Computational Complexity Investigations for High Dimensional Model Representation Algorithms Used in Multivariate Interpolation Problems

M. ALPER TUNGA
Bahçeşehir University
Department of Computer Engineering
Beşiktaş, 34100, İstanbul
TURKEY (TÜRKİYE)

METİN DEMİRALP
İstanbul Technical University
Informatics Institute
Maslak, 34469, İstanbul
TURKEY (TÜRKİYE)

Abstract: In multivariate interpolation problems, increase in both the number of independent variables of the sought function and the number of nodes appearing in the data set cause computational and mathematical difficulties. It may be a better way to deal with less variate partitioned data sets instead of an N–dimensional data set in a multivariate interpolation problem. New algorithms such as High Dimensional Model Representation (HDMR), Generalized HDMR, Factorized HDMR, Hybrid HDMR are developed or rearranged for these types of problems. Up to now, the efficiency of the methods in mathematical sense were discussed in several papers. In this work, the efficiency of these methods in computational sense will be discussed. This investigation will be done by using several numerical implementations.

Key–Words: Data Partitioning, Multivariate Approximation, High Dimensional Model Representation, Computational Complexity

1 Introduction

If the values of a multivariate function $f(x_1, ..., x_N)$ are given for only a finite number of points in the space of its arguments and it is asked to determine an analytical structure for the sought multivariate function, standard multivariate routines may become cumbersome as the dimensionality grows. This urges us to use a divide–and–conquer algorithm which approximates the function for the mentioned multivariate interpolation problems. Hence, the given multivariate data is partitioned into low-variate data and then an analytical structure is determined with the aid of these partitioned data.

For this purpose, two new data partitioning methods were developed by using the philosophy given in High Dimensional Model Representation (HDMR) method which was first proposed by I. M. Sobol in 1993[1]. The equation given by Sobol for this method is as follows.

$$f(x_1, ..., x_N) = f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) +$$

$$\sum_{\substack{i_1, i_2 = 1 \\ i_1 < i_2}}^{N} f_{i_1 i_2}(x_{i_1}, x_{i_2}) + \cdots + f_{1...N}(x_1, ..., x_N) \quad (1)$$

This expansion is a finite sum and is composed of a constant term, univariate terms, bivariate terms and so on. These are the HDMR components of a given multivariate function.

Then, several other new algorithms based on this method were proposed in more comprehensive forms for different types of engineering problems by H. Rabitz, M. Demiralp and their groups[2-11].

A multivariate function can be given by its values at a finite number of nodes of a hyperprismatic regular grid. These nodes can be represented by N tuples which are the elements of a cartesian product of the given individual sets of values for each independent variable. High Dimensional Model Representation is used to approximately partition this given multivariate data into low–variate data[7].

On the other hand, data need not to be given at all nodes of hyperprismatic regular grid. Instead, it can be given at certain randomly chosen nodes. Hence, certain level of incompleteness may be encountered in HDMR method for such data sets. This time Generalized High Dimensional Model Representation (GHDMR), which is based on the HDMR expansion, is used as a data partitioning technique[8].

At this point, the nature of the given data, in other words the nature of the sought function, and the construction features of the data set affect the behavior

of the interpolation problem and the structure of the high dimensional model representation method. To this end, HDMR or Generalized HDMR (GHDMR) can be used to partition the multivariate data and to determine an approximate analytical structure for the sought function. These methods work well for the sought functions having additive nature as a result of the additive structure of the HDMR expansion. For the sought functions having dominantly or entirely multiplicative nature Factorized HDMR (FHDMR) is used[9, 10]. Hybrid HDMR (HHDMR) method is used when the sought function has intermediate nature, that is, it has neither a dominantly additive nor a dominantly multiplicative nature[11].

These abovementioned methods were developed and published in several journals. In this work, we will discuss CPU times spent for each algorithm in different types of multivariate interpolation problems. There exists a chapter related to the numerical testing implementations for this investigation. The results are obtained by using certain program codes (scripts) written in MuPAD 4.0, Multi Processing Algebra Data tool[12, 13]. This software is developed by the MuPAD Research Group at the University of Paderborn in Germany. MuPAD is a general purpose computer algebra system for symbolic and numerical computations. Additionally, PERL Scripting Language, Practical Extraction and Report Language, is used for making the given multivariate data amenable for MuPAD program codes[14]. MuPAD program codes run in a 20 digits precision environment. These results are obtained on a PC of P-IV 2400MHz CPU speed and 512MB RAM.

2 Data Partitioning via HDMR

HDMR is constructed as an expansion for a given multivariate function such that its components are ordered starting from a constant component (zeroth order multivariance) and continuing in ascending multivariance, that is, univariate, bivariate, trivariate components and so on. The main step of the algorithm is to determine the right hand side components of the HDMR expansion given in (1). To obtain the structure of the constant term, the following operator is defined.

$$\mathcal{I}_0 F(x_1,\ldots,x_N) \equiv \int\limits_{a_1}^{b_1} dx_1 W_1(x_1) \cdots$$
$$\times \int\limits_{a_N}^{b_N} dx_N W_N(x_N) F(x_1,\ldots,x_N) \qquad (2)$$

Similarly, the following operator is defined to built a way to determine the structure of the univariate HDMR term of the given multivariate function

$$\mathcal{I}_m F(x_1,\ldots,x_N) \equiv \int_{a_1}^{b_1} dx_1 W_1(x_1) \cdots$$
$$\times \int_{a_{m-1}}^{b_{m-1}} dx_{m-1} W_{m-1}(x_{m-1})$$
$$\times \int_{a_{m+1}}^{b_{m+1}} dx_{m+1} W_{m+1}(x_{m+1}) \cdots$$
$$\times \int_{a_N}^{b_N} dx_N W_N(x_N) F(x_1,\ldots,x_N) \qquad (3)$$

where $1 \leq m \leq N$. The function, $F(x_1,\ldots,x_N)$, appearing in these two relations is an arbitrary square integrable function. When the abovementioned operators are applied to the both sides of the HDMR expansion given in (1), the structures of the constant and univariate terms are obtained[7]. Other operators can be defined in a similar philosophy to determine the structures of the other HDMR terms, such as bivariate terms and so on.

Additionally, to uniquely determine these components, the following vanishing conditions are used in evaluation of the integrals appearing in the abovementioned operators.

$$\int_{a_1}^{b_1} dx_1 \cdots \int_{a_N}^{b_N} dx_N W(x_1,...,x_N) f_i(x_i) = 0 \quad (4)$$

where $1 \leq i \leq N$. Since we need to perform a multivariate interpolation on a finite number of discrete points we can extend the domain of HDMR variables to the entire space without imposing any extra conditions. Hence, we assume that the interval for each independent variable is $(-\infty, \infty)$. It is assumed that, the structure of the function, $f(x_1,\ldots,x_N)$, is not given analytically. Instead it is specified by values on a finite number of points of the Euclidean space defined by the independent variables, $x_1,\ldots,x_N$. These points are defined through a cartesian product. For this definition, first the data of the variable x_j is defined as the following set

$$\mathcal{D}_j \equiv \left\{ \xi_j^{(k_j)} \right\}_{k_j=1}^{k_j=n_j} = \left\{ \xi_j^{(1)},\ldots,\xi_j^{(n_j)} \right\} \qquad (5)$$

where $1 \leq j \leq N$. The cartesian product mentioned above can be constructed from these sets as follows.

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N \qquad (6)$$

The weight function appearing in the vanishing conditions is assumed to be a product of univariate functions each of which depends on a different independent variable. The structure which needs to be created through the interpolation must include the values of the function $f(x_1,\ldots,x_N)$ on the given points

only. This structure can be obtained by formatting the weight function for this purpose. In this sense the necessary action is to define the weight function as a linear combination of several Dirac delta functions[15]. Hence, the following univariate weight functions are selected.

$$
W_j(x_j) \equiv \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta\left(x_j - \xi_j^{(k_j)}\right),
$$
$$
x_j \in [a_j, b_j], \qquad 1 \le j \le N \qquad (7)
$$

Using this weight function the operators mentioned in this section can be applied to the both sides of the HDMR expansion by the help of the vanishing conditions and the given multivariate data is partitioned into low-variate data sets. In this work we deal with constant, univariate and at most bivariate terms.

After several integrations a constant value, univariate partitioned data set and bivariate partitioned data set are obtained. To this end, we have a constant value, n_m ordered pairs for the univariate function $f_m(x_m)$ and $n_{m_1} n_{m_2}$ ordered pairs for the bivariate function, $f_{m_1 m_2}(x_{m_1}, x_{m_2})$[7]. Next step is to determine analytical structures for these partitioned data sets and built the HDMR expansion of the sought function by using these structures. This step will be given in the fourth section of the paper. Next section is about another data partitioning technique.

3   Data Partitioning via GHDMR

If a multivariate data is given for the determination of a multivariate function, the location of data points in hyperspace of the independent variables gains a lot of importance. If they are located at the points of a set which is constructed as a direct product of univariate sets; High Dimensional Model Representation (HDMR) can be successfully used to partition the data into less variate data. On the other hand HDMR becomes unemployable when the data are random or not given at all points of a grid which is constructed via direct product of univariate meshes due to the incompleteness of the data. Hence, for these cases, a new high dimensional model representation method is needed. Generalized High Dimensional Model Representation (GHDMR) is used for this purpose. In this method a general multivariate weight function is used instead of a product type weight function. The algorithm uses the HDMR components of this general weight function. The steps of the method include first the determination of the HDMR components of the general multivariate weight function by using a prod-

uct type auxiliary weight function.

$$
\Omega(x_1, ..., x_N) \equiv \prod_{j=1}^N \Omega_j(x_j) \qquad (8)
$$

Then, these components are employed in the formulae to obtain the GHDMR components of the given multivariate function. In this way, the multivariate data is partitioned into low variate data. Here, the constant and the univariate terms of GHDMR expansion are obtained to get an approximation. Similar operators as given in the first section are used for this purpose. This time, the integrations will be evaluated by also using the HDMR components of the general weight function under the auxiliary weight function. The following orthogonality conditions are employed in these evaluations

$$
\int_{a_1}^{b_1} dx_1 \quad \cdots \quad \int_{a_N}^{b_N} dx_N \Omega(x_1, ..., x_N)
$$
$$
\times \quad W(x_1, ..., x_N) f_i(x_i) = 0 \qquad (9)
$$

where $1 \le i \le N$. As a result constant and univariate GHDMR terms are obtained. Relation for the univariate terms correspond to an integral equation system whose unknowns are the univariate GHDMR terms[8].

When we use this method to partition the multivariate random data, the following general weight function is selected

$$
W(x_1, ..., x_N) \equiv \sum_{j=1}^m \alpha_j \delta(x_1 - x_1^{(j)}) \cdots \delta(x_N - x_N^{(j)})
$$
$$
(10)
$$

where α_j parameters are used for making it possible to give different importance to each individual datum.

Using this general weight function and the orthogonality conditions given in (9) when applying the abovementioned operators to the HDMR expansion, a constant value and a number of linear equations whose unknowns are the univariate component values at the given data points of N dimensional space are obtained. Final step of this algorithm is to determine the unknowns of this equation set. This completes the construction of the univariate components at the data points[8].

At this point, approximate analytical structure should be determined by using this partitioned data. For this purpose, Lagrange interpolation formula will be used. Next section is about this subject.

4   Interpolation

Partitioning the given multivariate data via HDMR or GHDMR a table of pairs of data can be obtained

instead of an analytical structure for the function $f_m(x_m)$. This table provides an opportunity to determine the function $f_m(x_m)$ under an assumed structure, that is, to interpolate the corresponding data. By this way, multivariate interpolation, at least for these functions, can be approximately reduced to a set of univariate interpolations. To determine the overall structure of the function, an analytical structure should be defined or a calculation rule should be imposed on the interpolation. If the function to be determined by HDMR or GHDMR is sufficiently smooth, then the function can be represented with a multinomial of all independent variables over the continuous region produced by the cartesian product of the related intervals. For this purpose, first a multinomial representation should be built for $f_m(x_m)$.

$$p_m(x_m) = \sum_{k_m=1}^{n_m} L_{k_m}(x_m) f_m\left(\xi_m^{(k_m)}\right),$$
$$\xi_m^{(k_m)} \in \mathcal{D}_m, \qquad 1 \le m \le N \qquad (11)$$

Here $L_{k_m}(x_m)$s are Lagrange coefficient polynomials[16] which are independent of the structure of the function. The structures of these polynomials are given below

$$L_{k_m}(x_m) \equiv \prod_{\substack{j=1 \\ j \ne k_m}}^{n_m} \frac{\left(x_m - \xi_m^{(j)}\right)}{\left(\xi_m^{(k_m)} - \xi_m^{(j)}\right)},$$
$$\xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \le k_m \le n_m, \quad 1 \le m \le N \quad (12)$$

As Lagrange polynomials are constructed, univariate functions given by the relation (11) are uniquely determined within continous polynomial interpolation. These functions can be considered as univariate components of HDMR or GHDMR for the multivariate function, $f(x_1, ..., x_N)$. The expansion formed by the summation of these functions and the constant term provides the following multinomial approximation which is called "Univariate Approximation".

$$s_1(x_1, ..., x_N) = f_0 + \sum_{m=1}^{N} p_m(x_m) \qquad (13)$$

Same relations for the higher variate approximations can be defined in a similar way.

5 Factorized HDMR

We have observed that the truncations of both HDMR and GHDMR work well as long as the multivariate function under consideration has additive nature. If it is completely additive then data partitioning is exact,

otherwise a certain level of truncation error is encountered. Additivity is one end of the behavior of the multivariate function. The other hand is multiplicativity where all HDMR components contribute to the function at similar orders. Therefore, truncation approximation fails to describe the multivariate function under consideration. In those cases we need to formulate a different truncation approximation which somehow takes all components of HDMR or GHDMR into consideration. First step is to write this new equality (FHDMR) for this method.

$$f(x_1, ..., x_N) = r_0 \left[\prod_{i_1=1}^{N} \left(1 + r_{i_1}(x_{i_1})\right) \right]$$
$$\times \left[\prod_{\substack{i_1, i_2=1 \\ i_1 < i_2}}^{N} \left(1 + r_{i_1 i_2}(x_{i_1}, x_{i_2})\right) \right]$$
$$\times \cdots \times \left[\left(1 + r_{1...N}(x_1, ..., x_N)\right)\right] \qquad (14)$$

The right hand side components of the above relation can be determined by making comparisons between the right hand side of equation (1) and the additive form of the right hand side in (14). To make comparisons, idempotent operators will be used as auxiliary tools. These operators satisfy the following relations

$$\mathcal{I}_j^{(id)} \mathcal{I}_k^{(id)} \equiv \mathcal{I}_k^{(id)} \mathcal{I}_j^{(id)}, \qquad \left[\mathcal{I}_j^{(id)}\right]^2 \equiv \mathcal{I}_j^{(id)} \quad (15)$$

where $j, k = 1, ..., N$. Using these operators HDMR and FHDMR expansions are replaced by the following generalized ones

$$\mathcal{S}(x_1, ..., x_N) \equiv f_0 I + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) \mathcal{I}_{i_1}^{(id)} + \cdots$$
$$\mathcal{R}(x_1, ..., x_N) \equiv r_0 \left[\prod_{i_1=1}^{N} \left(I + r_{i_1}(x_{i_1}) \mathcal{I}_{i_1}^{(id)}\right) \right]$$
$$\times \cdots \qquad (16)$$

These two entities represent the same multivariate function. Hence, their right hand sides must match for all idempotent operators. This permits us to determine the constant term, the univariate terms and higher order terms of the FHDMR expansion.

As a result, constant, univariate and bivariate FHDMR terms are obtained in terms of HDMR or GHDMR terms as follows.

$$r_0 = f_0$$
$$r_{i_1}(x_{i_1}) = \frac{f_{i_1}(x_{i_1})}{f_0}$$
$$r_{i_1 i_2}(x_{i_1}, x_{i_2}) = \frac{f_0 f_{i_1 i_2}(x_{i_1}, x_{i_2}) - f_{i_1}(x_{i_1}) f_{i_2}(x_{i_2})}{(f_0 + f_{i_1}(x_{i_1}))(f_0 + f_{i_2}(x_{i_2}))}$$
$$(17)$$

6   Hybrid HDMR

In most cases the given multivariate data and the sought multivariate function have neither purely additive nor purely multiplicative nature. They have a hybrid nature. So, a new method is used to obtain better results and it is called Hybrid High Dimensional Model Representation (HHDMR). This new expansion includes both the HDMR (or GHDMR) and the FHDMR expansions through a hybridity parameter, γ.

$$
f(x_1, ..., x_N) = \gamma \left(f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \cdots \right)
$$
$$
+ (1 - \gamma) \left(r_0 \left[\prod_{i_1=1}^{N} \left(1 + r_{i_1}(x_{i_1}) \right) \right] \times \cdots \right) \quad (18)
$$

Using the equation given in (18) an HHDMR approximant can be defined as follows by using the HDMR and the FHDMR approximants

$$
h_{jk}(x_1, ..., x_N; \gamma) \equiv \gamma s_j(x_1, ..., x_N)
$$
$$
+ (1 - \gamma) \pi_k(x_1, ..., x_N), \quad 0 \le j, k \le N \quad (19)
$$

where $s_j(x_1, ..., x_N)$ stands for the j-th HDMR approximant and $\pi_k(x_1, ..., x_N)$ stands for the k-th FHDMR approximant which is a truncated product including at most k-variate factors.

The most important step here is to determine the hybridity parameter, γ. For this purpose, a functional is defined as

$$
F(\gamma) \equiv ||f_{org} - f_{HHDMR}(\gamma)||^2 \quad (20)
$$

where f_{org} and f_{HHDMR} stand for the original function and the function obtained from the HHDMR expansion respectively. We need to obtain the γ value that minimizes the value of this norm. This minimization criterion can be written as

$$
\frac{\partial F}{\partial \gamma} = 0 \quad (21)
$$

Using this criterion best value for that parameter can be obtained[11]. By this way the best representation for the sought multivariate function can be determined via Hybrid HDMR.

7   Error Analysis

According to the abovementioned methods, HDMR or GHDMR, FHDMR and HHDMR, several representations can be obtained approximately by using the constant, univariate and bivariate terms of the mentioned expansions. For obtaining these several representations there exist questions, that is, how to find the best expansion for the sought multivariate function or whether the obtained representations are or are not the acceptable solutions for the given engineering problems. For this purpose, the following relative norm

$$
\mathcal{N} = \frac{||f_{org} - f_{new}||}{||f_{org}||} \quad (22)
$$

will be evaluated. Here, f_{new} stands for the multivariate function obtained via a high dimensional model representation expansion.

The minimum norm value obtained by using this relation through all the evaluated norm values will show the best representation for the sought multivariate function. This result is assumed to be the best representation for the multivariate function.

8   Numerical Implementations

In this section, the numerical implementations are classified into two main parts. The first part includes the examples in which the HDMR method is used as a data partitioning technique. In this part FHDMR and HHDMR algorithms are used the partitioned data obtained through HDMR. In the second part, the examples are constructed by using GHDMR method. FHDMR and HHDMR algorithms are used the partitioned data which are obtained through the GHDMR method.

The results are obtained by using MuPAD 4.0. The CPU time results for each implementation are evaluated by using "time()" function which returns the total CPU time in milliseconds that was spent by the current MuPAD process. Only the relative error values and CPU times spent for the evaluations are given in this work.

It is assumed that the following $(N + 1)$–tuples are taken as data to describe a multivariate function $f(x_1, ..., x_N)$

$$
d_j \equiv \left(x_1^{(j)}, ..., x_N^{(j)}, \varphi_j \right), \quad 1 \le j \le m \quad (23)
$$

where φ_j is the value of $f(x_1, ..., x_N)$, the sought function, at the point described by the first N components of d_j in the N–dimensional space we are concerned. That is,

$$
\varphi_j \equiv f(x_1^{(j)}, ..., x_N^{(j)}), \quad 1 \le j \le m \quad (24)
$$

To construct the information for the data set which are the values of the sought multivariate function at the nodes of the grid, analytical structures of known multivariate functions are used.

8.1   HDMR Based Implementations

The first example considered here is a multivariate function which is completely additivite, that is, the sum of univariate functions as follows.

$$f(x_1, \ldots, x_{10}) = \sum_{i=1}^{10} a_i x_i, \quad a_i = 2i - 1 \quad (25)$$

This function has 10 independent variables and it is assumed that the given data set has 16384 nodes in it. The relative error value obtained for the univariate HDMR approximant and the CPU time spent for this approximation are

$$\mathcal{N}_{s_1} = 2.84 \times 10^{-25}, \qquad t_{s_1} = 4.48 \ mins \quad (26)$$

respectively. Because the programming environment has 20 decimal digit accuracy, this result can be assumed to be zero and it means that the representation obtained is exact for the multivariate function dealt with.

In the second example, the selected multivariate function has five independent variables where the function is of purely multiplicative nature

$$f(x_1, x_2, x_3, x_4, x_5) = x_1 \ x_2 \ x_3 \ x_4 \ x_5 \quad (27)$$

and there are 640 nodes in the given hyperprismatic regular grid. The results of the relative error analysis and the CPU times spent for each algorithm are obtained as follows

$$\mathcal{N}_{s_1} = 3.16 \times 10^{-1}, \quad t_{s_1} = 1.56 \ secs$$
$$\mathcal{N}_{s_2} = 8.66 \times 10^{-2}, \quad t_{s_2} = 6.44 \ secs$$
$$\mathcal{N}_{\pi_1} = 8.37 \times 10^{-25}, \quad t_{\pi_1} = 6.47 \ secs \quad (28)$$

where s_1, s_2, and π_1 correspond to the univariate HDMR, bivariate HDMR and univariate FHDMR approximants.

The analytical structure of the multivariate function is defined as follows as the third example with six independent variables.

$$f(x_1, \ldots, x_6) = (x_1 + x_2 + x_3 + x_4 + x_5 + x_6)^5 \quad (29)$$

In this example the given data set is constructed by using 6400 nodes. The relative error values and the CPU times are obtained as follows.

$$\mathcal{N}_{s_1} = 7.30 \times 10^{-2}, \quad t_{s_1} = 9.32 \ secs$$
$$\mathcal{N}_{s_2} = 7.43 \times 10^{-3}, \quad t_{s_2} = 22.79 \ secs$$
$$\mathcal{N}_{\pi_1} = 1.92 \times 10^{-2}, \quad t_{\pi_1} = 9.34 \ secs$$
$$\mathcal{N}_{\pi_2} = 1.14 \times 10^{-3}, \quad t_{\pi_2} = 22.93 \ secs$$
$$\mathcal{N}_{h_{11}} = 5.13 \times 10^{-3}, \quad t_{h_{11}} = 19.37 \ secs$$
$$\mathcal{N}_{h_{22}} = 6.06 \times 10^{-4}, \quad t_{h_{22}} = 391.15 \ secs \quad (30)$$

8.2   GHDMR Based Implementations

In the following example we know the nodes of the mesh and the values of the sought function at the nodes of the given mesh. Hence, the domains for the independent variables are known. For the following numerical implementation there are 4976640 nodes in the mesh. 100 nodes from this mesh are selected randomly. Using these nodes and the values of the following selected multivariate function at these nodes a multivariate data set is constructed.

$$f(x_1, ..., x_{10}) = \sum_{i=1}^{10} i x_i \quad (31)$$

The relative error value and the CPU time spent for this generalized form HDMR method are obtained as follows

$$\mathcal{N}_{\bar{s}_1} = 1.76 \times 10^{-25}, \qquad t_{\bar{s}_1} = 7.69 \ secs \quad (32)$$

where $\bar{s}_1$ corresponds to the univariate GHDMR approximant.

The last example is given to discuss the performance results of GHDMR, FHDMR and HHDMR methods for the following multivariate interpolation problem. The analytical structure of the sought function is selected as

$$f(x_1, x_2, x_3, x_4, x_5) = \prod_{i=1}^{5} (1 + 4x_i) \quad (33)$$

where the problem has 100 nodes. It has both additive and multiplicative features. Hence, it is expected that the HHDMR approximants will give better results than GHDMR and FHDMR. To make this comparison the following relative error values of all approximants obtained through GHDMR, FHDMR and HHDMR are calculated and needed CPU times for these calculations are measured.

$$\mathcal{N}_{\bar{s}_1} = 1.84 \times 10^{-1}, \qquad t_{\bar{s}_1} = 2.10 \ secs$$
$$\mathcal{N}_{\pi_1} = 1.03 \times 10^{-1}, \qquad t_{\pi_1} = 2.16 \ secs$$
$$\mathcal{N}_{h_{11}} = 7.52 \times 10^{-2}, \qquad t_{h_{11}} = 8.25 \ secs \quad (34)$$

9   Concluding Remarks

In this work, the basic idea is to partition the given data to less variate data and then to interpolate them individually to fit an analytical structure to the multivariate function to be determined. The elements of data set are assumed to be given at the nodes of a hyperprismatic grid. Certain nodes may be missing to locate data or entire nodes are used to specify the values of the multivariate function under consideration.

If data is given at all nodes of a hyperprismatic grid then classical HDMR can be used for partitioning. On the other hand, GHDMR should be used instead of HDMR when the data has no datum for certain nodes. The nature of the sought multivariate function also affects the method in use. Since the HDMR expansion has an additive structure, these two methods seem to be effective for additive type functions. As the sought function has not purely additive but also multiplicative nature, the obtained representation via HDMR or GHDMR for the sought function gets worse. Hence, certain new methods are needed to determine better representations for the functions having multiplicative or intermediate natures. For this purpose, FHDMR and HHDMR methods are used.

As a result, we have HDMR, GHDMR, FHDMR and HHDMR methods to deal with the functions whose nature is additive or multiplicative or intermediate type.

When the results given in the previous section are examined carefully depending on the nature of the sought multivariate function the reults get better while we use the method that best fits. However, when the number of nodes or the number of HDMR terms taken into consideration increase, more time periods are needed to obtain the better results. This brings much more CPU time need for the mentioned algorithms.

This means that if you want the best solution for your problem you have to wait much more for the results. On the other hand, if a result obtained by using an approximant having less variate terms is sufficient for the given problem, then you may spend less CPU time for your work.

References:

[1] I.M. Sobol, Sensitivity Estimates for Nonlinear Mathematical Models, *Mathematical Modelling and Computational Experiments (MMCE)*, 1, 1993, No.4.407.

[2] H. Rabitz and Ö. Alış, General Foundations of High Dimensional Model Representations, *J. Math. Chem.* 25, 1999, pp. 197–233.

[3] Ö. Alış and H. Rabitz, Efficient Implementation of High Dimensional Model Representations, *J. Math. Chem.* 29, 2001, pp. 127–142.

[4] G. Li, C. Rosenthal and H. Rabitz, High Dimensional Model Representations, *J. Math. Chem. A* 105, 2001, pp. 7765–7777.

[5] M. Demiralp, High Dimensional Model Representation and its Application varieties, *Mathematical Research* 9, 2003, pp. 146–159.

[6] B. Tunga and M. Demiralp, Hybrid High Dimensional Model Representation Approximats and their Utilization in Applications, *Mathematical Research* 9, 2003, pp. 438–446.

[7] M. Demiralp and M.A. Tunga, High Dimensional Model Representation of Multivariate Interpolation via Hypergrids, *The Sixteenth International Symposium on Computer and Information Sciences (ISCIS XVI)*, 2001, pp. 416–423.

[8] M.A. Tunga and M. Demiralp, Data Partitioning via Generalized High Dimensional Model Representation (GHDMR) and Multivariate Interpolative Applications, *Mathematical Research* 9, 2003, pp. 447–462.

[9] M.A. Tunga and M. Demiralp, A Factorized High Dimensional Model Representation on the Partitioned Random Discrete Data, *Appl. Num. Anal. Comp. Math.* 1, 2004, pp. 231-241.

[10] M.A. Tunga and M. Demiralp, A Factorized High Dimensional Model Representation on the Nodes of a Finite Hyperprismatic Regular Grid, *Applied Mathematics and Computation* 164, 2005, pp. 865–883

[11] M.A. Tunga and M. Demiralp, Hybrid High Dimensional Model Representation (HHDMR) on the Partitioned Data, *Journal of Computational and Applied Mathematics* 185, 2006, pp. 107–132.

[12] W. Oevel, F. Postel, S. Wehmeier and J. Gerhard, *The MuPAD Tutorial*, Springer, 2000.

[13] http://www.mupad.de

[14] H.M. Deitel, P.J. Deitel, T.R. Nieto and D.C. McPhie, *Perl How to Program*, Prentice Hall, 2001.

[15] A.H. Zemanian, *Distribution Theory and Transform Analysis, An Introduction to Generalized Functions, with Applications*, Dover Publications Inc. New York, Reprint 1987.

[16] J.L. Buchanan and P.R. Turner, *Numerical Methods and Analysis*, McGraw-Hill, 1992.