

Accurate And Efficient Global Approximation using Adaptive Polynomial RSM for Complex Mechanical and Vehicular Performance Models

Y. Z. Wu, Z. Dong, and S. K. You

Abstract—Global approximation using metamodel for complex mathematical function or computer model over a large variable domain is often needed in sensibility analysis, computer simulation, optimal control, and global design optimization of complex, multi-physics systems. To overcome the limitations of the existing response surface (RS), surrogate or metamodel modeling methods for complex models over large variable domain, a new adaptive and regressive RS modeling method using quadratic functions and local area model improvement schemes is introduced. The method applies an iterative and Latin hypercube sampling based RS update process, divides the entire domain of design variables into multiple cells, identifies rougher cells with large modeling error, and further divides these cells along the roughest dimension direction. A small number of additional sampling points from the original, expensive model are added over the small and isolated rough cells to improve the RS model locally until the model accuracy criteria are satisfied. The method then combines local RS cells to regenerate the global RS model with satisfactory accuracy. An effective RS cells sorting algorithm is also introduced to improve the efficiency of model evaluation. Benchmark tests are presented and use of the new metamodeling method to replace complex hybrid electrical vehicle powertrain performance model in vehicle design optimization and optimal control are discussed.

Keywords—Global approximation, polynomial response surface, domain decomposition, domain combination, multiphysics modeling, hybrid powertrain optimization

I. INTRODUCTION

TODAY design optimization and optimal control of a mechanical, electrical, mechatronics or vehicular system are often carried using complex and computation intensive multi-physics models.

Professor Y. Z. Wu is with the National CAD Supported Software Engineering Centre, Huazhong University of Science & Technology, Wuhan, China.

Professor S. K. You is with the Department of Mechanical Engineering, University of Guangzhou, Guangzhou, China.

Professor Wu and You were both Visiting Professors at the Department of Mechanical Engineering, University of Victoria, British Columbia, Canada in 2009.

*Professor Z. Dong is with the Department of Mechanical Engineering, University of Victoria, British Columbia, Canada. (e-mail: zdong@uvic.ca).

Due to the demanding and lengthy process for evaluating these original models, and the added complexity of global optimization, a simplified model or metamodel of the original model is often used. This approximation model substitutes the original complex nonlinear function or computer simulation model to make sensibility analysis, computer simulation, optimal control, and global design optimization of the complex, multi-physics systems feasible and efficient. For instance, the use of a metamodel to approximate the large finite element analysis model of a complex mechanical structure can considerably speed up the search of optimal design parameters in global design optimization. Another example is the model based powertrain design optimization of hybrid electrical vehicles. The complex powertrain performance model, which is constructed using various powertrain components, including internal combustion engine (ICE), electric motor/generators, continuous variable transmission (CVT), vehicle control system, etc., and their interactions determined by different powertrain architectures, requires extensive computation using special powertrain performance modeling and simulation programs. These programs, often implemented using multi-physics modeling and simulation software, such as MATLAB Simulink, or special powertrain modeling tools, such as Powertrain Simulation and Analysis Tools (PSAT), require extensive, recursive computation through a “black-box” simulation process. It is very difficult to carry out vehicle powertrain design optimization using these black-box simulations as objective or constraint functions. The use of computation efficient metamodels to approximate these complex black-box functions makes system level design analysis and optimization more feasible. The research on global design optimization of next-generation hybrid electric vehicle powertrain system, carried out by the authors, calls for these metamodels. The aim of this work is to introduce a new method that can generate the needed metamodel efficiently with high modeling accuracy. The approximate model represents the inputs-outputs relations of the modeled complex vehicle system. Different from a simple look-up table that only functions well for low-dimension problems; the metamodel is introduced to capture more complex, non-linear, large dimensional input-output relations.

Response surface method (RSM) is one of the well established and widely used global approximation tools, in which the approximate substitutes the original model is called a response surface (RS), a surrogate or a metamodel. Researches on RSM developments can be further divided into several key techniques [1], Kriging, Radial Basis Functions (RBF), Polynomial Response Surface (PRS), Support Vector Regression (SVR), Multivariate Adaptive Regression Spline (MARS) and their combinations. A brief overview of these techniques is presented below.

A. Kriging

Kriging technique was named after the South African mining engineer D. G. Krige [2] and was first used in computer experiments by Sacks et al [3]. Kriging model estimates the value of an original complex model using the combination of a known function and the departures captured by a stochastic process from a Bayesian perspective [4]. A conventional Kriging model interpolates the sampling points, which is an important characteristic for design and analysis of computer experiments. Kriging is also a flexible technique since different instances can be generated by choosing different pairs of regression and correlation functions [5]. In addition, Kriging is suitable for both high order functions and low dimensional problems, and it can approximate the original model smoothly with relatively fewer sampling points.

B. Radial Basis Functions

RBF technique was firstly introduced by Hardy R. L. to fit irregular topographic contours of geographical data [6]. A RBF model is expressed as the linear sum of a series of basis functions about the sampling points [7]. RBF performs well for high dimensional and high-order nonlinear problems. The accuracy of RBF falls between PRS and Kriging, while RBF model is easier to build than Kriging model [8]. Radial Basis Neural Network (RBNN) is a combination of RBS and the artificial neural network (ANN), which uses radial basis functions as the transfer functions of ANN [9]. Compared with the standard feed-forward or back-propagation networks, RBNN may require more neurons, but it can be generated in a fraction of the time that takes to train standard networks.

C. Polynomial Response Surface

PRS approximation is another well established meta-modeling technique, which is first developed and described by Box and Wilson [10]. In PRS modeling, a second-order polynomial function is commonly used to approximate the original model [11]. The coefficients of the polynomial functions can be obtained using least square approximation and the fitting is unbiased with minimum variance. PRS can uniquely identify the significance of different design factors directly from the coefficients in the normalized regression model [12].

D. Support Vector Regression

SVR, a special implementation of *Support Vector Machines* [13], was introduced as an alternative technique for

approximating complex engineering analyses [14, 15]. Searching for the optimal hyperplane of the training data, the aim of SVR is to find a fitting function that has minimum deviation from training data points. Mathematically, SVR is expressed as a constant and the sum of a series of kernel functions about the training points. The parameters are obtained from a fitting process during which SVR minimizes an upper bound on the expected risk using alternative loss functions. According to S. M. Clarke *et al* [16], SVR had the best overall performance for the test bed of 26 engineering analysis functions in comparison to other approximating techniques. SVR also gives a robust approximation, providing a good compromise between prediction accuracy and robustness.

E. Multivariate Adaptive Regression Spline

MARS is a non-parametric regression technique that constructs the relationship from inputs to outputs variables through spline curve [17]. The input space is divided into regions containing their own regression equation which contains knots vectors which need to be optimized. MARS is suitable for problems with high input dimensions and has been applied in stochastic dynamic programming [18] and global optimization [19]. S. Richardson also implemented a *multivariate adaptive regression B-spline algorithm* (BMARS) for solving a class of nonlinear optimal feedback control problems [20].

F. Analysis

In introducing a RS approximation, needed original model evaluation, as well as modeling accuracy, model update computation time, needed memory space for holding the model, and speed of model evaluation are major concerns. The number of the evaluations for the original model should be limited since these evaluations may be computation expensive. The approximation processes is carried out through refining iterations, and time spent in each iteration should be minimized.

Kriging technique works well for high order functions and low dimensional problems, but is less efficient for low order functions and high dimensional problems [21]. As an interpolation method, it is highly sensitive to the noisy data. In addition, when the number of sampling points goes beyond two thousands, the computation time needed to construct a *Kriging* model becomes unacceptable. Tests of the DACE code [22] were carried out. Improperly selected initial values of the correlation function parameters often led to poor results. Proper matching of the regression and correlation functions is another challenge.

RBF technique performs well for high dimension and high-order nonlinear problems. Meanwhile the technique has trouble in dealing with linear problems, and is less efficient for low dimensional problems due to its slow convergence [23]. Like *Kriging*, when the number of sampling points is large, the computation time needed in constructing a RBF model becomes too long and unacceptable. For RBNN, its

drawback for some applications is the need of a large number of training points [1].

PRS technique is particularly suitable for lower order functions and low-to-medium dimensional problems near a local area. Some adaptive response surface methods have been proposed [24, 25], but these improved algorithms still encounter performance challenges for high order functions and highly nonlinear models over a large global domain.

Due to the need of a search on the upper bound of expected risk, SVR technique appears to be less efficient due to the constructing and refining procedure of the fitting process.

All these approximation techniques needs sampling points to construct metamodel. Classical design of experiments based sampling methods, such as central composite design, fractional/full factorial design or D-optimal design, have difficulty in meeting the tight accuracy requirement when the domain of the inputs variables is large. Space filling sampling, such as grid design, random sampling or Latin hypercube sampling, cannot adaptively allocate sampling data points according to the nonlinear characteristic of the original model. In the work, we do not attempt to construct one single response surface upon a global large domain. Under most occasions, the global domain is split into a number of smaller cells to support adaptive sampling.

MARS algorithm uses decomposition to construct an interpolation response surface over the global domain. When there are not enough sampling points, precision of the MARS model becomes less dissatisfactory. The algorithm includes calculations of lack-of-fit procedure for finding the optimal knots vector. This computation expensive procedure leads to low modeling efficiency [17].

Armin Iske and Jeremy Levesley [26] studied a method for multilevel scattered data approximation by using compactly supported radial basis functions with adaptive domain decomposition. The numerical examples showed improvement on the accuracy of approximation over the well-established multilevel interpolation schemes. Other researchers studied the adaptive sampling and global approximation for RBF [27-28]. For a large domain and high dimension problem requiring a large number of sampling points, the RBF refining process remains is inevitably less efficient.

D. Busby *et al* [29] proposed a hierarchical nonlinear approximation method for experimental design and statistical data fitting. The iterative method combines suitable evaluation points using *Kriging* for statistical data analysis at each refinement step. Other recent improvements on *Kriging* technique can be found in [30-31]. Like domain decomposition with RBF, the computation demanding refining process appears to be less efficient, the metamodel occupies large memory space, and evaluation of given points is slow.

For PRS, D. Shahsavani and A. Grimvall [32] introduced an adaptive design and interpolation technique for extracting

highly nonlinear response surfaces from deterministic models. "A sequential design algorithm for cuboid domains is initiated by selecting an extended corner/centre point design for the entire domain, and then updated by decomposing this domain into disjoint cuboids and taking the corners and centre of these cuboids as new design points". The numbers of corner and center points for a p -dimension cubic is

$$n_1 = 1 + 2^p$$

while the quadratic polynomial items of the p -dimension is

$$n_2 = 1 + 2p + C_p^2$$

When $p \leq 3$, $n_1 < n_2$, interpolation can be done upon the cubic cell by adding additional points. But if $p > 3$; $n_1 > n_2$, it is impossible to interpolate the cell with a quadratic polynomial function, an issue in modeling multi-dimension problems.

G. Proposed method and organization of this paper

To overcome the drawbacks of the existing RS modeling methods for complex models over large variable domain, a new adaptive and regressive RS modeling method using quadratic functions and local area model improvement schemes is introduced in this work. The method applies an iterative and Latin hypercube sampling based RS update process, divides the entire domain of design variables into multiple cells, identifies rougher cells with large modeling error, and further divides these cells along the roughest dimension direction. A small number of additional sampling points from the original, expensive model are added over the small and isolated rough cells to improve the RS model locally until the model accuracy criteria are satisfied. The method then combines local RS cells to regenerate the global RS model with satisfactory accuracy. An effective RS cells sorting algorithm is also introduced to improve the efficiency of model evaluation. Benchmark tests are presented and applications of the new method in design optimization and optimal control are discussed.

II. MATHEMATICAL FOUNDATIONS OF THE METHOD

For an original model with k outputs and p inputs:

$$\mathbf{y} = f(\mathbf{x}),$$

where $\mathbf{y}: \Omega \subset \mathbb{R}^k$, $\mathbf{x}: \Omega \subset \mathbb{R}^p$; and $f: \mathbb{R}^k \rightarrow \mathbb{R}^p$.

The k outputs \mathbf{y} are expressed by k function sets:

$$\mathbf{y} = [y_1(\mathbf{x}), y_2(\mathbf{x}), \dots, y_k(\mathbf{x})]$$

For each of the i th scalar output ($1 \leq i \leq k$) y_i , a response surface can be built using the quadratic response surface of the form:

$$\hat{y}(\mathbf{x}) = \beta_0 + \sum_{i=1}^p \beta_{ii} x_i^2 + \sum_{i=1}^p \sum_{1 \leq i < j} \beta_{ij} x_i x_j \quad (1)$$

where β are coefficients determined using least squares regression.

For the entire global domain of the design variables, it is infeasible to construct a simple polynomial quadratic response

surface. A recursive partitioning procedure is then introduced to divide/decompose the cells with poor modeling accuracy along the roughest dimension into two separated cells. Additional sampling points are added to the newly formed cells. The recursive process continues until the accuracy of the metamodel meets desired criteria.

A. The Poor Performing Cells

In principle, the roughness of a cubic cell C can be evaluated according to the coefficients of the quadratic PRS and side dimensions of the cubic cell C [32]:

$$\hat{R}(C) = V(C) \left[\sum_{j=1}^p (2\beta_{jj}l_j^2)^2 + 2 \sum_{j < k} (\beta_{jk}l_jl_k)^2 \right]$$

where $V(C)$ represents the volume of C , and l_j is the length of the j th side of C .

However, in order to take the sampling data into account and to be compatible with other techniques or high order polynomial functions, a sample point container-box method is introduced. The worst performing cell is thus the one with the largest container box of the sampling points. As illustrated by the one dimension problem shown in Fig.1 (a), the area (2D volume) of the container box can be calculated by the distance a times height h , where a is the length of the vector \mathbf{a} which is measured from the left-down corner point to the right-up point of the cell; and h is the height of two line which are parallel to vector \mathbf{a} .

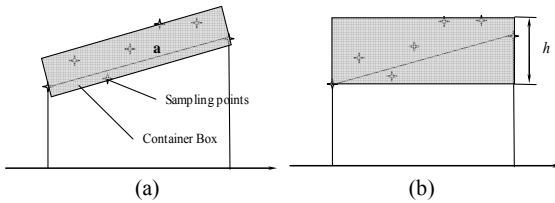


Fig.1 Roughness calculation of a cell

For the multi-dimensional problem, it is difficult to calculate the volume of the container box. Its value is then approximated using a simple expression as shown in Fig. 1 (b), i.e.

$$R(C) = hV(C) \quad (2)$$

where $h = \max_{1 \leq i \leq N} (y_i) - \min_{1 \leq i \leq N} (y_i)$, and y_i is the response value of i th point of the N sampling points.

B. The splitting dimension

Similarly, the roughest dimension is determined following the splitting direction criterion calculated using the coefficients of quadratic PRS [32],

$$\hat{S}(j) = (2\beta_{jj}l_j^2)^2 + \sum_{1 \leq k < j} (\beta_{kj}l_kl_j)^2 + \sum_{p \geq k > j} (\beta_{jk}l_jl_k)^2$$

and the splitting dimension j_s is:

$$j_s = \arg \max_{1 \leq j \leq p} (\hat{S}(j))$$

Using sampling data, the criterion can be simplified as

$$S(j) = \max_{1 \leq i \leq N_j} (h_i) \quad (3)$$

h_i is the height of the i th sample point along j th dimension. As shown in Fig 2, h_i can be calculated by

$$h_i = |y_i - y_i'|$$

y_i is the response value of the i th sampling point of the cell, y_i' is the interpolation of the two side points of the cell which have the same coordinates except the j th dimension, and can be evaluated by the PRS function.

Then the splitting dimension j_s is

$$j_s = \arg \max_{1 \leq j \leq p} (S(j)) \quad (4)$$

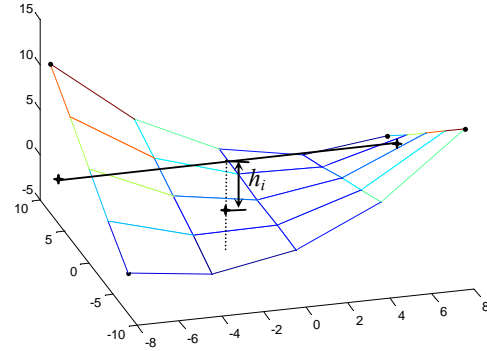


Fig.2 approximation of the splitting dimension

C. Re-composing the RS Set

After all the cells are refined and the models become accurate enough, some adjacent cells are combined into one cell to reduce the number of cells while maintaining modeling accuracy. As shown in Fig 3 (a), the original domain was divided into 5 cells during recursive partitioning. These cells are recombined into 3 cells later as shown in Fig 3 (b).

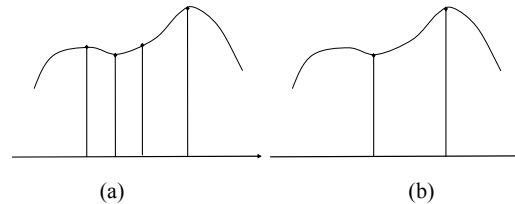


Fig.3 Cells recombining

Two cells can be combined or re-composed into one if the following two conditions are satisfied:

1) Domain adjacent constraint:

We assume that XL , XU are the left-down (with minimum values of all dimensions), and right-top (with maximum values of all dimensions) points of the cell container, respectively. If two combinable cells C_1 and C_2 should satisfies the following relations:

$$\forall j \in \{1, 2, \dots, p\}$$

$$XL_1(j) = XU_2(j) \text{ or } XL_2(j) = XU_1(j)$$

$$XL_1(i) = XL_2(i) \text{ and } XU_1(i) = XU_2(i) \quad (i \neq j)$$

2) Accuracy constraint

Assuming f_{new} is the regressive quadratic PRS function using all the N sampling points of the two cells; pt_i is the i th sampling points, then the *root square mean error (RSME)* of the new RS should be no larger than a given threshold $RSME0$:

$$RSME = \sqrt{\left(\sum_{i=1}^N (y_i - y_i')^2 \right) / N} \leq RSME0$$

where y_i is the response value of pt_i ;

$$y_i' = f_{new}(pt_i)$$

D. Sorting the RS set

The number of cells generated by the recursive RS scheme may be very large, ranging from hundreds to thousands, for a large variable domain and highly nonlinear model. Evaluation the metamodel value at a given point x using the RS set should use the proper cell (XL, XU), i.e.

$$XL \leq x \leq XU$$

For RS set with a large number of cells, locating the cell also demands computation time. A cell sorting binary search scheme has been introduced to speed up the process.

In order to sort the cells, we can define the ' \geq ' operation,

Definition 1: For two cells C_1, C_2 , we have $C_1 \geq C_2$, when p_1 (the left-down corner point of the cell C_1) is greater and equal than p_2 (that of C_2), $p_1 \geq p_2$, if the coordinate $p_1(j) > p_2(j)$, where $1 \leq j \leq p$.

III. GENERATION OF RS SET THROUGH A STEP-BY-STEP PROCESS

Generation of the RS set, RS_0 , is initiated with the entire domain Ω – a RS model covers the whole domain hypercube. Sampling points include corners and center testing points.

Definition 2: A cell is a basic domain that includes a hypercube and a response surface.

The data structure of RS has the following fields:

- Cube:** scope of the RS defined by XL, XU ;
- B:** coefficients of RS;
- R:** residue of regression;
- FC:** number of fittings executed
- SD:** sampling data (sample points and response values) used to form the RS function through regression
- TD:** testing data not used in regression

A. Algorithm steps

The step-by-step refining procedure consists of:

- Step1: construct an initial RS_0 over Ω , and carry out the first recursive partitioning for RS_0 , and generate initial RS set $RSS = RSS_0$;
- Step2: exit if stopping criterion is satisfied; else go to next step;

- Step3: identify the worst cells CS from the RSS using Eq. (2); and divided it by the RS 's fitting counts powers of p ;

$$R(C) = \frac{R(C)}{p^{RS,FC}} \quad (5)$$

where C is one of CS , and RS is one element of RSS .

- Step4: for each cell C in CS , apply LHC sampling on C with p points, run the original model to obtain response points, $\langle vals, pts \rangle$, and add the pairs to TPS and TV of the RS ; check whether the sampling points and their response values satisfy the RS function of C . If satisfied, increase the fitting counts of RS upon C ; otherwise add the RS to the poor cells RCS ;

- Step5: for each cell in RCS , carry out recursive partitioning, obtaining subsets SS of RCS ;

- Step6: update the RSS with SS .

B. Recursive domain partitioning

If the RS is unable to meet the accuracy criterion, recursive domain partitioning is needed. The partitioning function has the following steps:

function **BinarySplit**

Input: rs – response surface

Output: $RSset$ – response surface set

- Step1: split domain Ω of rs into two equal ones: Ω_1 and Ω_2 along the roughest dimension according to Eqs. (3) and (4);

- Step2: distribute the sampling data and testing data of rs into the two sub-domains according to their coordinates;

- Step3: get the corners and centers of the two sub-domains, calculates their response values through running the original model if some of them are not calculated;

- Step4: call regression procedure with key data (corners and center), sampling and testing data to generate two RS s of the two sub-domains respectively; update the sampling data with all the above data and reset the testing data to null;

- Step5: for each one RS of the two RS s, if the residue error is not accurate, call **BinarySplit** function recursively;

- Step6: record all the RS s into $RSset$.

C. Regression or interpolation

The center and the corners are key points of a domain, which should be used for RS regression. These critical data points heavily influence the accuracy of the RS model.

For a domain with p dimensions, if $p \leq 3$ and there are not enough other sampling points over the domain, the regression procedure carries out interpolation automatically. For $p > 3$ or for cases with enough sampling points over the domain for regression, the RS will be regressed.

D. Stopping criterion

Four stop criteria are used in the refining procedure of RS set:

- i) Number of iterations.
- ii) Number of runs of the original model.

- iii) Dimensions of all the RS cells; when these values reach a small threshold, all cells are small enough and the iteration terminates.
- iv) Fitting count of all the RS in RS set. When it becomes greater than a given integer M , the iteration stops. The RSs in the RS set fitted well with the sampling data and passed M random sampling tests. All these RSs are considered accurate enough.

E. About the probable worst cells

In Step 3, the possible worst cells are selected from the RS set. The roughness of a cell is evaluated using Eq. (2). However, roughness is not a unique criterion, if the RS has passed FC tests, this RS may be accurate. Eq. (5) is thus used for roughness evaluation based upon results from many numerical experiments. Cells identified by Eq. (5), will go through additional sampling tests, and only when the candidates fail the sampling test, they are considered poor cell.

F. Testing the RS

Before splitting the potential poor cells, a sampling test is carried out to verify the conditions of these cells. The commonly-used random sampling technique, Latin hyper-cubic sampling, is used. Too many or too few sampling points are both undesirable. The number of sampling points is set to be p , the dimensions of the domain.

Four criteria are used to control the sampling test on the candidate cells: absolute error, relative error, cell dimension, and RSME of the sampling data.

G. Updating the RS set

After the recursive partitioning on a cell, a new subset of RSs is generated. The updating process substitutes the cell using the new subset.

During each step of the binary splitting process one cell is divided into two cells. After each splitting, a subset including two RSs will be generated and the original one will be substituted with the subset of two RSs.

IV. COMBINATION OF RS SET

The number of resulting RS set produced during domain approximation determines the needed memory and computation cost of the metamodel. After additional sampling data points have been added to the rough set and the accuracy of the modeling has been improved, some adjacent cells can be combined to reduce needed memory and computation while keeping the modeling accuracy. A sorting scheme is introduced to speed up the cell combination process:

Step1: scan all RS in the current RS set RSS

Step2: generate two testing points from the adjacent cells and introduce a small displace δ along the positive direction of all dimensions, as shown in Fig 4.

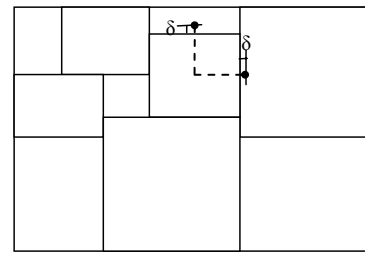


Fig.4 testing points to obtain the adjacent cells

Step3: get the two cells most likely contains the two testing points.

Step4: remove the cell(s) which has no common edge with the RS, RSsetA;

Step5: for each cell rs_i in RSsetA, if rs_i and RS_k can be combined according to 2-C, regress a new RS using the sampling and testing data upon the two cells; reset the sampling data of the combined cell and replace the two original cells with the new one; update the RSS.

Step6: if the combination process is activated, go to step1; otherwise, exit.

In the step3, we can determine whether a given point is in the RS cell according to the coordinate comparison. But because there may be many RS in the RS set, so we adopt a so-called quasi binary search to position the right RS quickly in the next part.

V. METAMODEL EVALUATION USING RS SET

Quick calculation of the model value is the main objective of the approximation using the RS set. It is straightforward to evaluate the model value of a given point for a quadratic PRS using the Eq. (1). However, when the variable domain is very large and the response surface is of high dimension, the number of the RSs in the resulting RS set may become very large, thus finding the corresponding RS model for a given point may be time-consuming, following the regular, sequential search process.

The following quasi-binary search scheme is used to quickly identify the corresponding RS for a given point:

Algorithm 1 (Binary-Positioning)

```

midlen ← length(RSset) / 2;
start ← 1 + midlen;
while loop
  RSMid ← RSset[start + midlen];
  if pt is in RSMid
    RSMid is found and return;
  elseif pt is after RSMid
    start ← start + midlen;
    midlen ← midlen / 2;
  else
    midlen ← midlen / 2;
  end if
end while loop.
end algorithm

```

During this procedure, A pt after a cell C means that the pt follows the left-corner of C , or the point with least value of all dimensions in the cell, similar to Definition 2. The algorithm is a typical binary search procedure. However, the right RS may not be found. As shown in the example of Fig. 5, point pt is after the RS cell 8, but the real RS set that we want is RS cell 5, before 8.

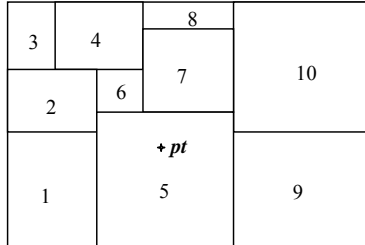


Fig.5 position the cell which contains a given point

The main reason of this problem is due to the fact that if a point pt is before a RS cell 10, the cell A containing pt is absolutely before 10; but if pt is after the cell B, A is not absolutely after B. The “absolute after” function from definition 2 is defined as:

Definition 3: for a point pt and a cell C ,
 $pt \geq C$

when $pt(j) \geq C.XL(j)$, where $1 \leq j \leq p$.

So we modify the binary search algorithm as following:

Algorithm 2 (Quasi_Binary_Positioning)

$midlen \leftarrow \text{length}(RSset) / 2;$

$start \leftarrow 1 + midlen;$

$nFrom \leftarrow 1;$

while loop

$RSMid \leftarrow RSset[start + midlen];$

if $(start + midlen > len)$ **or** $(midlen \leq 1)$

no results found and break;

if pt is in $RSMid$

$RSMid$ is found and return;

elseif pt is before $RSMid$

$nTo \leftarrow start + midlen;$

$midlen \leftarrow midlen / 2;$

else

if pt is absolutely after $RSMid$

$nFrom \leftarrow start + midlen;$

end if absolutely after

$start \leftarrow start + midlen;$

$midlen \leftarrow midlen / 2;$

end if

end of while loop.

if not found yet

Search from $nFrom$ **to** nTo **cells one-by-one;**

end if

end algorithm

From this algorithm, sometimes the result cell can be found during the binary search process. Even if no results are found during the binary search, the scope of search can be reduced.

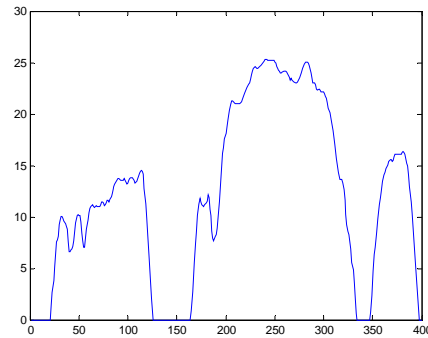
The next one-by-one search can be carried out from $nFrom$ to nTo after the binary search.

VI. CASE STUDIES

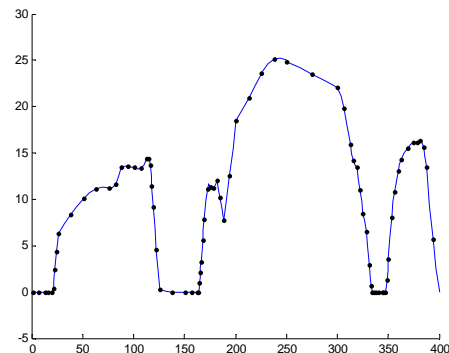
A. Approximation for 1-D data model

As shown in Fig. 6 shows, the Urban Dynamometer Driving Schedule (UDDS) vehicle driving cycle, mandated by the US Environmental Protection Agency (EPA) for vehicle dynamometer test is represented as a statistically obtained 1-D velocity-time curve. The UDDS driving cycle represents typical city driving conditions for light duty vehicles. Normally, the driving cycles are recorded as a look-up table. There are 400 data points to describe the driving cycle, as shown in Fig 6.

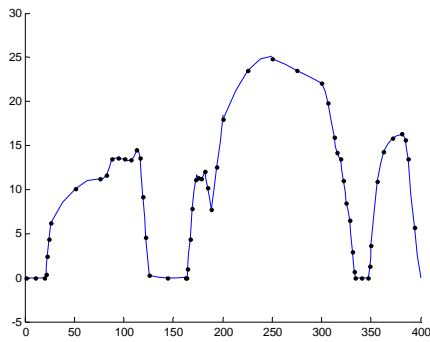
UDDS) vehicle driving cycle is used as input to vehicle performance modeling and simulation. Now, we use it to demonstrate use of the adaptive response surface domain decomposition to approximate the given drive cycle model-data. During the initial domain decomposition, the approximate RS set contains 39 cells, as shown in Fig 6 (b); and after executing the re-composing algorithm, with RSME = 0.1, the result RS set (29 cells) is shown in Fig 6 (c). Now, the number of the data points which can approximately record the original driving cycle is $29 \times 2 + 1 = 59$, which is much less than the original number 400.



(a) 1-D driving cycle – UDDS curve



(b) Approximate result after decomposing



(c) Approximate result after re-uniting

Fig.6 approximation for 1-D problem

Note that the result may be different with different runs because of the LHC random sampling.

B. Approximation for 2-D function

Two commonly used benchmark mathematical (2-D) functions for testing global modeling/optimization algorithms are also used to verify the proposed algorithm.

1) Schaffer's Function

The approximation result of the Schaffer's function is shown in Fig. 7 (41 cells with RSME=0.0132)

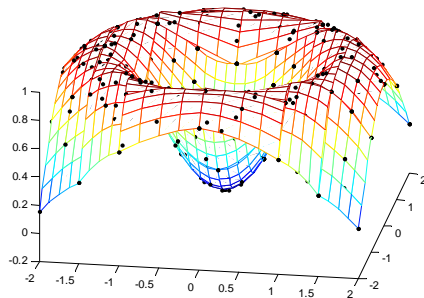


Fig.7 Approximation for Schaffer's Function

2) Rastrigin Function

$$f(x_1, x_2) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2)$$

$$x_1, x_2 \in [-1, 1]$$

The approximation result of the Rastrigin function is shown in Fig. 8. (470 cells with RSME=0.0743)

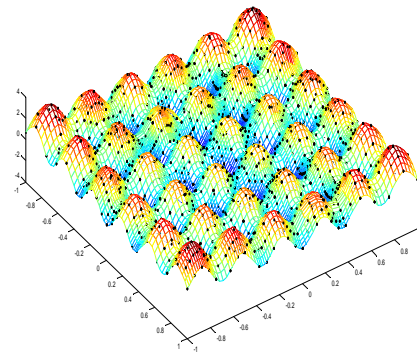


Fig.8 The approximation for the Rastrigin function

3) Peaks Function $x = [-3 -3; 3 3]$

$$f(x, y) = 3(1-x)^2 e^{-(x^2-(y+1)^2)} - 10(x/5 - x^3 - y^5) e^{(-x^2-y^2)} - \frac{1}{3} e^{(-(x+1)^2-y^2)}$$

The approximation result of the Peaks function is shown in Fig. 9. (139 cells with RSME=0.062)

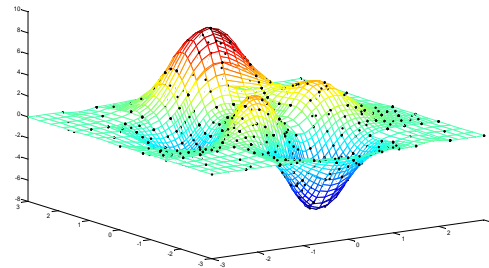


Fig.9 The approximation for the Peaks function

C. Approximation for multi-dimension function

Two other commonly benchmark mathematical functions for testing multi dimensional, global modeling/optimization algorithms are also used to verify the proposed algorithm.

1) Butterfly-like function in [29]

$$f(x, y, z) = 7 \frac{\sin(\sqrt{x^2 + y^2}) + 10^{-7}}{\sqrt{x^2 + y^2}} + 3|x - y|^{1/2} + 0.01z$$

$$x, y \in [-8, 8]; z \in [0, 2]$$

The approximation result for this three-dimension function is: Cells = 608, RSME=0.202. If we set the value of Z to 0, then the result can be shown in Fig. 10. (Cells=201, RSME=0.236)

2) Alpine function ($p=5$)

$$f(x_1, x_2, \dots, x_p) = \sin x_1 \cdot \sin x_2 \cdot \dots \sin x_p \sqrt{x_1 x_2 \dots x_p}$$

$$x_1, x_2, \dots, x_p \in [0, 10]$$

The approximation result for this five-dimensional function is: Cells = 59049, RSME=0.097.

When $x_3 = x_4 = \dots = x_p = \frac{\pi}{2}$, the result can be shown in

Fig. 11. (Cells = 237, RSME = 0.0551)

VII. APPLICATIONS: APPROXIMATION FOR A POWERTRAIN MODEL

During the development of new hybrid electric vehicles, high fidelity performance model of the hybrid powertrain is the foundation for both powertrain optimization in vehicle design and the optimal control during vehicle operation, to achieve best fuel economy and minimum emission. The multi-physics model is normally built using MATLAB Simulink tools or special vehicle powertrain simulation software built on top of multi-physics modeling and simulation package. In all cases, the modeling and simulation are implicit, complex, computationally intensive and high nonlinear. To carry out optimal control in real time, the complex hybrid powertrain model needs to be evaluated quickly. For vehicle powertrain design optimization, a large number of evaluations of the hybrid powertrain model are required in searching for the design optimum.

A quick, efficient and accurate metamodel of the original complex hybrid powertrain model can facilitate the design optimization and optimal control tasks. At present, Determinative Dynamic Program (DDP) is commonly used. The powertrain performance simulation model is called repeatedly to calculate the vehicle performance parameters based upon given vehicle input variables (control variables and state variables) for each small given time step, for instance, one second. To cover a driving cycle or continuous operation, the approach is inevitably calculation intensive. In this work, the newly introduced metamodeling method is introduced to address the problem. The method is demonstrated here using the Toyota Prius hybrid powertrain model example. The MATLAB Simulink Prius powertrain model that consists of models of one internal combustion engine (ICE) and two motors/generators (M/G) and the power split hybrid architecture is shown in Fig.12.

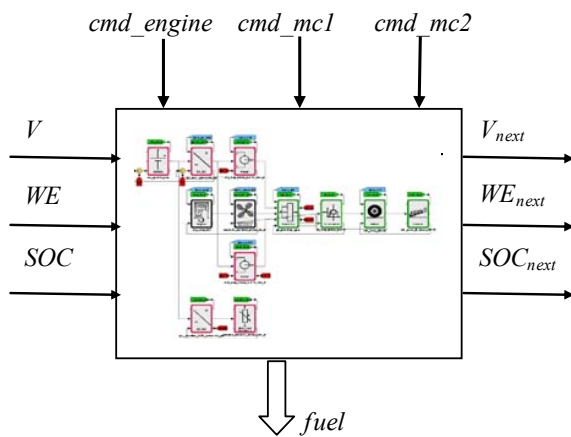


Fig.12 Powertrain control model of Prius

There are four system outputs in the model: *fuel* –fuel consumed, V_{next} –vehicle speed of the next time step, WE_{next} – engine speed of the next time step, SOC_{next} –state of charge of battery of the next time step. The system inputs include three control variables (*cmd_engine* - throttle of engine, command of the two motors: *cmd_mc1* and *cmd_mc2*) and three state variables (V – current vehicle speed, WE – current engine speed, SOC –current SOC of the battery).

For the four output variables, four corresponding response surface sets are generated respectively with respect to the same six inputs. Using the stopping criterion with the maximum iterations $N = 100$, we have obtained four different RS sets of the outputs: *fuel*, V_{next} , WE_{next} and SOC_{next} . The number of formed modeling cells and the RSME are given in Table 1. The metamodel supports the calculation of the vehicle performance parameters under different conditions quickly and accurately.

TABLE I
APPROXIMATION RESULT OF POWERTRAIN CONTROL MODEL OF PRIUS

	<i>fuel</i>	V_{next}	WE_{next}	SOC_{next}
Cells	1293	4323	3234	2344
RSME	1E-6	0.10	10	0.001

VIII. DISCUSSION AND CONCLUSION

Global approximation using metamodel for complex mathematical function or computer model over a large variable domain is often needed in sensibility analysis, computer simulation, optimal control, and global design optimization of complex, multi-physics systems.

Today, design optimization of complex mechanical and vehicular systems requires repeated evaluation of the computationally extensive system performance model and simulation, demanding an effective metamodeling method to access the system performance quickly and accurately over the entire variable domain. This need of high fidelity and highly efficient global approximation method led to this work on adaptive polynomial RSM.

The proposed approximate method is able to meet the five requirements: satisfactory modeling accuracy, low model update computation time, modest needed memory space for holding the model, and fast model evaluation. To overcome the drawbacks of the existing RS modeling methods for complex models over large variable domain, the new adaptive and regressive RS modeling method uses new quadratic function and local area model improvement schemes. The method applies an iterative and Latin hypercube sampling based RS update process, divides the entire domain of design variables into multiple cells, identifies rougher cells with large modeling error, and further divides these cells along the roughest dimension direction. A small number of additional sampling points from the original, expensive model are added over the small and isolated rough cells to improve the RS model locally until the model accuracy criteria are satisfied. The method then combines local RS cells to regenerate the global RS model with satisfactory accuracy. An effective RS cells sorting algorithm is also introduced to improve the efficiency of model evaluation.

Benchmark tests are presented and use of the new metamodeling method to replace complex hybrid electrical vehicle powertrain performance model in vehicle design optimization and optimal control are discussed.

The new method also tolerates few bizarre sampling data since regression instead of interpolation is used, and utilizes all available sampling data, obtained through expensive calculations, in forming the metamodel. The comprehensive stopping criteria make recursive partitioning and global modeling. The decomposition method can also be used in other approximation techniques such as Kriging or RBF.

At present there still exist some drawbacks in the method. The metamodel is discontinuous at the sub-region boundaries due to regression. It cannot guarantee that all the points on the edge are continuous. Another research might be done in developing a mechanism to handle multiple outputs using all sampling data to generate each RS set in parallel. Further improvement of the technique may also include searching for the optimal split position along the roughest dimension instead of binary splitting; and heuristically sampling using the history sample data instead of LHD in random.

ACKNOWLEDGEMENTS

This work was supported by the National High Technology Research and Development Program of China (863 Program) under the grant of No. 2006AA04Z121, the National Natural Science Foundation of China under the grant of No. 50775084, as well as supports from Natural Science and Engineering Research Council (NSERC) of Canada, China Scholarship Council and University of Victoria. We are also grateful to the anonymous referees for their insightful comments.

REFERENCES

- [1] Surrogates Toolbox, <http://fchegury.googlepages.com>
- [2] G. Matheron, "Principles of Geostatistics," *Economic Geology*, vol. 58, pp. 1246–1266, 1963.
- [3] Sacks, J., Welch, W. J., Mitchell, T.J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989.
- [4] Cressie, N., "Spatial Prediction and Ordinary Kriging," *Math. Geol.*, vol. 20, no. 4, pp. 405–421, 1988.
- [5] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, "DACE - A MATLAB Kriging Toolbox," Technical Report IMM-TR-2002-12, Informatics and Mathematical Modelling, Technical University of Denmark, 2002.
- [6] Hardy, R. L., "Multiquadratic Equations of Topography and Other Irregular Surfaces," *J. Geophys. Res.*, vol. 76, pp. 1905–1915, 1971.
- [7] M. D. Buhmann, *Radial Basis Functions*, Cambridge University Press, Cambridge, UK, 2003.
- [8] Martin, J. D., and Simpson, T. W., "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models," *Proceedings of 2003 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, Illinois USA, 2003.
- [9] Smith, M., "Neural Networks for Statistical Modeling," *Von Nostrand Reinhold*, 1993.
- [10] Box, G. E. P., and Wilson, K. B., "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society. Series B Methodological*, vol. 13, no. 1, pp. 1–45, 1951.
- [11] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters*, John Wiley & Sons, New York, 1978.
- [12] Myers, R. H., and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, INC., Toronto, 2002.
- [13] S. R. Gunn, "Support Vector Machines for Classification and Regression," Technical Report, Image Speech and Intelligent Systems Research Group, University of Southampton, UK, 1997.
- [14] Vapnik, V., Golowich, S., and Smola, A., "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing," *Advance in Neural Information Processing Systems*, MIT Press, Cambridge, MA, pp. 281–287, 1997.
- [15] Prakasvudhisarn, C., Trafalis, T. B., and Raman, S., "Support Vector Regression for Determination of Minimum Zone," *ASME J. Manuf. Sci. Eng.*, vol. 125, no. 4, pp. 736–739, 2003.
- [16] S. M. Clarke, J. H. Griebisch, and T. W. Simpson, "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," *Journal of Mechanical Design*, vol. 127, pp. 1077–1087, 2005.
- [17] Friedman, J. H., "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.
- [18] Chen, V. C. P., Ruppert, D., and Shoemaker, C. A., "Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming," *Oper. Res.*, vol. 47, pp. 38–53, 1999.
- [19] Scott Crino, Donald E. Brown, "Global Optimization With Multivariate Adaptive Regression Splines," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 37, no. 2, pp. 333–340, April 2007.
- [20] Steven Richardson, Song Wang, Les S. Jennings, "A Multivariate Adaptive Regression B-Spline Algorithm (BMARS) for Solving a Class of Nonlinear Optimal Feedback Control Problems," *Automatica*, vol. 44, pp. 1149 – 1155, 2008.
- [21] Gu, J., Li, G. Y., and Dong, Z., "Hybrid and Adaptive Metamodel Based Global Optimization," *Proc. Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, IDETC/CIE 2009, DETC2009-87121*, San Diego, California, USA, Aug 30 - Sep 2, 2009.
- [22] S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *Aspects of the Matlab Toolbox DACE. Report IMM-REP-2002-13, Informatics and Mathematical Modelling, DTU. (2002)*, 44 pages, available at <http://www.imm.dtu.dk/~hbn/publ/TR0213.ps>
- [23] Younis, A., Gu, J., Dong, Z., and Li, G., "Trends, Features, and Tests of Common and Recently Introduced Global Optimization Methods," *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and*

- Optimization Conference, Victoria, British Columbia Canada, Paper No. AIAA-2008-5853, 2008.
- [24] James R. Simpson, Drew Landman, Rupert Giroux et al, Adapting Second-order Response Surface Designs to Specific Need. *Quality and Reliability Engineering International*. Qual. Reliab. Engng. Int., vol. 24, pp. 331–349, 2007.
- [25] Xuan Son Nguyen, Alain Sellier and Frédéric Duprat. “Adaptive Response Surface Method Based on a Double Weighted Regression Technique,” *Probabilistic Engineering Mechanics*, vol. 24, pp. 135–143, 2009.
- [26] Armin Iske and Jeremy Levesley. “Multilevel Scattered Data Approximation by Adaptive Domain Decomposition,” *Numerical Algorithms*, vol. 39, pp. 187–198, 2005.
- [27] Tobin A. Driscoll, Alfa R.H. Heryudono. “Adaptive Residual Subsampling Methods for Radial Basis Function Interpolation and Collocation Problems,” *Computers and Mathematics with Applications* vol. 53, pp. 927–939, 2007.
- [28] Hongbing Fang and Mark F. Horstemeyer. “Global Response Approximation with Radial Basis Functions,” *Engineering Optimization*. vol. 38, no. 4, pp. 407–424, June 2006.
- [29] Daniel Busby, Chris L. Farmer, and Armin Iske. “Hierarchical Nonlinear Approximation for Experimental Design and Statistical Data Fitting,” *SIAM J. Sci. Comput.* Vol. 29, No. 1, pp. 49–69.
- [30] Jay D. Martin. “Computational Improvements to Estimating Kriging Metamodel Parameters,” *Journal of Mechanical Design*. vol. 131 / 084501-1, August 2009.
- [31] Jack P.C. Kleijnen. “Kriging Metamodeling in Simulation: A Review,” *European Journal of Operational Research*, Vol. 192, pp. 707–716, 2009.
- [32] D. Shahsavani a, and A. Grimvall. “An Adaptive Design and Interpolation Technique for Extracting Highly Nonlinear Response Surfaces from Deterministic Models,” *Reliability Engineering and System Safety*, vol. 94, pp. 1173–1182, 2009.