Improving the scaling and performance of multiple time stepping-based molecular dynamics with hybrid density functionals

Sagarmoy Mandal, Ritama Kar, Tobias Klöffel, Bernd Meyer, Nisanth N. Nair

Department of Chemistry, Indian Institute of Technology Kanpur (IITK), Kanpur, India
Interdisciplinary Center for Molecular Materials and Computer Chemistry Center, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany
Erlangen National High Performance Computing Center (NHR@FAU), Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany

Correspondence
Nisanth N. Nair, Department of Chemistry, Indian Institute of Technology Kanpur (IITK) 208016 Kanpur, India. Email: nnair@iitk.ac.in

Funding information
“Competence Unit for Scientific Computing” (CSC); Bundesministerium für Bildung und Forschung, Grant/Award Number: NHR; Cluster of Excellence “Engineering of Advanced Materials” (EAM); Council of Scientific & Industrial Research (CSIR), India; Deutsche Forschungsgemeinschaft, Grant/ Award Number: 182849149; National Supercomputing Mission, India; Subgroup Materials and Computational Chemistry; State of Bavaria; Federal Ministry of Education and Research; Research Foundation; Science and Engineering Research Board, Grant/Award Number: MTR/2019/000359

1 INTRODUCTION

Kohn-Sham density functional theory (KS-DFT) and plane wave (PW) based ab initio molecular dynamics (AIMD) techniques are widely used in investigating structural and dynamical properties of condensed matter systems. The accuracy of the KS-DFT calculations crucially depends on the choice of exchange-correlation (XC) functionals. Owing to improved accuracy, hybrid functionals are preferred over the commonly used XC functionals using the generalized gradient approximation (GGA). Hybrid functionals incorporate a certain fraction of the Hartree-Fock (HF) exchange to the GGA exchange. They give better prediction of energies, structures, electronic properties, reaction barriers, band gaps of solids and dynamical properties of liquids. However, the prohibitively high computational cost associated with HF exchange energy evaluation makes the hybrid functionals and PW based AIMD simulations extremely time consuming. This limits the routine use of hybrid functionals and PW based AIMD simulations for large condensed matter systems.

A number of promising strategies have been proposed so far to speed-up such calculations, which can be broadly divided into two
categories. The first set of techniques introduces some approximations in the evaluation of the HF exchange energy, thereby reducing the computational cost. Several works have been reported in this direction using localized orbitals,\textsuperscript{16,25–27,31,33–41} multiple time step (MTS) algorithms,\textsuperscript{29,30,42–45} coordinate scaling,\textsuperscript{46,47} and other strategies.\textsuperscript{48–51} The other group of methods improve the performance by employing massively parallel algorithms.\textsuperscript{52–55} A combination of both strategies has been also used to achieve remarkable speed-up.\textsuperscript{40,41}

Recently, we proposed an efficient and robust method\textsuperscript{29,45} for performing hybrid functionals and PW based AIMD. We employed a MTS integrator\textsuperscript{2} scheme based on the adaptively compressed exchange (ACE)\textsuperscript{56,57} operator formalism. To take advantage of the ACE operator formalism, we partitioned the ionic forces into computationally cheap fast forces using an approximated ACE operator and computationally costly slow forces due to corrections to the approximated ACE operator. We denote this method as MTACE hereafter. This approach provided a significant speed-up in AIMD simulations by decreasing the number of HF exchange evaluations. Subsequently, we improved the efficiency of this method\textsuperscript{30} by employing localized orbitals. In particular, we used the selected columns of the density matrix (SCDM)\textsuperscript{58} method to obtain localized orbitals, and we used these localized orbitals to build the ACE operator.\textsuperscript{59} We will be denoting this method as s-MTACE. These methods could speed-up the calculations up to an order of magnitude without compromising on accuracy.\textsuperscript{29,30,45} Both MTACE and s-MTACE methods are found to reproduce the structure and dynamics of bulk water and free energetics of chemical reactions in solutions correctly.

In PW based KS-DFT codes, wavefunctions and KS potentials can be in real and/or reciprocal space, and these representations are interconverted with the help of three dimensional (3D) fast Fourier transforms (FFTs). For the optimal performance of the 3D parallel FFTs, PW implementations use a slab decomposition of the 3D real space FFT grids to distribute the data. For instance, the 3D FFT grids are distributed along the X-direction and the YZ planes are distributed among the MPI tasks (or compute cores). Weak scaling performance of such implementations is limited by the number of grid points along the X-direction. For typical DFT calculations today, the number of grid points along any direction is few hundreds, while the number of available compute cores on any modern day supercomputing resource is of the order of few thousands to millions. Thus, slab decomposition based FFTs cannot take the full advantage of the large computational resources available today. To overcome this, a task-group based parallelization strategy was proposed.\textsuperscript{52,60} This strategy implemented in the CPMD\textsuperscript{43} program is called CP Group. In the CP Group implementation of HF exchange computation, the available processors are divided into several task groups and the array that holds the wavefunctions is replicated among these groups. The total workload of the HF exchange energy computation is divided into several parts and they are distributed evenly among these task groups. Finally, a global summation across these groups provides the total contribution to the HF exchange energy.

In the present work, we report the implementations of the MTACE and s-MTACE methods together with CP Group within the CPMD program and we present their performance. In particular, we present the scaling behavior of the MTACE and s-MTACE methods on a large number of CPU compute cores. We will demonstrate here that a significant improvement in the performance of these methods can be achieved through such an approach.

## 2 | METHODS

### 2.1 | HF exchange operator

In conventional KS-DFT calculations using hybrid density functionals and plane waves, evaluation of the HF exchange contributes the most to the total computational time. The HF exchange operator \( V_X \) is defined as

\[
V_X = -\sum_{ij} \langle \psi_i | V_{12} | \psi_j \rangle / r_{12},
\]

(1)

where \( \{ | \psi_i \rangle \} \) is the set of occupied KS orbitals. \( N_{\text{orb}} \) is the total number of occupied orbitals and \( r_{12} = |r_1 - r_2| \). The \( V_X \) operator is applied on a KS orbital \( | \psi_i \rangle \) as

\[
V_X | \psi_i \rangle = -\sum_{ij} \langle \psi_j | V_{12} | \psi_i \rangle / r_{12} | \psi_j \rangle
\]

(2)

\[
= -\sum_{ij} v_{ij}(r) | \psi_j \rangle, \quad i = 1, \cdots, N_{\text{orb}}
\]

with

\[
v_{ij}(r) = \langle \psi_j | v_{\text{HF}}(r) | \psi_i \rangle.
\]

(3)

The HF exchange energy is calculated as

\[
E_{\text{HF}}^X = -\sum_{ij} \langle \psi_i | v_{ij}(r) | \psi_j \rangle.
\]

(4)

For an optimal performance, \( v_{ij}(r) \) is usually evaluated in reciprocal space.\textsuperscript{52,53} The computational cost for doing Fourier transform scales as \( N_{\text{PW}} \log N_{\text{PW}} \) using the FFT technique, where \( N_{\text{PW}} \) is total number of PWs. Therefore, the total computational cost for the evaluation of exchange energy scales as \( N_{\text{orb}}^2 N_{\text{PW}} \log N_{\text{PW}} \).\textsuperscript{32} as \( v_{ij}(r) \) has to be evaluated \( N_{\text{orb}}^2 \) times. This scaling is the reason behind the exorbitant computational time requirement for hybrid functional calculations.

### 2.2 | ACE operator

Lin Lin\textsuperscript{56,57} introduced a low rank decomposition of the \( V_X \) operator in the form

\[
V_X^{\text{ACE}} = -\sum_k | P_k \rangle \langle P_k |.
\]
Here, the set of ACE projection vectors \( \{ \langle P_k \rangle \} \) can be computed by a decomposition of the \( V_X \) operator, see References [45, 56] for details. The construction of \( \{ \langle P_k \rangle \} \) requires evaluation of \( \{ V_X | \psi \rangle \} \), which follows the same computationally demanding procedure requiring \( N_{\text{orb}}^2 \) evaluations of \( v_{ij}(r) \) as discussed in the previous section. However, once the \( V_X \) operator is constructed, it can be easily applied on KS orbitals through the evaluation of \( N_{\text{orb}}^2 \) inner products as

\[
V_X^{\text{ACE}} | \psi \rangle = - \sum_{k}^{N_{\text{orb}}} \langle P_k | P_k | \psi \rangle, \quad i = 1, \ldots, N_{\text{orb}}.
\]

The advantage of the ACE approach lies in the fact that the application of the \( V_X^{\text{ACE}} \) operator on each KS orbitals consumes much less time as compared to the \( V_X \) operator. Such a low rank decomposition can be used in multiple ways to speed-up the HF exchange energy calculations.

### 2.3 | MTACE method

The MTACE method introduced by some of the authors of this paper uses the ACE formalism in the framework of the MTS scheme for speeding-up hybrid functional based AIMD. The self-consistent field (SCF) procedure is modified to exploit the benefits of the ACE operator. In the first SCF step, the ACE operator is constructed after the decomposition of \( V_X \), which is a computationally demanding step as it involves the computation of \( V_X \). However, for the remaining SCF steps, we use the same ACE operator constructed at the first SCF step without recalculating it. After reaching complete SCF convergence, we compute the ionic forces \( F^{\text{ACE}} \). It is to be noted that the optimized wavefunction is most certainly different to the wavefunction which one would obtain if the ACE operator is updated every SCF step. As a result, \( F^{\text{ACE}} \neq F^{\text{exact}} \), where \( F^{\text{exact}} \) is the ionic force computed using the \( V_X \) operator. We take care of this difference in forces within the MTS algorithm as explained later. The flowchart for this modified SCF procedure is shown in Figure 1A.

![Flowchart for MTACE and s-MTACE methods](image-url)
In the MTACE scheme, the ionic forces are written as

\[ F_{\text{exact}}^K = F_{\text{ACE}}^K + \Delta F_{\text{K}}, \quad K = 1, \ldots, 3N_{\text{atom}}, \]  

(6)

with \( \Delta F_{\text{K}} = F_{\text{exact}}^K - F_{\text{ACE}}^K \). In our earlier works,\textsuperscript{29,45} we have shown that differences in the ionic forces \( F_{\text{exact}} \) and \( F_{\text{ACE}} \) are very small and it is justified to consider \( F_{\text{ACE}} \) as fast and \( \Delta F \) as slow:

\[ F_{\text{slow}}^K \equiv \Delta F_{\text{K}} = F_{\text{exact}}^K - F_{\text{ACE}}^K, \quad \text{and} \quad F_{\text{fast}}^K \equiv F_{\text{ACE}}^K. \]  

(7)

Finally, we employ the reversible reference system propagator algorithm (r-RESPA) scheme,\textsuperscript{62} which allows us to compute the computationally costly \( \Delta F \) less frequently, in fact every \( n \) MD steps, as compared to the computationally cheaper \( F_{\text{ACE}} \), resulting in speeding-up the calculations as shown in Figure 1B. For more details, see Reference [45]. We could achieve a speed-up of about seven times using this method for a periodic system containing \( \sim 100 \) atoms employing only 120 compute cores. In passing we note that an optimal choice of \( n \), which has to be determined by monitoring the total energy drift, is crucial for the efficiency and the accuracy of the MTACE method. In our earlier studies,\textsuperscript{29,30,45} we identified the optimal value of \( n \) to be
15 and thus we used the same value of $n$ for the calculations presented in this paper.

### 2.4 s-MTACE method

A modification of the MTACE method, namely s-MTACE, was subsequently proposed, wherein localized SCDM orbitals are used for the construction of the $V^\text{ACE}$ operator, see Figure 1A. Based on a rank-revealing QR factorization of $\Psi^\top \Psi'$, where $\Psi$ is the matrix with all the occupied KS orbitals, the SCDM method constructs linearly independent columns of the density matrix $\Gamma_{\text{ACE}}$ without computing the full $\Gamma$ matrix. The selected columns of $\Gamma$ are then used to construct the set of orthonormal localized selected column of the density matrices $\phi_{i\text{fg}}$. By screening $\phi_{i\text{fg}}$ based on their spatial overlap, it is possible to achieve a substantial reduction in the number of orbitals.

| Table 3 | The number of BOMD steps over which the average compute times were calculated |
|---------|-------------------------------------------------|
| $N_{\text{water}}$ | $N_{\text{PBE}}$ | $N_{\text{PBE0}}$ | $N_{\text{MTACE}}$ | $N_{s\text{-MTACE}}$ | $N_{s0\text{-MTACE}}$ | $\rho_{cut}$ | $n_{\text{core}}$ |
| 32      | 500    | 300    | 300    | 300    | 300    | $2.5 \times 10^{-2}$ | 120     |
| 64      | 300    | 50    | 150    | 150    | 150    | $1.0 \times 10^{-2}$ | 144     |
| 128     | 300    | 20    | 75    | 75    | 75    | $2.0 \times 10^{-3}$ | 192     |

Notes: $N_{x\text{MD}}$ is the total number of MD steps with method $x$. $\rho_{cut}$ is the cutoff used for the screening of the SCDMs in s-MTACE and $s'\text{-MTACE}$ runs. $n_{\text{core}}$ is the number of compute cores per task group.

| Table 4 | Average computational time per BOMD step and average computational time per SCF step for periodic systems containing 32, 64, and 128 water molecules |
|---------|-------------------------------------------------|
| $N_{\text{water}}$ | $N_{\text{core}}$ | $N_{\text{node}}$ | $N_{\text{CPG}}$ | $t_{\text{PBE}}$ | $t_{\text{PBE0}}$ | $t_{\text{MTACE}}$ | $t_{s\text{-MTACE}}$ | $t_{s0\text{-MTACE}}$ | $t_{\text{SCF exact}}$ | $t_{\text{SCF s\text{-ACE}}}$ | $t_{\text{SCF s0\text{-ACE}}}$ |
| 32      | 120    | 3    | 1    | 0.68  | 101.19 | 16.20  | 11.05  | 11.12  | 9.92   | 2.14   | 2.27   |
| 64      | 240    | 6    | 2    | 0.58  | 77.88  | 11.62  | 8.61   | 8.06   | 7.00   | 2.17   | 1.67   |
| 128     | 480    | 12   | 4    | 0.43  | 37.38  | 5.71   | 4.88   | 4.15   | 3.35   | 1.60   | 0.87   |

Notes: $N_{\text{water}}$ is the number of water molecules, $N_{\text{core}}$ is the total number of CPU compute cores, $N_{\text{node}}$ is the total number of compute nodes, $N_{\text{CPG}}$ is the number of task groups. $t_{\text{PBE}}$ is the average computing time per MD step using GGA (PBE) functional, $t_{\text{PBE0}}, t_{\text{MTACE}}, t_{s\text{-MTACE}}$ and $t_{s0\text{-MTACE}}$ are the average computing time per MD step using PBE0, MTACE, s-MTACE and $s'\text{-MTACE}$ methods. $t_{\text{SCF exact}}$ is the average computing time per SCF step during the computation of $F_{\text{exact}}$, $t_{\text{SCF s\text{-ACE}}}$ and $t_{\text{SCF s0\text{-ACE}}}$ are the average computing time for the first SCF step during the computation of $F_{\text{s\text{-ACE}}}$ and $F_{\text{s0\text{-ACE}}}$. All the times reported are in seconds. Calculations were done with 48 compute cores per node, except for the case with $N_{\text{water}} = 32$ and $N_{\text{CPG}} = 1$ case where 24 compute cores per node were used.
involved in the evaluation of $\langle V_x | \psi_i \rangle$ in Equation (2). We screen the orbitals using the criteria

$$\int |\phi_i(r)|^2 \, r \geq \rho_{\text{cut}}.$$ 

We have reported that this procedure substantially reduces the cost of $V_{\text{XACE}}$ operator construction.\textsuperscript{30} It has been shown that s-MTACE can achieve one order of magnitude speed-up for a system containing $\sim 100$ atoms using 120 compute cores.

### 2.5 s'-MTACE method

The rank-revealing QR factorization required for the construction of SCDMs is the most time consuming step of the SCDM procedure. To speed-up these calculations, we employ parallel ScaLAPACK routines. However, these procedures scale poorly when using a large number of processors.

Giannozzi and co-workers\textsuperscript{59} proposed a way to improve the computational efficiency of finding localized SCDMs. In this method, one pre-selects a column of the density matrix based on the electron density and the gradient of the electron density. A column with index $i$ is selected only if

$$\rho(r_i) > \langle \rho \rangle \quad \text{and} \quad \nabla \rho(r_i) < \langle \nabla \rho \rangle.$$ 

Here, $\langle \rho \rangle$ and $\langle \nabla \rho \rangle$ are the average electron density and average gradient of the electron density over the grid points. This pre-screening scheme substantially reduces the number of grid points, which are to be considered for QR factorization. Now, the call to the ScaLAPACK routine involves a smaller size matrix $\Psi'$, thereby improving the performance. A successful implementation of this procedure is already available in the Quantum ESPRESSO code.\textsuperscript{63,64} We have implemented the same approach in the CPMD program to improve the performance of the s-MTACE method. This will be referred as s'-MTACE method hereafter.

### 2.6 CP Groups approach

Let us consider that each of the $N_{\text{orb}}$ KS orbitals possesses $N_{\text{PW}}$ PW coefficients and $N_{\text{core}}$ compute cores are available, then each compute core stores $N_{\text{PW}} / N_{\text{core}}$ rows of the wavefunction matrix $\Psi$ in a typical implementation of the slab decomposition, see Figure 2A. In the CP Group approach, as shown in Figure 2B, the total number of available compute cores $N_{\text{core}}$ is divided into $N_{\text{CPG}}$ groups. Each of such a CP Group possesses $N_{\text{core}} / N_{\text{CPG}}$ compute cores. A copy of the whole $\Psi$ matrix is kept with every group, distributed among the $n_{\text{core}}$ compute cores within that group. As a result, each compute core of a group keeps $N_{\text{PW}} / n_{\text{core}}$ rows of the $\Psi$ matrix. The workload across the task groups is parallelized over the orbital pairs entering the exchange integral in such a manner that computations within each CP Group is restricted to a subset of orbital pairs. Through this, computation of $v_{ij}(r)$ (in Equation 3), which has to be performed for all the orbital pairs, can now be done in chunks across the CP Group. Finally, a...
The replication of the whole $\Psi$ matrix among the CP Groups minimizes the inter group communication. It has been shown that the CP Group approach can be made use to achieve excellent scaling performance in hybrid functional based calculations on several thousands of compute cores.52,60

### RESULTS AND DISCUSSION

We are presenting here the results of scaling tests of the MTACE, s-MTACE and s’-MTACE methods using the CP Groups implementation. Benchmark calculations were carried out for periodic supercells with 32, 64, and 128 water molecules (Table 1).

All the methods and algorithms presented earlier were implemented in a modified version of the CPMD 4.3 program,61,65 and adapted for the existing CP Group implementation within the program. We used PBE0 (hybrid)13 and PBE66 (GGA) functionals for all the computations. Core electrons were accounted by using norm-conserving Troullier Martin pseudopotentials.67 A cutoff energy of 80 Ry was used to expand the wavefunctions in the PW basis set. We carried out Born–Oppenheimer molecular dynamics (BOMD) simulations and the wavefunction convergence criteria in SCF calculations was set to $10^{-6}$ au for the wavefunction gradients. At every MD step, the initial guess of the wavefunctions was constructed based on the Always Stable Predictor Corrector extrapolation scheme68 with order 5. For standard PBE and PBE0 calculations, the standard Velocity Verlet scheme was employed with a time step of $\Delta t = 0.48$ fs. However, for the PBE0 runs with the MTS scheme, $\Delta t = 7.2$ fs (i.e., $n = 15$) were taken as the smaller and larger time steps, respectively. We use the labels PBE, PBE0, MTACE, s-MTACE, and s’-MTACE to indicate different methods used in this work, see Table 2.

All the benchmark calculations presented here were performed on SuperMUC-NG located at Leibniz Supercomputing Center (LRZ). The compute nodes are equipped with two Intel® Skylake Xeon Platinum 8174 processors (24 compute cores per processor). Each compute node has 48 compute cores and 96 GB memory. The nodes are interconnected through a fast Intel® OmniPath network with 100 Gbit/s speed.

In Table 3, we give the number of steps for which average compute timings were calculated and the values of $\rho_{cut}$ for the s-MTACE runs. For an ideal load balancing, the number of compute cores per CP Group ($n_{core}$) is chosen to be equal to the number of grid points in the X-direction. We chose $n_{core}$ as 120, 144, and 192 for systems containing 32, 64, and 128 water molecules, respectively, see Table 3.

The average computational time per BOMD step is reported in Table 4 for the PBE, PBE0, MTACE and s-MTACE methods. In addition, the scaling is shown in Figure 3. First, we observe that the PBE calculations have poor scaling with CP Groups for all the systems. This is expected as these calculations lack enough computationally scalable work that can be distributed over the CP Groups. In fact, due to the extra overhead of communication and synchronization, we notice small increase in computational time per MD step with large number of CP Groups. However, for all the other three methods, which use hybrid functionals, we notice considerable improvements in the performance with increase in CP Groups. Noticeably, PBE0 scales almost perfectly with the number of CP Groups. The scaling behavior of MTACE is also as good as PBE0. It is clear that the scaling of MTACE without the CP Group approach is poor when more than 120 compute cores are used for the 32 water system (Figure 4).

From Table 4, we observe that the ratio of the average computing time per MD step for MTACE and PBE ($t_{MTACE}/t_{GGA}$) decreases with

### TABLE 5 Average number of SCF cycles, $N_{SCF}^{ACE}$ and $N_{SCF}^{exact}$, required to compute $F^{ACE}$ and $F^{exact}$, respectively, using different methods

| Method     | $N_{SCF}^{ACE}$ | $N_{SCF}^{exact}$ |
|------------|-----------------|-------------------|
| PBE0       | –               | 11                |
| MTACE      | 11              | 9                 |
| s-MTACE    | 14              | 12                |
| s’-MTACE   | 14              | 12                |

Note: Here the averages are calculated over 300 MD steps.

Figure 6: Performance measured in units of ps per day for periodic systems containing (A) 32, (B) 64, and (C) 128 water molecules. X axis denotes number of compute cores ($N_{core}$) and Y axis is indicating different methods described in this work. Z axis is ps of trajectory that can be generated per day using all of these schemes.
increasing number of CP Group. This is a consequence of the fact that the MTACE calculations are scaling well with CP Groups as compared to PBE. Depending on the system size, MTACE is only 2–4 times slower than PBE runs by employing the combination of MTACE and CP Groups when a sufficiently large number of processors is used. The best case scenarios for the systems with 32, 64, and 128 water molecules are having MTACE runs clocking only 2.7, 3.5, and 4.3 times slower than PBE, respectively. Further, MTACE is giving a 4–5-fold speed-up compared to PBE0 when using the highest number of CP Groups. For a 32 water system, the computing time for one MD step is now only 0.86 s for hybrid functional based BOMD with the MTACE method when 7680 compute cores were taken. Figure 5 shows the improved performance of the method with and without the CP Group approach. Similar enhancement in performance is also seen for systems with 64 and 128 water molecules. These results are encouraging as we can generate long trajectories at the level of hybrid functionals within a shorter time by making use of large computing resources. The best computing performances we obtained are 48, 15, and 4 ps of trajectory per day for systems with 32, 64, and 128 water molecules, respectively (see also Figure 6). We would like to emphasize that $\delta t = 0.48$ fs was taken considering that H atoms were assigned 1 amu mass. By using a deuterium mass for H atoms, a $\delta t = 1$ fs could be used, resulting in doubling the simulation performance (ps/day).

Although the average number of SCF cycles per force calculation is higher for s-MTACE than MTACE (Table 5), the former is faster than the latter. However, our calculations show that the scaling behavior of s-MTACE deteriorates beyond a certain number of compute cores. In order to understand this poor scaling behavior, we have looked at the average computing time per SCF step during different modes of force calculations. We label the average computing times per SCF step during the computation of $t_{\text{SCF}}$, $t_{\text{SCD}}$, $t_{\text{MC}}$ as $t_{\text{SCF}}$, $t_{\text{SCD}}$, and $t_{\text{MC}}$, respectively. We observe that $t_{\text{SCF}}$ scales poorly when CP Groups is large (Table 4). To scrutinize the poor scaling of $t_{\text{SCF}}$, we decomposed the time for various stages of computation (Table 6 and Figure 7). The parallel QR factorization with the ScaLAPACK routines (Table 6) contributes mostly to the computational overhead for the SCDM localization procedure ($t_{\text{SCDM}}$). It is clear that $t_{\text{SCDM}}$ scales poorly with $N_{\text{CPG}}$, resulting in an overall poor scaling of $t_{\text{SCDM}}$. On the other hand, computation of the HF exchange integrals ($t_{\text{comp}}$) scales well with $N_{\text{CPG}}$.

| $N_{\text{water}}$ | $N_{\text{CPG}}$ | $t_{\text{SCF}}$ | $t_{\text{ACE}}$ | $t_{\text{SCD}}$ | $t_{\text{comp}}$ | $t_{\text{SCF}}$ | $t_{\text{ACE}}$ | $t_{\text{SCD}}$ | $t_{\text{comp}}$ |
|-------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 32                | 1               | 2.14            | 0.46            | 0.04            | 1.65            | 2.27            | 0.00            | 0.03            | 2.23            |
|                   | 2               | 2.17            | 0.93            | 0.06            | 1.18            | 1.67            | 0.01            | 0.05            | 1.62            |
|                   | 4               | 1.60            | 0.94            | 0.06            | 0.60            | 0.87            | 0.01            | 0.05            | 0.82            |
|                   | 8               | 1.39            | 0.93            | 0.06            | 0.40            | 0.62            | 0.01            | 0.05            | 0.56            |
|                   | 16              | 1.24            | 0.95            | 0.05            | 0.25            | 0.39            | 0.01            | 0.05            | 0.33            |
|                   | 32              | 1.15            | 0.93            | 0.05            | 0.17            | 0.28            | 0.01            | 0.05            | 0.22            |
|                   | 64              | 1.12            | 0.94            | 0.05            | 0.13            | 0.22            | 0.01            | 0.05            | 0.16            |
| 64                | 1               | 19.61           | 5.10            | 0.24            | 14.27           | 20.27           | 0.01            | 0.20            | 20.06           |
|                   | 2               | 12.52           | 5.00            | 0.23            | 7.29            | 10.46           | 0.01            | 0.19            | 10.25           |
|                   | 4               | 8.66            | 4.92            | 0.21            | 3.52            | 5.11            | 0.02            | 0.19            | 4.90            |
|                   | 8               | 6.90            | 4.90            | 0.22            | 1.78            | 2.61            | 0.02            | 0.19            | 2.40            |
|                   | 16              | 6.16            | 4.89            | 0.22            | 1.05            | 1.54            | 0.02            | 0.19            | 1.33            |
|                   | 32              | 5.95            | 4.89            | 0.21            | 0.85            | 1.28            | 0.03            | 0.18            | 1.07            |
|                   | 64              | 5.72            | 4.90            | 0.21            | 0.61            | 0.93            | 0.04            | 0.18            | 0.70            |
|                   | 128             | 5.62            | 4.90            | 0.21            | 0.51            | 0.76            | 0.04            | 0.18            | 0.53            |
| 128               | 1               | 183.01          | 32.57           | 1.26            | 149.18          | 212.59          | 0.13            | 1.03            | 211.43          |
|                   | 2               | 104.11          | 32.01           | 1.24            | 70.87           | 102.83          | 0.13            | 1.02            | 101.68          |
|                   | 4               | 65.23           | 31.72           | 1.19            | 32.32           | 47.26           | 0.13            | 1.01            | 46.11           |
|                   | 8               | 49.24           | 31.67           | 1.19            | 16.38           | 23.61           | 0.13            | 1.01            | 22.46           |
|                   | 16              | 41.70           | 31.64           | 1.18            | 8.89            | 12.94           | 0.14            | 1.01            | 11.79           |
|                   | 32              | 39.44           | 31.63           | 1.17            | 6.65            | 9.95            | 0.15            | 0.99            | 8.81            |
|                   | 64              | 37.41           | 31.64           | 1.17            | 4.60            | 6.81            | 0.16            | 0.99            | 5.66            |
|                   | 128             | 36.32           | 31.65           | 1.17            | 3.50            | 5.20            | 0.18            | 0.98            | 4.03            |
| 256               | 35.92           | 31.69           | 1.17            | 3.06            | 4.61            | 0.26            | 0.99            | 3.36            |

Notes: Various contributions to $t_{\text{SCF}}$, $t_{\text{ACE}}$, and $t_{\text{SCD}}$ are reported. $t_{\text{SCDM}}$ or $t_{\text{SCDM}}$ is the time for the localization procedure. $t_{\text{SCF}}$ is the time for the QR factorization and $t_{\text{ACE}}$ is the compute time for the rest of the part. $t_{\text{comp}}$ is the compute time for the actual computation of the HF exchange energy. All compute times reported here are in seconds.
To overcome this and to improve the scalability, we adopted the s'-MTACE method. The reported results in Table 6 and Figure 7 suggest that the QR factorization with the ScaLAPACK routine now consumes negligible amount of computing time. As a result, the poor scaling of $t_{QR}$ has no significant effect on the overall scaling of the method. It has to be noted that the pre-screening procedure used in the s'-MTACE method slightly deteriorates the localization properties of the computed SCDMs, resulting in a large number of overlapping pairs during the construction of the ACE operator. Consequently, $t_{compute}$ turns out to be higher than that of the s-MTACE scheme. The overall scaling behavior of MD timings ($t_{s'-MTACE}$) is satisfactory as can be seen in Table 4 and Figure 3. From Table 5, it is clear that the average number of SCF cycles required for the s'-MTACE is the same as that of the s-MTACE.

4 | SUMMARY

We have presented a detailed benchmarking study on the computational performance of the MTACE and s-MTACE algorithms with task groups (CP Group) for performing AIMD simulations with hybrid density functionals and plane waves. In our implementations of the MTACE and s-MTACE methods using the CP Group environment in the CPMD program, orbital pairs are distributed across the processor groups to achieve a better scaling performance. Through this implementation, we are able to accomplish excellent scaling behavior beyond $\sim$100 compute cores, even for typical system sizes with $\sim$100 atoms. Further, excellent speed-up has been also seen while using this implementation. In the best performance achieved for a model system containing 32 water molecules, computational overhead for doing hybrid density functional based AIMD is only three times more expensive than with GGA. Our implementation has resulted in boosting the performance of hybrid functional based AIMD of this system by a factor of 121 (see Figure 5). The performance of the s-MTACE method was better than the MTACE for a small number of CP Groups, however, it deteriorated with increasing number of the CP Groups. This problem was overcome by the implementation of the s'-MTACE method. Our results suggest that either the MTACE or s'-MTACE method in combination with CP Groups is ideal for running hybrid density functional based AIMD simulations on high-performance computers. For system with finite band gap, s'-MTACE should perform better than MTACE.

ACKNOWLEDGMENTS

Financial support from National Supercomputing Mission (Subgroup Materials and Computational Chemistry), from Science and Engineering Research Board (India) under the MATRICS (Ref. No. MTR/2019/000359), from the German Research Foundation (DFG) through SFB 953 (project number 182849149), from the Federal Ministry of Education and Research (BMBF) and the state of Bavaria as part of the NHR Program, and from the Cluster of Excellence “Engineering of Advanced Materials” (EAM) and the “Competence Unit for Scientific Computing” (CSC) at the University of Erlangen-Nürnberg (FAU) are gratefully acknowledged. RK thanks the Council of Scientific & Industrial Research (CSIR), India, for her PhD fellowship. Computational resources were provided by SuperMUC-NG (project pn97fa) at Leibniz Supercomputing Centre (LRZ), Param Shivay (IIT BHU) and Param Sanganak (IIT Kanpur).

REFERENCES

[1] D. Marx, J. Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*, Cambridge University Press, Cambridge 2009.
[2] M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation*, Oxford University Press, Oxford 2010.
[3] A. P. Leach, *Molecular Modelling. Principles and Applications*, Pearson Education Limited, Upper Saddle River, NJ 2001.
[4] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Academic Press, San Diego, CA 2002.
[5] M. E. Tuckerman, G. J. Martyna, *J. Phys. Chem. B* 2000, 104, 159.
[6] A. J. Cohen, P. Mori-Sánchez, W. Yang, *Science* 2008, 321, 792.
[7] P. Mori-Sánchez, A. J. Cohen, W. Yang, *Phys. Rev. Lett.* 2008, 100, 146401.
[8] W. Koch, M. C. Holthausen, *A Chemist’s Guide to Density Functional Theory*, WILEY-VCH, New York, NY 2001.
[9] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge 2004.
[10] A. D. Becke, *J. Chem. Phys.* 1993, 98, 5648.
[11] J. P. Perdew, M. Ernzerhof, K. Burke, *J. Chem. Phys.* 1996, 105, 9982.
[12] J. Heyd, G. E. Scuseria, M. Ernzerhof, *J. Chem. Phys.* 2003, 118, 8207.
[13] C. Adamo, V. Barone, *J. Chem. Phys.* 1999, 110, 6158.
[14] C. J. Cramer, D. G. Truhlar, *Phys. Chem. Chem. Phys.* 2009, 11, 10757.
[15] B. G. Janesko, T. M. Henderson, G. E. Scuseria, *Phys. Chem. Chem. Phys.* 2009, 11, 443.
[16] Q. Wan, L. Spanu, F. Gygi, G. Galli, *J. Phys. Chem. Lett.* 2014, 5, 2562.
[17] A. J. Cohen, P. Mori-Sánchez, W. Yang, *Chem. Rev.* 2012, 112, 289.
[18] H. Xiao, J. Tahir-Kheli, W. A. Goddard III., *J. Phys. Chem. Lett.* 2011, 2, 212.
[19] Q. Zhao, H. K.ulk, *J. Phys. Chem. Lett.* 2019, 10, 5090.
[20] N. Gerrits, E. W. F. Smeets, S. Vuckovic, A. D. Powell, K. Dobihoff-Dier, G.-J. Kroe, *J. Phys. Chem. Lett.* 2020, 11, 10552.
