Parallel Version of the Combined Quantum Classical Molecular Dynamics Code for Complex Molecular and Biomolecular Systems.

Piotr Bała[1,2], Terry Clark[3], Paweł Grochowski[1], Bogdan Lesyng[1] and James Andrew McCammon[4]

[1] Interdisciplinary Centre for Mathematical and Computational Modelling, Warsaw University, Pawińskiego 5a, 02-106 Warsaw, Poland
[2] Institute of Physics, N. Copernicus University, Grudziądzka 5/7, 87-100 Toruń, Poland
[3] Nichols Research Corp/SCT, 2435 5th Street WPAFB, OH 45433-7802, USA
[4] Department of Chemistry and Biochemistry, and Department of Pharmacology, University of California at San Diego, La Jolla, CA 92093-0365, USA

Abstract. The parallel versions of the Quantum Classical Molecular Dynamics code is presented. The execution time scales almost linear with the number of processors. The measured overhead of the parallelization paradigm is extremally small which ensures the high efficiency of the presented method.

1 Combined Quantum Classical Molecular Dynamics

Molecular dynamics (MD) is an important tool for studying biomolecular systems. However, conventional Newtonian MD methods with analytical force fields are limited; in particular, they cannot be applied for the description of chemical or biochemical reactions in which bond-breaking or bond-forming processes occur.

To overcome this problem during the past few years, there has been an increased interest in development of quantum-classical models and their applications to chemical and physical problems at the molecular level. Such models contain both *classical* and *quantum* degrees of freedom. Electronic degrees of freedom are usually treated as quantum variables, whereas positions and momenta of heavy atoms are treated as classical ones.

We have developed a Quantum-Classical Molecular Dynamics model (QCMD), which allows one to simulate dynamical processes in quantum-classical systems [1, 2, 3]. In particular, this model can be used to simulate the motion of quantum particles, protons or electrons, in molecular systems and in the solid state. The ultimate targets of our studies are enzymatic reactions.

In the QCMD model, the proton dynamics in enzyme active sites are described by the time-dependent Schroedinger equation, and the dynamics of the remaining atoms is performed using classical molecular dynamics. The potential energy function appearing in the time-dependent Schroedinger equation is numerically computed using an approximate valence bond (AVB) method [4]. The method was designed and parametrized by us for proton transfer processes [3]. included in the calculations, and the AVB matrix elements are parameterized to reproduce *ab initio* MP2/6-31G* and/or density functional calculations.

1.1 QCMD model

In the QCMD model, selected degrees of freedom are treated quantum dynamically, and the others classically or stochastically. The model consists of a system of coupled equations which govern the dynamics of the quantum and classical subsystems.

The dynamics of the quantum subsystem is described by the time-dependent Schroedinger equation, while the rest of the system is described by the Newtonian equations of motion. The coupling between the quantum and classical domains is described by the time-dependent potential function $V = V(x, \{R_\alpha(t)\})$ in the Schroedinger equation, where V is a function of the instantaneous atomic positions R_α, and by extended Hellmann-Feynman forces [5] modifying the classical forces in the Newtonian equations of motion (Eq. (2) and (3) below). These coupling terms govern the energy transfer between the quantum and classical domains, which influences the time evolution of the whole system [5, 6].

Assuming that the positions of the classical nuclei change slowly in time, one can treat them as parameters in the quantum Hamiltonian, and in the simplest representation the QCMD equations are the following:

$$i\hbar \frac{\partial \Psi(x,t;\{R_\alpha\})}{\partial t} = \left(-\frac{\hbar^2}{2m} \Delta_x + V(x;\{R_\alpha\}) \right) \Psi(x,t;\{R_\alpha\}), \tag{1}$$

$$M_\alpha \ddot{R}_\alpha = F_\alpha + F_\alpha^{HF} \tag{2}$$

with:

$$F_\alpha^{HF} = -\sum_k \left(b_k^2 \langle \Phi_k | \frac{\partial V(x;\{R_\alpha\}}{\partial R_\alpha} | \Phi_k \rangle + E_k \frac{\partial b_k^2}{\partial R_\alpha} \right), \tag{3}$$

where Φ_k and E_k are instantaneous eigenfunctions and eigenvalues of the Hamiltonian $H = -\frac{\hbar^2}{2m}\Delta_x + V(x;\{R_\alpha\})$. b_k are the expansion coefficients of the wavefunction in the basis of the eigenfunctions of the Hamiltonian H: $\Psi(x,t;\{R_\alpha\}) = \sum_k b_k e^{-\frac{i}{\hbar}E_k t} \Phi_k(x;\{R_\alpha\})$.

In the limit of a small time step the extended Hellmann-Feynman forces are equal to the conventional Hellmann-Feynman forces:

$$F_\alpha^{HF} = -\left\langle \Psi(x,t;\{R_\alpha\}) \left| \frac{\partial V(x;\{R_\alpha\}}{\partial R_\alpha} \right| \Psi(x,t;\{R_\alpha\}) \right\rangle. \tag{4}$$

The forces between classical atoms, F_α are computed using standard potential energy functions. The forces between quantum particles and classical atoms located sufficiently far from the quantum subsystem are calculated in the classical limit; in practice, when the distance to a remote atom is much larger than the hydrogen bond length, the classical forces can be used instead of the rigorous extended Hellmann-Feynman forces. The flowchart of the model is presented in the Fig. 1.

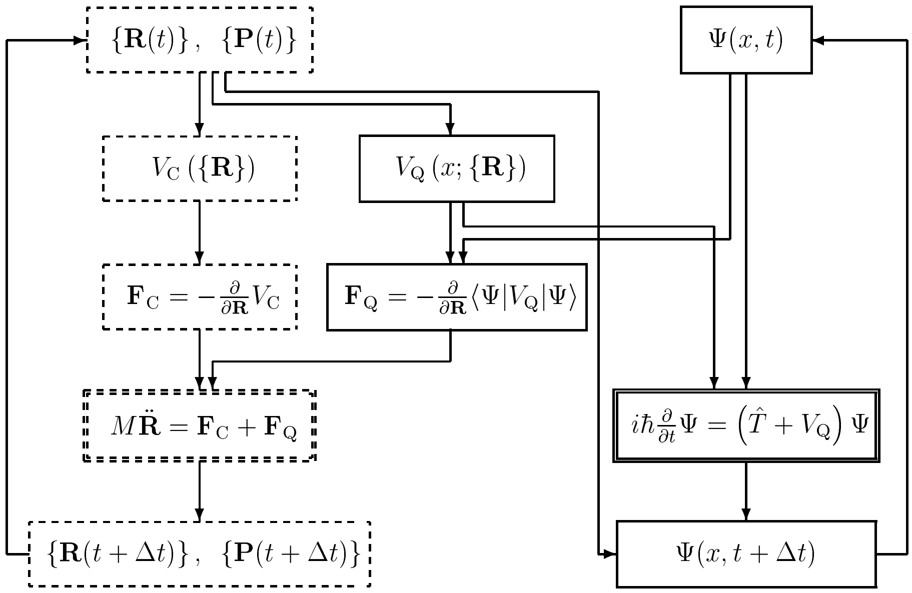

Fig. 1. Scheme of the QCMD model. $\mathbf{R}$ and $\mathbf{P}$ denote positions and momenta of the classical particles. Ψ is the wave function of the quantum particle(s). V_{C} and V_{Q} are the potential energy functions of the classical and quantum regions, respectively. $\mathbf{F}_{\mathrm{C}}$ is the classical force and $\mathbf{F}_{\mathrm{Q}}$ are Hellmann-Feynman forces. The solid boxes indicated parts of the code executed in parallel.

1.2 Potential energy surface

In the case of proton transfer, the potential energy function $V(x;\{R_\alpha\})$ is the Born-Oppenheimer potential energy surface which is obtained numerically by solving the time-independent electronic Schroedinger equation, varying the proton position and computing the energy of the system. For this purpose conventional *ab initio*. semi-empirical molecular orbital methods or density functional methods could be applied. In practical applications for proton transfer processes in molecular complexes or in the active site of phospholipase A_2 the fast AVB method is used [4]. The influence of remote classical atoms on the enzyme active site is included by explicit inclusion of electrostatic interactions with remote atomic charges in the AVB Hamiltonian's diagonal elements. The discretized energy surface changes its shape in time according to the current atomic positions and the current value of the electrostatic field generated in the active site region by the remote atoms. Changes of the shape of the potential energy surface play a key role in the quantum proton dynamics which leads to the proton transfer.

1.3 Algorithm

The QCMD code, because of complexity of the model, consists of different parts characterized by a variety of numerical methods involved. The MD dynamics is performed using a*leap frog* algorithm. However in order to calculate the AVB potential the electrostatic potential acting on the active site atoms must be computed. This task is performed based on standard MD subroutines used to compute nonbonded interactions. The quantum dynamics described by the time-dependent Schroedinger equation is performed based on the discrete representation of the proton's wavefunction on a uniform Cartesian grid. Chebychev polynomial method is used for the propagation of the wavefunction [7, 8, 9]. For this purpose the potential $V(x; \{R_\alpha\})$ must be calculated at each grid point. Moreover, because of the evaluation of the Hellmann-Feynman forces, which is performed using numerical derivatives of the potential with respect to the classical atoms positions R_α, additional evaluations of the potential at all grid points must be performed. The calculation of the potential at each grid points involves three main steps:

1. **Evaluation of the electrostatic field generated by the charged atoms.** This task is performed with help of standard MD routines for evaluation of nonbonded interactions. The electrostatic interaction energy of the unit charge is performed.
2. **Evaluation of the valence band matrix elements.** AVB matrix elements consist of a sum of several empirical potentials described by analytical functions (harmonic, Lenard-Jones type, etc.) Because of a high number of short loops this part of the code cannot be effectively vectorized.
3. **Diagonalization of the AVB matrix.** In the practical applications the dimension of the valence matrix does not exceed 20 and therefore this part of the code cannot be vectorized and parallelized efficiently.

Since the steps 1–3 must be performed at each time-step several times for each point of the 3D grid this part of the code is the most time consuming (see Fig. 2) and together gains about 99% of the total CPU time. One should note, that the potential energy surface could, in principle, be calculated on fly with any *ab initio* method, but since the fastest quantum mechanical code is at least 100 times more time consuming than the developed AVB method, such task would exceed any available computer power.

2 Parallelization of the code

Since the most time-consuming part of the code is performed on a 3 dimensional spatial grid and evaluations of the potential at each grid point require only local information, the all variables evaluated on the grid, e.g. the potential $V(x; \{R_\alpha\})$ and the wavefunction $\Psi(x, t)$, can be distributed over the processor set and evaluated in parallel. One should note that in this case large portions of the code must be performed in parallel, which requires careful distribution of variables and prohibits parallelization scheme based on the distribution of the innermost loops.

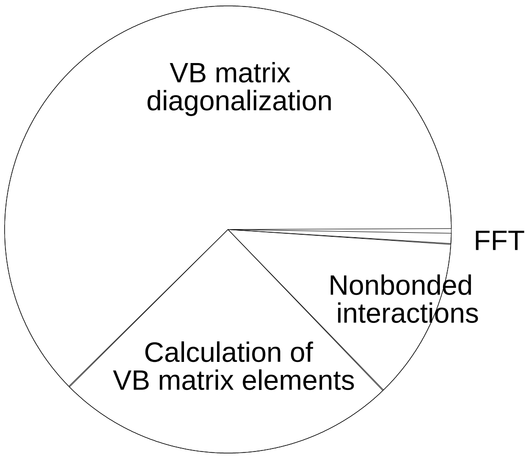

Fig. 2. The time spent in different parts of the QCMD code in the case of sequential execution. The most time consuming steps: evaluation of the electrostatic potential (nonbonded interactions), calculation of the AVB matrix elements, and diagonalization which are performed at each evaluation of the AVB potential.

2.1 Parallelization paradigm

The QCMD code consist of about 20,000 lines of the Fortran 77 code [2]. The dynamics of the classical atoms together with evaluation of the electrostatic potential required for the approximated valence bond are performed by the modified GROMOS [10] routines. The parallelization of the code is performed based on the PFortran package [11] together with the Cray MPI package. PFortran, used as preprocessor, allows for the parallelization of the code together with its easy portability. The developed PFortran code can be preprocessed for using with different message passing packages, such as MPI, PVM or TCGMSG.

The IO is performed from the designated processor. All files are open and read by this processor and than communicated to all processors. The outputted variables, if not stored at every processor, are first exchanged and than outputted from the designated processor. Since disk operations takes insignificant amount of the CPU time, this approach does not effect the code performance.

2.2 Distribution of the arrays

The potential $V(x;\{R_\alpha\})$ defined on the discrete grid $N_x \times N_y \times N_z$ is distributed on the processor grid $np_x \times np_y \times np_z$. The uniform mapping between Cartesian grid and the processor grid is performed.

Distribution of the calculations of the potential function results in parallelization of the majority of the code. However, the potential is evaluated very often during simulations and time spent in the single potential evaluation is relatively small. This requires parallelization of all operations performed of the grid. Otherwise, the distribution of the potential for each processor significantly lowers the performance due to the vast communication. In order to avoid such effects the all operations performed independently at the grid points are performed in parallel, and in consequence all variables represented on the grid (potential, wavefunction, gradient of the potential) are distributed in the same way as the potential.

Evaluation of the Hellmann-Feynman forces (4) requires calculation of the sum over the grid. However partial sums on the distributed grids are performed at each processor and than partial results are communicated and summed over the processor set. This reduces significantly the communication costs and improves the program performance. In the analogous way calculations of all averages involving the wavefunction are performed.

Because of the identical operations performed at each grid point during evaluation of the potential an equal workload can be easily obtained with uniform distribution of the Cartesian grid over the processor set.

Unfortunately not all operations can be easily distributed. The propagation of the protons wavefunction requires evaluation of the Hamiltonian acting on the wavefunction. Since both the potential and the wavefunction are distributed in the same way the potential part of the Hamiltonian $\hat{V}(x)\Psi(x)$ can be easily computed.

2.3 Parallelization of the kinetic energy evaluations

The evaluation of the kinetic part of the Hamiltonian $\frac{1}{2m}\Delta_x\Psi(x)$ is more complicated. In the standard applications this operation is performed using a Fast Fourier Transform (FFT) [12]. In the QCMD the Chebychev polynomial method is used to propagate the wavefunction and Hamiltonian must be evaluated $2N_{CZ}$ times for each time-step, where N_{CZ} is length of the Chebychev polynomial expansion.

Since parallelization of the FFT algorithm is not straightforward and has been of long interest we have applied two simple solutions:

a) **3-dimensional parallel Fast Fourier Transform, PCFFT3D**, form the Cray T3E Scientific libraries is used. This approach requires proper distribution of the matrix over the processor set, eg. $np_x = 1$.
b) **The sequential FFT from Netlib** is used. This approach, however, requires accumulation of the whole wavefunction at one processor, than FFT is performed and the resulting matrix is redistributed over the processor set.

The major limitation of the first approach is its applicability only to a 3-dimensional grid and lack of portability. The second approach requires a significant number of message passing. Although the evaluation of the FFT takes only 0.1% of the CPU time, it requires however, $2N_{CZ}$ communications of the wavefunction matrix, which significantly increases the communication time and reduces the code performance, especially with the increase of the number of processors (see Fig. 3).

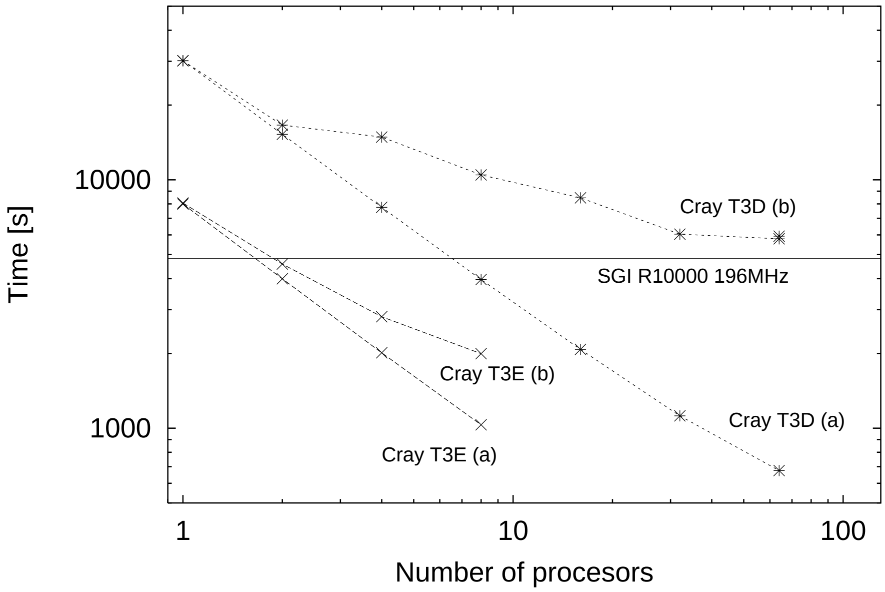

Fig. 3. Performance of the QCMD code on the Cray T3D and T3E computers with FFT executed in parallel (a) and at one processor (b).

3 Results

The QCMD performance measured on Cray T3D and T3E machines with the MPI library exhibits almost linear scaling with the number of processors (Fig. 3). which ensures a high level of the achieved parallelism. The overhead of the performed parallelization was measured by the comparison of the execution time of the sequential code without usage of PFortran and the parallel version executed at one processor. The overhead time is equal to 0.22 % of the execution time.

4 Conclusions

The Quantum Classical Molecular Dynamics code has been successfully parallelized using the PFortran preprocessor which allows a wide portability of the code. The execution time scales almost linear with the number of processors. The measured overhead of the parallelization paradigm is extremally small which ensures the high efficiency of the presented method. The developed code opens new fields for practical applications of the QCMD model. In particular it allows for studies of enzymatic reactions such as electron and proton transfer and proton transfer phenomena in condensed phase. The parallel QCMD code allows for an economic usage of scalable parallel architectures.

Acknowledgements The computations were performed using the Cray T3D computer in SDSC, San Diego (USA) and Cray T3E at ICM, Warsaw University. The research was supported by Polish State Committee for Scientific Research.

References

1. Bala P., Lesyng B., Truong T. N., McCammon J. A.: Ab initio studies and quantum – classical molecular dynamics for proton transfer in model systems and in enzymes. J. Bertran, editor, *The Role of Computational Models and Theories in Biotechnology*, NATO ASI Series, pages 299–326. Kulsweir Press, Dordrecht, 1992.
2. Bala P., Grochowski P., Lesyng B., McCammon J. A.: Quantum – Classical Molecular Dynamics and its computer implementation. Computers & Chemistry **19** (1995) 155–160
3. Bala P., Grochowski P., Lesyng B., McCammon J. A.: Quantum-classical molecular dynamics simulations of proton transfer processes in molecular complexes and in enzymes. J. Phys. Chem. **100** (1996) 2535–2545
4. Grochowski P., Bala P., Lesyng B., McCammon J. A.: Density functional based parametrization of a valence bond method and its applications in quantum-classical molecular dynamics simulations of enzymatic reactions. Int. J. Quant. Chemistry **60** (1996) 1143–1164
5. Bala P., Lesyng B., McCammon J. A.: Extended Hellmann-Feynman theorem for non–stationary states and its application in Quantum-Classical Molecular Dynamics simulations. Chem. Phys. Lett. **219** (1994) 259–266
6. Bala P., Grochowski P., Lesyng B., McCammon J. A.: Quantum–classical molecular dynamics. models and applications. In M. Field, editor, *Quantum Mechanical Simulations Methods for Studying Biological Systems*, pages 115–196. Springer-Verlag Berlin Heidelberg and Les Editions de Physique Les Ulis, 1996.
7. Tel-Ezer H. Kosloff R.: An accurate and efficient scheme for propagating the time dependent Schroedinger equation. J. Chem. Phys. **81** (1984) 3967–3971
8. Leforestier C., Bisseling R. H., Cerjan C., Feit M. D., Freisner R., Guldberg A., Hammerich A., Jolicard G., Karrlein W., Meyer H.-D., Lipkin N., Roncero O., Kosloff R.: A comparison of different propagation schemes for the time dependent Schrodinger equation. J. Comput. Phys. **94** (1991) 59–80
9. Truong T. N., Tanner J. J., Bala P., McCammon J. A., Kouri D. J., Lesyng B., Hoffman D.. A comparative study of time dependent quantum mechanical wavepacket evolution methods. J. Chem. Phys. **96** (1992) 2077–2084
10. Van Gunsteren W.: GROMOS (Groningen Molecular Simulation Computer Program Package). Biomos, Laboratory of Physical Chemistry, University of Groningen, 1987.
11. Bagheri B., Clark T. W., Scott L. R.: PFortran: a parallel dialect of Fortran. ACM Fortran Forum **11** (1992) 20–31
12. Kosloff D., Kosloff R.: A fourier method solution for the time dependent schroedinger equation as a tool in molecular dynamics. J. Comput. Phys. **52** (1983) 35–53