\varphi(\boldsymbol{x}) = \frac{a_1 + a_2 x_1 + a_3 x_2 + ... + a_{n+1} x_n}{1 + a_{n+2} x_1 + a_{n+3} x_2 + ... + a_{2n+1} x_n} \tag{12}$$

This type of approximations was studied before by, e.g. Burgee et al. (1994) and Salazar Celis et al. (2007). Due to rapidly growing number of coefficients for large N (that is the main objective of this work), the function structure has to be limited to low degree polynomials (e.g. linear) and small data sets

The coefficients in (12) are determined using a least squares approach which reduces to a nonlinear optimization problem with a constraint on the sign of the denominator (positive or negative). The latter is necessary in order to prevent denominator from crossing the zero axis within a specified trust region. One may note that this formulation yields the objective function with many local minima. Currently this problem is resolved using optimization restarts from a specified number of initial guesses randomly generated in the trust region.

Tests results demonstrated that, although the above functions (7-12) may describe the global behavior rather poorly, such approximations proved to be efficient in the mid-range approximation framework of MAM.

It should be mentioned that functions φ_l may also represent global approximations such as PRS, KRG, RBF, etc. This issue will be addressed to our next work.

IV. Design of Experiments

In the present work, new sampling points are generated randomly. The added points are checked for calculability of the response function and, if the check fails, a new set of points is generated until a required number of sampling points (all passing the check) are obtained. To improve the quality of the random plan, a constraint on the minimal distance between the points is imposed using the following expression:

$$\frac{dist^e}{Diag} \geq r\,, \tag{13}$$

where

$$Diag = \sqrt{\sum_{i=1}^{N} \left(B_i - A_i\right)^2}\,,$$

$$dist^e = \sqrt{\sum_{i=1}^{N} \left(x_i^e - x_i^p\right)^2} \qquad (e, p = 1,...,P; e \neq p)\,.$$

In (13) *Diag* is a characteristic size of the trust region defined by lower A_i and upper B_i limits, e is number of a new sampling point, p is number of a previously generated point, and P is the total number of sampling points in the search sub-region.

The parameter r is initially set to 0.9. However if the condition (13) is not satisfied after a prescribed number of random generations for a new point, the value of the threshold ratio is iteratively reduced

$$r = r * coeff\ , \quad 0.9 \le coeff < 1$$

until the constraint is satisfied.

Figure 1 demonstrates the quality of design of experiments using a proposed technique. As an example, two patterns (20 and 100 points) were generated for the two-bar truss optimization test problem; see Toropov et al. (1993).

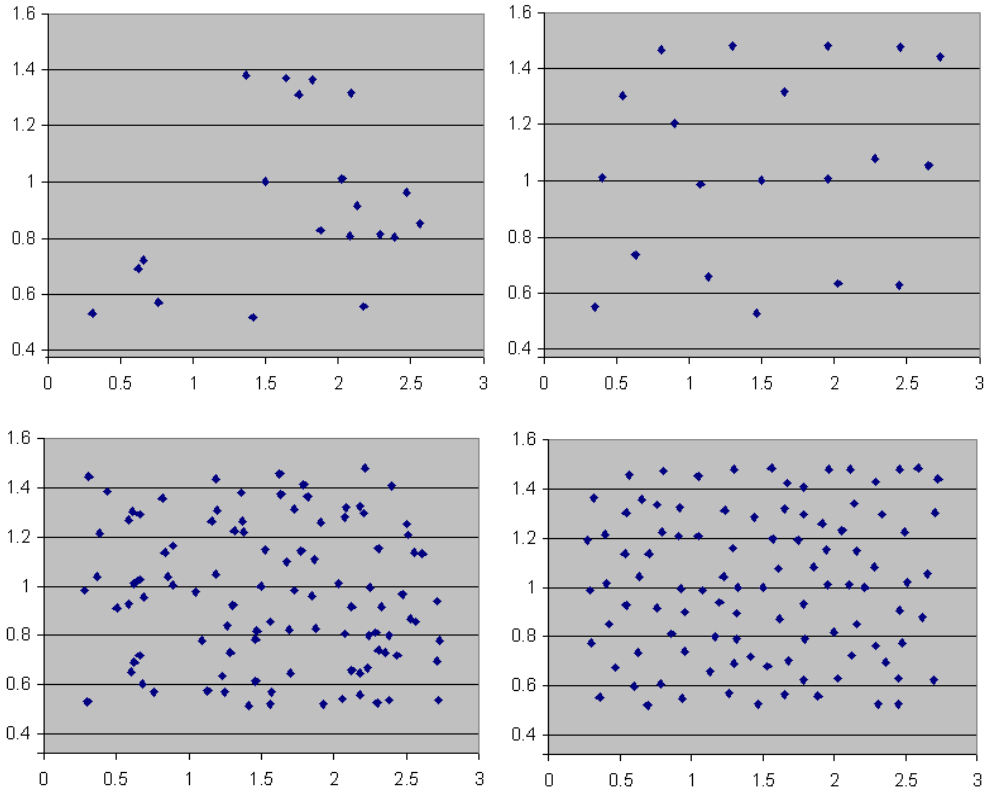

Figure 1: Designs of experiments generated randomly without (left) and with (right) a constraint imposed on the minimal distance between sampling points

Analytical tests have shown that the above technique may improve the quality of the approximate functions and therefore reduce number of MAM optimization steps.

V. Parallel implementation

Currently parallel processing has been developed without a CPU load testing mechanism (for example, using uptime option). MAM sets up the parallel jobs using a specified number of free processors (or nodes) *NP* available on a Linux cluster. It is assumed that this number is known *a priory* and is not changed during the optimization run, as for instance in implementation of van Keulen and Toropov (1999) for the case of heterogeneous computing environment.

In order to accelerate the performance of the algorithm, the number of sampling points generated in each step (starting from the second step) may be set equal to $k \times NP$ number where $k \geq 1$. In terms of the wall clock time, the latter will be equivalent to k function evaluations per step.

To ensure that the head node is not overloaded, a submission script was implemented based on performing the individual jobs in local directories of the slave nodes for each DOE point. Otherwise such situation is likely to occur when a massive parallel run is submitted involving simulations that need reading and writing frequently to a disk.

VI. Trust region strategy

After having solved the approximate optimization problem, a new trust region must be defined, i.e. its new size and its location have to be specified. This is done on the basis of a set of parameters that estimate the quality of the approximations ("bad", "reasonable" or "good") and the location of the sub-optimum point in the current trust region. Once the parameters have been determined, the trust region is moved and resized.

If the sub-optimum point does not pass the check for calculability of the response functions, the trust region is reduced and the approximated problem is solved again. The only essential assumption here is that all functions of the optimization problem exist at the starting point.

In a parallel mode, a set of NP points (that includes the obtained optimum point of the iteration) is submitted for evaluation after each iteration.

VII. Optimization examples and discussions

The proposed method has been tested on several structural optimization problems. The results obtained for three test cases are presented in order to give some insight into the approach.

Vanderplaats scalable beam

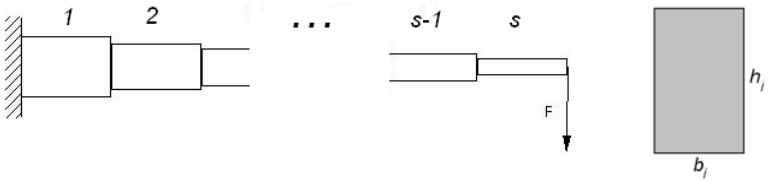

Figure 2: Scalable beam with rectangular cross sections

The problem is formulated as follows: minimize the volume of a cantilever beam

$$V = \sum_{i=1}^{S} b_i h_i l_i$$

under stress, aspect ratio, and tip deflection constraints

$$\sigma_i / \bar{\sigma} \le 1; \quad h_i /(20 b_i) \le 1; \; y_S / \bar{y} \le 1;$$

with the lower limits on cross section sizes

$$b_i \ge 1; \; h_i \ge 5;$$

Three cases were considered depending on the number of elements in the beam: (a) S=5 resulting in N=10 design variables and 11 constraints; (b) S=50 resulting in N=100 and 101 constraints; (c) S=500 resulting in N=1000 and 1001 constraints.

Following the proposed procedure (3) - (6) for building the approximate models, the next intrinsically linear functions were included in the model bank to solve the optimization problem:

1: linear
2: linear in squared variables x_i^2
3: multiplicative
4: linear in reciprocal variables $1/x_i$
5: linear in reciprocal squared variables $1/x_i^2$

It was found that a multiplicative function (function 3 in the list) was given preference in the model building for the stress and aspect ratio constraints. As an example, the coefficients b_i (i=1…5) obtained for one of the stress and aspect ratio constraints are shown below

$$b_1\text{=-0.86E-05} \quad b_2\text{=-0.12E-04} \quad b_3\text{=1.00003} \quad b_4\text{=0.32E-04} \quad b_5\text{=-0.61E-04}$$

As can be seen, all the parameters except b_3 actually equal to zero, i.e. the algorithm automatically selects a right model from the bank for a function whose behaviour is exactly described by that model.

In contrast to the stress and aspect ratio constraints, the models for the objective function and displacement constraint have non-zero coefficients b_i for all the available regressors in the bank. An example of coefficients determined for the displacement constraint during the optimization search is given below

$$b_1\text{=-0.119} \quad b_2\text{=0.15E-01} \quad b_3\text{=0.453} \quad b_4\text{=0.207} \quad b_5\text{=0.444}$$

It is interesting to note that the normalization condition (4), i.e. $\sum_{l=1}^{5} b_l = 1$ in this case, is automatically satisfied without explicit imposing a corresponding constraint on the regression coefficients.

MAM's optimization result for the case (a) is V=61914 (cm^3) with the corresponding vector of design variables {2.99; 2.77; 2.52; 2.20; 1.75; 59.84; 55.55; 50.47; 44.09; 34.99} (cm). All stress and aspect ratio constraints are active in the optimal point except the displacement constraint which is fulfilled with a big capacity. For comparison, Vanderplaats's solution obtained using exterior penalty function method is V=66169 (cm^3) with the vector {3.24; 2.90; 2.52; 2.26; 2.24; 56.77; 53.81; 50.30; 44.87; 41.71} (cm).

The optimization results obtained for all three cases (a-c) using MAM method are summarized in Fig. 3. An important observation is that number of MAM steps did not depend on the number of design variables while number of calls for analysis depended linearly on *N*.

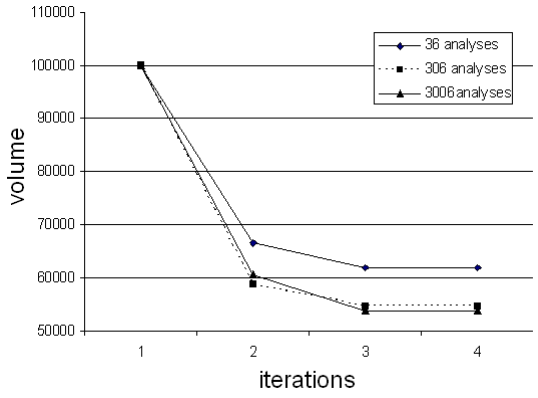

Figure 3: Convergence plots for the optimization cases with (a) *N*=10, (b) *N*=100, and (c) *N*=1000 design parameters. The number of analyses for each case is 36, 306, and 3006 respectively

A cantilever scalable thin-wall beam

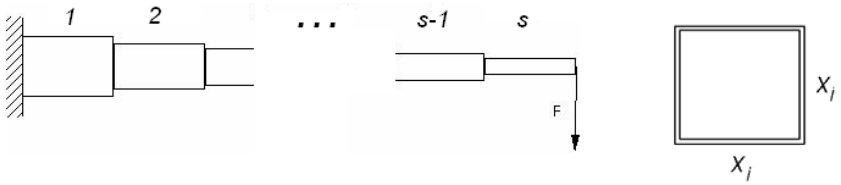

Figure 4: Scalable beam with hollow square cross-sections

In this test, a cantilever beam is built up of *S* elements with hollow square cross sections. The objective function is the weight of the beam that has to be minimized. There is a constraint imposed on the tip displacement. The design variables are heights (widths) of the square cross sections, Fig. 4.

Based on the discretization of five elements the optimization problem was formulated by Svanberg (1987) in a closed form:

minimize

$$F_0(x) = 0.0624 \cdot \sum_{i=1}^{5} x_i$$

subject to

$$F_1(x) = 61/x_1^3 + 37/x_2^3 + 19/x_3^3 + 7/x_4^3 + 1/x_5^3 \le 1$$

with a feasible starting point x_i=5 (i=1…5).

In order to solve the problem, the same set of 5 regressors was used as for the previous example. Typical values of the regression coefficients obtained during the optimization search for the displacement constraint were

$$b_1 = 1.828 \quad b_2 = -0.746 \quad b_3 = 0.15E\text{-}02 \quad b_4 = 3.419 \quad b_5 = -3.502$$

that consistently satisfied the normalization condition (4).

For the objective function the algorithm always selected b_1 =1 and b_i =0 (i=2…5). The solution was obtained after 5 MAM iterations and 31 analyses. The optimum point is {6.015; 5.309; 4.493; 3.502; 2.152}. The corresponding value of the objective function is 1.339.

It is worth to note that this problem seemed to be tight for solving by approximation techniques. For instance, Svanberg's MMA method converged after four iterations after some preliminary tuning while Fleury's CONLIN optimizer didn't converge at all. Using an earlier version of Toropov's MAM (1993) with a multiplicative approximation (as a default type) for the constraint function the solution was achieved after 17 steps (104 analyses) that we considered as unsatisfactory.

In order to verify the performance of the algorithm on the large-scale level, the problem was extended to 100 and 500 beam elements resulting in N=100 and N=500 design variables. The corresponding solutions are compared in Fig.5.

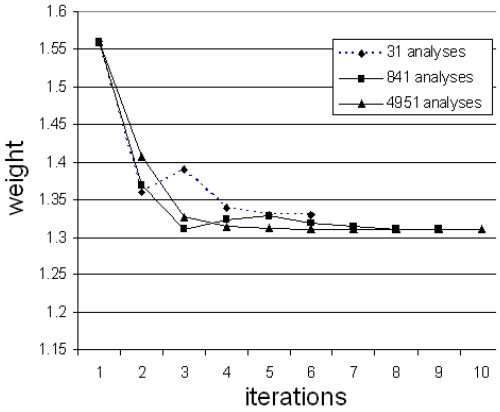

Figure 5: Convergence plots for the optimization cases with N=5, N=100, and N=500 design parameters. The number of analyses for each case is 31, 841, and 4951 respectively

Note that including a regressor $\varphi(\boldsymbol{x}) = a_0 + \sum_{i=1}^{N} a_i / x_i^3$ in the model bank can considerably improve the performance of the algorithm as the solution of the problem in this case requires one MAM iteration only.

Unconstraint minimization problem

The robustness of the proposed technique was tested on the unconstraint minimization problem considered by Vanderplaats (1999) for two-spring system. The objective is to find an equilibrium position of the springs by minimizing the total potential energy of the system

$$PE = 0.5 * K_1\left[\sqrt{x_1^2 + (l_1 - x_2)^2} - l_1\right]^2 + 0.5 * K_2\left[\sqrt{x_1^2 + (l_2 + x_2)^2} - l_2\right]^2 - P_1 x_1 - P_2 x_2$$

The constants K_i are spring stiffnesses, P_i are loads, l_i are the original spring lengths, and x_i are displacements where K_1=8 N/cm, K_2=1 N/cm, P_1=P_2=5 N, l_1=l_2=10 cm. The two-variable function space is shown in Fig. 6. In order to consider a positive range of variations for the function and design variables, the following scaling has been applied: PE=PE+100; x_i=x_i+6. Then the exact minimum of the scaled problem is {14.63; 10.45} (cm) with PE=58.19 (N·cm).

In should be mentioned here that the method presented in this work is primarily developed for solving constraint optimization problems where optimum always belongs to the boundary of the feasible search domain. This is because simple monotonic functions were used as the metamodels in the mid-range approximation concept. As it will be shown below, based on the proposed approximation scheme (3-5) the method is now capable to solve this sort of problems too.

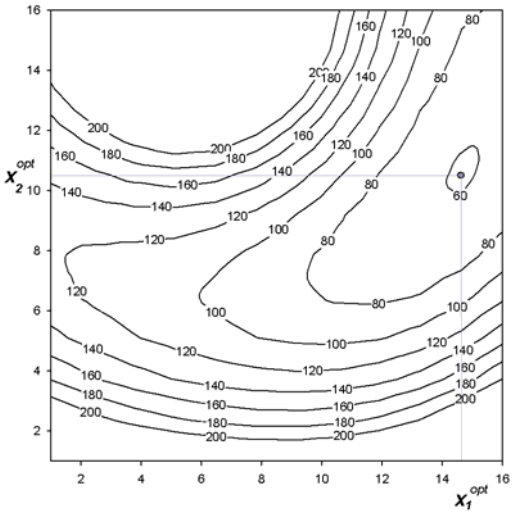

Figure 6: Two-variable function space for the spring-force system

Depending on number of sampling points generated in the trust region, three solutions were obtained. The results are summarised in Table 1. The optima were identified as internal points of the trust region. Note that the minimum number of sampling points required to build the approximation model (3) is five as five regressors were used in the model bank for this case.

sampling points per iteration	x_1^{opt}	x_2^{opt}	PE	iterations	calls for function
6	15.0	11.06	58.44	8	49
15	14.77	10.72	58.22	8	121
20	14.65	10.47	58.20	8	161

Table 1: Optimization results depending on the number of sampling points in the trust region

Typical regression coefficients for the model built in the first and last 8th iteration using 20 sampling points are

1st iteration: b_1 =-5.86 b_2 = 3.87 b_3 =5.67 b_4 =-3.40 b_5 =0.73
8th iteration: b_1 = -49603.9 b_2 = 24828.1 b_3 = 63.74 b_4 =-24755.0 b_5 =49468.1

The meaning of the negative coefficients can now be illustrated. Figures 7-8 show how the technique defines different (positive/negative) slopes for monotonic functions φ_i from the model bank in order to assemble the adequate approximation with non-monotonic behaviour in the trust regions generated in the first and the last (8th) iterations during the optimization search.

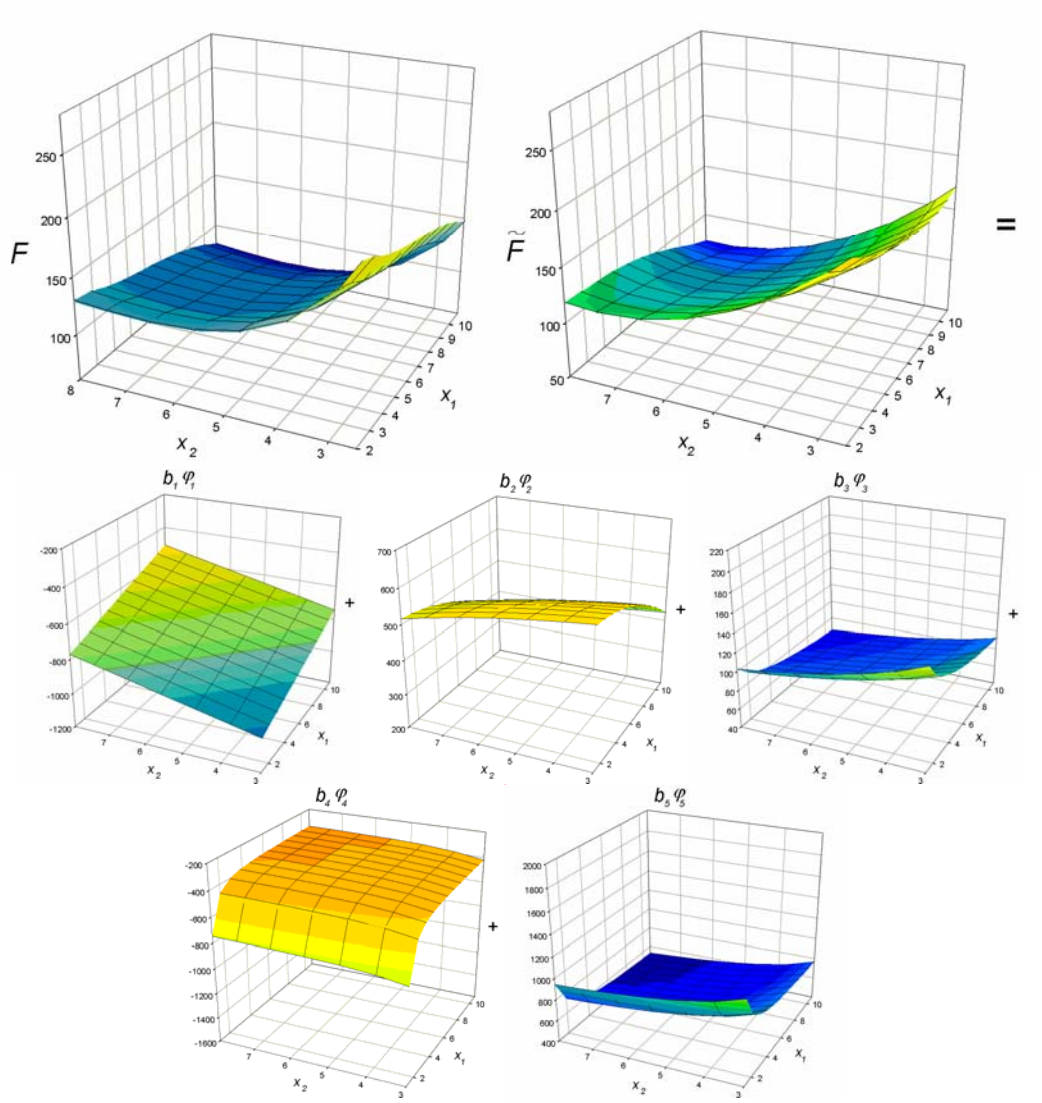

Figure 7: Actual function and metamodel assembly built of five regressors in the first MAM iteration

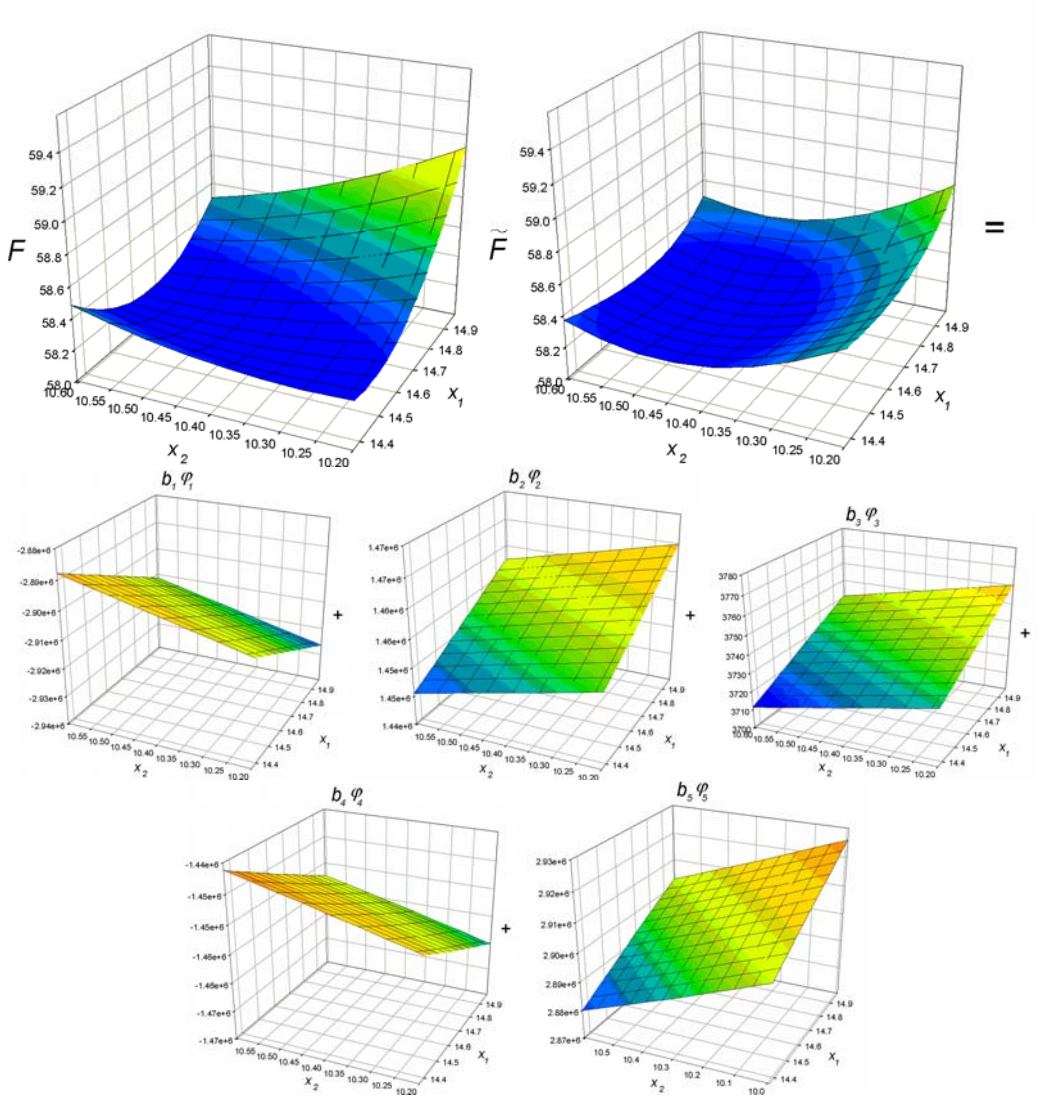

Figure 8: Actual function and metamodel assembly built of five regressors in the 8th MAM iteration

In order to compare the accuracy of different components φ_l with the performance of the assembled model $\tilde{F}$, *RMSE* of scaled response values was used

$$RMSE = \sqrt{\frac{1}{K_{test}} \sum_{i=1}^{K_{test}} \left(\frac{\tilde{F}_i - F_i}{F_i} \right)^2}$$

where K_{test} is number of testing points randomly generated in the trust region; $\tilde{F}_i$ and F_i are model and actual function values at test points. For this case 500 test points were generated.

Values of *RMSE* depending on the number of sampling points are given in Tables 2 and 3.

Sampling points per iteration	$\tilde{F}$	φ_1	φ_2	φ_3	φ_4	φ_5
6	0.18	0.21	0.22	0.19	0.27	0.19
15	0.12	0.21	0.23	0.18	0.19	0.17
20	0.11	0.22	0.23	0.18	0.19	0.17
30	0.12	0.21	0.22	0.18	0.19	0.17
40	0.11	0.21	0.23	0.18	0.19	0.17

Table 2: *RMSE* for metamodel assembly and individual regressors in the first MAM iteration

The main finding from these results is that the accuracy of the assembled model is always higher than the accuracy of its components. The trend is kept for the different number of sampling points and is observed in different iterations.

Sampling points per iteration	$\tilde{F}$	φ_1	φ_2	φ_3	φ_4	φ_5
6	3.00e-3	6.28e-3	6.23e-3	6.23e-3	6.32e-3	6.38e-3
15	3.41e-3	3.88e-3	3.86e-3	3.88e-3	3.92e-3	3.90e-3
20	3.25e-3	4.57e-3	4.58e-3	4.54e-3	4.54e-3	4.56e-3
30	3.24e-3	3.56e-3	3.58e-3	3.53e-3	3.53e-3	3.54e-3
40	1.02e-3	1.97e-3	1.96e-3	1.97e-3	1.99e-3	1.99e-3

Table 3: *RMSE* for metamodel assembly and individual regressors in the 8th MAM iteration

VIII. Acknowledgment

The authors are grateful to the UK Department of Trade and Industry, Rolls-Royce plc and Airbus UK Ltd for the support of this work within the CFMS R&D programme.

References

Acar, E., Rais-Rohani, M. (2008). Ensemble of metamodels with optimized weight factors. 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials 7-10April 2008, Schaumburg, IL, AIAA 2008-1884

Box, G.E.P., Draper, N.R. (1987). Empirical model-building and response surfaces. New York: John Wiley and Sons

Fleury, C. (1989). CONLIN: An efficient dual optimizer based on convex approximation concepts. Structural Optimization, 1:81-89

Salazar, C., Cuyt, A., and Verdonk, B. (2007). Rational approximation of vertical segments. Numerical algorithms, 45:375-388

Svanberg, K. (1987). The method of moving asymptotes - a new method for structural optimization. Int. J. Num. Meth. Eng., 24: 259-373

Toropov, V.V. (1989). Simulation approach to structural optimization. Structural Optimization, 1:37-46

Toropov, V.V., Filatov, A.A., and Polynkin, A.A. (1993). Multiparameter structural optimization using FEM and multipoint explicit approximations. Structural Optimization 6:7-14

Vanderplaats, G,N. (1999). Numerical optimization tecniques for engineering design. 3rd edition, VR&D

van Keulen, F., and Toropov, V.V. (1997). New developments in structural optimization using adaptive mesh refinement and multi-point approximations. Engineering Optimization 29:217-234

van Keulen, F. and Toropov, V.V. (1999). The multipoint approximation method in a parallel computing environment. ZAMM, 79 (supp. 1):67-70

Viana, F.A.C., Haftka, R.T. (2008). Using multiple surrogates for metamodeling. 7th ASMO-UK/ISSMO International Conference on Engineering Design Optimization, Bath, UK, July 7-8, 2008:132-137