Data Reduction Through Increased Data Utilization in Chemical Dynamics Simulations

Misha Ahmadian a, Yu Zhuang a, William L. Hase b, Yong Chen a

a Texas Tech University, Department of Computer Science, Lubbock, TX, 79409–3104, United States
b Texas Tech University, Department of Chemistry and Biochemistry, Lubbock, TX, 79409–1061, United States

A R T I C L E   I N F O

Article history:
Received 25 May 2016
Received in revised form 11 November 2016
Accepted 8 June 2017
Available online 23 August 2017

Keywords:
Data utilization
Machine learning
Approximation
Interpolation
Chemical dynamic simulation
VENUS

A B S T R A C T

Many scientific applications consist of heavy computational and analysis workload on data, and often require producing intermediate data for ongoing calculations. For instance, chemical dynamics simulations are known as heavy workload applications in terms of calculation in many aspects. There is a strong desire of seeking a solution to minimize expensive calculations by replacing them with lightweight ones. VENUS is one of these chemical dynamic simulation software packages known as classical chemical dynamics simulation, with scalar executing code and heavy calculation process. In this research, we introduce an innovative approximation method by storing, managing, and leveraging intermediate data (results) in order to speed up expensive calculations. The key idea is a newly introduced data interpolation method that leverages data points from previous calculations. The newly proposed method is a general approach that can be applied to a variety of scientific applications and disciplines. In this research, we focus on chemical dynamics simulations and the VENUS code and have developed a prototype of the data interpolation method for reduced computations. The proposed computation reduction method through increased data re-use can increase the efficiency and productivity of scientific simulations, thus can have an impact on scientific discovery powered by high performance computing simulations.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Scientific applications increasingly utilize large-scale data in various fields including physics, astrophysics, climate studies, bioinformatics [1], and chemistry. In many of these science and engineering investigations, there are data that are of critical importance but highly expensive to generate, e.g. obtainable through time-consuming experimental or computational processes. Examples of highly expensive data include those produced by heavy scientific calculations, which might take hours, days, or even weeks to compute. If such critical and expensive datasets are also very large, there is an incentive to reduce the amount of such data, since reduction of expensive computed data also means reduction of computation time. However, what is critical is that the reduction of data should not lead to, within a problem-dependent threshold, the loss of the information that is carried by the original larger set of data. Thus, for scientific applications with large computationally expensive datasets, the objective of this research is to develop a computationally efficient data modeling procedure to replace the computation of some, hopefully as many as possible, data computed through highly expensive processes, and this data modeling procedure must be accurate so that the model data carries almost the same scientific information as the original expensive data.

In this research, we concentrate our approach of replacing expensive data computation by an efficient procedure of data modeling in chemical dynamics simulations, where the potential energy needed in every time step of a simulation has to be computed using a highly expensive computation procedure. Potential energy computation is the most expensive part in each step of a chemical dynamics simulation, whether using empirical formulas or using quantum mechanical electronic structure theories. Parallelization of potential energy calculation is a widely adopted approach to reducing the computation time, as done in many electronic structure calculation packages including NWChem [2,3] and in analytic empirical formulas [4]. Potential energy data calculated based on a quantum mechanical electronic structure theory are called ab initio potential energy data and they are far more computation intensive than empirical formula-based calculations. In this paper, the ab initio data are our targets to be replaced by modeled data at
as many time steps as possible while trying to maintain a desired simulation accuracy.

While data modeling methods are problem dependent and require specific domain science knowledge to design an effective one, this research focuses on machine learning techniques for problems where the originally expensive data have a broadly existent property. Using our techniques, we have developed a method for modeling ab initio potential energy data, and implemented our data modeling method in VENUS, a chemical dynamics simulation software package. VENUS [2,5,6] has its own in-package suit of analytic potential energy models and is also linked with several electronic structure calculation packages, including NWChem [2,3], for generating accurate ab initio potential energy data. In the implementation, our model data replace the ab initio data generated by NWChem.

The rest of this paper is organized as follows. We will explain the chemical dynamics simulations and the VENUS simulation software in Section 2. The method of using modeling data to replace expensively generated data, including its application to chemical dynamics simulations, is described in Section 3. Section 4 presents experimental results, and Section 5 concludes this study.

2. Classical trajectory chemical dynamics simulation

Classical trajectory chemical dynamics simulation provides a useful and generally applicable investigation tool for dynamics studies including gas–surface collisions [12], energy transfer and chemical reaction in gas-phase [13], intramolecular vibrational energy distribution [7], unimolecular decomposition and conformational change [14,15], and, intramolecular energy transfer and chemical reaction [16,17]. For these calculations, the potential energy function \( V \), the potential gradient \( \frac{dV}{dq} \), and in some cases the Hessian \( H \), are required in the process of calculating an ensemble of trajectories, and each trajectory will be determined by numerically integrating the classical equations of motions [2].

A general and accurate approach used in chemical dynamics simulation is to calculate the potential energy data directly from electronic structure theory. VENUS contains a set of analytic potential energy functions and is also integrated with electronic structure calculation packages such as NWChem [3], MOLPRO [18], GAMASS [19], etc.

In chemical dynamics simulations, initial conditions of the reactants for chemical reactions are given for calculating an ensemble of trajectories. Each trajectory is evaluated by numerically integrating either Hamilton’s or Newton’s equations of motion. In an ab initio chemical dynamics simulation, the potential energy data including the energy gradients are calculated by an electronic structure program (e.g. NWChem).

VENUS software package [2,6] is a general Monte Carlo classical trajectory program, when calculating a classical trajectory, the execution of the program begins by reading Cartesian coordinates \( q \) and moments of inertia \( \rho \). The selection of initial conditions calls a subroutine that integrates classical equations of motion, using the potential energy \( V \), and its gradient \( \frac{dV}{dq} \) and Hessian \( H \) to produce dynamics results including the vibrational energy within the molecule. At the final stage of program, cross sections, scattering angels, product energy distribution, rate constant, etc. will be analyzed. The flow of VENUS process is shown in Fig. 2.1 [6].

Based on physical systems of interest in molecular dynamics simulations, VENUS obtains ab initio potential energy data from electronic structure calculation software, like NWChem, and it also provides a variety of analytic potential energy functions to build blocks of potential surfaces, and molecular model systems.

3. Modeling-enabled data reduction through increased data utilization

3.1. The general framework

Data reduction in this work refers to the reduction of data generated by expensive process of calculations, not covering general data reduction techniques like de-duplication, compressions, dimension reduction. We focus on expensive data since, in many cases, inexpensive data can be re-generated with very low costs.

Our strategy for reducing the costly computations for data generation is to replace some of them by data modeling. While data modeling methods do need domain specific knowledge to achieve high modeling accuracy, there is a feature that exist widely in scientific data. Many data generated in scientific or engineering processes consist of data from controllable parameters of the scientific or engineering processes, called input parameter data, and other data resulting from processes with the controllable parameters as input parameter data. In many cases, the resulting data are continuous functions of the input parameter data. In another word, the causal relationship between the input data and the resulting data may have a relationship that is of a continuous function. Even for discrete data in input–output pairs, relations resembling continuous functions also exist. For instance, many discrete optimization methods, e.g. the simulated annealing and the genetic algorithm, are based on the assumption that close input data will, in high
probability, lead to close output data. In this paper, we restrict our investigation only to input and output data related as continuous functions. Investigation on continuous function-like relations for discrete input–output data will be carried out in future research.

For data generated through an expensive computational or experimental process, usually the controllable parameter data incur almost no cost but the resulting output data consume a huge cost to produce. Thus, our data modeling procedure is to characterize the resulting data using the input parameters as input variables of the unknown continuous function that is to be modeled at desirable points. This strategy was introduced in earlier work [8–11] for ab initio chemical dynamics studies, but the data modeling in [8–11] used only ab initio data from the most recent time steps to model the potential data needed at the current time step. In this paper, we investigate the possibility of using ab initio data from any earlier time steps as long as those saved from earlier time steps are close enough to the to-be-modeled function data in the space of the input parameter data. Thus, the range of data selection for the modeling is increased as compared with the modeling procedure in earlier work [8–11]. With enlarged data selection range, an ab initio data item will hence have higher chance to be reused, as compared with possible re-uses only for a few close time steps in the work reported in [8–11]. Thus, increased utilization of the expensive data is one of the new features of our modeling–replacing-generation approach. From the perspective of the data to be modeled, the larger data selection range makes it more likely for the modeling procedure to find close data for accurate modeling, leading to replacement of the expensive data generation by the efficient data modeling at more time steps, and consequently resulting in a higher reduction of the expensive data.

While in this paper, we apply the strategy only to ab initio dynamics simulations where the data is are computed, the approach is also applicable to applications where the original data are generated via non-computational expensive processes. Thus, for both computational and non-computational data generation processes, the amount of such expensive data itself is a good measure of the cost for the data generation. In addition, in many applications where such expensive and large sets of data are also one of the most important, if not the most important, investigation results, as is the case of the ab initio potential energy data in most chemical and material science investigations. Hence, the modeling-replacing-computation approach investigated in this paper is termed as modeling-enabled data reduction through increased data utilization.

The general assumption we impose on the input data and the expensively generated output data is that the output data are continuous functions of the input data. We use mathematical properties of continuous functions, e.g. Taylor expansion or similar expansions, into supervised machine learning where the data to be modeled are expressed as combinations of known data with an optimization process to minimize the differences between the modeled data and the measured data at some observation points. The goal of this optimization process is to choose the optimal, or near-optimal, parameters in the modeling to reach a desired modeling accuracy.

Parameters to be optimized in the optimization process include coefficients in the expression of the to-be-modeled data as combination of the known data, as well as distances of the expensive data used in the data modeling procedure to the data to be modeled. The common approach used in data modeling is to choose a set of available data within a threshold distance to the data to be modeled, and usually the number of data points within the threshold distance is less than adequate to produce a high accuracy approximation, and in such cases the least square optimization is a method often employed to find the coefficients for the formula in the expression of the to-be-modeled data as combination of the known data. In this study, we choose a different approach. We assume that an adequate number of data within a threshold distance will be available or will be made available as expensive data accumulate. Instead, our data-modeling procedure is to find a good threshold distance – good in terms of producing adequate modeling accuracy as well as not requiring a too small threshold distance (a too small threshold distance would require high density of expensive data and lead to high cost for data generation). For a generally applicable data modeling procedure as a convenient investigation tool, in general the threshold distance for the modeling procedure to reach a desired problem-dependent modeling accuracy is unknown, since such a distance is not only problem-dependent, it also depends on the modeling scheme. Thus, a domain expert may not know a good threshold distance for a new modeling method, and the expert of the modeling scheme may not have sufficient knowledge of the domain problem to produce a desired estimation of the threshold distance. In this paper, our supervised learning process for data modeling targets the discovery of a desired threshold distance.

Our threshold distance discovery process starts by an initial estimation of the threshold distance, and uses all expensive data generated in the earliest stage (say, data from the first 1% or 2% of the entire scientific investigation process) to model data in subsequent steps of the investigation. This set of data generated in the earliest stage of the scientific investigation is called the training set. Data in this training set that are used to model the data at a subsequent step are within the threshold distance to the data to be modeled. As we compute the modeled datum using data from the training set, we also let the data generation procedure to produce the “true” data against which the modeled data will be compared to for assessing the modeling error. If the modeling error is too large, we reduce the threshold distance, and the generated “true” data are added to the training set. This iterative refinement process continues until a desired threshold distance for adequate modeling accuracy is found.

3.2. Application to ab initio chemical dynamics simulations

Chemical dynamics simulation is one of those scientific applications that generate expensive ab initio potential energy data as output data from the atom nuclear coordinates. Hence, in this paper we focus on how to re-use those ab initio data for a reduction in some of the expensive data computations.

To increase utilization of the expensive ab initio data, all computed ab initio data have to be saved so that they can be used at later time steps in modeling. After the saved ab initio data reach a certain amount, at every time step of the dynamics simulation, the ab initio dataset is searched to decide if there are data that are close enough to produce accurate data modeling. If it is determined that an accurate modeling is achievable, the potential energy data are modeled and the expensive ab initio calculation is not performed; otherwise, the ab initio calculation is carried out to generate potential energy data. This procedure in using modeling to replace computation in the chemical dynamics simulation is illustrated by the flow chart in Fig. 3.1, and technical details of this procedure are described in following paragraphs.

According to our methodology for data modeling of ab initio data from scientific applications, there are three essential steps to be added to the original procedures of software package to replace the high-cost calculations with new data modeling algorithms at as many time steps as possible:

- Store the ab initio data for future use: The ab initio data generated in the dynamics simulation would be re-used in the data modeling algorithm as an alternative to expensive calculations.
It is needed to memorize these ab initio data by storing them in a proper designed database for future use.

- **Search for close enough data from database:** After storing certain number of ab initio data in database, a procedure needs to be called for each step of calculation to decide whether there is any close data in the database with respect to current calculated data of that step. Then those data will be retrieved from the database to be used by the data modeling algorithm. This search procedure should be called on each step of calculation, after having enough data in the database and before calling any expensive calculation algorithm. In terms of finding close data from database, search algorithm should be designed precisely to let data modeling algorithm output more accurate results, without losing any information. In these cases, applying a training set would be essential to gain accurate resulting data. The problem-dependent machine learning procedure using the ab initio dataset as the training set is described in Section 4.

- **Replace highly expensive calculation with data modeling algorithm:** The final step for this approach is to utilize those ab initio data, which are found by the search algorithm from database, in a data modeling algorithm. If there are enough close data from the database, then we can prevent an expensive original calculation algorithm from being called, and trigger data modeling procedures from using those data. In the other hand, if no data could be found in a step, then expensive calculations should be called, and the resulting data must be stored. Since the data modeling algorithm is a sort of approximation, we do not store the output from data modeling.

3.2.1. The interpolation method for data modeling

As we have discussed earlier, our data modeling technique has three main steps storing ab initio data in a database on each step where ab initio calculation is carried out, searching the database to find the closest data to be re-used for the data modeling algorithm, and finally replacing expensive calculations with a computation-light data modeling algorithm, which uses stored data from the database. Before going through each of these steps, a mathematical technique will be discussed, which is used in data modeling algorithm for replacing expensive calculations.

It is simple sometimes to obtain the value of function $f(x)$ at a set of points such as $x_1, x_2, \ldots, x_n$ ($x_1 < x_2 < \cdots < x_n$). However, sometimes $f(x)$ is more complex that might be a result from long numerical calculation or some physical measurement, which would be very hard to setup a simple functional form. In this case, $f(x)$ needs to be estimated for an input variable value $x$. If the desired $x$ is between the largest and smallest of $x_i$’s where function values are known, then the approximation method is called interpolation. Otherwise, if the $x$ is out of the range then it is called extrapolation [20].

An interpolation scheme is developed and used as our machine learning technique for our data modeling process. The interpolation is to approximate the potential and its gradient with respect to Cartesian coordinates of the atom locations of the current time step using stored ab initio potential energy data. To reduce data amount, a dimension reduction is applied to reduce the dimension of Cartesian coordinates to inter-atomic distances $R$. The dimension reduction reduces the data amount for the coordinates and the gradient of the potential with respect to the coordinates (the coordinates in the reduced-dimension space is $R$). Then we need to transform $dv/dr$, the potential gradient with respect to Cartesian coordinates $Q$, to $dv/DR$, the gradient with respect to distance $R$. The conversion is illustrated by the code in Fig. 3.2 for the case of two atoms.

As it can be observed from this code, the calculation of $Dv/DR$ is fairly light in terms of calculation. It is worth mentioning that this dimension-reduction is lossless up to machine precision, and hence leads to no loss of information. With the dimension reduction, all data are stored in the dimension-reduced form, and interpolation is also carried out in the reduced-dimension space. With $R_1$ and $R_2$ denoting the atom positions of two adequately close stored ab initio data records whose $R_1$ and $R_2$ values are closest to the $R$ value of the current location, the interpolation formulas for ap-
proximiting the potential energy $V$ and its gradient $G$ are given by

$$V = \frac{(R_2 - R) \cdot V_1 + (R - R_1) \cdot V_2}{R_2 - R_1},$$

and

$$G = \frac{(R_2 - R) \cdot G_1 + (R - R_1) \cdot G_2}{R_2 - R_1},$$

where $V_i$ and $G_i$, are the potential energy and its gradient at $R_i$, respectively.

### 3.2.2. Store ab initio data in a dynamic in-memory database

In-memory database refers to storing data in RAM to have fast data accesses. In many Big Data applications, quick access to large amount of data is highly desired, and in-memory processing is the method of choice whenever it is possible. In other words, in-memory processing is being used to speed up the data accesses for data-intensive problems. Since quick access to our stored dataset is highly desirable, we store our data in a simple two-dimensional array dataset.

We have designed a dynamic in-memory table (simply called MemTable) to keep ab initio data for future re-use. Fig. 3.3 shows the structure of this MemTable.

As shown in this figure, MemTable consists of records of distance between atoms $R_i$, which comes from Cartesian coordinates in the $i$th step. $dv_{R_i}$ with respect to $dv_{R_i}$ is potential energy $V_i$ of the $i$th step in trajectory integration. Since the number of bounds between atoms could be different according to the input parameters, and because the number of records per table is unpredictable, we cannot define the number of calculation preemptively. Therefore, MemTable should be able to adjust its size dynamically. This dynamical allocation feature is implemented to let MemTable grow on the fly, which means that after a specific number of records are inserted, if there is no more space to keep the data in the table, an additional amount of memory space, defined as a default in the program, would be allocated.

The goal is to perform calculation for $V$ and $dv/dq$ at least for two steps, and store the distance $R$, potential energy $V$, and $dv/dr$ into the MemTable. Consider that on each data insertion process, records should be sorted based on $Rs$. To achieve the highest efficiency on the algorithm of sorting while inserting data dynamically and sorted into the MemTable after each calculation, we implement the Heapsort.

In this particular problem that we introduce in the current work the memory usage is not very high to be concerned. However, as the problem scales up the memory cost increases, but the cost of expensive computations will be decreased. This could be considered as a fair trade-off between memory and CPU usage.

### 3.2.3. Search for data from the dataset

Before interpolation is performed, it is necessary to search through the dataset of calculated ab initio data. After the first two time steps, the program automatically starts to search for the two closest $Rs$ to the current $R$. To make this clear, consider that at each step of the trajectories, atoms move to new coordinates which result in a new $R$, let’s call it $R^*$. We change the traditional flow of the program in such a way that before performing any further calculation, MemTable should be searched for two inter-atomic distance, say $R_1$ and $R_2$, which are:

$$R_1 < R^* < R_2$$

$R_1$ and $R_2$ will be opted in such a way that there must be no closer $R$ to $R^*$. If $R_1$ and $R_2$ are both found successfully, we would say enough data is found from the database, and we can retrieve all the data in the same way as $R_1$, and $R_2$ to re-use them for data modeling. However, the criteria for finding two closest data could be different for different cases, especially for different number of atoms, but this rule works well for calculation of every two atoms. Another point that should be considered is the accuracy of data: not finding close enough data from the database, may cause loss of information in further steps of calculation while the database grows up and the number of calls for data modeling increases. To solve this issue, we need to design a robust machine learning technique with the stored data as the training set to reduce the chance of information loss, and increase the accuracy of resulting modeled data up to the final step.

Training set is a part of our machine learning procedure to discover potentially predictive relationships between resulting data. Based on the type of algorithm and the number of atoms, this training test could be different. However, devising a good training set to achieve more accurate results would be based on empirical rules. We will show our experimental training set in Section 4. Therefore, if $R_1$ and $R_2$ are found from the database, and they meet the training set criteria, then the data modeling procedure would be called on, and the expensive calculation for potential energy $V$ and $dv/dr$ will be prevented. However, if neither $R_1$, nor $R_2$ exists in MemTable nor they are satisfied by the criteria provided training set, then the original calculation would be called and resulting data should be stored in MemTable. For better and more precise results, we do not store interpolated outputs to the MemTable, and we only keep ab initio data.

### 3.2.4. Data modeling in-place of expensive ab initio calculation

For interpolating potential energy and potential energy gradient, before calling this function, two closest $Rs$ from the MemTable should be found, which meet the condition of $R_1 < R^* < R_2$. If any of these $Rs$ couldn’t be found, the traditional calculation is triggered instead of the interpolation method. After obtaining $R_1$ and $R_2$, data in the same records of $R_1$, and $R_2$ should be retrieved and
used as input of interpolation function. In other words, if we assume $R_i$ and $R_j$ are two records in $i$th and $j$th level of MemTable, the $dv/dr_i$ and $dv/dr_j$ in addition to $V_i$ and $V_j$ would be retrieved from MemTable to be used as input data, then potential energy $V$ and $dv/dq$ are being interpolated as the outputs.

Fig. 3.4 shows the flow of the chemical dynamics simulation process after implementing the data modeling to replace some of the ab initio data calculations.

4. Experimental results and analysis

This section presents our experimental results with the method implemented in the chemical dynamics simulation code VENUS/NWChem. We chose two atoms of $H_2$ with 1000 time steps for one trajectory integration. We have executed the VENUS/NWChem codes on a system with the following configurations:

- $2 \times$ Intel Xeon Quad core 2.66 GHz CPU
- 16 GB RAM

4.1. Results without training set

We first show the importance of a training set in terms of the accuracy of the final results. Figs. 4.1 and 4.2 illustrate the results for the calculation of potential energy $V$ and gradient potential energy $dv/dr$ of two hydrogen atoms ($H_2$) with the implementation of the data modeling, but without using training set of data. It can be observed that the lack of training set would affect the final results in terms of a desired accuracy in the data interpolation.

4.2. Experimental training sets for $H_2$ data modeling in VENUS

As shown in Figs. 4.1 and 4.2, the results are not accurate at all for simulation fidelity, which means that we need to define a specific range for $R_1$ and $R_2$ to obtain desired interpolation accuracy. To find this range a new restriction is applied in the search function described below.

First, we try to find the maximum difference $M = \max(|R_1 - R^*|, |R_2 - R^*|)$ in those steps where two closest $R_1$ and $R_2$ are found, and then we calculate the average of all the maximum results ($M$) up to $k$ steps:

$$\text{Avg} = \frac{1}{k} \sum_{i=1}^{k} \max(|R_1 - R_i|, |R_2 - R_i|)$$

After $k$ steps, we start comparing the maximum values with a Threshold Constant $C$ times final calculated average value (Avg).

$$M_i \leq (C \times \text{Avg}) \rightarrow \text{Interpolation}$$
$$\text{Otherwise} \rightarrow \text{Expensive Calculation}$$

This is the process of threshold distance discovery by using all expensive data generated in earliest stages, which is called the training set. In the case of trajectory calculation for two atoms of hydrogen ($H_2$), $C$ is the threshold variable between 1.0 and 0.05, and $k = 100$ which is the 1% of 1000 total trajectory integration time steps for ($H_2$). If $M_i$ after $k$ steps was in the range of $C$ times $\text{Avg}$ then interpolation will be triggered; otherwise the original expensive calculation would be invoked.

This process mines useful data out of our MemTable database, which leads the interpolation to achieve an accurate approximation. Increasing this threshold constant $C$ increments the number of interpolation and decreases the total calculation number. However, it causes less accurate results in further steps. On the other hand, a small $C$ increases the number of calculations but gives out a more precise result.

Figs. 4.3–4.8 show the results for potential energy ($V$) and potential energy gradient ($dv/dr$) of two hydrogen atoms ($H_2$) atoms in 1000 steps, with different Threshold Constant $C$ values of 0.5, 0.1, and 0.05.

As shown in Figs. 4.3–4.8, a smaller threshold constant leads to better results, and with threshold 0.05, the data modeling results are very close to the original calculation for potential energy ($V$) and potential energy gradient ($dv/dr$), even with an overlap of the original results up to 340 steps out of 1000. These results show that the training set plays an important role in terms of data modeling and learning from data. More precise training sets lead the data modeling process to more accurate result.

Fig. 4.9 plots the number of original expensive calculations versus the number of interpolation calls as the data modeling process
Fig. 4.1. Original calculation vs. Interpolation (No Training set): Potential Energy ($V$) – 1000 steps of trajectory calculation for two atoms of Hydrogen (H$_2$) – VENUS/NWChem.

Fig. 4.2. Original calculation vs. Interpolation (No Training set): Potential Energy gradient (DVDR) – 1000 steps of trajectory calculation for two atoms of Hydrogen (H$_2$) – VENUS/NWChem.

Fig. 4.3. Original calculation vs. Interpolation (Training Set with threshold 0.5): Potential Energy ($V$) – 1000 steps of trajectory calculation for two atoms of Hydrogen (H$_2$) – VENUS/NWChem.

Fig. 4.4. Original calculation vs. Interpolation (Training Set with threshold 0.5): Potential Energy gradient (DVDR) – 1000 steps of trajectory calculation for two atoms of Hydrogen (H$_2$) – VENUS/NWChem.
Fig. 4.5. Original calculation vs. Interpolation (Training Set with threshold 0.1): Potential Energy ($V$) – 1000 steps of trajectory calculation for two atoms of Hydrogen ($H_2$) – VENUS/NWChem.

Fig. 4.6. Original calculation vs. Interpolation (Training Set with threshold 0.1): Potential Energy gradient (DVDR) – 1000 steps of trajectory calculation for two atoms of Hydrogen ($H_2$) – VENUS/NWChem.

Fig. 4.7. Original calculation vs. Interpolation (Training Set with threshold 0.05): Potential Energy ($V$) – 1000 steps of trajectory calculation for two atoms of Hydrogen ($H_2$) – VENUS/NWChem.

Fig. 4.8. Original calculation vs. Interpolation (Training Set with threshold 0.05): Potential Energy gradient (DVDR) – 1000 steps of trajectory calculation for two atoms of Hydrogen ($H_2$) – VENUS/NWChem.
for two atoms of hydrogen in 1000 steps of one trajectory integration. As shown in this figure, increasing the threshold in the training set for data modeling algorithm causes more reduction in the number of expensive calculations, even though the accuracy of final output results would be decreased. Although the current training set is devised precisely to provide very good data modeling results, it should be considered that it leads the interpolation process into a sort of trade-off between the number of calculation and the level of accuracy.

5. Conclusion

Scientific computing simulation is an essential part of the discovery process of many modern sciences, such as physics, astrophysics, climate studies, and chemistry. Many scientific applications in these fields generate large-scale output data with high data generation costs, which may take long time to be finished, and also may consume a huge space for final output data. There is an incentive to reduce those costly calculations in scientific applications. In this paper we have introduced a machine learning-based data modeling procedure to reduce highly expensive scientific calculations, by re-using data at intermediate steps. The fundamental idea is a newly introduced data interpolation method that leverages data points from previous calculations. This process was applied to a chemical dynamics simulation software package (VENUS), and the experimental results confirm the reduction of computation cost for simulations of hydrogen molecules while maintaining the desired accuracy of simulations.

The newly proposed data interpolation method and computation reduction through increased data re-use are general approaches that can be applied to a variety of scientific applications and domains. They can increase the efficiency and productivity of scientific simulations and can have an impact on scientific discovery powered by high performance computing simulations.

Conflict of interest statement

We, authors of the article, state that there is no financial or personal interest or belief that could affect our objectivity, and no potential conflict of interest exist concerning the scientific discoveries described in this article.

Acknowledgements

This research work is support in part by the National Science Foundation under Grant Nos. CHE-0957521 and OISE-0730114 and CNS-1526055, and the Welch Foundation under Grant No. D-0005.

References

