Theoretical Chemistry Accounts manuscript No.
(will be inserted by the editor)

An efficient implementation of a QM-MM method in SIESTA

Carlos F. Sanz-Navarro · Rogeli Grima · Alberto García · Edgar A. Bea · Alejandro Soba · José M. Cela · Pablo Ordejón

the date of receipt and acceptance should be inserted later

Abstract We present the major features of a new implementation of a QM-MM method that uses the DFT code SIESTA to treat the quantum mechanical subsystem and the AMBER force field to deal with the classical part. The computation of the electrostatic interaction has been completely revamped to treat periodic boundary conditions exactly, using a real-space grid that encompasses the whole system. Additionally, we present a new parallelization of the SIESTA grid operations which provides near-perfect load-balancing for all the relevant operations and achieves a much better scalability, which is important for efficient massive QM-MM calculations in which the grid can potentially be very large.

Keywords Density Functional Theory · Molecular Mechanics · Parallelization · Load Balancing

1 Introduction

Despite the advances in methods, algorithms and computers, currently it is still not possible to provide an accurate quantum-mechanical (QM) treatment of physico-chemical or biological systems involving large numbers of atoms. In certain cases, like when studying a chemical reaction which occurs locally, an alternative is to split the system into a (typically small) part which is treated with QM methods and another treated with molecular mechanics (MM) methods, using force fields (for a recent review, see Ref. [1]). A QM-MM approach of this kind was described by Crespo et al. [2] and implemented using the Density Functional Theory (DFT) code SIESTA [3] for the QM part and the Wang *et al.* force field parametrization [4] within the AMBER scheme [5]. This QM-MM implementation was intended for the description of molecular effects in which periodic boundary conditions (PBC) did not play a significant role [6,7,8]. However, in many cases, and certainly for extended systems, it is desirable to employ PBC, and to account correctly for all the interactions involved, particularly the long-range Coulomb forces.

We have recently implemented a QM-MM approach that builds on the original work of Crespo et al [2] and extends it with the full consideration of periodic boundary conditions. The new functionality can be used with most first-principles codes acting as the QM back-end, but still finds its most efficient expression in combination with the SIESTA DFT code, due to its use of localized orbitals.

The efficiency of SIESTA makes the approach suitable for very large scale calculations using parallel computers, in which the QM part could contain up to a few thousands of atoms, which is a significant breakthrough with respect to other QM-MM approaches. Nevertheless, the new approach still presents some computational challenges for the description of large systems, even when the QM subsystem is relatively small, since the real-space grid used to represent the QM charge density and potential must be extended to cover the whole system,

C.F. Sanz-Navarro, P. Ordejón
Centre d'Investigació en Nanociència y Nanotecnologia (CIN2-CSIC)
E-mail: csanz@cin2.es

R. Grima, E.A. Bea, A. Soba, J.M. Cela
Barcelona Supercomputer Center (BSC)

A. García
Institut de Ciència de Materials de Barcelona (ICMAB-CSIC)

instead of being restricted to the QM region only. In addition to an increased operation count, the typical spatial inhomogeneity of the quantum/classical atomic distribution implies that issues of load balancing and communication scheduling become very important when executing the simulation in parallel. To address these issues, together with the development of our QM-MM approach, our work has focused on a careful optimization of the parallel implementation of the SIESTA routines involved in the real-space grid operations. This has resulted in a much more efficient and scalable parallel performance of the QM calculations with SIESTA, which has a big impact on the performance of the new QM-MM method.

In this paper we present the basic ideas behind our QM-MM implementation using PBC, and we document, in the context of the QM-MM method, the improvements made in the parallel execution of SIESTA. We stress that these improvements in parallel performance are not restricted to QM-MM simulations, but are of general usefulness also for fully QM calculations, particularly for systems which exhibit a high degree of inhomogeneity, such as clusters, nanowires, and surfaces.

2 An overview of the QM-MM method

We give here only a brief summary of the method, concentrating on the key new aspects, and on its validation with a small example. Full details, including its application to larger systems, will be presented elsewhere.

2.1 Theory and programming scheme

The QM/MM methodologies aim to accurately describe the chemistry of a specific region of interest using a QM method, while the typically much larger surrounding region is modelled by faster MM approaches. The total energy of the combined system is given by [9]:

$$E^{tot} = E_{QM} + E_{MM} + E_{QM/MM}. \quad (1)$$

E_{QM} is the internal Kohn-Sham energy of the quantum subsystem calculated by SIESTA (Eq. (53) of Ref. [3]). The pure classical term, E_{MM}, includes two-, three- and four-body bonded terms as well as non-bonded terms such as van der Waals and Coulomb interactions (Eq. (10) of Ref. [2]) between classical atoms. The mixed interaction between the QM and MM regions, $E_{QM/MM}$, only contains van der Waals and Coulomb interactions between the classical and quantum atoms (and electrons). It should be noted, however, that the presence of the classical atoms is taken into account when computing E_{QM} through its effect on the selfconsistent electronic charge density, which is different from that of the isolated quantum subsystem.

As in our previous works, [2] in the present QM/MM implementation, the system can be partitioned in QM and MM regions in such a way that a covalent bond is in the boundary (*i.e.*, one atom in the bond belongs to the QM part and the other to the MM part), which is useful, for instance, in the study of biomolecules. To deal with these cases, we have developed a variant of the scaled position link atom method (SPLAM) [10]. Details of our implementation will be given elsewhere.

The additive character of the energy terms in Eq. 1 allows us to setup the code in a modular way. A front-end driver program takes care of the classical subsystem, and makes calls to the QM back-end module (SIESTA in our case) to solve the quantum mechanical problem and return the relevant quantities to the driver, which then computes the total energy and gathers the forces to complete the molecular dynamics step and proceed to the next one. The key steps in the operation of the method are:

i) The calculation of the electrostatic potential generated on the real-space grid by the classical point charges of the MM region. This will act as an additional external potential $V_{MM}(\mathbf{r})$ for SIESTA.

ii) The SIESTA calculation in the presence of the external potential. The real-space grid operations performed by SIESTA during the self-consistent loop include the computation of the Hartree and exchange-correlation potentials (routines `poison` [*sic*] and `cellxc`, respectively), of the grid contribution to the Hamiltonian matrix elements (routine `vmat`), and, after diagonalization, the calculation of the charge density from the density matrix (routine `rhoofd`). All these operations are now carried out using the extended grid. This does not in itself increase dramatically the operation count (except in routine `poison`) because the basis orbitals are still localized in the QM region, but a proper load balancing is essential, as we shall see below.

After reaching self-consistency, SIESTA determines the forces on the QM atoms, computes E_{QM} and the electrostatic part of $E_{QM/MM}$, and sends back to the driver program the electronic charge density $\rho(\mathbf{r})$ on the grid.

iii) The driver program will compute the forces on the classical atoms due to the QM atoms (via $\rho(\mathbf{r})$) and to the other classical atoms (via the force field), as well as the remaining MM and QM/MM energy terms.

The driver program communicates with the SIESTA process via FIFO pipes [11].

2.2 Periodic boundary conditions

The SIESTA code works by default with periodic boundary conditions, in particular computing the Hartree potential from the charge density using a Fourier transform (routine `poison`). To have consistent periodic boundary conditions for the QM and MM regions, it is convenient to extend the QM simulation box to coincide with the total simulation box (this is not strictly necessary [12], but it is the simplest option for a code such as SIESTA). To deal explicitly with the point charges in the system we have implemented the Ewald method, both for the calculation of the potential created by the classical ions on the QM grid points, $V_{MM}(\mathbf{r})$, and to compute the electrostatic interaction between the classical ions. The shorter-range bonded interactions and the fast-decaying van der Waals term also take into account PBC, although they do not need the Ewald approach and are implemented using cut-offs.

As the electrostatic interaction of the QM and MM calculations takes place only through the exchange of $V_{MM}(\mathbf{r})$ and $\rho(\mathbf{r})$ on the real-space grid, this approach works for most first-principles codes, which in one way or another implement such a grid. However, SIESTA is particularly suitable since its basis set is associated to the atoms, and hence non-QM regions do not need to be explicitly accounted for in the computation of the electronic structure of the QM sub-system, as would be the case with a plane-wave basis set, for example. In addition, the computation of $V_{MM}(\mathbf{r})$ can be accelerated since it is only needed on those grid points touched by a QM orbital. Furthermore, the short-ranged real-space Ewald contribution of those classical ions far enough from the QM region can be skipped.

2.3 Validation of the QM/MM methodology: Water molecules

As an illustration and validation of the QM-MM methodology we present some calculations in systems containing water molecules. We employed the GGA exchange-correlation functional of Perdew-Burke-Ernzerhof (PBE) [13], and a double-ζ plus polarization basis set optimized for water [14] with relatively short orbital cutoffs (6.1 a.u. and 4.2 a.u. for the longest orbitals of O and H, respectively). This functional and basis set provide a very good description of the properties of liquid water [14] with a low computational cost, although it yields a dipole for the free water molecule of 2.04 D, which is somewhat higher than the one obtained experimentally in the gas phase, i.e. 1.85 D [15] (more complete basis sets are needed to obtain a converged value of the dipole moment [16,17]). The classical water molecules were modelled by the TIP3P potential [18] with additional bond and bond angle terms taken from AMBER [5]. This potential produces a dipole of 2.35 D, which largely overestimates the experimental one for the free molecule.

We first considered a small system comprised of two water molecules [19,20,21,22,23]. The hydrogen bond in this simple system can already put the accuracy of the implemented QM-MM methodology to the test. Table 1 shows the equilibrium O-O distance, angles and binding energy of the water dimer (see Fig. 1 for the definition of the angles). The QM calculation using the GGA-PBE functional provides results which compare very well to experiment [24]. The MM calculation, on the other hand, is less accurate, producing a short O-O distance, and overestimating the binding energy of the dimer. Besides, the ϕ angle is too small: the right angle is obtained only when the directional charges in the sp^3 lone pairs of the acceptor O atom are properly described, and these effects are absent in the TIP3P potential (which only includes point charges centred on the atoms). The QM-MM calculations with QM description of the donor molecule give similar results to the MM calculation: the O-O distance is too short, the ϕ angle is too small and the binding energy is too large. However, when the acceptor molecule is described on the QM level, the QM-MM results are close to those of the full QM calculation. This highlights the importance of describing properly the electronic distribution around the acceptor O atom for the description of the hydrogen bond. For comparison, we also show the results of a fully QM calculation, but in which the LDA functional

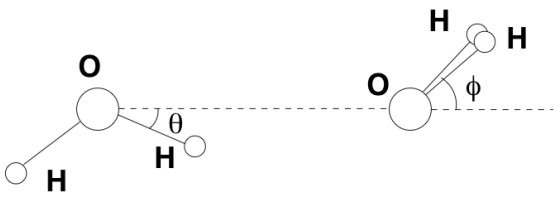

Fig. 1 Geometry of the H_2O dimer with the definition of the angles and distances. The donor and acceptor molecule are the right and left one, respectively.

	O-O distance (Å)	ϕ (deg)	θ (deg)	Eb (kcal/mol)
Exp. (ref. [24])	2.98	57.0	-1.0	-5.44±0.7
QM(GGA)	2.91	62.1	4.6	-5.42
MM	2.73	23.4	3.1	-6.73
QM-MM (QM donor)	2.81	14.5	0.8	-8.11
QM-MM (QM acceptor)	2.64	57.5	-2.3	-5.9
QM(LDA)	2.76	69.5	7.1	-8.99

Table 1 Computed O-O distances, angles (defined in Fig. 1) and binding energies for a water dimer. QM (GGA) and QM(LDA): A pure QM water dimer using a counterpoise correction to compensate the basis set superposition error (BSSE), using the GGA and LDA functionals, respectively. MM: A pure MM water dimer. We also present two sets of QM/MM results (both using GGA for the QM part), in which the QM molecule is the hydrogen acceptor and donor, respectively.

of Perdew-Wang [25] was used, which highlight the difficulty of a proper description of the dimer, even using fully quantum mechanical calculations. In conclusion, the QM/MM implementation performs well in the description of the water dimer, with errors that reflect the basic limitations of the MM model used. Our results are similar to those of previous QM-MM implementations [20,26,27].

As a second test, we assess the influence of the MM region on the QM region by calculating the induced dipole moment of a QM molecule in a liquid environment of 215 MM water molecules and under periodic boundary conditions. The dipole of the QM molecule is obtained by a time average during a 10 ps molecular dynamics simulation at 300 K. We obtain a value of 2.73 D, which subtracting the value for the free molecule yields a dipole change induced by the MM region of 0.69 D. This value is close to the experimental one obtained for the difference between the dipole of a water molecule in the gas phase and that in ice, i.e. 0.75 D. [28].

In summary, our method gives very reasonable results for systems comprised of both QM and MM water molecules.

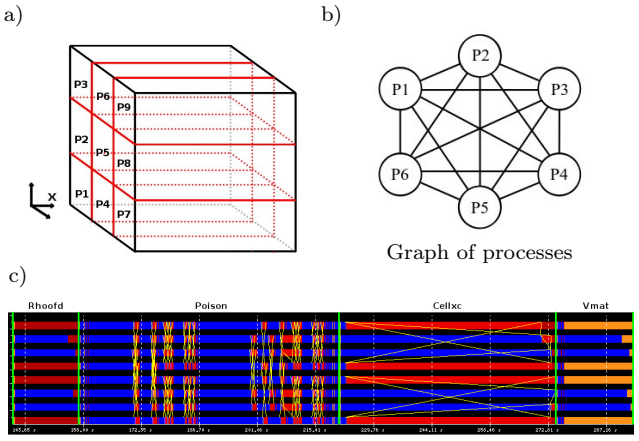

Fig. 2 Original parallelization of SIESTA. a) Sketch of the uniform 2-D real-space domain decomposition; b) Graph showing the all-to-all (or global) communications pattern; c) Execution trace of the four main routines involved in the real-space grid operations. The eight horizontal bars represent eight processes which at a given instant can be computing (blue) or communicating (red and orange). This trace corresponds to a test case involving a set of 262 water molecules distributed inhomogeneously in the simulation box, as shown in Fig. 3(a).

3 Optimization of the parallel execution of grid operations

In the original SIESTA parallelization the distribution of the real-space mesh data among processors was done in a uniform way. The mesh points were divided in the Y and Z directions (more precisely, along the second and third lattice vectors) over the processors in a 2-D grid, so that each processor was assigned a parallelepipedic sub-mesh that extended along the X (first lattice vector) direction (see Fig. 2(a)). A highly unbalanced workload resulted for cases with a inhomogeneous ionic distribution (for example, for a cluster centred at the origin, or for a slab perpendicular to the Z direction). In QM-MM calculations one typically has a rather localized QM region immersed in the classical system, so workload imbalance problems are likely to be the norm. To exemplify the problem we use as one of our test cases a system of liquid water with a total of 7161 molecules (262 QM and 6899 MM). The QM molecules are confined to the central region of a cubic box of 60 Å side (Fig. 3(a)), and surrounded by the MM molecules. We will compare this case with a system of the same size and number of molecules, but where the QM and MM molecules are uniformly distributed throughout the simulation cell (Fig. 3(b)).

The imbalance problems can be visualized using the PARAVER tool [29], which processes trace data obtained during the execution of an instrumented version of the code [30] and displays the information in a convenient way. Figure 2(c) shows a computation with eight processors for the inhomogeneous system of Fig. 3(a). The blue, orange and red colours represent computing, global communication and

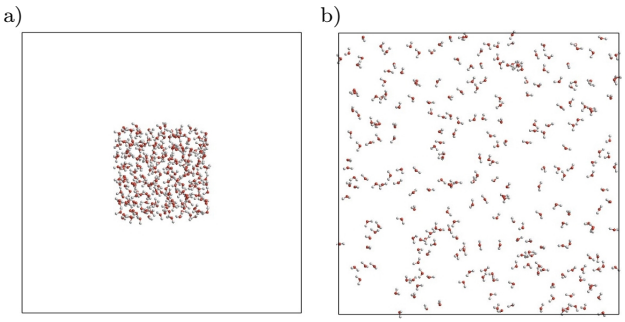

Fig. 3 QM-MM simulation boxes with liquid water at uniform density. Only the QM molecules are shown. In (a), the QM molecules are located in the central area, while the MM molecules are in the surrounding region. In (b), the distribution of QM and MM molecules is uniform.

point-to-point communication events, respectively. Global communications and start-end times of the four key SIESTA routines described above have been marked. Some problems are immediately obvious from the trace. First, the computation and waiting times are different for each processor. This is a symptom that the workload is unbalanced among the processors. In particular, processors 1, 4, 5, and 8 seem to have been assigned mostly empty regions of the box, with very little computation in all routines except `poison`. Furthermore, the imbalance is different for every routine, except for `rhoofd` and `vmat`, which show similar behaviour. Second, there are too many global communications. These all-to-all communications are trying to make the relevant parts of the distributed data structures available to those processors that need them, but in a quite inefficient way. This is clearly seen in routines `rhoofd` and `vmat`, in which pieces of the density matrix and the Hamiltonian, respectively, are passed around. Figure 2(b) represents this state of affairs as a graph of processes with fully-connected nodes. The edges represent portions of data held by each processor which are sent to others. In this case the data are sent to *all* other processors, so that many unnecessary data transfers are carried out in the network.

Workload imbalance, even if relatively small, can lead to gross inefficiencies in parallel operation, typically manifested in a reduced speed-up when the number of processors is increased, i.e., reduced scalability. The same is true of the abuse of all-to-all communication patterns. To correct these inefficiencies we have developed new approaches to the problems of mesh distribution and communication scheduling.

3.1 Balanced mesh distribution

The key to the choice of an adequate mesh distribution among processors is the use of a weight function which represents the amount of work associated to each mesh point. Seen in this light, the uniform distribution used in the original version is appropriate only if the weight is the same for all points, as is the case in the `poison` routine, which basically performs an FFT on the data on the grid. In general, though, the amount of calculation in each mesh point is different and, crucially, depends on the type of operation to be performed. Therefore, a properly load-balanced calculation will need not just one, but several distributions, which will alternate during the execution of the program. Routine `cellxc` involves a bi-valued weight function: 1 if the mesh point is touched by any basis orbital, and 0 otherwise (when there is no charge density to process). Routines `vmat` and `rhoofd` need a weight function proportional to the square of the number of orbitals touching the point, since the operations to be performed involve pairs of orbitals. Only `poison` has a flat weight, as described above.

So SIESTA needs three different distributions for the grid operations. For a given weight function, each processor is assigned a parallelepipedic portion of the real-space grid, determined using a recursive bisection algorithm [31], (see Fig. 4(a)), which at each step creates new sub-domains corresponding to regions of approximately equal computational cost.

3.2 Efficient communication scheduling

In order to improve the efficiency of the communications these are pre-scheduled. The pattern of communication can be represented as before as a graph (of processes) in which nodes represent processes and edges communication between them (Fig. 4(b)). Rather than using indiscriminate all-to-all broadcasts, as in Fig. 2(b), it pays to consider in detail the specific communication events really needed to redistribute the appropriate pieces of the density matrix and the Hamiltonian among the processors that need them to complete mesh operations (in routines `rhoofd` and `vmat`, respectively). With the use of these point-to-point communications the graph is no longer fully connected.

Furthermore, it becomes possible to schedule communications in such a way that those involving disjoint sets of processors can take place at the same time. Our scheduling algorithm uses the dual graph (of communications) pictured to the right in Fig. 4(b),

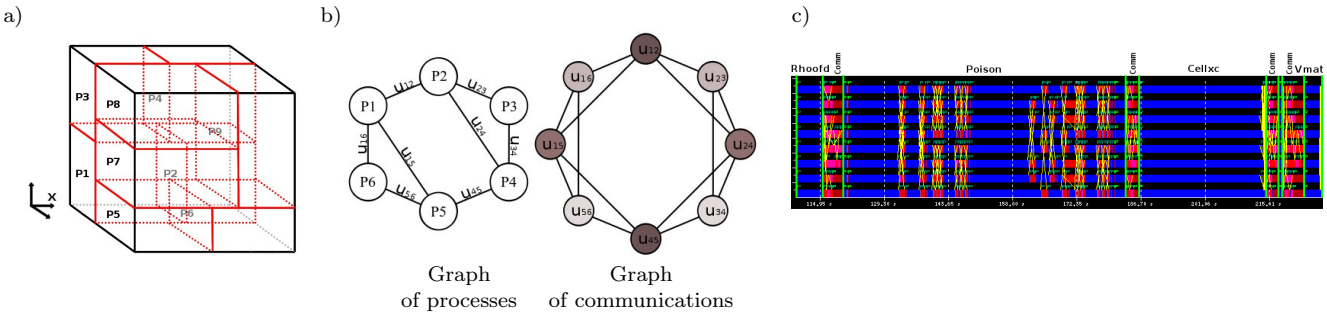

Fig. 4 New parallelization of SIESTA. a) Balanced real-space domain decomposition by using a recursive bisection algorithm; b) communication scheduling generated by applying a colouring algorithm to the graph of communications. The same colour means communications that happen at the same time; c) idem Fig. 2(c).

in which nodes now represent communication events and edges the processors involved, so two nodes are connected by an edge if the same processor is needed for the two communication events. The search for concurrency opportunities in communication is now equivalent to the problem of colouring the graph with the minimum number of colours in such a way that nodes (communication events) connected by lines (processors) do not share the same colour. Communications (nodes) of the same colour can then take place simultaneously (see figure 4(b)). Graph colouring is an NP-complete problem [32], but a heuristic can be used to find closely optimal colourings. We use the iterative largest-first algorithm [33], in which at every step or iteration a non-coloured node is chosen and painted with a different colour from its adjacent nodes. The selected node is one that has the greater number of non-coloreud adjacent nodes.

4 Showcase for parallelization results

4.1 Execution trace

Figure 4(c), obtained for the same inhomogeneous system as Fig. 2(c), shows the reduction in global communications and the much better balanced workload for all routines that are achieved using the new parallelization of SIESTA. Note that these improvements will be observed in general by all systems, even if they are not intrinsically very inhomogeneous, due to the operation-dependent workload distribution.

4.2 Scalability tests

To analyze the improvements on the parallel performance of SIESTA, we have used the two systems shown in Fig. 3. While the typical QM-MM calculations have traditionally used a relatively small QM part, in order to analyze the scalability improvements we have considered a moderately large QM subsystem of 262 water molecules. With this benchmark we can employ up to 128 processors while keeping a reasonable load on each processor. The system sizes shows the potential for QM-MM simulations with large QM parts which we aim at performing in the future. In addition, the tests performed will be relevant for more general kinds of systems, and for fully-QM calculations.

For both cases in Fig. 3 (inhomogeneous and homogeneous distributions of QM molecules) we compare the performances of the old version of SIESTA and of the new version implementing the parallelization improvements. As a measure of performance we use the relative speed-up, conceptually $S_p = T_1/T_p$, which measures how much faster the calculation is when using p processors instead of one. To avoid artefacts stemming from different memory access patterns [34], we actually use $S_p = 8T_8/T_p$, taking as reference a calculation with eight processors.

As the focus of these benchmarks is on the scalability properties of the new parallelization of SIESTA, we do not take into account any classical atoms in the calculations.

The grid-related operations performed by SIESTA are called in the program from a parent routine `dhscf`, and in what follows we use its overall performance as the approximate figure of merit for the benchmarks, while still discussing the individual performance of the four worker routines already introduced. Figure 5 shows the speed-up curves for the old (upper row) and new (lower row) versions, for both the homogeneous (left side) and inhomogeneous (right side) QM sub-systems.

In all cases one observes a progressive degradation of the parallel efficiency (S_p/p) with processor count, but the performance reduction of the original parallelization is markedly worse, particularly, as expected, for the

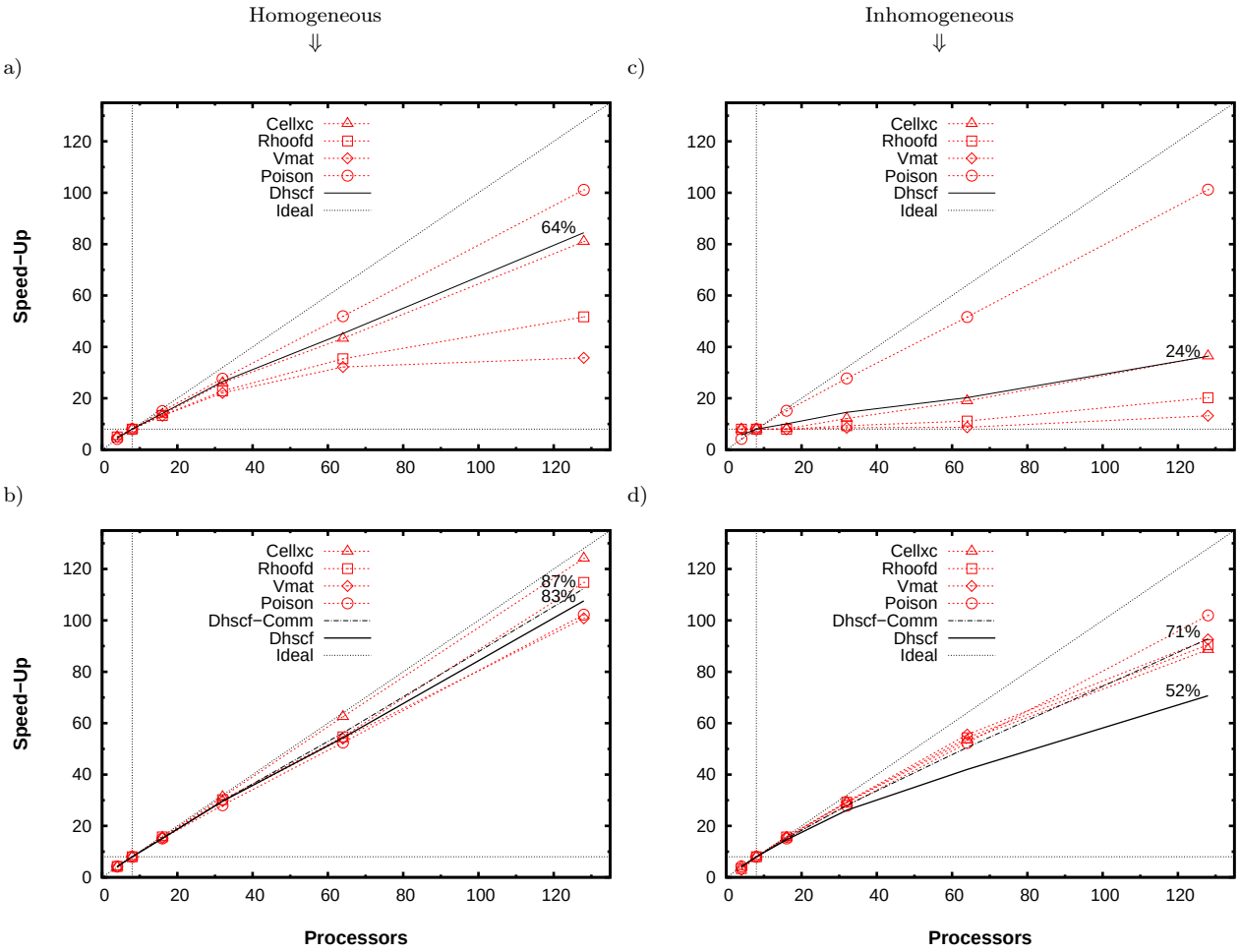

Fig. 5 Speed-up and efficiency comparison, with reference to eight processors, of the original (top row) and new (bottom row) Siesta parallelization schemes. Benchmark cases correspond to two water boxes with homogeneous (left side) and inhomogeneous (right side) molecular distributions, as shown in Fig. 3. The overall `dhscf` speed-up for the new parallelization includes three data re-distributions (communications) for routine pre-scheduling. For direct comparison with the original parallelization, the contribution of these communications is removed from the global speed-up in the curves marked `Dhscf-Comm`. All calculations use the PBE GGA exchange-correlation functional.

Procs.	**Homogeneous**			**Inhomogeneous**		
	Original	**New**		**Original**	**New**	
		Total	**Comm.**		**Total**	**Comm.**
4	296.23	305.17	22.74	210.45	220.60	22.23
8	171.52	162.33	17.50	150.37	115.37	14.28
16	97.21	86.11	10.21	119.43	63.22	9.54
32	51.82	43.85	4.87	82.72	35.34	6.51
64	30.34	23.91	3.29	59.40	21.98	5.98
128	16.25	12.07	1.76	33.08	13.06	4.35

Table 2 Parallel execution time in seconds for grid operations (routine `dhscf`) for both the original and new Siesta parallelizations. The benchmarks are the same as in Fig. 5. The total `dhscf` time for the new parallelization includes the time spent in the data re-distributions for routine pre-scheduling, which is also shown under the `Comm.` heading.

inhomogeneous case, in which the efficiency drops to 24% (see Fig. 5(c)) due to the workload imbalance exemplified in Fig. 2, compared to a more reasonable 64% for the homogeneous case (see Fig. 5(a)). The new parallelization improves performance significantly, with the efficiency reaching 52% (see Fig. 5(d)) and 83% (see Fig. 5(b)) for the inhomogeneous and the homogeneous case, respectively.

Before discussing the performance of the individual routines, it should be noted that the use of three separate data distributions in the new parallelization scheme introduces extra communication needs for re-distribution of the data arrays before and after the relevant operations. The effect of these communications is included in the `dhscf` curves for the new parallelization (Fig. 5(b) and (d)). For completeness, we also present in the plots the curves obtained when the time spent in these communications is subtracted from the total count for `dhscf`. These curves show that these communications are of greater importance for the inhomogeneous case due to the inherent non-uniformity of data distribution. The parallel efficiency goes up to 71% and 87% when the extra communications are not counted. Since there is no meaningful and unambiguous way to assign the communication overhead to any particular sub-routine of `dhscf`, the individual-routine curves in the bottom plots of Fig. 4 refer to the net speed-up without re-distribution communications. While this communication overhead is obviously relevant for a global assessment of performance, the net speed-up curves are a good measure of the efficiency gains with respect to the old parallelization scheme.

The scalability of the `poison` routine is the same for both parallelizations and both test cases, homogeneous and inhomogeneous, since its uniform data distribution is intrinsically appropriate and was not modified. This routine, with an efficiency higher than 77%, exhibits the best performance in the original parallel version. In the new parallelization, in contrast, the rest of routines exhibit better or at least similar performance: the `cellxc` scalability is nearly perfect for the homogeneous case and similar to `poison` for the inhomogeneous case. Both `rhoofd` and `vmat` exhibit a similar scalability, slightly better than `poison`, as shown in Figs. 5(b) and (d).

Table 2 shows the parallel execution time of the `dhscf` routine. Timings are much improved with the new version in the inhomogeneous case, reaching a factor of 1/3 for 128 processors. So even if the scalability is still not near ideal in the inhomogeneous case, the execution time still decreases significantly, therefore improving overall performance. For the homogeneous case, the total `dhscf` time values are similar, as an homogeneous molecular distribution naturally leads to a nearly uniform mesh distribution. The data re-distribution time (`comm` on Table 2) reaches ∼33% of the total time for inhomogeneous case and ∼15% for the homogeneous case as the number of processors is increased up to 128.

Overall, Figure 5 and Table 2 show that the new parallelization has a much better parallel performance in both homogeneous and inhomogeneous cases.

5 Conclusions

We have presented the design and performance results of a new parallelization strategy for the grid operations in Siesta that is particularly appropriate to the demands of a new QM-MM hybrid implementation which deals exactly with periodic boundary conditions in the electrostatic interaction and thus employs large real-space meshes. The new parallelization will be useful for all systems, and particularly for those exhibiting some degree of inhomogeneity in the ionic distribution.

Acknowledgements This work was supported by the Spanish Ministry of Science and Innovation (MICINN) through grants CSD2007-00050 (Supercomputing and e-Science), and FIS2009-12721-C04. C.S.-N. acknowledges support from MICINN through the Ramon y Cajal Program.

References

1. Zhang R, Lev B, Cuervo JE, Noskov SY, Salahub DR (2010) A Guide to QM/MM Methodology and Applications. In: Advances in Quantum Chemistry, Vol 59, Elsevier Accademic Press, San Diego, pp. 353–400.
2. Crespo A, Scherlis DA, Martí MA, Ordejón P, Roitberg AE, Estrin DA (2003) A DFT-based QM-MM approach designed for the treatment of large molecular systems: Aplication to chorismate mutase. J Phys Chem B 107:13728–13736
3. Soler JM, Artacho E, Gale JD, García A, Junquera J, Ordejón P, Sánchez-Portal D (2002) The Siesta method for ab initio order-N materials simulation. J Phys: Condens Matter 14:2745–2779
4. Wang J, Cieplak P, Kollman PA (2000) How well does a restrained electrostatic potential (resp) model perform in calculating conformational en ergies of organic and biological molecules. J Comput Chem 21:1049–1074
5. Cornell WD, Cieplak P, Bayly CI, Gould IR, Jr KMM, Ferguson DM, Spellmeyer DC, Fox T, Caldwell JW, Kollman PA (1995) A 2nd generation force-field for the simulation of proteins, nucleic-acids, and organic-molecules. J Am Chem Soc 117:5179–5197
6. Scherlis DA, Martí MA, Ordejón P, Estrin DA (2002) Environment effects on chemical reactivity of heme proteins Int. J. Quantum Chem 90:1505–1514

7. Martí MA, Scherlis DA, Doctorovich FA, Ordejón P, Estrin DA (2003) Modulation of the NO trans effect in heme proteins: implications for the activation of soluble guanylate cyclase J Biol Inorg Chem 8:595–600.
8. Martí MA, Capece L, Crespo A, Doctorovich F, Estrin DA (2005) Nitric Oxide Interaction with Cytochrome c' and Its Relevance to Guanylate Cyclase. Why Does the Iron Histidine Bond Break? J Am Chem Soc 127:7721–7728.
9. Senn HM, Thiel W (2009) QM/MM methods for biomolecular systems. Angew Chem Int Ed 48:1198–1229
10. Eichinger M, Tavan P, Hutter J, Parrinello M (1999) A hybrid method for solutes in complex solvents: Density functional theory combined with empirical force fields J Chem Phys 110:10452-10467
11. A. Garcia, E. Anglada, and J.M. Soler (unpublished).
12. Laino T, Mohamed FI, Laio A, Parrinello M (2006) An Efficient Linear-Scaling Electrostatic Coupling for Treating Periodic Boundary Conditions in QM/MM Simulations. J Chem Theory Comput 2:1370–1378.
13. Perdew JP, Burke K, Ernzerhof M (1996) Generalized gradient approximation made simple. Phys Rev Lett 77:3865–3868
14. Fernández-Serra MV, Artacho E (2006) Electrons and hydrogen-bond connectivity in liquid water. Phys Rev Lett 96:016404
15. Clough A, Beers Y, Klein GP, Rothman LS (1973) Dipole moment of water from stark measurements of H_2O, HDO, and D_2O. J Chem Phys 59:2254–2259
16. Wei D, Salahub DR (1994) A combined density functional and molecular dynamics simulation of a quantum water molecule in aqueous solution. Chem Phys Lett 224:291–296
17. The dipole obtained with Siesta using a basis set with triple-ζ plus double polarization orbitals, and cutoff radii of around 9 a.u. is 1.86 D, very close to the experimental value of 1.85 D.
18. Jorgensen WL (1981) Quantum and statistical mechanical studies of liquids. 10. Transferable intermolecular potential functions for water, alcohols, and ethers. Application to liquid water. J Am Chem Soc 103:335-340
19. Eichinger M, Tavan P, Hutter J, Parrinello M (1999) A hybrid method for solutes in complex solvents: Density functional theory combined with empirical force fields. J Chem Phys 110:10452–10467
20. Takahashi H, Hori T, Hashimoto H, Nitta T (2001) A hybrid QM/MM method employing real space grids for QM water in the TIP4P water solvents. J Comp Chem 22:1252–1261
21. Tu Y, Laaksonen A (1999) On the effect of Lennard-Jones parameters on the quantum mechanical and molecular mechanics coupling in a hybrid molecular dynamics simulation of liquid water. J Chem Phys 111:7519–7525
22. Lofere MJ, Loeffler HH, Liedl KR (2003) A QM-MM interface between CHARMM and Turbomole: Implementation and application to systems in bulk phase and biologically active systems. J Comp Chem 24:1240–1249
23. Tuñón I, Martins-Costa MTC, Millot C, Ruiz-López MF, Rivail JL (1996) A coupled density functional-molecular mechanics monte carlo simulation method: The water molecule in liquid water. J Chem Phys 17:19–29
24. Curtis LA, Frurip DJ, Blander M (1979) Studies of molecular association in H_2O and D_2O vapors by measurement of thermal conductivity. J Chem Phys 71:2703–2711
25. Perdew JP, Wang Y (1992) Accurate and simple analytic representation of the electron-gas correlation energy. Phys Rev B 45:13244–13249
26. Biswas PK, Gogonea V (2005) A regularized and renormalized electrostatic coupling Hamiltonian for hybrid quantum-mechanical-molecular-mechanical calculations. J Phys Chem 123:164,114
27. Lyne PD, Hodoscek M, Karplus M (1999) A hybrid QM-MM potential employing hartree-fock or density functional methods in the quantum region. J Phys Chem A 103:3462–3471
28. Coulson CA, Eisenberg D (1966) Interactions of H_2O molecules in ice .I. Dipole moment of an H_2O molecule in ice. Proc Roy Soc A (London) 291:445–453
29. V. Pillet, J. Labarta, T. Cortés, S. Girona, "Paraver: A tool to visualize and analyze parallel code", Transputer and occam Developments', (1995) 17-32, http://www.bsc.es/paraver.
30. MPItrace instrumentation package, http://www.bsc.es/plantillaA.php?cat id=492.
31. H. D. Simon, S. Teng, "How good is recursive bisection?", SIAM Journal on Scientific Computing (1995).
32. M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness" W.H. Freeman, New York, (1979).
33. D. Welsch, M. B. Powel, "An upper bound for the chromatic number of a graph and its application to timetabling problems", Computer Journal, (1967).
34. The large system size implies a large total memory requirement, which reflects in a slow-down for execution in a small number of processors due to swapping, cache misses, etc.