On the Prediction of Properties of Benzene using MP4 Method Executed on High Performance Computing with Heterogeneous Platform

Norma Alias[1], Riadh Sahnoun[2]
Ibnu Sina Institute for Fundamental Science Studies
Universiti Teknologi Malaysia, Johor, Malaysia

Nur Fatin Kamila Zanalabidin[3]
Department of Mathematical Sciences, Faculty of Science
Universiti Teknologi Malaysia, Johor, Malaysia

***Abstract*—High computational complexity problem, high computational cost and deal with a big data are the motivation to study the physical and chemical properties of benzene. Based on the limitation of memory system, processor speed and huge time step computation, we propose the implementation of parallel Gaussian suites of program, particularly the program dealing with high order Møller–Plesset perturbation theory, on high performance homogeneous computing platform (HPC) for predicting the physical and chemical properties of small to medium size molecules, such as benzene, the subject of the present work. Besides high accuracy of the geometrical parameters that can be offered by MP4 simulation, orbital shapes, HOMO-LUMO energy gaps and spectral properties of the molecule are among the properties that can be obtained with accurate prediction. In order to achieve high performance indicators, we need to execute the program in multiple instruction and data stream (MIMD) paradigm using homogenous processors architecture. At the end of this paper, it is shown that Parallel algorithm of Gaussian program using the Linda software can be executed and is well suited in both homogenous and heterogeneous processors. The performance evaluation is essentially based on run time, temporal performance, effectiveness, efficiency, and speedup.**

Keywords*—MP4; parallel computing; homogenous platform; properties of benzene; high performance computing

I. INTRODUCTION

Quantum chemists running electronic structure calculations seem to have an insatiable need for CPU resources in order to get their results faster, or run bigger jobs, or both. Performing high theoretical level calculation is always the most desired target for quantum chemists as performed in paper [1] and [2]. With the recent advancement in computing facilities, calculation of medium size molecules using high theoretical level has become affordable.

In the field of computational chemistry, one of a few quantum chemistry post-Hartree–Fock methods is Møller–Plesset perturbation theory, or MP. By adding electron correlation effects of Rayleigh–Schrödinger perturbation theory, Hartree–Fock method is enhanced [3]. Second (MP2), third (MP3) [4; 5], and fourth (MP4) [6; 7] order Møller–Plesset calculations are enforced in many computational chemistry codes and are common levels implied in calculating modest systems. MP2 imitates dispersion-corrected MP2 coupled model, which is outstanding in a framework with five global empirical parameters and new analogous uncoupled Hartree–Fock [8]. Also, MP2 is a preferable theory due to its relatively moderate computational cost and is known as one of the most extensively used methods for studying noncovalent interactions in quantum chemistry [9]. Paper [10] also stated that single point energy calculation proposes an accurate and fast compromise by combining a MP2 with a Hartree–Fock geometry optimization.

While single point energy calculation at the MP4 level is affordable, full geometry optimizations are rarely reported because of the computational cost, complexity and time-consuming calculations. A single point energy calculation only calculates the energy of the atoms in their current positions, that is, with absence of molecular vibration. Geometry optimizations is a method to determine the minimum energy structure of a system, by modifying the geometry at every stage until a lowest-energy arrangement is found and performing an iterative series of single point calculations. Geometry optimizations mainly rely on the gradient of the first derivative of the energy with respect to atomic positions. Instead of using only a single-point determination, an optimization inevitably requires a lot more computational resources.

Gaussian program package [11] is one of the most popular commercial quantum chemistry codes for which a wide variety of quantum chemistry calculations, including MPn (2-5), can be executed. Depending on the type of operating system and interface, the software can be executed either in single processor, or in parallel execution using multiple processors.

The Gaussian 09 (or simply G09) can be installed for parallel use (shared memory, multi-processor) within a single computer node, serial use (single processor), and distributed use (multi-processor, distributed memory) using TCP Linda. This model implicates a process involving master and a number of workers where in a master process, the programming runs on the current processor. Meanwhile, the programming can run on other nodes of the network in workers processes. In order to execute Gaussian 09/Linda parallel computing, one should fix the list of processors including the jobs that should be executed, job parameters, and the number of processors to use.

Apart from that, every nodes requires some access to the Gaussian 09 directory tree. In our custom configured HPC we used the suggested configuration, where each system will using

G09 for the parallel work. Note that the Linda binaries need to have the same path on each machine. Parallel performance evaluations are established on temporal performance, effectiveness, efficiency, speedup and time execution [12].

II. Gaussian Modeling

In calculating MP2 and beyond, some local disk must be inserted in each node so that Gaussian 09 can put temporary files. The details are in the GAUSS_SCRDIR environment variable, which should be set in the .cshrcor and .profile files for the user account on each node.

Our custom configured workstation is equipped with dual-processor (double threaded) quad-core nodes. In a single node, one can use a maximum of 8-processors (cores) to run parallel Gaussian Job. In MPn calculations it is important to request multiple processors, since the processing time for this type of calculation is very long. To perform parallel Gaussian 09 job in single-node multi-processor, the TCP Linda uses shared memory. The specifications of the number of processors using shared memory need to be thorough in the Gaussian input file. This can be specified by embedding the following line at the beginning of the Gaussian input file:

%NProcShared=n

The above line indicates the "n" processors that will share the memory within the computer node assigned to execute the job. In our calculation, the number of processors was varied from one (1) to the maximum number of eight (8) processors.

III. Sequential Algorithm

All calculations were performed using the Gaussian 09 suites of programs. The model system used was the benzene molecule, for which unconstrained geometry optimization was carried out employing the standard 6–311G(3df,3dp) basis set as implemented in Gaussian 09 program package. The fourth order Møller Plesset perturbation theory employing the space quadruple, double and single substitutions, MP4(SDQ) was used throughout the calculations.

Fig. 1 summarizes the sequence of the steps carried out during MP4(SDQ) geometry optimization.

Steps from "Berny optimizations to locate minimum and transition state" until the last step of "Process information for optimizations and frequencies" will be repeated iteratively until a local minimum is located in the potential energy surface (minimum energy is found). In our calculations, the local minimum was found after four iterations.

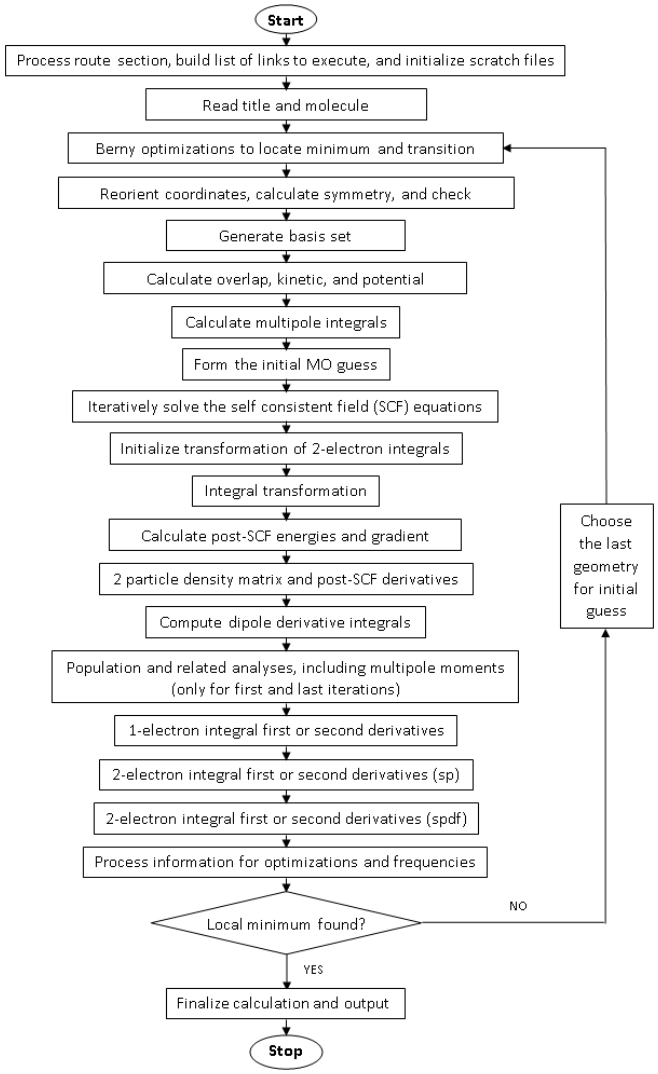

Fig. 1. Flowchart of Sequential Algorithm MP4 Calculation using Gaussian Program.

IV. Parallel Algorithm

Fig. 2 shows the flowchart of parallel algorithm MP$ calculations using Gaussian Method. The calculations were carried out in two types of machine: one is HP and the other one is custom-configured. Both machines have 16 GB of RAM, for which only 2 GB was specified in the input stream of the Gaussian job. As for the hard disk, the HP type has 3 x 1 TB of hard disk while the custom-configured has 3 x 2 TB.

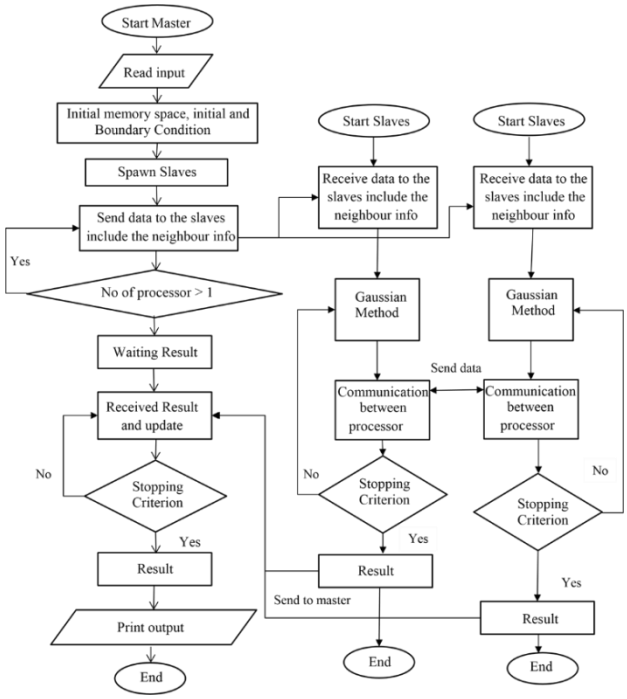

Fig. 2. Flowchart of Parallel Algorithm MP4 Calculation using Gaussian Method.

V. Performance Measure

Run Time, Speedup, Effectiveness, Efficiency, and Temporal Performance are used to measure the performances of parallel strategies.

Run time is the period of time obligated for a program routine to be completed and the time needed for a computer to interpret and carry out an accumulated command. Also, by measuring an algorithm's speedup we can determine its performance since we are occasionally concerned about the performance gains from the algorithm. Amdahl's law emphasized that Speedup can be described as the time taken for a computer to enact in serial divided by the time taken to enact in parallel. The measurements of Speedup are defined as follows;

$$\text{Speedup } (S_P) = t_1 / t_p \quad (1)$$

Now, a calibration of processor usage is called the Efficiency of a parallel algorithm and can be utilized to determine its effectiveness. It is easily explained that the Efficiency is the Speedup divided by the number of used processors. The measurements of Efficiency are defined as follows:

$$\text{Efficiency } (E_p) = \text{Speedup } (S_P) / P \quad (2)$$

Next, by evaluating the Speedup and Efficiency, Effectiveness of a method using parallel algorithm can be determined. The measurements of Effectiveness are defined as follows:

$$\text{Effectiveness } (F_p) = S_P / Pt_p = E_p / t_p \quad (3)$$

The Temporal Performance is a parameter to evaluate the performance of a parallel algorithm, defined as:

$$\text{Temporal} = 1/\text{Time } (p), \quad (4)$$

where Time (p) = time execution using p processor.

VI. Results and Discussion

To predict the physical and chemical properties of small to medium size molecules, such as benzene, the subject of this work, at MP4(SDQ) level of theory, we have implemented the Gaussian suite of program on high performance homogeneous computing platform.

The Gaussian software, particularly the program dealing with high order Møller–Plesset perturbation theory including MP4(SDQ), supports sending and receiving data activities between processors. Note that the parallel computing's performance was evaluated on the prospect of Temporal Performance, Effectiveness, Efficiency, Speedup and Run Time based on the numerical results attained. Here, Table I and Table II show the results of parallel performance evaluation.

Fig. 3 and Fig. 4 show the run time in seconds for 8 types of number of processors – 1, 2, 3, 4, 5, 6, 7 and 8. Observing the graph, execution time decreases with increasing number of processors. This might be attributed to the fact that the task from the master is split and shared between all slaves. It means that the higher the number of utilized processors, the higher the number of slaves the master can involve in the execution of the task which contributes in lowering the execution time. Hence, with the increasing number of processors used, the execution time decreases.

TABLE. I. Run Time, Speedup, Efficiency, and Effectiveness with Respect to the Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM)

No. of Processors	Run time	Speedup	Efficiency	Effectiveness	Temporal Performance
1	49466	1.00	1.00	2.02159E-05	2.02159E-05
2	23174	2.13	1.07	4.60548E-05	4.31518E-05
3	15459	3.20	1.07	6.89928E-05	6.46858E-05
4	11764	4.20	1.05	8.93625E-05	8.50069E-05
5	10867	4.55	0.91	8.37756E-05	9.20217E-05
6	9273	5.33	0.89	9.5884E-05	0.000107844
7	8271	5.98	0.85	0.000103298	0.000120904
8	7638	6.48	0.81	0.000105999	0.000130931

TABLE. II. Run Time, Speedup, Efficiency, and Effectivenesswith Respect to the Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM)

No. of Processors	Run time	Speedup	Efficiency	Effectiveness	Temporal Performance
1	41800	1.00	1.00	2.392E-05	2.392E-05
2	21156	1.98	0.99	4.670E-05	4.727E-05
3	14052	2.97	0.99	7.056E-05	7.116E-05
4	10584	3.95	0.99	9.329E-05	9.448E-05
5	9171	4.56	0.91	9.940E-05	1.090E-04
6	7765	5.38	0.90	1.155E-04	1.288E-04
7	7034	5.94	0.85	1.207E-04	1.422E-04
8	6558	6.37	0.80	1.215E-04	1.525E-04

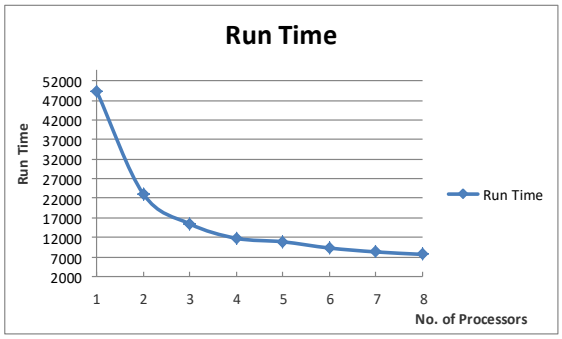

Fig. 3. Run Time vs. Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM).

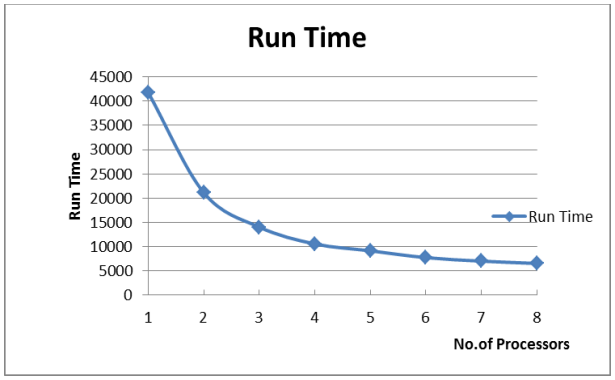

Fig. 4. Run Time vs. Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM).

Inspection of Fig. 5 and Fig. 6 show that the parallel algorithm's speedup increases with increase in the number of processors used. These results are expected because by increasing the number of processors, the calculation will, generally, perform faster.

From the graphs plotted in Fig. 7 and Fig. 8, one can conclude that, the parallel algorithm's efficiency decreases with the increasing number of processors. It is known that the efficiency is the ratio of speedup with respect to the number of processors. Therefore, the efficiency is a performance that is closely related to the speedup.

In Fig. 9 and Fig. 10 are plotted the effectiveness against the number of processors. It can be noticed that as the processors used increases, the effectiveness also escalates. As shown in Equation (3) the effectiveness is proportional to the speedup.

Finally, the graphs plotted in Fig. 11 and Fig. 12 show that as the number of processors is increasing, the temporal performance increases. It is escalating with respect to the increase in the number of processors used. Overall, from these prospects mentioned above, it can be seen that by using more processors, the parallel algorithm can be substantially improved.

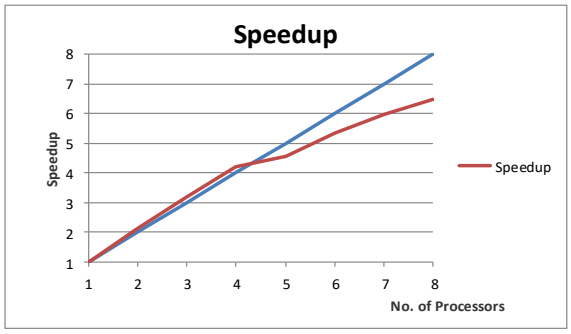

Fig. 5. Speedup vs. Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM).

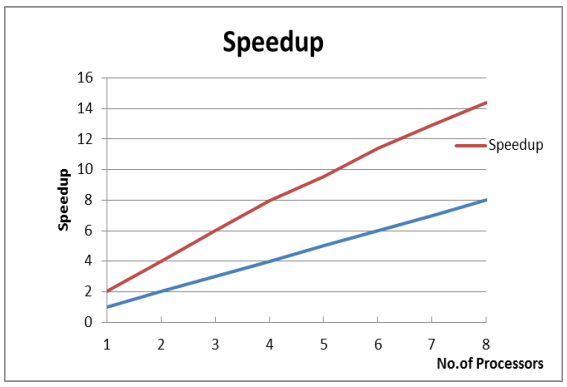

Fig. 6. Speedup vs. Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM).

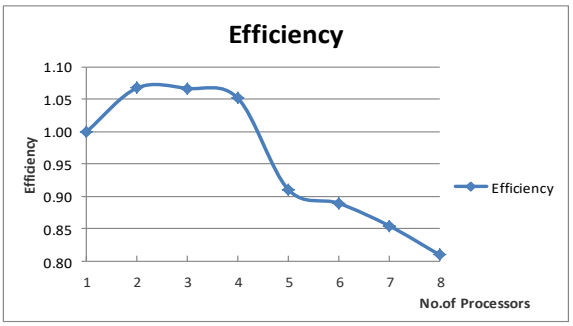

Fig. 7. Efficiency vs. Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM).

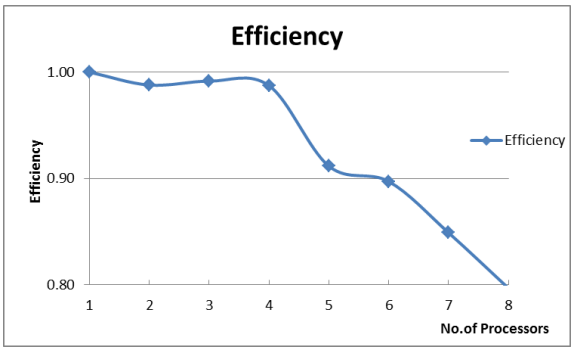

Fig. 8. Efficiency vs. Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM).

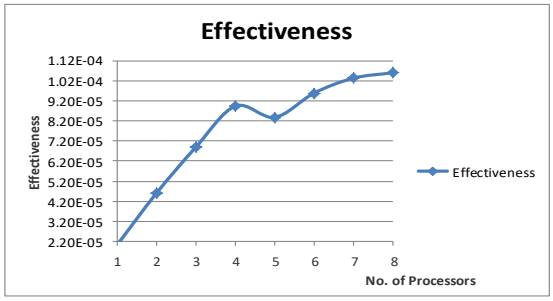

Fig. 9. Effectiveness vs. Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM).

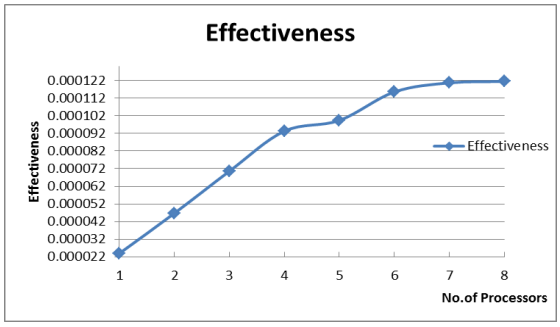

Fig. 10. Effectiveness vs. Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM).

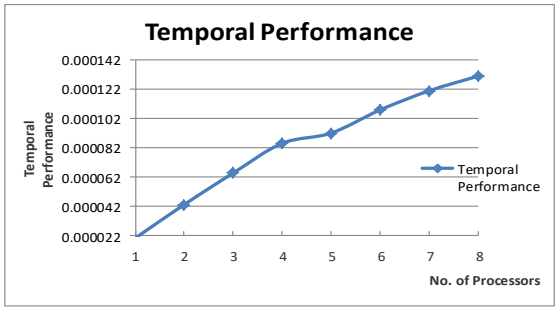

Fig. 11. Temporal Performance vs. Number of Processors (RSN003, 1X3TB, 1 Hitachi + 2 WD BLACK, 16GB RAM).

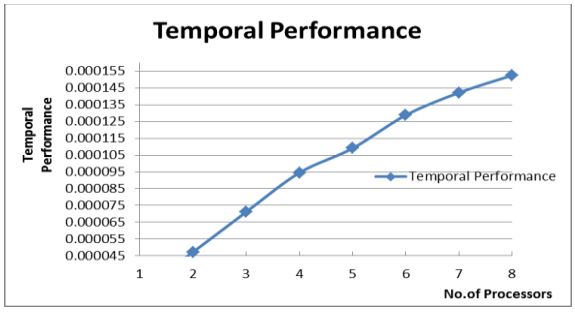

Fig. 12. Temporal Performance vs. Number of Processors (RSN015, 2X3TB, WD BLACK CAVIAR, 16GB RAM).

VII. Conclusion

In conclusion, the parallel algorithms of Gaussian program are well suited for solving the large sparse problem of physical and chemical properties of benzene based on MP4 method. The program is available executed on high performance computing heterogeneous computing platform with different operating systems. Some parallel performance measurements show that the proposed model is accurate prediction of physical and chemical properties of benzene. For a future research, it is recommended to implement the Gaussian program based on MP4 method in solving other large sparse problem.

References

[1] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, & A. Mezzacapo. (2018). Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions. Physical Review A, 98(2), 022322.

[2] A. Kuc, & T. Heine. (2015). The electronic structure calculations of two-dimensional transition-metal dichalcogenides in the presence of external electric and magnetic fields. Chemical Society Reviews, 44(9), 2603-2614.

[3] K. Simeonov. (2009). Higher fullerenes: isolation, halogenation and structural studies.

[4] H. Yamada, Y. Mochizuki, K. Fukuzawa, Y. Okiyama, & Y. Komeiji. (2017). Fragment molecular orbital (FMO) calculations on DNA by a scaled third-order Møller-Plesset perturbation (MP2. 5) scheme. Computational and Theoretical Chemistry, 1101, 46-54.

[5] J. A. Pople, J. S. Binkley, & R. Seeger. (1976). Theoretical models incorporating electron correlation. International Journal of Quantum Chemistry, 10(S10), 1-19.

[6] P. Goel, & J. F. Stanton. (2018). Semiclassical transition state theory based on fourth order vibrational perturbation theory: Model system studies beyond symmetric Eckart barrier. The Journal of chemical physics, 149(13), 134109.

[7] J. Z. Gong, D. A. Matthews, P. B. Changala, & J. F. Stanton. (2018). Fourth-order vibrational perturbation theory with the Watson Hamiltonian: Report of working equations and preliminary results. The Journal of chemical physics, 149(11), 114102.

[8] J. Rezac, C. Greenwell, & G. J. Beran. (2018). Accurate Noncovalent Interactions via Dispersion-Corrected Second-Order Møller–Plesset Perturbation Theory. Journal of chemical theory and computation, 14(9), 4711-4721.

[9] R. T. McGibbon, A. G. Taube, A. G. Donchev, K. Siva, F. Hernández, C. Hargus, K.-H. Law, J. L. Klepeis, & D. E. Shaw. (2017). Improving the accuracy of Møller-Plesset perturbation theory with neural networks. The Journal of chemical physics, 147(16), 161725.

[10] B. D. Sellers, N. C. James, & A. Gobbi. (2017). A comparison of quantum and molecular mechanical methods to estimate strain energy in druglike fragments. Journal of chemical information and modeling, 57(6), 1265-1275.

[11] J. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, & O. Kitao. Gaussian 09, Revision C. 01. In. 2010. Gaussian, Inc.: Wallingford, CT.

[12] N. Alias, H. F. S. Saipol, A. Ghani, A. Che, & M. N. Mustaffa. (2014). Parallel performance comparison of alternating group explicit method between parallel virtual machine and matlab distributed computing for solving large sparse partial differential equations. Advanced Science Letters, 20(2), 477-482.