J Math Chem (2012) 50:1711–1726
DOI 10.1007/s10910-012-0001-2

ORIGINAL PAPER

Multivariate data modelling through Piecewise generalized HDMR method

M. Alper Tunga · Metin Demiralp

Received: 11 January 2012 / Accepted: 23 March 2012 / Published online: 5 April 2012

Abstract This work aims to develop a new High Dimensional Model Representation (HDMR) based method which can construct an analytical structure for a given multivariate data modelling problem. Modelling multivariate data through a divide-and-conquer method stands for multivariate data partitioning process in which we deal with a number of less variate data sets instead of a single N dimensional problem. Generalized HDMR is one of these methods used to model a multivariate data set which has a number of scattered nodes with associated function values. However, Generalized HDMR includes a linear equation system with huge number of unknowns and equations to be solved. This equation sometimes has linearly dependent equations in it and this is an undesirable situation. This work offers a new method named Piecewise Generalized HDMR method which bypasses this disadvantage as well as reducing the mathematical complexity and CPU time needed to complete the algorithm of the previous method. Our new method splits the given problem domain into subdomains, applies the Generalized HDMR philosophy to each subdomain and superpositions the information coming from these subdomains. The algorithm of this new method and a number of numerical implementations are given in this paper.

Keywords High dimensional model representation · Multivariate data modelling · Interpolation · Multidimensional problems · Approximation

M. A. Tunga (✉)
Software Engineering Department, Faculty of Engineering, Bahçeşehir University,
Beşiktaş, 34349 Istanbul, Turkey
e-mail: alper.tunga@bahcesehir.edu.tr

M. Demiralp
Informatics Institute, Computational Science and Engineering Program, İstanbul Technical University,
Maslak, 34469 Istanbul, Turkey
e-mail: metin.demiralp@be.itu.edu.tr

1 Introduction

High Dimensional Model Representation (HDMR) is a divide-and-conquer method and was first proposed by I. M. Sobol in 1993 [1]. HDMR has a finite expansion including the summation of a constant, N number of univariate components, $N(N-1)/2$ number of bivariate components and so on which has a total of 2^N components. In literature, scientists usually use the very first few components, going at most to the bivariate ones to approximately represent a multivariate problem in terms of a number of less-variate ones.

After Sobol's work, H. Rabitz and his group have developed HDMR based methods and used these methods in different areas of engineering problems [2–4]. During the same time period, M. Demiralp and his group worked on different problems and developed many HDMR based methods for solving these problems [5–10]. Many other scientists from different research areas work on HDMR and use it for various scientific problems [11–20].

This representation technique is used as a data partitioning method in multivariate data modelling problems also [21]. When an N dimensional data modelling problem ($N > 3$) is given and the purpose is to construct an analytical structure for that problem, HDMR allows us to partition the given N dimensional data set into a constant, N number of univariate data sets and so on. The data partitioning process through an HDMR based method provides us with a method to deal with a number of less-variate interpolations such as univariate ones instead of an N dimensional interpolation and finally to determine the target analytical structure. This philosophy can be applied to any problem of mathematics, chemistry, physics or of any area in which a data partitioning process is needed to model the real life problems.

Generalized HDMR method [22] is based on HDMR philosophy. HDMR based methods use a product type weight to obtain a representation for the given multivariate problem while the Generalized HDMR method uses a general weight function. The product type weight inside the HDMR algorithm allows us only to apply this method to the problems having orthogonal geometry in which the function values are known at all nodes of the whole problem domain [21]. The reason for a general type weight usage is to bypass the restrictions of an orthogonal geometry on the algorithm since a multivariate data modelling problem has a multivariate data set as a training data set which has a number of nodes with an associated function value and the total number of these nodes are very small part of the whole domain. That is, this type of a problem has a nonorthogonal geometry. Hence, we need a general weight in our algorithm and the Generalized HDMR method allows us to model these types of problems [22]. However, we know that at most univariate approximation through Generalized HDMR can be obtained for the given multivariate data modelling problem since only the general structure of univariate Generalized HDMR components are given in literature and we know that it is not easy to develop a general structure for the higher variate components since that structure will have an equation system of integral equations [22]. However, the general structure of univariate Generalized HDMR components include a system of linear equations in which the unknowns are the univariate components of the multivariate function under consideration. Finding a unique solution for this type of a system is not usually easy and sometimes it is

not possible to get a unique solution because of the number of linearly dependent equations included by the system [22].

The main purpose of this work is to bypass the disadvantage of solving a system of linear equations and to determine an approximation through Generalized HDMR which has better presentation than the classical Generalized HDMR method for a multivariate data modelling problem. The proposed method of this paper, which is named as Piecewise Generalized HDMR, splits the given N dimensional domain into many N dimensional subdomains and then evaluates the constant Generalized HDMR component in each subdomain. Finally, an analytical structure is determined approximately by using the coordinates of each subdomain with the associated constant component value through a multivariate interpolation technique. The very beginning steps of such an algorithm are given for univariate data modelling problems in [23].

Another HDMR based data modelling method is Indexing HDMR [24,25]. This method constructs an index space to obtain an orthogonal structure to have the ability of using the plain HDMR method directly for data partitioning purpose. The method makes a one-to-one matching between the original problem domain and the index space. To evaluate the unknown function values of the given problem, the method needs to use similarity metrics to specify the location of each testing node in the index space [24]. The only disadvantage of this method occurs when the similarity metric chosen for the given problem fails and the appropriate index node for each testing node cannot be determined as successfully as it is expected. The main advantage of the method given in this work is the usage of the original problem domain in modelling process.

In this work, first the details of the Generalized HDMR method and data partitioning procedure through this method are given very briefly in Sects. 2 and 3, respectively. Section 4 covers our proposed method, Piecewise Generalized HDMR method, while a numerical example with important steps of this new method is given in the computational procedure section as Sect. 5 of the manuscript. A number of numerical implementations to show the performance of the new method are designed in Sect. 6. Finally, Sect. 7 is about the concluding remarks on the proposed method.

2 The generalized HDMR method

The expansion of HDMR for a multivariate function is given as follows [1]

$$f(x_1,\ldots,x_N) = f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \sum_{\substack{i_1,i_2=1 \\ i_1<i_2}}^{N} f_{i_1 i_2}(x_{i_1},x_{i_2}) + \cdots + f_{1\ldots N}(x_1,\ldots,x_N) \tag{1}$$

where N is the number of independent variables of the function under consideration. The main task of the HDMR based algorithms is to uniquely determine the right hand side components of this expansion. A product type weight function is needed for this determination process in HDMR philosophy [5,21]. However, to have the ability of dealing with scattered data, the Generalized HDMR method uses a general type weight

with a product type auxiliary weight function [22]. In this sense, we need to obtain the HDMR components of that general weight in order to evaluate the general structures of the Generalized HDMR components of the multivariate function, $f(x_1, \ldots, x_N)$ given in (1). For this purpose, the HDMR expansion of the general weight can be written as follows

$$W(x_1, \ldots, x_N) = W_0 + \sum_{i_1=1}^{N} W_{i_1}(x_{i_1}) + \sum_{\substack{i_1, i_2=1 \\ i_1<i_2}}^{N} W_{i_1 i_2}(x_{i_1}, x_{i_2}) + \cdots + W_{1\ldots N}(x_1, \ldots, x_N) \tag{2}$$

The following product type auxiliary weight function is defined to determine the components of the general weight with normalization criterion which helps us to easily determine these components [22]

$$\Omega(x_1, \ldots, x_N) \equiv \prod_{j=1}^{N} \Omega_j(x_j), \quad \int_{a_j}^{b_j} dx_j \Omega(x_j) = 1, \quad 1 \le j \le N \tag{3}$$

The following vanishing conditions are defined to obtain the HDMR components of the general weight function

$$\int_{a_{i_j}}^{b_{i_j}} dx_{i_j} \Omega_{i_j}(x_{i_j}) W_{i_1 \ldots i_k}\left(x_{i_1}, \ldots, x_{i_k}\right) = 0, \quad 1 \le i_j \le i_k \tag{4}$$

while the vanishing conditions to determine the components of the mutivariate function given in (1) are as follows [22]

$$\int_{a_1}^{b_1} dx_1 \cdots \int_{a_N}^{b_N} dx_N \Omega(x_1, \ldots, x_N) W(x_1, \ldots, x_N) f_i(x_i) = 0, \quad 1 \le i \le N \tag{5}$$

Using the general and auxiliary weight functions, the components of the multivariate function can be determined by taking the vanishing conditions given in (4) and (5) into consideration. For this purpose, the following operator is defined to obtain the constant HDMR component

$$\mathcal{I}_0 F(x_1, \ldots, x_N) \equiv \int_{a_1}^{b_1} dx_1 \Omega_1(x_1) \cdots \int_{a_N}^{b_N} dx_N \Omega(x_N) F(x_1, \ldots, x_N) \tag{6}$$

while the univariate components can be obtained through the following operator

$$\mathcal{I}_i F(x_1,\ldots,x_N) \equiv \int_{a_1}^{b_1} dx_1 \Omega_1(x_1) \cdots \int_{a_{i-1}}^{b_{i-1}} dx_{i-1}\Omega_{i-1}(x_{i-1}) \times \int_{a_{i+1}}^{b_{i+1}} dx_{i+1}\Omega_{i+1}(x_{i+1}) \cdots \int_{a_N}^{b_N} dx_N \Omega_N(x_N) F(x_1,\ldots,x_N) \tag{7}$$

where $1 \le i \le N$ and $F(x_1,\ldots,x_N)$ is an arbitrary square integrable multivariate function.

When we apply the operator $\mathcal{I}_0$ to both sides of the expansion given in (1) the general relation is obtained as follows [22]

$$\mathcal{I}_0\,[\,W(x_1,\ldots,x_N) f(x_1,\ldots,x_N)\,] = \mathcal{I}_0 \left[\left(W_0 + \sum_{i_1=1}^{N} W_{i_1}(x_{i_1}) + \cdots \right) \times \left(f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \cdots \right) \right] \tag{8}$$

Using the vanishing conditions given in (4) and (5), the constant component of the HDMR expansion, which is f_0, is evaluated as

$$\mathcal{I}_0\,[\,W(x_1,\ldots,x_N) f(x_1,\ldots,x_N)\,] = W_0 f_0 \tag{9}$$

and when the normalization criterion given in (3) is taken into consideration, we get the result, $W_0 = 1$. Hence, the general structure of the constant component is obtained as follows [22]

$$f_0 = \mathcal{I}_0\,[\,W(x_1,\ldots,x_N) f(x_1,\ldots,x_N)\,] \tag{10}$$

The same philosophy can be used to determine the univariate components. The operator $\mathcal{I}_{i_1}$ is applied to both sides of the HDMR expansion under the vanishing conditions and the final relation for the univariate components is obtained as

$$\begin{aligned}
&\mathcal{I}_i\,[\,W(x_1,\ldots,x_N) f(x_1,\ldots,x_N)\,] \\
&\quad = (1 + W_i(x_i))\, f_0 + (1 + W_i(x_i))\, f_i(x_i) \\
&\qquad + (1 + W_i(x_i)) \sum_{\substack{i_1=1 \\ i_1 \ne i}}^{N} \int_{a_{i_1}}^{b_{i_1}} dx_{i_1} \Omega_{i_1}(x_{i_1}) \left(1 + W_{i_1}(x_{i_1})\right) f_{i_1}(x_{i_1}) \\
&\qquad + \sum_{\substack{i_1,i_2=1,i_1<i_2 \\ (i_1=i)\vee(i_2=i)}}^{N} \int_{(1-\delta_{i_1 i})a_{i_1}+\delta_{i_1 i}a_{i_2}}^{(1-\delta_{i_1 i})b_{i_1}+\delta_{i_1 i}b_{i_2}} \left[\left(1-\delta_{i_1 i}\right) dx_{i_1} \Omega_{i_1}(x_{i_1}) + \delta_{i_1 i} dx_{i_2} \Omega_{i_2}(x_{i_2}) \right] \\
&\qquad \times \left[W_{i_1 i_2}(x_{i_1}, x_{i_2}) - W_{i_1}(x_{i_1}) W_{i_2}(x_{i_2}) \right] \left[\left(1-\delta_{i_1 i}\right) f_{i_1}(x_{i_1}) + \delta_{i_1 i} f_{i_2}(x_{i_2}) \right]
\end{aligned} \tag{11}$$

where $1 \leq i \leq N$ and $\delta_{i_1 i}$ stands for Kronecker's Delta [22]. This relation includes a set of integral equations whose unknowns are the univariate HDMR components.

3 Data partitioning through generalized HDMR

Multivariate data with its associated function value can be defined as follows

$$d_j \equiv \left(\xi_1^{(j)}, \ldots, \xi_N^{(j)}, \varphi_j\right), \quad \varphi_j \equiv f(\xi_1^{(j)}, \ldots, \xi_N^{(j)}) \quad 1 \leq j \leq m \tag{12}$$

where m is the total number of nodes of the training data set of the given problem.

Since we are dealing with multivariate data through an HDMR based method, to partition the multivariate data set into less-variate data sets, we need to specify a general weight function that can take only the given nodes and the associated function values into consideration. For this purpose, a weight function, which is a linear combination of Dirac delta functions, is selected as [22]

$$W(x_1, \ldots, x_N) \equiv \sum_{j=1}^{m} \alpha_j \delta(x_1 - \xi_1^{(j)}) \cdots \delta(x_N - \xi_N^{(j)}) \tag{13}$$

where α_j parameters are used for assigning a different importance to each individual datum. To evaluate the value of each α_j parameter, the normalization criterion given in (3) is taken into consideration and finally the following relation is used

$$W_0 = \mathcal{I}_0\,[\,W(x_1, \ldots, x_N)\,] = \sum_{j=1}^{m} \alpha_j \overline{\Omega}_j = 1 \tag{14}$$

The product type auxiliary weight function will be used implicitly as given in (3). Using the general weight given in (13) and the general structure for the constant component given in (10), the constant component for the data partitioning process is obtained as follows [22]

$$f_0 = \sum_{j=1}^{m} \alpha_j \overline{\Omega}_j \varphi_j, \qquad \overline{\Omega}_j \equiv \prod_{k=1}^{N} \Omega_k(\xi_k^{(j)}), \quad 1 \leq j \leq m \tag{15}$$

To determine the univariate components under the weight function given in (13), we obtain a system of linear equations instead of integral equations as given in (11) [22]. However, it is sometimes impossible to get a unique solution of such a system. In addition, because there may sometimes exist huge number of unknowns as univariate components and linear equations depending on the data set of the given problem, it takes too much CPU time to have an acceptable solution. For these reasons, the main purpose of this work is to bypass the univariate Generalized HDMR components and to propose a new algorithm that uses only the constant Generalized HDMR component to approximately construct an analytical structure for the given data modelling problem.

4 The Piecewise generalized HDMR method

This section includes the steps of the proposed method which is named as Piecewise Generalized HDMR. This new method aims to use only the constant component of the Generalized HDMR method to model the given multivariate problem. It is clear that a single constant value obtained for the whole problem domain does not have the ability of representing the analytical structure of the problem under consideration. Hence, instead of using the whole domain, we split the N dimensional problem domain into many N dimensional subdomains and evaluate a constant value for each subdomain through the Generalized HDMR method. Using a multivariate interpolation technique applied to these constant values to determine an approximate analytical structure for the given multivariate problem, we will have an approximation as a result at the end. This philosophy constructs the Piecewise Generalized HDMR method and bypasses the disadvantages of solving a linear equation system coming from classical univariate Generalized HDMR approximation in which there may exist linearly dependent equations that cause having no unique solution.

A multivariate data modelling problem can be defined through a relation as given in (12) and the domain of each independent variable of such a problem is described as follows

$$x_i \in [a_i, b_i], \qquad 1 \le i \le N \tag{16}$$

where each a value stands for the minimum value and each b value stands for the maximum value that the related independent variable can take.

To split the problem domain into subdomains first we split each interval, which describes the domain of each independent variable, into subintervals

$$\begin{gathered} x_i^{(1)} \in \left[c_i^{(1)}, c_i^{(2)}\right), \quad x_i^{(2)} \in \left[c_i^{(2)}, c_i^{(3)}\right), \quad \cdots, \quad x_i^{(t_i)} \in \left[c_i^{(t_i)}, c_i^{(t_i+1)}\right], \\ c_i^{(1)} \equiv a_i, \quad c_i^{(t_i+1)} \equiv b_i, \quad 1 \le i \le N \end{gathered} \tag{17}$$

where t_i is the number of subintervals assigned to the ith independent variable, c values are the lower and the upper bounds of each subinterval. The value of each t_i will be chosen by the user and the subintervals of each independent variable will be equivalent subintervals in this work. Of course, the selection process of the number of subintervals and the properties of these subintervals can be organized by using different methods. These can be the subject of a future work.

To construct subdomains of the whole problem domain, the cartesian product of the subintervals of the independent variables are used. This process can be expressed as follows

$$\begin{gathered} \mathcal{D}^{(s)} \equiv x_1^{(j_1)} \times x_2^{(j_2)} \times \cdots \times x_N^{(j_N)}, \\ 1 \le j_1 \le t_1, \ldots, \quad 1 \le j_N \le t_N, \quad 1 \le s \le \ell, \quad \ell \equiv t_1 \times \cdots \times t_N \end{gathered} \tag{18}$$

where ℓ is the total number of the subdomains.

Next step is to define a product type auxiliary weight in each subdomain which satisfies the criterion given in (3). In this work, the following constant auxiliary weight function structure is selected

$$\Omega(x_1,\ldots,x_N)^{(s)}=\prod_{j=1}^{N}\frac{1}{\omega_j^{(s)}-\theta_j^{(s)}},\qquad 1\le s\le \ell \tag{19}$$

where $\omega_j^{(s)}$ and $\theta_j^{(s)}$ stand for the upper and lower bounds of each independent variable of each subdomain, respectively.

Then, using relation (14), the α values should be specified. Finally, a constant Generalized HDMR component is obtained for each subdomain by using the relation given in (15).

Now, it is time to construct an analytical structure through the subdomains to obtain an approximate representation of the given multivariate data modelling problem. To this end, the first step is to evaluate the arithmetic means of the value set of each independent variable in each subdomain. This results in the following data sets

$$\left(\mu_1^{(s)},\ldots,\mu_N^{(s)},f_0^{(s)}\right),\quad 1\le s\le \ell \tag{20}$$

where $f_0^{(1)},\ f_0^{(2)},\ \ldots,\ f_0^{(\ell)}$ stand for the constant Generalized HDMR component of the first subdomain, second subdomain and finally the last subdomain, respectively. In addition, μ values are the arithmetic means of the values that each independent variable can take in the corresponding subdomain. The final step of the method is to determine an analytical structure through the abovementioned data sets. For this purpose, one can use a multivariate polynomial interpolation method.

In this work, we use MuPAD [26], which is a multi-processing algebra data tool, to develop the program codes needed for the calculations and the command *interpolate* allows us to apply the multivariate polynomial interpolation technique to the data sets given in (20).

5 Computational procedure

This section covers a numerical example to describe the important steps of the Piecewise Generalized HDMR method. For this purpose, the following testing function is selected

$$f(x_1,x_2,x_3)=\prod_{i=1}^{3}(1+x_i) \tag{21}$$

where there exist 3 independent variables. Let the domains of the independent variables of the designed example be as follows

$$\xi_1\in\{1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0\},\quad \xi_2\in\{1.5,2.5,3.5,4.5,5.5\},$$
$$\xi_3\in\{1.3,1.6,2.2,2.4,2.9\} \tag{22}$$

The number of nodes of the whole problem domain is 200 and it is assumed that the function values at 130 nodes of 200 are known in our testing implementation. The training and the testing data sets have 80 and 50 nodes, respectively.

When we apply the steps of the proposed method to the testing function within the problem domain given in (21) and (22), the results obtained with the related steps are itemized as follows:

– Specify the number of subintervals for each independent variable

$$t_1 = 3, \qquad t_2 = 2, \qquad t_3 = 2 \tag{23}$$

– Identify the subintervals of each independent variable's domain

$$\begin{gathered} x_1^{(1)} \in [1.0, 3.33)\,, \quad x_1^{(2)} \in [3.33, 5.67)\,, \quad x_1^{(3)} \in [5.67, 8.0] \\ x_2^{(1)} \in [1.5, 3.5)\,, \quad x_2^{(2)} \in [3.5, 5.5] \\ x_3^{(1)} \in [1.3, 2.1)\,, \quad x_3^{(2)} \in [2.1, 2.9] \end{gathered} \tag{24}$$

– Construct the cartesian product of these subintervals. There should be $3\times2\times2 = 12$ elements in this set. Each element stands for a subdomain of the given problem domain.

$$\begin{aligned} \mathcal{D}^{(1)} &\equiv [1.0, 3.33) \times [1.5, 3.5) \times [1.3, 2.1) \\ \mathcal{D}^{(2)} &\equiv [1.0, 3.33) \times [1.5, 3.5) \times [2.1, 2.9] \\ &\vdots \\ \mathcal{D}^{(11)} &\equiv [5.67, 8.0] \times [3.5, 5.5] \times [1.3, 2.1) \\ \mathcal{D}^{(12)} &\equiv [5.67, 8.0] \times [3.5, 5.5] \times [2.1, 2.9] \end{aligned} \tag{25}$$

– Determine the training nodes appearing in each element of this cartesian product set, that is, the nodes appearing in each subdomain.

$$\begin{gathered} (3.0, 2.5, 1.3, 32.2) \in \mathcal{D}^{(1)}, \quad \ldots, \quad (4.0, 2.5, 2.2, 56) \in \mathcal{D}^{(6)}, \quad \ldots, \\ (4.0, 4.5, 2.2, 88) \in \mathcal{D}^{(8)}, \quad \ldots, \quad (6.0, 3.5, 2.9, 122.85) \in \mathcal{D}^{(12)} \end{gathered} \tag{26}$$

– Define the auxiliary weight function as given in (19) for each subdomain.

$$\begin{aligned} \Omega(x_1, \ldots, x_N)^{(1)} &= 0.267857142857143125, \quad \ldots, \\ \Omega(x_1, \ldots, x_N)^{(7)} &= 0.267857142857141875, \quad \ldots \end{aligned} \tag{27}$$

– Evaluate α values as given in (14) for each subdomain. In this work, we set α values equal in the related subdomain.

$$\begin{aligned}
\alpha_1^{(1)} &= \cdots = \alpha_8^{(1)} = 0.466666666666666 \\
\alpha_1^{(2)} &= \cdots = \alpha_8^{(2)} = 0.466666666666666 \\
&\vdots \\
\alpha_1^{(11)} &= \cdots = \alpha_3^{(11)} = 1.24444444444444 \\
\alpha_1^{(12)} &= \cdots = \alpha_5^{(12)} = 0.746666666666666
\end{aligned} \tag{28}$$

Here the superscripts in αs stand for the subdomain number. The subscripts are the indices of the training nodes appearing in that subdomain. That is, we have an α value for each node of that subdomain.

– Determine a constant value for each subdomain by using the relation (15).

$$\begin{aligned}
f_0^{(1)} &= 21.475, \quad f_0^{(2)} = 26.99375, \quad \ldots, \\
f_0^{(11)} &= 94.7666666667, \quad f_0^{(12)} = 153.05
\end{aligned} \tag{29}$$

– Evaluate the arithmetic means of the values that each independent variable takes in the corresponding subdomain and construct a data set as given in (20).

$$\begin{aligned}
&(1.9545454546, 1.9516129032, 1.425, 21.475)\,, \\
&(1.9545454546, 1.9516129032, 2.4818181818, 26.99375)\,, \\
&\vdots \\
&(7.1052631579, 4.5408163265, 1.425, 94.7666666667)\,, \\
&(7.1052631579, 4.5408163265, 2.4818181818, 153.05)
\end{aligned} \tag{30}$$

There exist 12 nodes in this set since there are 12 subdomains.

– Interpolate these nodes to construct an analytical structure for the given data modelling problem.

– The arithmetic means of relative error values obtained for 30 randomly constructed problems by using the testing function given in (21) are as follows

$$\begin{aligned}
Training\ Part &\Longrightarrow \mathcal{N}_{s_0} = 0.1095142846, \quad \mathcal{N}_{s_1} = 0.1425208816 \\
Testing\ Part &\Longrightarrow \mathcal{N}_{s_0} = 0.1096262314, \quad \mathcal{N}_{s_1} = 0.1697872029
\end{aligned} \tag{31}$$

where s_0 and s_1 stand for constant Piecewise Generalized HDMR approximant and univariate Generalized HDMR approximant, respectively. The reason to run the random problem producing process 30 times is to examine the general tendency of these two methods for this type of an analytical structure which has a hybrid nature. It is clearly seen that our new method works better than the classical Generalized HDMR method, which is composed of univariate components, even when we use the constant approximation here.

6 Numerical implementations

In this section, several numerical implementations are constructed through a number of testing functions to examine the performance of the proposed method of this work. To obtain the numerical results, Perl programming language [27] is used to prepare the given multivariate data set and the related subdomains to be used in the Piecewise Generalized HDMR algorithm and to evaluate the constant component in each subdomain while MuPAD [26] is used for multivariate interpolation. The evaluations done through both Perl and MuPAD are within 15-digits precision while the numerical results are given to within 10-digits precision for simplicity.

The selected multivariate testing functions are as follows

$$
\begin{aligned}
f_1(x_1,\ldots,x_5) &= \sum_{i=1}^{5} x_i, & f_2(x_1,\ldots,x_5) &= \left[\sum_{i=1}^{5} x_i\right]^3,\\
f_3(x_1,\ldots,x_5) &= \left[\sum_{i=1}^{5} x_i\right]^5, & f_4(x_1,\ldots,x_5) &= \prod_{i=1}^{5} x_i,\\
f_5(x_1,\ldots,x_5) &= \mathrm{e}^{\left(\sum_{i=1}^{5} x_i\right)}, & f_6(x_1,\ldots,x_5) &= \log\left(\sum_{i=1}^{5} x_i\right),\\
f_7(x_1,\ldots,x_5) &= \cos\left(\pi \sum_{i=1}^{5} x_i\right), & f_8(x_1,\ldots,x_5) &= \cos\left(\pi \sum_{i=1}^{5} (-1)^{i+1} x_i\right),\\
f_9(x_1,\ldots,x_5) &= \sin\left(\pi \sum_{i=1}^{5} x_i\right), & f_{10}(x_1,\ldots,x_5) &= \sin\left(\pi \sum_{i=1}^{5} (-1)^{i+1} x_i\right)
\end{aligned}
\tag{32}
$$

where there exist 5 independent variables in each testing function. The testing functions have polynomial, exponential, logarithmic and trigonometric natures. In addition, the domains consisting of real values are defined as follows

$$
\begin{aligned}
\xi_1 &\in \{0.11, 0.20, 0.28, 0.31\}, \quad \xi_2 \in \{0.21, 0.25, 0.40, 0.45, 0.50\},\\
\xi_3 &\in \{0.33, 0.36, 0.39, 0.44, 0.49\}, \quad \xi_4 \in \{0.05, 0.09, 0.12, 0.17, 0.22, 0.27\},\\
\xi_5 &\in \{0.35, 0.47, 0.52, 0.56, 0.64, 0.68, 0.72, 0.77\}
\end{aligned}
\tag{33}
$$

Here, the whole grid of this type of problem includes 4,800 nodes while we assume that we know the function values at 1,000 of them and to examine the performance of our new method we construct a training data set and a testing data set having 800 and 200 nodes, respectively. The nodes of both training and testing data sets are selected randomly through a Perl script written by the authors.

The number of subintervals of each independent variable are selected as 2 and this results in 32 subdomains for each modelling problem since each has 5 independent

Table 1 The best relative error values of the approximants for the testing functions

	Training part		Testing part	
	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\overline{s}_1}$	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\overline{s}_1}$
f_1	0.0109919219	**0.0**	0.0112274833	**0.0**
f_2	**0.0325112908**	0.0388924534	**0.0336871316**	0.0476513996
f_3	**0.0596076556**	0.1078946643	**0.0647403980**	0.1474977725
f_4	**0.0611614189**	0.2407281089	**0.0653195357**	0.2766371114
f_5	**0.0185261244**	0.0225271274	**0.0191496923**	0.0260201576
f_6	0.0184182533	**0.0151683960**	0.0202359248	**0.0186316009**
f_7	**0.0905035636**	0.1625787772	**0.1024028999**	0.2005737889
f_8	**0.1010035060**	0.1600716107	**0.1100409221**	0.1856552323
f_9	**0.1038960737**	0.2322699389	**0.1108429075**	0.2344865627
f_{10}	**0.1048591760**	0.2203229438	**0.1050227575**	0.2438759154

variables. The number of nodes appearing in subdomains effect the performance of our new method since subdomains having nodes in a sparse structure cause unacceptable approximations through Piecewise Generalized HDMR. Hence, it is important to specify an appropriate subinterval number for each independent variable. This work does not include any predefined subinterval specification process in the proposed algorithm. A decision mechanism definition for this topic is left as future work, that is, in this work, this specification process is executed through the experiences learned from several numerical implementations.

The performance of the constant Piecewise Generalized HDMR approximation is compared with the univariate Generalized HDMR approximation since the aim of this work is to use only the constant component in Piecewise Generalized HDMR and we can use at most univariate components in the modelling process through the Generalized HDMR method. For this purpose, 30 different modelling problems are constructed randomly for each multivariate testing function.

Table 1 includes the best relative error values obtained through the constant Piecewise Generalized HDMR approximant, s_0, and the univariate Generalized HDMR approximant, $\overline{s}_1$ for the training and the testing parts of the problem, respectively. The boldface highlighted values in Table 1 stand for the best result obtained for the multivariate problem under consideration.

Table 2 consists of arithmetic means of relative error values obtained through 30 randomly constructed modelling problems. The results of Table 2 are given for the training and the testing parts obtained through the constant Piecewise Generalized HDMR and the univariate Generalized HDMR approximations. The best results are given in bold.

When these two tables, Tables 1 and 2 are examined, it is clearly seen that our new method works better than the older one for almost all testing functions except functions having purely additive or logarithmic nature. This means that now we have the ability of bypassing the disadvantages of classical Generalized HDMR method

Table 2 Arithmetic means of relative error values obtained for randomly constructed 30 multivariate problems

	Training part		Testing part	
	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\bar{s}_1}$	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\bar{s}_1}$
f_1	0.0136005169	**0.0**	0.0149067626	**0.0**
f_2	**0.0442491193**	0.0479828619	**0.0451884669**	0.0543374863
f_3	**0.0840845905**	0.1374036419	**0.0889496653**	0.1547915059
f_4	**0.0849817164**	0.2779633469	**0.0920292074**	0.2903352615
f_5	**0.0254465190**	0.0255362150	**0.0267617939**	0.0277463250
f_6	0.0264971042	**0.0180839432**	0.0269884863	**0.0205135060**
f_7	**0.1175592924**	0.2014863179	**0.1179943688**	0.2162797644
f_8	**0.1143780378**	0.1957304465	**0.1390591119**	0.2306487978
f_9	**0.1145178785**	0.2640597012	**0.1289285875**	0.2675223927
f_{10}	**0.1136234318**	0.2425035407	**0.1174034322**	0.3017587626

Table 3 Standard deviations of relative error values obtained for randomly constructed 30 multivariate problems

	Training part		Testing part	
	σ_{s_0}	$\sigma_{\bar{s}_1}$	σ_{s_0}	$\sigma_{\bar{s}_1}$
f_1	0.0017210622	0.0	0.0022270719	0.0
f_2	0.0047254780	0.0046854751	0.0080472807	0.0021862020
f_3	0.0121135434	0.0105582739	0.0153428464	0.0043137058
f_4	0.0172878899	0.0183765665	0.0202931827	0.0089198164
f_5	0.0035788492	0.0015314992	0.0047565953	0.0011041052
f_6	0.0039544633	0.0013622409	0.0030395826	0.0008066659
f_7	0.0130928187	0.0220949536	0.0082133786	0.0112058423
f_8	0.0078205979	0.0131263630	0.0172688205	0.0254884261
f_9	0.0069457151	0.0096310899	0.0113841212	0.0189647284
f_{10}	0.0045294470	0.0090307162	0.0081369943	0.0243044198

coming from the determination process of univariate components which stands for finding a unique solution for a linear equation system.

The results given in Table 3 correspond to standard deviation of relative error values obtained in each modelling problem and they show us that the relative error values obtained for 30 randomly constructed modelling problems are very close to each other since the standard deviation values are very close to 0. That is, our proposed method works stable. In addition, besides reducing the mathematical complexity of Generalized HDMR, we reduce the CPU time needed to complete the modelling process.

Table 4 shows us these CPU times in seconds. It is clear that our new method works faster than the classical Generalized HDMR method. This difference in the speed of

Table 4 CPU time in seconds needed to complete the modelling of each mutivariate modelling problem through Piecewise generalized HDMR and generalized HDMR

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
t_{s_0}	1.323	1.336	1.336	1.338	1.343	1.340	1.348	1.344	1.378	1.352
$t_{\bar{s}_1}$	4.504	4.567	4.588	4.546	4.523	4.561	4.543	4.572	4.526	4.542

Table 5 Numerical results for the testing function, $f_{11}(x_1, \ldots, x_6)$

	Training part		Testing part		CPU time	
	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\bar{s}_1}$	$\mathcal{N}_{s_0}$	$\mathcal{N}_{\bar{s}_1}$	t_{s_0}	$t_{\bar{s}_1}$
Case 1	**0.0424525867**	0.1631813850	**0.0433984293**	0.1646072780	**17.731**	96.740
Case 2	**0.0249888157**	0.1619353364	**0.0278418769**	0.1626710704	**34.634**	351.359

these two methods can be better observed for the following testing function.

$$f_{11}(x_1, \ldots, x_6) = \prod_{i=1}^{6} x_i, \quad 1.5 < x_1 < 2.0, \quad 2.5 < x_2 < 3.0,$$
$$5.0 < x_3 < 10.0, \quad 0.5 < x_4 < 1.0, \quad 0.5 < x_5 < 4.5, \quad 0.1 < x_6 < 0.4 \quad (34)$$

There are two cases for this testing function having 6 independent variables. The first case consists of 5,000 training nodes and 2,000 testing nodes while the second case includes 10,000 training nodes and 4,000 testing nodes. The numerical results are given in Table 5. The best relative error values of the table are bold highlighted. It is seen that the best results are obtained through Piecewise Generalized HDMR. The CPU time needed to model a multivariate data modelling problem having huge number of training nodes through the Piecewise Generalized HDMR method is extremely much less than the one obtained through the classical Generalized HDMR method.

7 Concluding remarks

Generalized HDMR is an HDMR based divide-and-conquer method which aims to partition the given multivariate data set into a number of univariate data sets with a constant component and to obtain an approximate analytical structure through these partitioned data sets including the constant value for the multivariate data modelling problem that is under consideration. However, this data partitioning process through Generalized HDMR consists of a system of linear equations which has sometimes linearly dependent equations arising from the nature of the given data modelling problem. This results in an inability to find a unique solution for such a system. In addition, when the number of unknowns and equations increase rapidly, the CPU time needed to solve that system increases.

The purpose of this work is to develop a new method which has the ability to get rid of solving a linear equation system. This means that we have to bypass using the

univariate components of the HDMR expansion, that is, the constant component of the expansion is the only chance for approximating the multivariate problem. In addition, we know that a constant value cannot represent a multivariate function itself.

This work offers a new Generalized HDMR based method and this new algorithm uses only the constant component to construct an analytical model. Our new method splits the given problem domain into subdomains by creating subintervals for each independent variable's interval. Evaluating the constant component in each subdomain constructs an interpolation problem for the whole domain of the problem. In this work, MuPAD's *interpolate* command gives us the resulting analytical structure for the given multivariate data modelling problem. The relative error results given in both the computational procedure and the numerical implementations sections show us that the proposed method of this manuscript almost always works better than the classical Generalized HDMR method even when we are using only the constant component in our new method.

Splitting procedure of the interval of each independent variable, which includes the possible values that the related independent variable can take, is an important step of the Piecewise Generalized HDMR algorithm. Selecting correct number of subintervals for each independent variable's domain effects the performance of the method. In this work, we select the subinterval numbers by looking at the length of each interval. Each subdomain should have nearly equal number of nodes with respect to other subdomains to evaluate each subdomain's constant component efficiently. A criterion or an optimization process can be defined for this process as a future work.

Finally, the numerical results show us that we now have a new algorithm that bypasses the disadvantages of the Generalized HDMR method in terms of mathematical complexity and CPU time needed to complete the modelling process.

References

1. I.M. Sobol, Sensitivity estimates for nonlinear mathematical models. Math. Model. Comput. Exp. (MMCE) **1**, 407–414 (1993)
2. H. Rabitz, Ö.F. Alış, J. Shorter, K. Shim, Efficient input-output model representations. Comput. Phys. Commun. **117**, 11–20 (1999)
3. H. Rabitz, Ö. Alış, General foundations of high dimensional model representations. J. Math. Chem. **25**, 197–233 (1999)
4. G. Li, S.-W. Wang, H. Rabitz, Practical approaches to construct RS-HDMR component functions. J. Phys. Chem. A **106**, 8721–8733 (2002)
5. M. Demiralp, High dimensional model representation and its application varieties. Math. Res. **9**, 146–159 (2003)
6. B. Tunga, M. Demiralp, Hybrid high dimensional model representation approximants and their utilization in applications. Math. Res. **9**, 438–446 (2003)
7. M. Demiralp, Illustrative implementations to show how logarithm based high dimensional model representation works for various function structures. WSEAS Trans. Comput. **5**, 1339–1344 (2006)
8. İ. Yaman, M. Demiralp, A new rational approximation technique based on transformational high dimensional model representation. Numer. Algorithms **52**, 385–407 (2009)
9. B. Tunga, M. Demiralp, Constancy maximization based weight optimization in high dimensional model representation. Numer. Algorithms **52**, 435–459 (2009)
10. M.A. Tunga, M. Demiralp, A factorized high dimensional model representation on the partitioned random discrete data. Appl. Numer. Anal. Comp. Math. **1**, 231–241 (2004)
11. T. Ziehn, A.S. Tomlin, A global sensitivity study of sulfur chemistry in a premixed methane flame model using HDMR. Int. J. Chem. Kinet. **40**, 742–753 (2008)

12. T. Ziehn, A.S. Tomlin, GUI-HDMR—a software tool for global sensitivity analysis of complex models. Environ. Model. Softw. **24**, 775–785 (2009)
13. J. Sridharan, T. Chen, Modeling multiple input switching of CMOS gates in DSM technology using HDMR. Proc. Des. Autom. Test Eur. **1**(3), 624–629 (2006)
14. B.N. Rao, R. Chowdhury, Probabilistic analysis using high dimensional model representation and fast fourier transform. Int. J. Comput. Methods Eng. Sci. Mech. **9**, 342–357 (2008)
15. R. Chowdhury, B.N. Rao, Hybrid high dimensional model representation for reliability analysis. Comput. Methods Appl. Mech. Eng. **198**, 753–765 (2009)
16. M.C. Gomez, V. Tchijov, F. Leon, A. Aguilar, A tool to improve the execution time of air quality models. Environ. Model. Softw. **23**, 27–34 (2008)
17. I. Banerjee, M.G. Ierapetritou, Design optimization under parameter uncertainty for general black-box models. Ind. Eng. Chem. Res. **41**, 6687–6697 (2002)
18. I. Banerjee, M.G. Ierapetritou, Parametric process synthesis for general nonlinear models. Comput. Chem. Eng. **27**, 1499–1512 (2003)
19. I. Banerjee, M.G. Ierapetritou, Model independent parametric decision making. Ann. Oper. Res. **132**, 135–155 (2004)
20. S. Shan, G.G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct. Multidiscip. Optim. **41**, 219–241 (2010)
21. M.A. Tunga, M. Demiralp, A new approach for data partitioning through high dimensional model representation. Int. J. Comput. Math. **85**, 1779–1792 (2008)
22. M.A. Tunga, M. Demiralp, Data partitioning via generalized high dimensional model representation (GHDMR) and multivariate interpolative applications. Math. Res. **9**, 447–462 (2003)
23. M.A. Tunga, M. Demiralp, *Data Partitioning Through Piecewise Based Generalized GHDMR: Univariate Case, the 2nd International Symposium on Computing in Science and Engineering (ISCSE 2011), June 1–4, 2011* (Kuşadası, Aydın, Turkey, 2011), pp. 128–133
24. M.A. Tunga, An approximation method to model multivariate interpolation problems: indexing HDMR. Math. Comput. Model. **53**(9–10), 1970–1982 (2011)
25. M.A. Tunga, A matrix based indexing HDMR method for multivariate data modelling. J. Math. Chem. **49**(5), 1092–1114 (2011)
26. W. Oevel, F. Postel, S. Wehmeier, J. Gerhard, *The MuPAD Tutorial* (Springer, New York, 2000)
27. H.M. Deitel, P.J. Deitel, T.R. Nieto, D.C. McPhie, *How to Program Perl* (Prentice Hall, New Jersey, 2001)