Available online at www.sciencedirect.com

ScienceDirect

Computer Physics Communications 177 (2007) 339–347

Computer Physics Communications

www.elsevier.com/locate/cpc

Efficient first-principles calculations of the electronic structure of periodic systems

M.M.G. Alemany [a], Manish Jain [b], Murilo L. Tiago [c,*], Yunkai Zhou [d], Yousef Saad [e], James R. Chelikowsky [c]

[a] *Departamento de Física de la Materia Condensada, Facultad de Física, Universidad de Santiago de Compostela, E-15782 Santiago de Compostela, Spain*
[b] *Corporate Research Materials Laboratory, 3M Company, St Paul, MN 55144, USA*
[c] *Institute for Computational Engineering and Sciences, University Station, University of Texas at Austin, Austin, TX 78712, USA*
[d] *Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA*
[e] *Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455, USA*

Received 14 February 2007; received in revised form 10 April 2007; accepted 11 April 2007

Available online 27 April 2007

Abstract

We have recently presented a real-space method for electronic-structure calculations of periodic systems that is based on the Hohenberg–Kohn–Sham density-functional theory. The method allows the computation of electronic properties of periodic systems in the spirit of traditional plane-wave approaches. In addition, it can be implemented efficiently on parallel computers. Here we will show that the method's inherent parallelism, in conjunction with a newly designed approach for solving the Kohn–Sham equations, enables the accurate study of the ionic and electronic properties of periodic systems containing thousands of atoms from first principles.

PACS: 71.15.-m; 71.15.Dx; 71.15.Mb

Keywords: Electronic structure; Density-functional theory; Pseudopotentials; Iterative diagonalization; Real-space method

1. Introduction

Solving the ionic and electronic structure of matter from first principles is of great interest both from scientific and technological points of view. During the last decades, one of the most successful ways of obtaining such information has been the use of computational approaches based on the Hohenberg–Kohn–Sham density-functional theory (DFT) [1,2]. However, although DFT simplifies the problem enormously, the size of systems susceptible to current quantum computation methods is limited. As such, the development of efficient DFT-based methods is crucial for solving large-scale problems in condensed matter physics. Of special interest are those methods that can take advantage of massively parallel architectures. These architectures offer substantial improvements in terms of solution time and memory requirements.

Plane-wave pseudopotential methods have been widely used for electronic structure calculations of periodic systems [3]. Pseudopotential theory allows one to focus on the chemically active valence electrons by replacing the strong all-electron atomic potential by a weak pseudopotential, which effectively reproduces the effects of the core electrons on the valence states. This approximation significantly reduces the number of eigenpairs to be handled, especially for heavier elements. Moreover, since the core wave functions and the core oscillatory region of the valence wave functions are removed, the use of simple basis functions such as plane waves is straightforward. Expanding the electronic wave functions with respect to a plane-wave basis is the natural way of representing a system with periodicity (boundary conditions are periodic) and offers a number of advantages. We mention two crucial advantages: the basis does not depend on atomic positions; and only one

[*] Corresponding author.
E-mail address: mtiago@ices.utexas.edu (M.L. Tiago).

0010-4655/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2007.04.003

parameter, the wavelength of the highest Fourier mode used in the expansion, needs to be refined to control convergence. However, plane-wave codes make use of fast Fourier transform (FFT) to perform the matrix–vector product between the Hamiltonian matrix and trial wave vectors. Since FFTs involve nonlocal operations, its efficiency on parallel computer architectures is diminished by the need for global communications among processors.

During the past decade, there has been increasing interest in developing real-space pseudopotential methods [4]. Such methods have a number of points in their favor. First, implementation of these approaches is simple: there is no "formal" basis, calculations being performed directly on a real-space grid that does not depend on ion positions. The spacing of the grid is refined until the calculation converges. The grid spacing plays the role of the cutoff energy in the plane-wave approach. Second, real-space methods are semi-local, which facilitates implementation on parallel computers. In short, real-space methods not only share the main advantages of plane wave representations, but they can also be easily parallelized. This makes such methods highly attractive for computation of the electronic ground states of large, complex systems.

Although real-space methods are naturally formulated for localized systems (there is no periodicity implicit in the basis), their application is not limited to such systems [5–7]. Recently, we proposed a real-space pseudopotential approach for self-consistent first-principles calculations of periodic systems [7]. The method was presented as an approach which offers the same degree of accuracy as traditional plane-wave approaches, and that can also perform efficiently on parallel computer architectures following its inherent parallelism. In this paper we go in detail through the method illustrating its capabilities. In particular, we describe its *parallel version* and a newly implemented approach to solve the Kohn–Sham equations that avoids explicit diagonalization of the hamiltonian in each cycle of the self-consistent loop. With all, the aim of this paper is to present a method capable of addressing from first-principles challenging problems involving periodic systems. In Section 2 below we describe the formalism of the method and the numerical algorithms used to solve it, in Section 3 we illustrate its performance on different test problems, and in Section 4 we summarize our main conclusions.

2. Description of the method

According to DFT [1,2], the total energy E_{tot} of a system comprising electrons and ions (the latter in positions $\{\mathbf{R}_a\}$) can be written as a unique functional of the electron density ρ,

$$E_{\text{tot}}[\rho] = T[\rho] + E_{\text{ion}}\big(\{\mathbf{R}_a\}, [\rho]\big) + E_H[\rho] + E_{\text{xc}}[\rho] + E_{\text{ion–ion}}\big(\{\mathbf{R}_a\}\big), \tag{1}$$

where $T[\rho]$ is the kinetic energy, $E_{\text{ion}}(\{\mathbf{R}_a\}, [\rho])$ is the electron–ion energy, $E_H[\rho]$ is the electron–electron Coulomb energy or Hartree potential energy, $E_{\text{xc}}[\rho]$ is the exchange-correlation energy, and $E_{\text{ion–ion}}(\{\mathbf{R}_a\})$ is the classical electrostatic energy among the ions. Finding the electron density that minimizes the energy functional is equivalent to solving the set of one-particle Kohn–Sham equations

$$\left[-\frac{\nabla^2}{2} + V_{\text{ion}}(\mathbf{r}) + V_H(\mathbf{r}) + V_{\text{xc}}(\mathbf{r})\right]\psi_n(\mathbf{r}) = \epsilon_n\psi_n(\mathbf{r}) \tag{2}$$

and setting

$$\rho(\mathbf{r}) = \sum_n \big|\psi_n(\mathbf{r})\big|^2, \tag{3}$$

where the sum runs over the occupied states. V_{ion} and V_H are the ionic and Hartree potentials, respectively, and $V_{\text{xc}} = \delta E_{\text{xc}}/\delta\rho$. Here and in the rest of the text we use atomic units ($e = m = \hbar = 1$) unless otherwise stated. Solving Eqs. (2) and (3) requires finding a self-consistent solution for the charge density.

Structure optimization and molecular-dynamics simulations require accurate calculation of the ionic forces $\{\mathbf{F}_a\}$. If the system has been brought to the Born–Oppenheimer surface (i.e. if the single-particle wave functions are very close to the exact eigenstates), the forces can be calculated from the Hellmann–Feynman theorem [8],

$$\mathbf{F}_a = -\frac{\partial E_{\text{tot}}}{\partial \mathbf{R}_a}. \tag{4}$$

2.1. Setting up the Kohn–Sham equations

We represent wave functions, the electron density and potentials on a uniform, orthogonal three-dimensional real-space grid. For simplicity, we assume the grid to be cubic, but the extension to a general orthorhombic grid is straightforward. In order to construct the grid, only two parameters need to be specified: the grid spacing h (the distance between adjacent points in each of the three Cartesian directions) and the size L of the unit cell or supercell described by the cubic grid. The grid is then generated by the points

$$\mathbf{r}(i, j, k) \equiv (x_i, y_j, z_k) = (ih, jh, kh), \tag{5}$$

with the integers i, j and k running from 1 to $N_{\text{grid}} = L/h$. The system is made periodic by replicating the unit cell and the atoms it contains (the basis) throughout space, as illustrated in Fig. 1. Here we assume that all the atoms belong to the same species.

In order to model Eq. (2) on the real-space grid we use a higher-order finite difference expansion [9] for the Laplacian operator. We approximate the partial derivatives of the wave function at a given point of the grid by a weighted sum over its values at that point and its neighbors. The second partial derivative in the x-direction, for example, has the form

$$\left.\frac{\partial^2\psi}{\partial x^2}\right|_{\mathbf{r}(i,j,k)} = \sum_{n=-N}^{N} C_n \psi(x_i + nh, y_j, z_k), \tag{6}$$

where N is the order of the expansion (typically 6 in order to ensure convergence). Under the assumption that the wave function can be approximated accurately by a power series in h, this approximation is accurate to $\mathrm{O}(h^{2N+2})$. Algorithms are available that compute the coefficients C_n for arbitrary order in h [10].

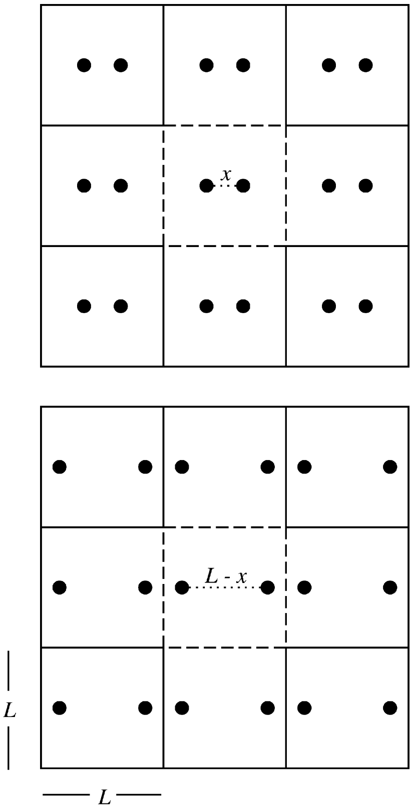

Fig. 1. Two-dimensional illustration of a supercell representation. Upper panel: supercell of size L (dashed frame) containing a diatomic molecule with interionic distance x. Lower panel: supercell of size L (dashed frame) containing a diatomic molecule with interionic distance $L - x$. (The molecules are oriented along the $\hat{x}$ direction of the supercells.) Owing the periodic nature of the supercell representation (the supercell and the basis are replicated trough the space) both examples correspond to the same physical system.

In each iteration of the algorithm for self-consistent solution of the Kohn–Sham equations, the Hartree and exchange-correlation potentials are set up directly on the real-space grid using the approximation to the electron density obtained in the previous iteration. For V_{xc} we use the local density approximation, according to which the value of V_{xc} at a given point is a function of the electron density at that point. To construct V_H, we solve the Poisson equation $[\nabla^2 V_H(\mathbf{r}) = -4\pi\rho(\mathbf{r})]$ using the matrix formalism corresponding to the higher-order finite difference method, first setting the total charge in the supercell to zero in order to prevent the system from becoming infinitely charged due to the required periodicity. In particular, we opt to use a readily available conjugate gradient solver from the sparse matrix package SPARSKIT (the sparsity of the square matrix representation of the Laplacian operator is clear from the values we normally use for N, see above) [11]. In its non-prediconditioned form, this algorithm accesses the matrix only by performing matrix–vector products. In fact, the matrix is not explicitly required, the matrix–vector products being done by using the finite difference "stencil" of the operator which acts directly on the input vector. This results in significant savings in memory usage.

The remaining potential term in Eq. (2), the ionic term, is determined using pseudopotential theory. We employ nonlocal norm-conserving ionic pseudopotentials cast in the Kleinman–Bylander form [12]. The ionic contribution due to one atom of the system, V^a_{ion}, is obtained as the sum of a local term and a nonlocal term, the latter corresponding to an angular-momentum-dependent projection [12,13]. Its effect on the wave function in Eq. (2) is

$$V^a_{\mathrm{ion}}(\mathbf{r})\psi_n(\mathbf{r}) = V_{\mathrm{loc}}(r_a)\psi_n(\mathbf{r}) + \sum_{lm} G^a_{n,lm} u_{lm}(\mathbf{r}_a)\Delta V_l(r_a), \tag{7}$$

where $\mathbf{r}_a = \mathbf{r} - \mathbf{R}_a$; u_{lm} is the atomic pseudopotential wave function corresponding to the angular momentum quantum numbers l and m; $\Delta V_l = V_l - V_{\mathrm{loc}}$ is the difference between V_l (the lth component of the ionic pseudopotential) and the local potential V_{loc}; and the projection coefficients $G^a_{n,lm}$ given by

$$G^a_{n,lm} = \frac{1}{\langle \Delta V^a_{lm} \rangle} \int u_{lm}(\mathbf{r}_a)\Delta V_l(r_a)\psi_n(\mathbf{r})\,\mathrm{d}^3r \tag{8}$$

include the normalization factor

$$\langle \Delta V^a_{lm} \rangle = \int u_{lm}(\mathbf{r}_a)\Delta V_l(r_a)u_{lm}(\mathbf{r}_a)\,\mathrm{d}^3r. \tag{9}$$

The local and nonlocal terms in Eq. (7) must in principle be evaluated and accumulated for all the atoms in the system, i.e. for both the atoms in the basis and their periodic images (see Fig. 1). However, the summation of nonlocal terms is actually performed over a finite number of atoms because, at distances greater than the pseudopotential core radius (a fraction of a bond length) V_l is $-Z/r$ for all l, where Z is the number of electrons acting as valence electrons in the pseudopotential (see the left-hand side panel of Fig. 2); this makes ΔV_l short-ranged, so that the nonlocal terms need only be evaluated for atoms belonging to the basis (and possibly its nearest replicas). Furthermore, the integrals in Eqs. (8) and (9) can be efficiently calculated in real space by direct summation over the grid points surrounding each atom.

The situation is different for the local contribution to the ionic potential, which involves a divergent summation of the long-range Coulomb term $-Z/r$. However, this divergence can be avoided by making use of the fact that the pseudopotentials are short-ranged functions in reciprocal space (see the right-hand panel of Fig. 2). The local ionic potential, $V_{\mathrm{ion,loc}}$, can be calculated efficiently in reciprocal space and transferred to the real-space grid by an FFT. We obtain the local ionic potential in reciprocal space as in a plane wave calculation with an energy cutoff of $\pi^2/2h^2$, the cutoff for which FFTs of the wave functions and potentials require a grid of size N^3_{grid} [14]. We first calculate the structure factor $S_{\mathrm{ion}}(\mathbf{q})$ at wave vector $\mathbf{q} = (2\pi/L)(n_x, n_y, n_z)$ where n_x, n_y and n_z are integers,

$$S_{\mathrm{ion}}(\mathbf{q}) = \sum_a \exp(i\mathbf{q}\cdot\mathbf{R}_a), \tag{10}$$

where the sum is taken over the positions of all the atoms in a single unit cell [15]. $V_{\mathrm{ion,loc}}$ is then calculated as

$$V_{\mathrm{ion,loc}}(\mathbf{q}) = S_{\mathrm{ion}}(\mathbf{q})V_{\mathrm{loc}}(q) \tag{11}$$

and transferred to the real-space grid by FFT. Note that we need to perform this transformation once, just before we enter the

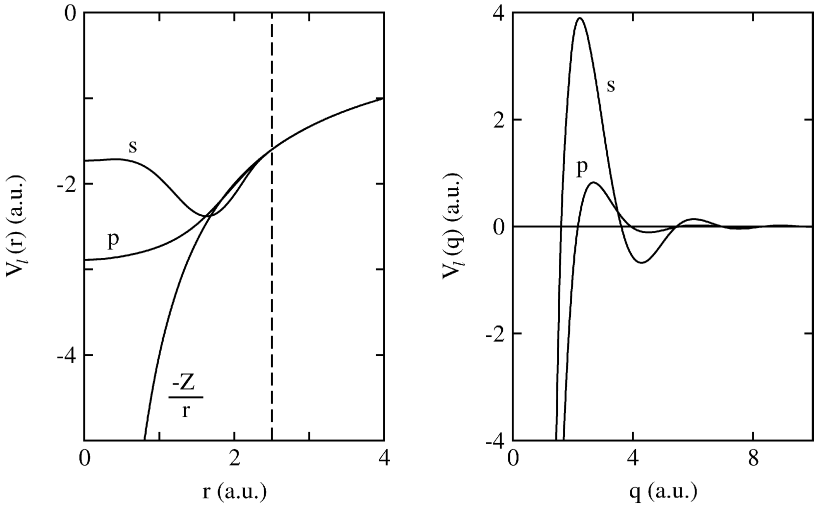

Fig. 2. Real-space (left panel) and reciprocal-space (right panel) representations of the silicon pseudopotentials employed in the tests of our real-space approach. The pseudopotentials were constructed using the Troullier–Martins prescription (Ref. [25]); the dashed line in the left panel corresponds to the core radius used in their generation, 2.5 a.u.

loop for self-consistent solution of the Kohn–Sham equations; since the local ionic potential is determined by the positions of the ions, it does not change during the process of finding a self-consistent solution for ρ.

When discretized as above, Eq. (2) adopts the form

$$-\frac{1}{2}\Bigg[\sum_{n_1=-N}^{N} C_{n_1}\psi_n(x_i+n_1h, y_j, z_k) + \sum_{n_2=-N}^{N} C_{n_2}\psi_n(x_i, y_j+n_2h, z_k) + \sum_{n_3=-N}^{N} C_{n_3}\psi_n(x_i, y_j, z_k+n_3h)\Bigg] + \big[V_{\text{ion}}(x_i, y_j, z_k) + V_H(x_i, y_j, z_k) + V_{\text{xc}}(x_i, y_j, z_k)\big]\psi_n(x_i, y_j, z_k) = \epsilon_n\psi_n(x_i, y_j, z_k). \tag{12}$$

Since ΔV_l differs from zero only inside the pseudopotential core radius and the Laplacian operator extends only to a few neighbors around each grid point, the matrix representation of Eq. (12) is very sparse.

The parallel implementation of the method detailed above uses the Message Passing Interface (MPI) standard [16]. All processors work during the most computationally intensive parts, i.e. obtaining the eigenvalue/eigenvector pairs and solving the Poisson equation. In order to obtain a good load balancing among processors, we opt for a domain decomposition approach. Within this approach the rectangular physical domain is naturally divided in nearly equally-sized sub-cubes that are ascribed as working units or local subdomains one by one to the processors. They also compute the nonlocal component of the forces. This allows for good memory balance since all the major arrays, which store wave functions, are distributed.

2.2. Solving the Kohn–Sham equations

The bottleneck of self-consistent DFT-based methods is in the solution of the Kohn–Sham equations (Eq. (2)). The ability of such methods to make use of efficient parallel algorithms for extracting the eigenvalue/eigenvector pairs is essential for tackling complex problems in condensed-matter physics. Traditional plane-wave representations of the Kohn–Sham equations require intensive use of FFTs for their solution. This implicitly entails a serious degradation of the performance of such codes on massively parallel environments since FFTs' need for global communications among processors. In constrast, our method only needs one FFT for the local part of the pseudopotential. During the solution of Eq. (12), there is little communication among processors. Two approaches can be followed in order to solve the Kohn–Sham equations: explicit diagonalization, using for instance the ARPACK package, or subspace filtering techniques, which are very efficient if an initial guess of eigenvectors is available. In either approach, the hamiltonian matrix is never stored explicitly. Instead, it is defined implicitly through matrix–vector products. We give details in the following.

ARPACK (and its parallel version PARPACK) [17] is an iterative method which requires the matrix only in the form of matrix–vector products. When the matrix is symmetric, as in Eq. (12), this package is based upon an algorithm variant of the Arnoldi process called the implicit restarted Lanczos method (for more details on the method consult Ref. [17]). In order to obtain the eigenvalues and eigenvectors, a loop until convergence is performed with successive calls to the PARPACK's routine, *pdsaupd*, which requires a reverse communication interface. The matrix–vector routine is provided by us [18]. The construction of the Arnoldi factorization inside *pdsaupd* requires only two communication points: computation of the norm of the distributed residual vector and its orthogonalization to the basis vectors. The matrix–vector multiplication takes three steps itself [19]. First, the contribution of the diagonals in Eq. (12) (i.e. the potential and Laplacian diagonal) are computed in parallel on all processors according to the partitioning of the physical domain explained above. Secondly, the contribution to the Laplacian is considered on the rows of each processor by using the stencil information. Since some of the neighbors of the local subdomain may reside on different processors, communication between nearest-neighbors processors is necessary. During the preprocessing phase, each processor locate which of their rows are needed in the stencils of other processors. In the second step of the matrix–vector multiplication this information is exchanged among the processors and the stencil multiplication proceeds completely in parallel. Finally, each of the rank-one updates of the nonlocal components is computed as a distributed dot product; this last step first computes all local dot products and then globally sums their values. Thus, matrix–vector multiplication requires just two communication/synchronization points among the processors.

Since Eq. (12) must be solved self-consistently, a typical numerical solution involves calculating eigenvalues and eigenvectors of the Kohn–Sham hamiltonian for an initial charge density and using them to construct a better charge density and hamiltonian. The cycle is iterated until the difference between the input and output charge densities is smaller than an accuracy threshold. If a new cycle is necessary, one can then use the recently calculated eigenpairs as initial guess for the eigenpairs of the hamiltonian in the new cycle, and refine them for the new hamiltonian. This is the basic idea behind subspace filtering [20].

Subspace filtering can be efficiently implemented by using Chebyshev polynomials of the first kind [21]. These polynomials are known for their rapid growth property: the k-order polynomial $C_k(t)$ is bound in the $[-1, 1]$ interval but it quickly diverges if $|t| > 1$, the growth speed increasing with order. Consider a hamiltonian for which the N lowest eigenvectors span a subspace $\mathcal{S}$. Also, assume that the highest possible eigenvalue is b and the highest eigenvalue to be computed is a. Operating a polynomial $p_k(H)$ where

$$p_k(H) = C_k\left(\frac{1}{b-a}[2H - b - a]\right) \tag{13}$$

on an arbitrary vector effectively suppresses its projection outside $\mathcal{S}$. It is important to mention that the filter polynomial $p_k(H)$ only suppresses the projection onto the subspace associated to the energy interval $[a, b]$. If b is not an upper bound for eigenvalues, then components with very high energy are also magnified, which can produce unwanted effects.

In practice, subspace filtering can be easily implemented once good estimates for the bounds a and b are available. The lower bound can be adjusted for the desired number of eigenvalues, and it is simply taken from the eigenvalues calculated in the previous iteration. The upper bound can be estimated in various ways [20,22]. In the present implementation, we use an inexpensive Lanczos process with a safeguard. The polynomial order is adjustable, and we observed that choices $8 < k < 20$ usually give very good performance. More details about this subspace filtering technique can be found in Refs. [20,22]. Applying $p_k(H)$ to wave functions is easily done with successive matrix–vector operations. The performance of subspace filtering is very good in parallel environment, and it can be more than one order of magnitude faster than explicit diagonalization, as we discuss below.

2.3. Calculation of the forces

The total ground-state energy (Eq. (1)) is given by

$$E_{\text{tot}}[\rho] = T[\rho] + \int \rho(\mathbf{r}) V_{\text{ion,loc}}(\mathbf{r})\, \mathrm{d}^3 r + \sum_{a,n,lm} \langle \Delta V^a_{lm}\rangle \left[G^a_{n,lm}\right]^2 + E_H[\rho] + E_{\text{xc}}[\rho] + E_{\text{ion–ion}}\big(\{\mathbf{R}_a\}\big) + \alpha, \tag{14}$$

where the sum on n is performed over the occupied states and α is the contribution of the non-Coulomb part of the pseudopotential at $\mathbf{q} = 0$,

$$\alpha = \frac{Z N_{\text{a}}^2}{L^3} \int \left(V_{\text{loc}}(r) + \frac{Z}{r}\right) 4\pi r^2\, \mathrm{d}r. \tag{15}$$

From Eq. (4), the force on ion a is

$$\mathbf{F}_a = -\int \rho(r) \frac{\partial V_{\text{loc}}(r_a)}{\partial \mathbf{R}_a}\, \mathrm{d}^3 r - 2 \sum_{n,lm} \langle \Delta V^a_{lm}\rangle G^a_{n,lm} \frac{\partial G^a_{n,lm}}{\mathbf{R}_a} - \frac{\partial E_{\text{ion–ion}}}{\mathbf{R}_a}. \tag{16}$$

The first term on the right-hand side of Eq. (16) is the contribution from the local ionic potential, $\mathbf{F}_{a,\text{loc}}$. It involves the integral of a long-range function (Z/r^2), but it is easily calculated in reciprocal space, where there is no long-range tail [23],

$$\mathbf{F}_{a,\text{loc}} = -iL^3 \sum_{\mathbf{q}} \mathbf{q} \exp(i\mathbf{q} \cdot \mathbf{R}_a) V_{\text{loc}}(q) \rho(\mathbf{q}), \tag{17}$$

where $\rho(\mathbf{q})$ is obtained by an FFT from the solution of the Kohn–Sham equations on the real-space grid. The other electronic contribution to the force is due to the nonlocal components of the pseudopotential. Taking advantage of its short range, we calculate this term in real space. The remaining term in Eq. (16) is the force exerted on the ion by other ions. As usual for periodic systems [3], we evaluate this term by performing two convergent summations, one over lattice vectors and the other over reciprocal lattice vectors, using Ewald's method [23].

The procedure we use to evaluate the expression given in Eq. (16) gives very accurate values of the ionic forces, as we demonstrate in Section 3. Note that Eq. (16) contains no term representing the derivative of the basis set with respect to the position of the ion (the “Pulay force” [24]).

3. Test of the method

3.1. Periodic boundary conditions

As a first test of our method, we studied the silicon dimer. A diatomic molecule is an appropriate system for testing the implementation of the periodic boundary conditions because its geometry is controlled by just one degree of freedom: the distance between ions or bond length. There are two main advantages associated to this system. First, the adequate implementation of the periodic boundary conditions for the ground-state energy (Eq. (14)) can be easily checked. Suppose we obtain the energy of the dimer as a function of the bond length for supercells of increasing size L. Since the interactions between the atoms that conform the basis and their replicated images diminish as L increases (more vacuum is put in between them, see Fig. 1), the energy curves should approach the energy curve obtained for the isolated molecule, which can be obtained with the implementation of the finite-difference real-space pseudopotential approach without periodic boundary conditions [13]. Secondly, the force acting on each ion of the dimer can be calculated "formally" by taking the numerical derivative of the total energy of the dimer with respect to the bond length. The resulting force can be compared with the one obtained from the evaluation of the explicit Hellmann–Feynman expression of Eq. (16). This comparison will allow us to test the accuracy of the procedure that we employ to calculate the ionic forces.

For each size of the supercell considered, the real-space grid was constructed with a spacing of $h = 0.6$ a.u. The core electrons were represented by norm-conserving pseudopotentials generated for the reference configuration [Ne]$3s^2 3p^2$ using the Troullier–Martins prescription [25] (see Fig. 2), with a radial cutoff of 2.5 a.u. for both s and p angular components. The potential was made separable by the procedure of Kleinman and Bylander [12], with the s potential chosen to be the local component. The local density functional of Ceperley and Alder [26] was used as parameterized by Perdew and Zunger [27], and the single Γ point was employed in sampling the Brillouin zone [28], as is appropriate for calculations on non-extended systems.

The binding energy curves obtained as a function of the bond length for supercells of size 12.0, 13.2, 14.4, 15.6, and 16.8 a.u. (from bottom to top; solid lines) are plotted in Fig. 3. The binding energy of the isolated molecule obtained using the same pseudopotential and grid used in the supercell calculations is also shown (dashed line) [13]. Binding energies were computed by subtracting the energy of the single atoms, calculated within the same approach, from the energy of the molecule. Fig. 3 confirms that our real-space method is consistent with a supercell representation. The curves obtained within the supercell representation clearly trend to the curve obtained for the isolated molecule when the size of the supercell increases. The binding energy and equilibrium bond length of the silicon dimer as obtained from our theoretical calculations are 0.178 and 4.16 a.u., respectively. These values can be compared with available experimental data for the silicon dimer, 0.110 and 4.23 a.u. [29]. The value for the bond length is consistent with the value reported in a previous work performed at the same level of approximation to DFT considered here [30], and agrees well with experiment. The result for the cohesive energy shows the typical overbinding associated with the local-density approximation [29]. Our motivation here is not to improve on this formalism, but to test the implementation of the periodic boundary conditions on our real-space method. Nevertheless, the current approach is not limited to the local-density approximation. Working with generic, orbital-independent functionals is straightforward since the local Hamiltonian is built only once.

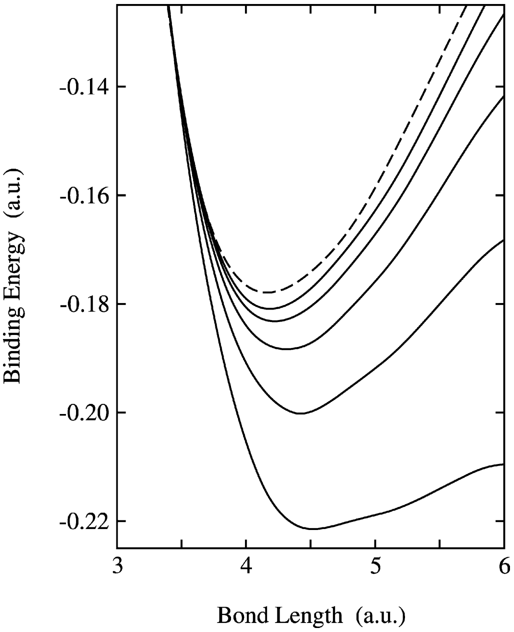

Fig. 3. Binding energy curves of the silicon dimer as obtained from our real-space code (solid lines). The energy curves correspond to calculations done for supercells of size (from bottom to top) 12.0, 13.2, 14.4, 15.6, and 16.8 a.u. The binding energy curve obtained for the isolated molecule is also shown (dashed line).

The calculation of the total energy of dimers with interionic distance x and $L - x$ within our approach should give the same result since both problems correspond to different representations of the same physical system (see Fig. 1). This means that the energy curves obtained within our approach must be symmetric with respect to $L/2$. Although we do not show in Fig. 3 results for energy curves at interionic distances bigger than $L/2$, the obtained curves are perfectly symmetric. However, it is interesting to notice how the shape of the energy curves obtained for the smaller supercells (lower part of Fig. 3) clearly depart from a parabolic-like shape and try to "match" the required symmetry when the interionic distances approach $L/2$. Such a "deformation" in the shape of the curves can be used to perform a stringent test of the accuracy of the method we employ for evaluating the ionic forces. In Fig. 4, we plot the ionic forces corresponding to the supercell calculations with $L = 12$ a.u. as obtained from the numerical derivative of the energy curve (bottom line in Fig. 3), and from the evaluation of the explicit Hellmann–Feynman expression of Eq. (16). The agreement between the ionic forces is very good (differences are less than 0.003 a.u.), keeping in mind the errors that are inherent to the three point rule employed in the evaluation of the derivative [31]). Both curves cross the zero constant-value line at 4.52 a.u. and 6.00 a.u. The first interionic distance exactly corresponds to the equilibrium bond length as extracted from the binding energy curve; the second one corresponds to $L/2$, in according to the extreme value (in this particular case a maximum) that the energy curve has at $L/2$ due to its required symmetry.

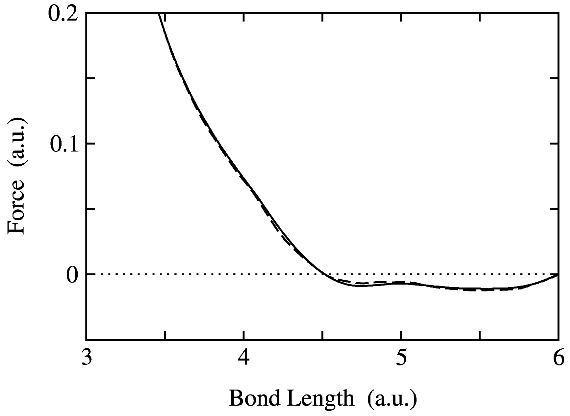

Fig. 4. Ionic force of the silicon dimer as obtained from our real-space code (Eq. (16)) for a supercell of size 12.0 a.u. (solid line). The ionic force obtained from the numerical derivative of the corresponding energy curve (bottom line in Fig. 3) is also shown (dashed line).

3.2. Efficiency

In order to test the performance of the parallel implementation of our code, we study a system composed of a cubic supercell of size 29.7 a.u. containing 216 silicon atoms. It has been recently shown that first-principles molecular-dynamic techniques are suitable for the calculation of the static and dynamic properties of liquids when the number of atoms involved in the simulation is of the order of hundreds [32]. In this size range of simulations, an effective use of parallel architectures is very advantageous. The supercell was created by randomly disrupting a 216-atom simple cubic cell in order to mimic the liquid state. (This also avoids a symmetric distribution of load among the processors.) The number of particles and size of the supercell chosen give a number density that corresponds to the experimental number density of liquid silicon close to its melting point. The real-space mesh was constructed for a grid spacing of 0.7 a.u., and the Brillouin zone was sampled at the Γ point (this sampling was shown to be adequate for describing the liquid state in a recent first-principles molecular-dynamic simulations involving a system of similar size than that considered here [32]). Our test involved a matrix of size 74 088 requiring 480 eigenpairs, and it was executed in two IBM power4 nodes (each node containing thirty-two 1.7 GHz-processors) of the Minnesota Supercomputer Institute (http://www.msi.umn.edu). As one can see in Fig. 5, the deviation of the obtained speed up curve from the ideal behavior only starts to be appreciable when a large number of processors are used. It is in this region of the plot where the communication cost between processors starts to outweigh the effectiveness of dividing the supercell into local subdomains assigned to processors.

The cohesive energy of this system was calculated for various choices of grid spacing, and it is shown in Fig. 6. As expected, explicit diagonalization and subspace filtering provide essentially the same cohesive energy. In addition, both methods are consistent with results obtained with a plane wave-based DFT code [33] and employing the same pseudopotential and parameterization [27] of the local density functional [26]. Ideally, real-space and plane-wave calculations should predict exactly the same cohesive energy when the precision parameter (grid spacing in the former, energy cut-off in the latter) is converged. From the properties of Fourier transforms, these two parameters are linked to each other by the approximate relationship $(\pi/2h)^2 < 2E_{\text{cut}} < (\pi/h)^2$. Fig. 6 shows that the rate of convergence follows the rule above.

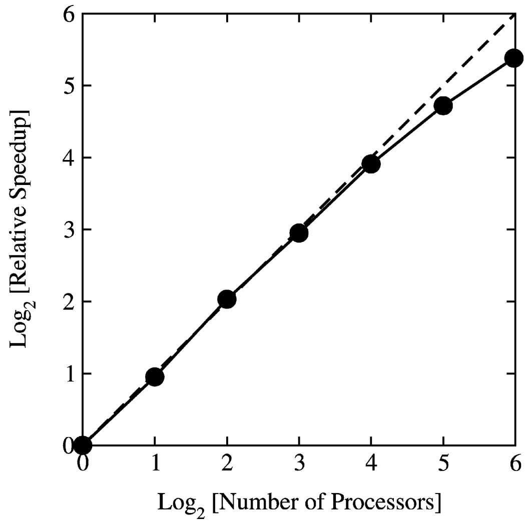

Fig. 5. Relative speedup of the parallel implementation of our real-space method (see the text).

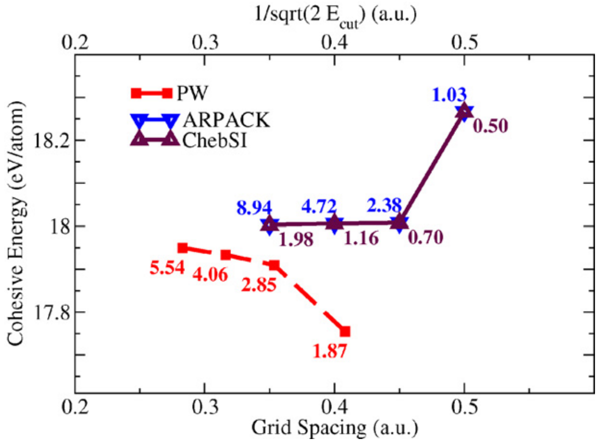

Fig. 6. Cohesive energy of the 216-atom silicon system, calculated using explicit diagonalization ("ARPACK") and subspace filtering ("CheFSI"), for various choices of grid spacing. For comparison, results obtained using a plane-wave code ("PW") are also shown. The numbers correspond to the run-times (in hours) obtained for each of the tests (see the text).

Table 1 shows the time spent during each self-consistent iteration. Explicit diagonalization shows very little decrease in run-time between the first iteration and the subsequent ones. In contrast, subspace filtering shows a speedup of more than one order of magnitude, which clearly shows the advantages of avoiding explicit diagonalization after the first iteration. Similar performance improvements have been observed in other test systems [22]. Calculations presented in Table 1 were obtained using a single IMP Power 3 node (16 processors, clock speed 375 MHz), located at the National Energy Research Scientific Computing Center (NERSC), http://www.nersc.gov/.

Table 1
Run time per self-consistent iteration for the 216-atom silicon system

Iteration #	Run-time per iteration (sec)	
	ARPACK	CheFSI
1	3612.75	3765.89
2	3599.85	86.35
3	3613.14	88.61

Exact diagonalization was done using the ARPACK package. Subspace iteration ("CheFSI") was performed starting from an initial exact diagonalization.

3.3. Addressing challenging problems

There are problems of fundamental interest that can only be addressed within supercell approaches. However, in some of these problems, the use of supercell approaches introduce critical limitations in their solution that are related to the small number of atoms that can be treated within the approaches. One of such problems is the characterization of defects and dopants in semiconductor bulk materials. The supercell approach entails the interaction between the defect or the dopant and its periodic images (see Fig. 1). In case the supercells chosen were not large enough to minimize such spurious interaction, the computational approach can significantly alter or fake the properties of interest under study.

In order to illustrate the finite-size errors introduced by the supercell approach we consider the study of bulk indium phosphide (InP) doped with zinc (Zn) [34]. This material is of great technological interest since it constitutes one of the most common materials in use within optoelectronics. The substitution of an In atom by a Zn, $\mathrm{Zn_{In}}$, gives an electrically active acceptor state that experiment locates 35 meV above the valence band edge (VBE) of the host material (the impurity has one less valence electron than the substituted atom, giving a hole as an electrical carrier). The characterization of such a shallow impurity state from first-principles is very challenging. Shallow states mainly consist of wavefunctions of the host bulk, thus being greatly delocalized in real space. Consequently, non-trivial precautions must be taken to avoid (or at least minimize) the spurious coupling between impurity wavefunctions corresponding to neighboring supercells.

Placing the $\mathrm{Zn_{In}}$ impurity in cubic supercells with different sizes resembling the host InP crystal allows us to take proper account of finite-size errors. In Fig. 7 we plot the charge density associated to the acceptor impurity state introduced by $\mathrm{Zn_{In}}$ as obtained from our real-space computational approach. In these calculations, the local density approximation is used for the exchange and correlation potential [26,27], and the Brillouin zone corresponding to each of the supercells is sampled at the Γ point. It is apparent from the figure that "small" supercells containing a few hundreds of atoms lead to significant overlap of the impurity wavefuntion and their images. Increasing the size of the supercell minimizes the coupling between states. In order to mimic the behavior of a truly isolated $\mathrm{Zn_{In}}$ impurity, we need to consider supercells containing thousands of atoms in our calculations. The results obtained for the position of the impurity state within the host band gap is in according with this discussion. The 216 and 512-atom supercells locate the impurity acceptor state 93 and 66 meV above the VBE, respectively [34]. The value obtained for the 2744-atom system is 39 meV, in rather good agreement with experiment (35 meV).

It is worth mentioning that the reported values for the binding energy of the $\mathrm{Zn_{In}}$ impurity state are well converged values within DFT. The different cubic supercells of size L employed to mimic the bulk crystal are obtained replicating the standard cubic cell (which contains 8 atoms) n times in each of the cartesian directions ($L = na$, with a being the bulk lattice constant of InP in the zinc-blende structure). Thus, sampling the Brillouin zone of each of the supercells at the Γ point corresponds to sampling the Brillouin zone of the standard cell over an equally spaced mesh of $n \times n \times n$ points centered at the Γ point. This

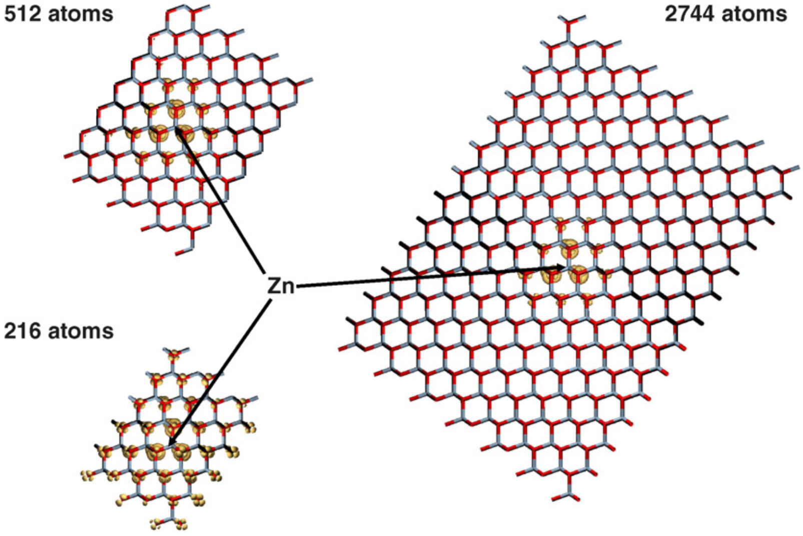

Fig. 7. Charge density associated to impurity state introduced in bulk InP by Zn-doping as obtained from our real-space first-principles approach for different sizes of the supercell. Gray (red) symbols stand for In (P) atoms. The charge density is plotted at the 20% of its maximum value in all the cases. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

gives very well converged values of the electronic band structure of the crystal. For example, the band gap energy calculated for the 216-atom supercell (the Brillouin zone of the standard cell is sampled over a mesh of $3 \times 3 \times 3$ points) agrees to better than 4 meV with the converged value. For the 512-atom supercell, such agreement is better than 1 meV (the sampling of the Brillouin zone is performed over $4 \times 4 \times 4$ points). We have also investigated the influence of the choice of exchange and correlation functional on the results. We have repeated the calculations for some of the systems studied including gradient corrections in the density functional representation of the potential. In particular, considering the Perdew–Burke–Ernzerhof form of the generalized gradient approximation for the exchange an correlation potential [35] we obtain binding energies of the impurity state that agree to better than 1 meV with those reported here.

4. Summary and conclusions

This paper presents a method for self-consistent DFT-calculations for the electronic structure of periodic systems. The method employs pseudopotentials to construct the electron-ion potential, and solves the Kohn–Sham equations on a uniform real-space grid. The only FFT performed is used to set up the local ionic potential on the real-space grid following its obtention in reciprocal space. Calculation of accurate forces on ions has also been implemented, which allows for the computation of ground state geometries and molecular dynamic simulations at finite temperature within DFT.

The Kohn–Sham equations are solved by subspace filtering using Chebyshev polynomials. This technique avoids direct diagonalization of the hamiltonian in each of the iterations of the self-consistent loop (but for the first one), which explains the significant speed-up of the code relative to diagonalization-based methods. Subspace filtering in combination with the method's inherent parallelism, makes it a tool capable of addressing challenging problems involving thousands of atoms from first-principles.

Acknowledgements

We would like to thank J.L. Martins and E.L. de la Grandmaison for helpful discussions. This work was funded by the National Science Foundation under grants ITR-0551195, ITR-0428774, and DMR-0551195, by the U.S. Department of Energy under grants DE-FG02-06ER46286 and DE-FG02-06ER15760, and by the Spanish Ministry of Education and Science (Program "Ramón y Cajal"). Computational support was provided by the National Energy Research Scientific Computing Center (NERSC), by the Minnesota Supercomputer Institute (MSI), by the Barcelona Supercomputer Center (BSC), and by the Galician Supercomputer Center (CESGA).

References

[1] P. Hohenberg, W. Kohn, Phys. Rev. B 136 (1964) 864.

[2] W. Kohn, L.J. Sham, Phys. Rev. A 140 (1965) 1133.

[3] See, for example, W.E. Pickett, Comp. Phys. Rep. 9 (1989) 115;
M.C. Payne, M.P. Teter, D.C. Allan, T.A. Arias, J.D. Joannopoulos, Rev. Mod. Phys. 64 (1992) 1045.

[4] See, for example, T.L. Beck, Rev. Mod. Phys. 72 (2000) 1041.

[5] E.L. Briggs, D.J. Sullivan, J. Bernholc, Phys. Rev. B 54 (1996) 14362.

[6] N.A. Modine, G. Zumbach, E. Kaxiras, Phys. Rev. B 55 (1997) 10289.

[7] M.M.G. Alemany, M. Jain, L. Kronik, J.R. Chelikowsky, Phys. Rev. B 69 (2004) 075101.

[8] H. Hellmann, Einführung in die Quantumchemie, Deuticke, Leipzig, 1937;
R.P. Feynman, Phys. Rev. 56 (1939) 340.

[9] G.D. Smith, Numerical Solutions of Partial Differential Equations: Finite Difference Methods, second ed., Oxford, New York, 1978.

[10] B. Fornberg, D.M. Sloan, in: A. Iserles (Ed.), Acta Numerica 94, Cambridge University Press, Cambridge, 1994.

[11] Y. Saad, SPARSKIT: A basic tool kit for sparse matrix computations, Technical Report RIACS-90-20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA, 1990.

[12] L. Kleinman, D.M. Bylander, Phys. Rev. Lett. 48 (1982) 1425.

[13] J.R. Chelikowsky, N. Troullier, Y. Saad, Phys. Rev. Lett. 72 (1994) 1240;
J.R. Chelikowsky, N. Troullier, K. Wu, Y. Saad, Phys. Rev. B 50 (1994) 11355.

[14] J.L. Martins, M.L. Cohen, Phys. Rev. B 37 (1988) 6134.

[15] The inclusion of a periodic image of an atom in the summation on the right-hand side of Eq. (10) would give the same contribution as that of the atom, since their exponential terms would differ by $2\pi n$, with n an integer.

[16] MPI: A Message-Passing Interface Standard, Message Passing Interface Forum, available from http://www.mpi-forum.org/docs/docs.html.

[17] R. Lehoucq, K. Maschhoff, D. Sorensen, C. Yang, ARPACK, available from http://www.caam.rice.edu/software/ARPACK/.

[18] Usually the eigenvectors are post-processed or "purified" by invoking just once PARPACK's routine *pdseupd*, but this step adds negligible computational cost to the algorithm.

[19] A. Stathopoulos, S. Öğüt, Y. Saad, J.R. Chelikowsky, H. Kim, Comput. Sci. Eng. 2 (2000) 19.

[20] Y. Zhou, Y. Saad, SIAM J. Matrix Anal. Appl. (2007), in press.

[21] Y. Saad, Numerical Methods for Large Eigenvalue Problems, John Wiley, New York, 1992.

[22] Y. Zhou, Y. Saad, M.L. Tiago, J.R. Chelikowsky, J. Comput. Phys. 219 (2006) 172;
Y. Zhou, Y. Saad, M.L. Tiago, J.R. Chelikowsky, Phys. Rev. E 74 (2006) 066704.

[23] J. Ihm, A. Zunger, M.L. Cohen, J. Phys. C 12 (1979) 4409.

[24] P. Pulay, Mol. Phys. 17 (1969) 197.

[25] N. Troullier, J.L. Martins, Phys. Rev. B 43 (1991) 1993.

[26] D.M. Ceperley, B.J. Alder, Phys. Rev. Lett. 45 (1980) 566.

[27] J.P. Perdew, A. Zunger, Phys. Rev. B 23 (1981) 5048.

[28] In Section 2 it was implicitly assumed that Brillouin-zone sampling would be restricted to the Γ point, but generalization of Eq. (2) to an arbitrary Bloch wave vector would not raise fundamental new points in the discussion of its discretization (see Ref. [5]).

[29] K.P. Kuber, G. Herzberg, Constants of Diatomic Molecules, Van Nostrand, New York, 1979.

[30] N. Binggeli, J.L. Martins, J.R. Chelikowsky, Phys. Rev. Lett. 68 (1992) 2956.

[31] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, New York, 1986.

[32] M.M.G. Alemany, L.J. Gallego, D.J. González, Phys. Rev. B 70 (2004) 134206.

[33] The code used was the PARATEC code, see, http://www.nersc.gov/projects/paratec/.

[34] M.M.G. Alemany, unpublished.

[35] J.P. Perdew, K. Burke, M. Ernzerhof, Phys. Rev. Lett. 77 (1996) 3865.