INSTITUTE OF PHYSICS PUBLISHING JOURNAL OF PHYSICS: CONDENSED MATTER

J. Phys.: Condens. Matter **14** (2002) 2745–2779 PII: S0953-8984(02)30737-9

The SIESTA method for *ab initio* order-N materials simulation

José M Soler[1], Emilio Artacho[2], Julian D Gale[3], Alberto García[4], Javier Junquera[1,5], Pablo Ordejón[6] and Daniel Sánchez-Portal[7]

[1] Dep. de Física de la Materia Condensada, C-III, Universidad Autónoma de Madrid, E-28049 Madrid, Spain
[2] Department of Earth Sciences, University of Cambridge, Downing St., Cambridge CB2 3EQ, UK
[3] Department of Chemistry, Imperial College of Science, Technology and Medicine, South Kensington SW7 2AY, UK
[4] Departamento de Física de la Materia Condensada, Universidad del País Vasco, Apt. 644, 48080 Bilbao, Spain
[5] Institut de Physique, Bâtiment B5, Université de Liège, B-4000 Sart-Tilman, Belgium
[6] Institut de Ciència de Materials de Barcelona, CSIC, Campus de la UAB, Bellaterra, 08193 Barcelona, Spain
[7] Dep. de Física de Materiales and DIPC, Facultad de Química, UPV/EHU, Apt. 1072, 20080 Donostia, Spain

Received 12 November 2001, in final form 16 January 2002
Published 8 March 2002
Online at stacks.iop.org/JPhysCM/14/2745

Abstract

We have developed and implemented a selfconsistent density functional method using standard norm-conserving pseudopotentials and a flexible, numerical linear combination of atomic orbitals basis set, which includes multiple-zeta and polarization orbitals. Exchange and correlation are treated with the local spin density or generalized gradient approximations. The basis functions and the electron density are projected on a real-space grid, in order to calculate the Hartree and exchange–correlation potentials and matrix elements, with a number of operations that scales linearly with the size of the system. We use a modified energy functional, whose minimization produces orthogonal wavefunctions and the same energy and density as the Kohn–Sham energy functional, without the need for an explicit orthogonalization. Additionally, using localized Wannier-like electron wavefunctions allows the computation time and memory required to minimize the energy to also scale linearly with the size of the system. Forces and stresses are also calculated efficiently and accurately, thus allowing structural relaxation and molecular dynamics simulations.

1. Introduction

As the improvements in computer hardware and software allow the simulation of molecules and materials with an increasing number of atoms N, the use of so-called order-N algorithms, in which the computer time and memory scales linearly with the simulated system size, becomes increasingly important. These $\mathcal{O}(N)$ methods were developed during the 1970s and 80s for long-range forces [1] and empirical interatomic potentials [2] but only in the last 5–10 years for the much more complex quantum mechanical methods, in which atomic forces are obtained by solving the interaction of ions and electrons together [3]. Even among quantum mechanical methods, there are very different levels of approximation: empirical or semiempirical orthogonal tight-binding methods are the simplest ones [4, 5]; '*ab initio*' nonorthogonal tight-binding and nonselfconsistent Harris-functional methods are next [6, 7] and fully selfconsistent density functional theory (DFT) methods are the most complex and reliable [8]. The implementation of $\mathcal{O}(N)$ methods in quantum mechanical simulations has also followed these steps, with several methods already well established within the tight-binding formalism [5], but much less so in selfconsistent DFT [9]. The latter also require, in addition to solving Schrödinger equation, the determination of the selfconsistent Hamiltonian in $\mathcal{O}(N)$ iterations. While this is difficult using plane waves, a localized basis set appears to be the natural choice. One proposed approach is the 'blips' of Hernandez and Gillan [10], regularly spaced Gaussian-like splines that can be systematically increased, in the spirit of finite-element methods, although at a considerable computational cost.

We have developed a fully selfconsistent DFT, based on a flexible linear combination of atomic orbitals (LCAO) basis set, with essentially perfect $\mathcal{O}(N)$ scaling. It allows extremely fast simulations using minimal basis sets and very accurate calculations with complete multiple-zeta and polarized bases, depending on the required accuracy and available computational power. In previous papers [11, 12] we have described preliminary versions of this method, that we call SIESTA (Spanish Initiative for Electronic Simulations with Thousands of Atoms). There is also a review [13] of the tens of studies performed with it, in a wide variety of systems, such as metallic surfaces, nanotubes and biomolecules. In this work we present a more complete description of the method, as well as some important improvements.

Apart from that of Born and Oppenheimer, the most basic approximations concern the treatment of exchange and correlation (XC), and the use of pseudopotentials. Exchange and correlation are treated within Kohn–Sham DFT [14]. We allow for both the local (spin) density approximation [15] (LDA/LSD) and the generalized gradient approximation (GGA) [16]. We use standard norm-conserving pseudopotentials [17, 18] in their fully nonlocal form [19]. We also include scalar-relativistic effects and the nonlinear partial-core correction to treat XC in the core region [20].

The SIESTA code has been already tested and applied to dozens of systems and a variety of properties [13]. Therefore, we shall just illustrate here the convergence of a few characteristic magnitudes of silicon, the archetypical system of the field, with respect to the main precision parameters that characterize our method: basis size (number of atomic basis orbitals); basis range (radius of the basis orbitals); fineness of the real-space integration grid and confinement radius of the Wannier-like electron states. Other parameters, such as the k-sampling integration grid, are common to all similar methods and we shall not discuss their convergence here.

2. Pseudopotential

Although the use of pseudopotentials is not strictly necessary with atomic basis sets, we find them convenient to get rid of the core electrons and, more importantly, to allow for the

expansion of a smooth (pseudo-) charge density on a uniform spatial grid. The theory and usage of first-principles norm-conserving pseudopotentials [17] is already well established. SIESTA reads them in semilocal form (a different radial potential $V_l(r)$ for each angular momentum l, optionally generated scalar-relativistically [21,22]) from a data file that users can fill with their preferred choice. We generally use the Troullier–Martins parametrization [23]. We transform this semilocal form into the fully nonlocal form proposed by Kleinman and Bylander (KB) [19]:

$$\hat{V}^{PS} = V_{local}(r) + \hat{V}^{KB} \tag{1}$$

$$\hat{V}^{KB} = \sum_{l=0}^{l_{max}^{KB}} \sum_{m=-l}^{l} \sum_{n=1}^{N_l^{KB}} |\chi_{lmn}^{KB}\rangle v_{ln}^{KB} \langle \chi_{lmn}^{KB}| \tag{2}$$

$$v_{ln}^{KB} = \langle \varphi_{ln} | \delta V_l(r) | \varphi_{ln} \rangle \tag{3}$$

where $r = |\boldsymbol{r}|$, $\hat{\boldsymbol{r}} = \boldsymbol{r}/r$ and $\delta V_l(r) = V_l(r) - V_{local}(r)$. $\chi_{lmn}^{KB}(\boldsymbol{r}) = \chi_{ln}^{KB}(r) Y_{lm}(\hat{\boldsymbol{r}})$ (with $Y_{lm}(\hat{\boldsymbol{r}})$ a spherical harmonic) are the KB projection functions

$$\chi_{ln}^{KB}(r) = \delta V_l(r) \varphi_{ln}(r). \tag{4}$$

The functions φ_{ln} are obtained from the eigenstates ψ_{ln} of the semilocal pseudopotential (screened by the pseudo-valence charge density) at energy ϵ_{ln} using the orthogonalization scheme proposed by Blöchl [24]:

$$\varphi_{ln}(r) = \psi_{ln}(r) - \sum_{n'=1}^{n-1} \varphi_{ln'}(r) \frac{\langle \varphi_{ln'} | \delta V_l(r) | \psi_{ln} \rangle}{\langle \varphi_{ln'} | \delta V_l(r) | \varphi_{ln'} \rangle} \tag{5}$$

$$\left[-\frac{1}{2r} \frac{\mathrm{d}^2}{\mathrm{d}r^2} r + \frac{l(l+1)}{2r^2} + V_l(r) + V^H(r) + V^{xc}(r) \right] \psi_{ln}(r) = \epsilon_{ln} \psi_{ln}(r). \tag{6}$$

V^H and V^{xc} are the Hartree and XC potentials for the pseudo-valence charge density, and we are using atomic units ($e = \hbar = m_e = 1$) throughout this paper.

The local part of the pseudopotential $V_{local}(r)$ is in principle arbitrary, but it must join the semilocal potentials $V_l(r)$, which, by construction, all become equal to the (unscreened) all-electron potential beyond the pseudopotential core radius r_{core}. Thus, $\delta V_l(r) = 0$ for $r > r_{core}$. Ramer and Rappe have proposed that $V_{local}(r)$ be optimized for transferability [25], but most plane wave schemes make it equal to one of the $V_l(r)$ for reasons of efficiency. Our case is different because $V_{local}(r)$ is the only pseudopotential part that needs to be represented in the real space grid, while the matrix elements of the nonlocal part $\hat{V}_{KB}$ are cheaply and accurately calculated by two-centre integrals. Therefore, we optimize $V_{local}(r)$ for smoothness, making it equal to the potential created by a positive charge distribution of the form [26]

$$\rho^{local}(r) \propto \exp[-(\sinh(abr)/\sinh(b))^2], \tag{7}$$

where a and b are chosen to provide simultaneously optimal real-space localization and reciprocal-space convergence[8]. After some numerical tests we have taken $b = 1$ and $a = 1.82/r_{core}$. Figure 1 shows $V_{local}(r)$ for silicon.

Since $V_l(r) = V_{local}(r)$ outside r_{core}, $\chi_{ln}^{KB}(r)$ is strictly zero beyond that radius, irrespective of the value[9] of ϵ_{ln}. Generally it is sufficient to have a single projector χ_{lm}^{KB}

[8] The local potentials constructed in this way usually have a strength (depth) that is an average of the different V_l and neither too deep nor too shallow. This tends to maintain the separable potentials free of ghost states [80].

[9] For some atoms, typically those with semicore states suitable for treating together with the valence states, $V_l(r)$ only assumes the asymptotic coulombic behaviour $-2Z_{val}/r$, and therefore only cancels out exactly with our $V_{local}(r)$, for r larger than certain $r_C > r_{core}$. In these cases, to avoid very extended KB projector functions, we generate the local potentials with a prescription different from that presented in the text: if $r_C > 1.3\, r_{core}$ we take $V_{local}(r) = V_l(r)$ for $r > r_{core}$ and $V_{local}(r) = \exp(v_1 + v_2 r^2 + v_3 r^3)$ for $r < r_{core}$, where v_1, v_2 and v_3 are determined by enforcing the continuity of the potential up to the second derivative. This simple prescription usually produces smooth local potentials with properties similar to those noted in the text (see footnote 8).

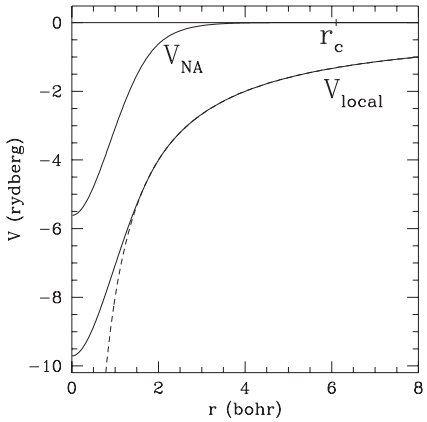

Figure 1. Local pseudopotential for silicon. V_{local} is the unscreened local part of the pseudopotential, generated as the electrostatic potential produced by a localized distribution of positive charge, equation (7), whose integral is equal to the valence ion charge ($Z = 4$ for Si). The dashed curve is $-Z/r$. V_{NA} is the local pseudopotential screened by an electron charge distribution, generated by filling the first-ζ basis orbitals with the free-atom valence occupations. Since these basis orbitals are strictly confined to a radius r^c_{max}, V_{NA} is also strictly zero beyond that radius.

for each angular momentum (i.e. a single term in the sum on n). In this case we follow the normal practice of making ϵ_{ln} equal to the valence atomic eigenvalue ϵ_l, and the function $\varphi_l(r)$ in equation (4) is identical to the corresponding eigenstate $\psi_l(r)$. In some cases, particularly for alkaline metals, alkaline earths and transition metals of the first few columns, we have sometimes found it necessary to include the semicore states together with the valence states[10]. In these cases, we also include two independent KB projectors, one for the semicore and one for the valence states. However, our pseudopotentials are still norm conserving rather than 'ultrasoft' [27]. This is because, in our case, it is only the electron density that needs to be accurately represented in a real-space grid, rather than each wavefunction. Therefore, the ultrasoft pseudopotential formalism does not imply in SIESTA the same savings as it does in PW schemes. Also, since the nonlocal part of the pseudopotential is a relatively cheap operator within SIESTA, we generally (but not necessarily) use a larger than usual value of l^{KB}_{max} in equation (2), making it one unit larger than the l_{max} of the basis functions.

3. Basis set

Order-N methods rely heavily on the sparsity of the Hamiltonian and overlap matrices. This sparsity requires either the neglect of matrix elements that are small enough or the use of strictly confined basis orbitals, i.e. orbitals that are zero beyond a certain radius [7]. We have adopted this latter approach because it keeps the energy strictly variational, thus facilitating the test of the convergence with respect to the radius of confinement. Within this radius, our atomic basis orbitals are products of a numerical radial function and a spherical harmonic. For atom I, located at $\boldsymbol{R}_I$,

$$\phi_{Ilmn}(\boldsymbol{r}) = \phi_{Iln}(r_I)Y_{lm}(\hat{\boldsymbol{r}}_I) \tag{8}$$

where $\boldsymbol{r}_I = \boldsymbol{r} - \boldsymbol{R}_I$. The angular momentum (labelled by l, m) may be arbitrarily large and, in general, there will be several orbitals (labelled by index n) with the same angular

[10] If there are both semicore and valence electrons with the same angular momentum, the pseudopotential is generated for an ion.

dependence, but different radial dependence, which is conventionally called a 'multiple-ζ' basis. The radial functions are defined by a cubic spline interpolation [28] from the values given on a fine radial mesh. Each radial function may have a different cutoff radius and, up to that radius, its shape is completely free and can be introduced by the user in an input file. In practice, it is also convenient to have an automatic procedure to generate sufficiently good basis sets. We have developed several such automatic procedures, and we shall describe one of them here for completeness, even though we stress that the generation of the basis set, like that of the pseudopotential, is to a large extent up to the user and independent of the SIESTA method itself.

In the case of a minimal (single-ζ (SZ)) basis set, we have found convenient and efficient the method of Sankey and Niklewski [7, 29]. Their basis orbitals are the eigenfunctions of the (pseudo-) atom within a spherical box (although the radius of the box may be different for each orbital; see below). In other words, they are the (angular-momentum-dependent) numerical eigenfunctions $\phi_l(r)$ of the atomic pseudopotential $V_l(r)$, for an energy $\epsilon_l + \delta\epsilon_l$ chosen so that the first node occurs at the desired cutoff radius r_l^c:

$$\left(-\frac{1}{2r}\frac{\mathrm{d}^2}{\mathrm{d}r^2}r + \frac{l(l+1)}{2r^2} + V_l(r)\right)\phi_l(r) = (\epsilon_l + \delta\epsilon_l)\phi_l(r) \tag{9}$$

with $\phi_l(r_l^c) = 0$ (we omit indices I and n here for simplicity). In order to obtain a well balanced basis, in which the effect of the confinement is similar for all the orbitals, it is usually better to fix a common 'energy shift' $\delta\epsilon$, rather than a common radius r^c, for all the atoms and angular momenta. This means that the orbital radii depend on the atomic species and angular momentum.

One obvious possibility for multiple-ζ bases is to use pseudopotential eigenfunctions with an increasing number of nodes [29]. They have the virtue of being orthogonal and asymptotically complete. However, the efficiency of this kind of basis depends on the radii of confinement of the different orbitals, since the excited states of the pseudopotential are usually unbound. Thus, in practice we have found this procedure rather inefficient. Another possibility is to use the atomic eigenstates for different ionization states [30]. We have implemented a different scheme [31], based on the 'split-valence' method, which is standard in quantum chemistry [32]. In that method, the first-ζ basis orbitals are 'contracted' (i.e. fixed) linear combinations of Gaussians, determined either variationally or by fitting numerical atomic eigenfunctions. The second-ζ orbital is then one of the Gaussians (generally the slowest-decaying one), which is 'released' or 'split' from the contracted combination. Higher-ζ orbitals are generated in a similar way by releasing more Gaussians. Our scheme adapts this split-valence method to our numerical orbitals. Following the same spirit, our second-ζ functions $\phi_l^{2\zeta}(r)$ have the same tail as the first-ζ orbitals $\phi_l^{1\zeta}(r)$ but change to a simple polynomial behaviour inside a 'split radius' r_l^s:

$$\phi_l^{2\zeta}(r) = \begin{cases} r^l(a_l - b_l r^2) & \text{if } r < r_l^s \\ \phi_l^{1\zeta}(r) & \text{if } r \geqslant r_l^s \end{cases} \tag{10}$$

where a_l and b_l are determined by imposing the continuity of value and slope at r_l^s. These orbitals therefore combine the decay of the atomic eigenfunctions with a smooth and featureless behaviour inside r_l^s. We have found it convenient to set the radius r_l^s by fixing the norm of $\phi_l^{1\zeta}$ in $r > r_l^s$. We have found empirically that a reasonable value for this 'split norm' is $\sim$0.15. Actually, instead of $\phi_l^{2\zeta}$ thus defined, we use $\phi_l^{1\zeta} - \phi_l^{2\zeta}$, which is zero beyond r_l^s, to reduce the number of nonzero matrix elements, without any loss of variational freedom.

To achieve well converged results, in addition to the atomic valence orbitals, it is generally necessary to also include polarization orbitals, to account for the deformation induced by bond

formation. Again, using pseudoatomic orbitals of higher angular momentum is frequently unsatisfactory, because they tend to be too extended, or even unbound. Instead, consider a valence pseudoatomic orbital $\phi_{lm}(\boldsymbol{r}) = \phi_l(r)Y_{lm}(\hat{\boldsymbol{r}})$, such that there are no valence orbitals with angular momentum $l + 1$. To polarize this, we apply a small electric field $\mathcal{E}$ in the z-direction. Using first-order perturbation theory

$$(H - E)\delta\phi = -(\delta H - \delta E)\phi, \tag{11}$$

where $\delta H = \mathcal{E}z$ and $\delta E = \langle\phi|\delta H|\phi\rangle = 0$ because δH is odd. Selection rules imply that the resulting perturbed orbital will only have components with $l' = l \pm 1$, $m' = m$:

$$\delta H\phi_{lm}(\boldsymbol{r}) = (\mathcal{E}r\cos(\theta))(\phi_l(r)Y_{lm}(\hat{\boldsymbol{r}})) = \mathcal{E}r\phi_l(r)(c_{l-1}Y_{l-1,m} + c_{l+1}Y_{l+1,m}) \tag{12}$$

and

$$\delta\phi_{lm}(\boldsymbol{r}) = \varphi_{l-1}(r)Y_{l-1,m}(\hat{\boldsymbol{r}}) + \varphi_{l+1}(r)Y_{l+1,m}(\hat{\boldsymbol{r}}). \tag{13}$$

Since in general there will already be orbitals with angular momentum $l - 1$ in the basis set, we select the $l + 1$ component by substituting (12) and (13) in (11), multiplying by $Y^*_{l+1,m}(\hat{\boldsymbol{r}})$ and integrating over angular variables. Thus we obtain the equation

$$\left[-\frac{1}{2r}\frac{\mathrm{d}^2}{\mathrm{d}r^2}r + \frac{(l+1)(l+2)}{2r^2} + V_l(r) - E_l\right]\varphi_{l+1}(r) = -r\phi_l(r) \tag{14}$$

where we have also eliminated the factors $\mathcal{E}$ and c_{l+1}, which only affect the normalization of φ_{l+1}. The polarization orbitals are then added to the basis set: $\phi_{l+1,m}(\boldsymbol{r}) = N\varphi_{l+1}(r)Y_{l+1,m}(\hat{\boldsymbol{r}})$, where N is a normalization constant.

We have found that the previously described procedures generate reasonable minimal SZ basis sets, appropriate for semiquantitative simulations, and double-ζ plus polarization (DZP) basis sets that yield high-quality results for most of the systems studied. We thus refer to DZP as the 'standard' basis, because it usually represents a good balance between well converged results and a reasonable computational cost. In some cases (typically alkali and some transition metals), semicore states also need to be included for good-quality results. More recently [33], we have obtained extremely efficient basis sets optimized variationally in molecules or solids. Figure 2 shows the performance of these atomic basis sets compared with plane waves, using the same pseudopotentials and geometries. It may be seen that the SZ bases are comparable to plane-wave cutoffs typically used in Car–Parrinello molecular dynamics simulations, while DZP sets are comparable to the cutoffs used in geometry relaxations and energy comparisons. As expected, the LCAO is far more efficient, typically by a factor of 10–20, in terms of number of basis orbitals. This efficiency must be balanced against the faster algorithms available for plane waves, and our main motivation for using an LCAO basis is its suitability for $\mathcal{O}(N)$ methods. Still, we have generally found that, even without using the $\mathcal{O}(N)$ functional, SIESTA is considerably faster than a plane-wave calculation of similar quality.

Figure 3 shows the convergence of the total-energy curve of silicon, as a function of lattice parameter, for different basis sizes, and table 1 summarizes the same information numerically. It can be seen that the 'standard' DZP basis offers already quite well converged results, comparable to those used in practice in most plane-wave calculations.

Figure 4 shows the dependence of the lattice constant, bulk modulus and cohesive energy of bulk silicon on the range of the basis orbitals. It shows that a cutoff radius of 3 Å for both s and p orbitals yields already very well converged results, specially when using a 'standard' DZP basis.

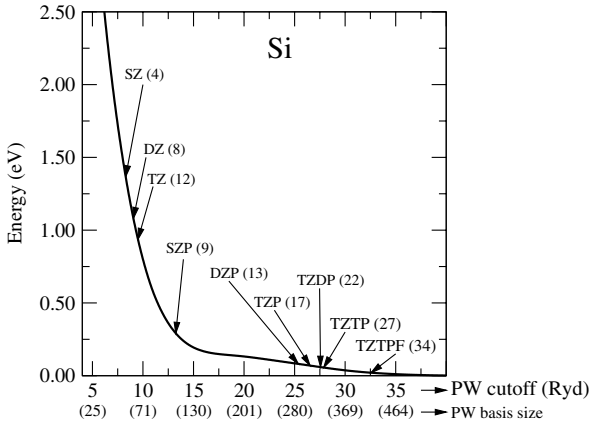

Figure 2. Comparison of convergence of the total energy with respect to the sizes of a plane-wave basis set and of the LCAO basis set used by SIESTA. The curve shows the total energy per atom of silicon versus the cutoff of a plane-wave basis, calculated with a program independent of SIESTA, which uses the same pseudopotential. The arrows indicate the energies obtained with different LCAO basis sets, calculated with SIESTA, and the plane-wave cutoffs that yield the same energies. The numbers in parentheses indicate the basis sizes, i.e. the number of atomic orbitals or plane waves of each basis set. SZ, single ζ (valence s and p orbitals); DZ, double ζ; TZ, triple ζ; DZP, double-ζ valence orbitals plus SZ-polarization d orbitals; TZP, triple-ζ valence plus SZ polarization; TZDP, triple-ζ valence plus double-ζ polarization; TZTP, triple-ζ valence plus triple-ζ polarization; TZTPF, the same as TZTP plus extra SZ-polarization f orbitals.

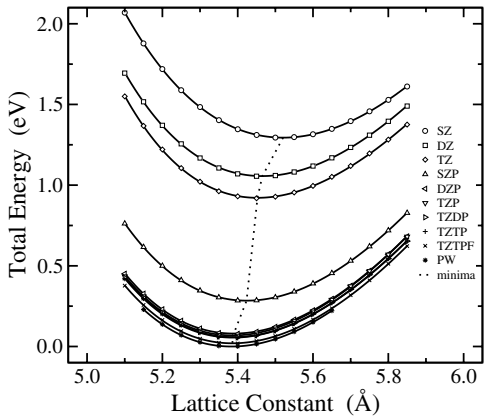

Figure 3. Total energy per atom versus lattice constant for bulk silicon, using different basis sets, denoted as in figure 2. PW refers to a very well converged (50 Ryd cutoff) plane-wave calculation. The dotted curve joins the minima of the different curves.

4. Electron Hamiltonian

Within the nonlocal-pseudopotential approximation, the standard Kohn–Sham one-electron Hamiltonian may be written as

$$\hat{H} = \hat{T} + \sum_I V_I^{local}(\boldsymbol{r}) + \sum_I \hat{V}_I^{KB} + V^H(\boldsymbol{r}) + V^{xc}(\boldsymbol{r}) \tag{15}$$

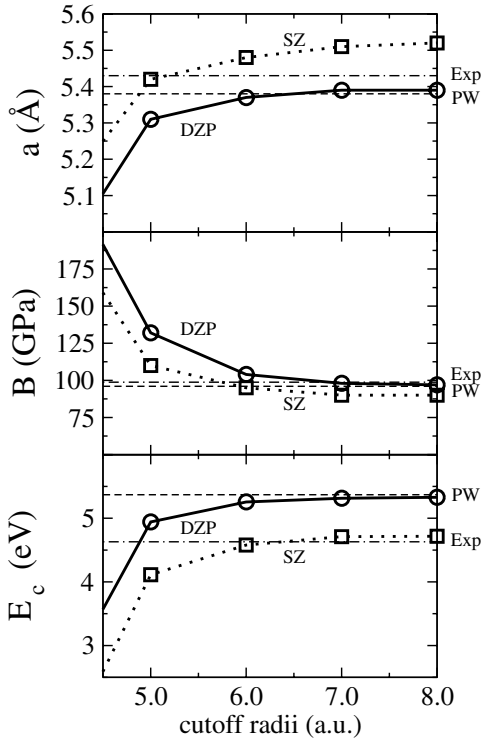

Figure 4. Dependence of the lattice constant, bulk modulus and cohesive energy of bulk silicon on the cutoff radius of the basis orbitals. The s and p orbital radii have been made equal in this case, to simplify the plot. PW refers to a well converged plane-wave calculation with the same pseudopotential.

Table 1. Comparisons of the lattice constant a, bulk modulus B and cohesive energy E_c for bulk Si, obtained with different basis sets. The basis notation is as in figure 2. PW refers to a 50 Ryd cutoff plane-wave calculation. The LAPW results were taken from [34], and the experimental values from [35].

Basis	a (Å)	B (GPa)	E_c (eV)
SZ	5.521	88.7	4.722
DZ	5.465	96.0	4.841
TZ	5.453	98.4	4.908
SZP	5.424	97.8	5.227
DZP	5.389	96.6	5.329
TZP	5.387	97.5	5.335
TZDP	5.389	96.0	5.340
TZTP	5.387	96.0	5.342
TZTPF	5.385	95.4	5.359
PW	5.384	95.9	5.369
LAPW	5.41	96	5.28
Expt	5.43	98.8	4.63

where $\hat{T} = -\frac{1}{2}\nabla^2$ is the kinetic energy operator, I is an atom index, $V^H(\boldsymbol{r})$ and $V^{xc}(\boldsymbol{r})$ are the total Hartree and XC potentials and $V_I^{local}(\boldsymbol{r})$ and $\hat{V}_I^{KB}$ are the local and nonlocal (KB) parts of the pseudopotential of atom I.

In order to eliminate the long range of V_I^{local}, we screen it with the potential V_I^{atom}, created by an atomic electron density ρ_I^{atom}, constructed by populating the basis functions with appropriate valence atomic charges. Notice that, since the atomic basis orbitals are zero beyond the cutoff radius $r_I^c = \max_l(r_{Il}^c)$, the screened 'neutral-atom' (NA) potential $V_I^{NA} \equiv V_I^{local} + V_I^{atom}$ is also zero beyond this radius [7] (see figure 1). Now let $\delta\rho(\boldsymbol{r})$ be the difference between the selfconsistent electron density $\rho(\boldsymbol{r})$ and the sum of atomic densities $\rho^{atom} = \sum_I \rho_I^{atom}$, and let $\delta V^H(\boldsymbol{r})$ be the electrostatic potential generated by $\delta\rho(\boldsymbol{r})$, which integrates to zero and is usually much smaller than $\rho(\boldsymbol{r})$. Then the total Hamiltonian may be rewritten as

$$\hat{H} = \hat{T} + \sum_I \hat{V}_I^{KB} + \sum_I V_I^{NA}(\boldsymbol{r}) + \delta V^H(\boldsymbol{r}) + V^{xc}(\boldsymbol{r}). \tag{16}$$

The matrix elements of the first two terms involve only two-centre integrals, which are calculated in reciprocal space and tabulated as a function of interatomic distance. The remaining terms involve potentials which are calculated on a three-dimensional real-space grid. We consider these two approaches in detail in the following sections.

5. Two-centre integrals

The overlap matrix and the largest part of the Hamiltonian matrix elements are given by two-centre integrals[11]. We calculate these integrals in Fourier space, as proposed by Sankey and Niklewski [7], but we use some implementation details explained in this section. Let us consider first overlap integrals of the form

$$S(\boldsymbol{R}) \equiv \langle\psi_1|\psi_2\rangle = \int \psi_1^*(\boldsymbol{r})\psi_2(\boldsymbol{r}-\boldsymbol{R})\,\mathrm{d}\boldsymbol{r}, \tag{17}$$

where the integral is over all space and ψ_1, ψ_2 may be basis functions ϕ_{lmn}, KB pseudopotential projectors χ_{lmn} or more complicated functions centred on the atoms. The function $S(\boldsymbol{R})$ can be seen as a convolution: we take the Fourier transform

$$\psi(\boldsymbol{k}) = \frac{1}{(2\pi)^{3/2}} \int \psi(\boldsymbol{r})\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\boldsymbol{r}}\,\mathrm{d}\boldsymbol{r} \tag{18}$$

where we use the same symbol ψ for $\psi(\boldsymbol{r})$ and $\psi(\boldsymbol{k})$, as its meaning is clear from the different arguments. We also use the plane-wave expression of Dirac's delta function, $\int \mathrm{e}^{\mathrm{i}(\boldsymbol{k}'-\boldsymbol{k})\boldsymbol{r}}\,\mathrm{d}\boldsymbol{r} = (2\pi)^3\delta(\boldsymbol{k}'-\boldsymbol{k})$, to find the usual result that the Fourier transform of a convolution in real space is a simple product in reciprocal space:

$$S(\boldsymbol{R}) = \int \psi_1^*(\boldsymbol{k})\psi_2(\boldsymbol{k})\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\boldsymbol{R}}\,\mathrm{d}\boldsymbol{k}. \tag{19}$$

Let us assume now that the functions $\psi(\boldsymbol{r})$ can be expanded exactly with a finite number of spherical harmonics:

$$\psi(\boldsymbol{r}) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} \psi_{lm}(r)Y_{lm}(\hat{\boldsymbol{r}}), \tag{20}$$

$$\psi_{lm}(r) = \int_0^{\pi} \sin\theta\,\mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\varphi Y_{lm}^*(\theta,\varphi)\psi(r,\theta,\varphi). \tag{21}$$

[11] Some integrals, such as $\langle\phi_{Ilmn}|V_I^{NA}|\phi_{I'l'm'n'}\rangle$ could also be calculated in this way, but this is not the case of $\langle\phi_{Ilmn}|V_{I''}^{NA}|\phi_{I'l'm'n'}\rangle$, which involve rather cumbersome three-centre integrals of arbitrary numerical functions. Therefore, it is simpler to find the total NA potential and to calculate a single integral $\langle\phi_{Ilmn}|V^{NA}(\boldsymbol{r})|\phi_{I'l'm'n'}\rangle$ in the uniform spatial grid.

This is clearly true for basis functions and KB projectors, which contain a single spherical harmonic, and also for functions such as $x\psi(\boldsymbol{r})$, which appear in dipole matrix elements. We now substitute in (18) the expansion of a plane wave in spherical harmonics [36]

$$\mathrm{e}^{\mathrm{i}\boldsymbol{k}\boldsymbol{r}} = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} 4\pi \mathrm{i}^l j_l(kr) Y_{lm}^*(\hat{\boldsymbol{k}}) Y_{lm}(\hat{\boldsymbol{r}}), \tag{22}$$

to obtain

$$\psi(\boldsymbol{k}) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} \psi_{lm}(k) Y_{lm}(\hat{\boldsymbol{k}}), \tag{23}$$

$$\psi_{lm}(k) = \sqrt{\frac{2}{\pi}} (-\mathrm{i})^l \int_0^{\infty} r^2 \,\mathrm{d}r \; j_l(kr) \psi_{lm}(r). \tag{24}$$

Substituting now (23) and (22) into (19) we obtain

$$S(\boldsymbol{R}) = \sum_{l=0}^{2l_{max}} \sum_{m=-l}^{l} S_{lm}(R) Y_{lm}(\hat{\boldsymbol{R}}) \tag{25}$$

where

$$S_{lm}(R) = \sum_{l_1 m_1} \sum_{l_2 m_2} G_{l_1 m_1, l_2 m_2, lm} S_{l_1 m_1, l_2 m_2, l}(R), \tag{26}$$

$$G_{l_1 m_1, l_2 m_2, lm} = \int_0^{\pi} \sin\theta \,\mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\varphi \; Y_{l_1 m_1}^*(\theta, \varphi) Y_{l_2 m_2}(\theta, \varphi) Y_{lm}^*(\theta, \varphi), \tag{27}$$

$$S_{l_1 m_1, l_2 m_2, l}(R) = 4\pi \mathrm{i}^{l_1 - l_2 - l} \int_0^{\infty} k^2 \,\mathrm{d}k \; j_l(kR) \mathrm{i}^{-l_1} \psi_{1, l_1 m_1}^*(k) \mathrm{i}^{l_2} \psi_{2, l_2 m_2}(k). \tag{28}$$

Notice that $\mathrm{i}^{-l_1}\psi_1^*(k)$, $\mathrm{i}^{l_2}\psi_2(k)$ and $\mathrm{i}^{l_1-l_2-l}$ are all real, since $l_1 - l_2 - l$ is even for all l for which $G_{l_1 m_1, l_2 m_2, lm} \neq 0$. The Gaunt coefficients $G_{l_1 m_1, l_2 m_2, lm}$ can be obtained by recursion from Clebsch–Gordan coefficients [7]. However, we use real spherical harmonics for computational efficiency:

$$Y_{lm}(\theta, \varphi) = C_{lm} P_l^m(\cos\theta) \begin{cases} \sin(m\varphi) & \text{if } m < 0 \\ \cos(m\varphi) & \text{if } m \geqslant 0 \end{cases} \tag{29}$$

where $P_l^m(z)$ are the associated Legendre polynomials and C_{lm} normalization constants [28]. This does not affect the validity of any of previous equations, but it modifies the value of the Gaunt coefficients. Therefore, we find it is simpler and more general to calculate $G_{l_1 m_1, l_2 m_2, lm}$ directly from equation (27). To do this, we use a Gaussian quadrature [28]

$$\int_0^{\pi} \sin\theta \,\mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\varphi \to 4\pi \frac{1}{N_\theta} \sum_{i=1}^{N_\theta} w_i \sin\theta_i \frac{1}{N_\varphi} \sum_{j=1}^{N_\varphi} \tag{30}$$

with $N_\varphi = 1+3l_{max}$, $N_\theta = 1+\mathrm{int}(3l_{max}/2)$, and the points $\cos\theta_i$ and weights w_i are calculated as described in [28]. This quadrature is exact in equation (27) for spherical harmonics Y_{lm} (real or complex) of $l \leqslant l_{max}$, and it can be used also to find the expansion of $\psi(\boldsymbol{r})$ in spherical harmonics (equation (21)).

The coefficients $G_{l_1 m_1, l_2 m_2, lm}$ are universal and they can be calculated and stored once and for all. The functions $S_{l_1 m_1, l_2 m_2, l}(R)$ depend, of course, on the functions $\psi_{1,2}(\boldsymbol{r})$ being integrated. For each pair of functions, they can be calculated and stored in a fine radial grid R_i, up to the maximum distance $R_{max} = r_1^c + r_2^c$ at which ψ_1 and ψ_2 overlap. Their value at an arbitrary distance R can then be obtained very accurately using a spline interpolation.

Kinetic matrix elements $T(\boldsymbol{R}) \equiv \langle \psi_1^* | -\frac{1}{2}\nabla^2 | \psi_2 \rangle$ can be obtained in exactly the same way, except for an extra factor of k^2 in equation (28):

$$T_{l_1 m, l_2 m_2, l}(R) = 4\pi \mathrm{i}^{l_1 - l_2 - l} \int_0^\infty \tfrac{1}{2} k^4 \, \mathrm{d}k \; j_l(kR) \mathrm{i}^{-l_1} \psi^*_{1, l_1 m_1}(k) \mathrm{i}^{l_2} \psi_{2, l_2 m_2}(k). \quad (31)$$

Since we frequently use basis orbitals with a kink [7], we need rather fine radial grids to obtain accurate kinetic matrix elements, and we typically use grid cutoffs of more than 2000 Ryd for this purpose. Once obtained, the fine grid does not penalize the execution time, because the interpolation effort is independent of the number of grid points. It also affects very marginally the storage requirements, because of the one-dimensional character of the tables. However, even though it needs to be performed only once, the calculation of the radial integrals (24), (28), and (31) is not negligible if performed unwisely. We have developed a special fast radial Fourier transform for this purpose, as explained in appendix B.

Dipole matrix elements, such as $\langle \psi_1 | x | \psi_2 \rangle$, can also be obtained easily by defining a new function $\chi_1(\boldsymbol{r}) \equiv x \psi_1(\boldsymbol{r})$, expanding it using (21) and computing $\langle \chi_1 | \psi_2 \rangle$ as explained above (with the precaution of using $l_{max} + 1$ instead of l_{max}).

6. Grid integrals

The matrix elements of the last three terms of equation (16) involve potentials which are calculated on a real-space grid. The fineness of this grid is controlled by a 'grid cutoff' E_{cut}: the maximum kinetic energy of the plane waves that can be represented in the grid without aliasing[12]. The short-range screened pseudopotentials $V_I^{NA}(\boldsymbol{r})$ in (16) are tabulated as a function of the distance to atoms I and easily interpolated at any desired grid point. The last two terms require the calculation of the electron density on the grid. Let $\psi_i(\boldsymbol{r})$ be the Hamiltonian eigenstates, expanded in the atomic basis set

$$\psi_i(\boldsymbol{r}) = \sum_\mu \phi_\mu(\boldsymbol{r}) c_{\mu i}, \quad (32)$$

where $c_{\mu i} = \langle \tilde{\phi}_\mu | \psi_i \rangle$ and $\tilde{\phi}_\mu$ is the dual orbital of ϕ_μ: $\langle \tilde{\phi}_\mu | \phi_\nu \rangle = \delta_{\mu\nu}$. We use the compact index notation $\mu \equiv \{Ilmn\}$ for the basis orbitals, equation (8). The electron density is then

$$\rho(\boldsymbol{r}) = \sum_i n_i |\psi_i(\boldsymbol{r})|^2 \quad (33)$$

where n_i is the occupation of state ψ_i. If we substitute (32) into (33) and define a density matrix

$$\rho_{\mu\nu} = \sum_i c_{\mu i} n_i c_{i\nu}, \quad (34)$$

where $c_{i\nu} \equiv c^*_{\nu i}$, the electron density can be rewritten as

$$\rho(\boldsymbol{r}) = \sum_{\mu\nu} \rho_{\mu\nu} \phi^*_\nu(\boldsymbol{r}) \phi_\mu(\boldsymbol{r}). \quad (35)$$

We use the notation ϕ^*_μ for generality, despite our use of real basis orbitals in practice. Then, to calculate the density at a given grid point, we first find all the atomic basis orbitals, equation (8), at that point, interpolating the radial part from numerical tables, and then we use (35) to calculate the density. Notice that only a small number of basis orbitals are nonzero at a given grid point, so that the calculation of the density can be performed in $\mathcal{O}(N)$ operations, once $\rho_{\mu\nu}$ is known.

[12] Notice that our grid cutoff to represent the density is not directly comparable to the energy cutoff in the context of plane-wave codes, which usually refers to the wavefunctions. Strictly speaking, the density requires a value four times larger.

The storage of the orbital values at the grid points can be one of the most expensive parts of the program in terms of memory usage. Hence, an option is included to calculate and use these terms on the fly, in the spirit of a direct-SCF calculation. The calculation of $\rho_{\mu\nu}$ itself with equation (34) does not scale linearly with the system size, requiring instead the use of special $\mathcal{O}(N)$ techniques to be described below. However, notice that in order to calculate the density, only the matrix elements $\rho_{\mu\nu}$ for which ϕ_μ and ϕ_ν overlap are required, and they can therefore be stored as a sparse matrix of $\mathcal{O}(N)$ size. Once the valence density is available in the grid, we add to it, if necessary, the nonlocal core correction [20], a spherical charge density intended to simulate the atomic cores, which is also interpolated from a radial grid. With it, we find the XC potential $V^{xc}(\boldsymbol{r})$, trivially in the LDA and using the method described in [**?**] for the GGA. To calculate $\delta V^H(\boldsymbol{r})$, we first find $\rho^{atom}(\boldsymbol{r})$ at the grid points, as a sum of spherical atomic densities (also interpolated from a radial grid) and subtract it from $\rho(\boldsymbol{r})$ to find $\delta\rho(\boldsymbol{r})$. We then solve Poisson's equation to obtain $\delta V^H(\boldsymbol{r})$ and find the total grid potential $V(\boldsymbol{r}) = V^{NA}(\boldsymbol{r}) + \delta V^H(\boldsymbol{r}) + V^{xc}(\boldsymbol{r})$. Finally, at every grid point, we calculate $V(\boldsymbol{r})\phi^*_\mu(\boldsymbol{r})\phi_\nu(\boldsymbol{r})\Delta\boldsymbol{r}^3$ for all pairs ϕ_μ, ϕ_ν which are not zero at that point ($\Delta\boldsymbol{r}^3$ is the volume per grid point) and add it to the Hamiltonian matrix element $H_{\mu\nu}$.

To solve Poisson's equation and find $\delta V^H(\boldsymbol{r})$ we normally use fast Fourier transforms in a unit cell that is either naturally periodic or made artificially periodic by a supercell construction. For neutral isolated molecules, our use of strictly confined basis orbitals makes it trivial to avoid any direct overlap between the repeated molecules, and the electric multipole interactions decrease rapidly with cell size. For charged molecules we suppress the $\boldsymbol{G} = 0$ Fourier component (an infinite constant) of the potential created by the excess of charge. This amounts to compensating this excess with a uniform charge background. We then use the method of Makov and Payne [38] to correct the total energy for the interaction between the repeated cells. Alternatively, we can solve Poisson's equation by the multigrid method, using finite differences and fixed boundary conditions, obtained from the multipole expansion of the molecular charge density. This can be done in strictly $\mathcal{O}(N)$ operations, unlike the fast Fourier transformations (FFTs), which scale as $N \log N$. However, the cost of this operation is typically negligible and therefore has no influence on the overall scaling properties of the calculation.

Figures 5 and 6 show the convergence of different magnitudes with respect to the energy cutoff of the integration grid. For orthogonal unit cell vectors this is simply, in atomic units, $E_{cut} = (\pi/\Delta x)^2/2$ with Δx the grid interval.

7. Noncollinear spin

In the usual case of a normal (collinear) spin-polarized system, there are two sets of values for $\psi_i(\boldsymbol{r})$, $\rho_{\mu\nu}$, $\rho(\boldsymbol{r})$, $V^{xc}(\boldsymbol{r})$ and $H_{\mu\nu}$, one for spin up and another for spin down. Thus, the grid calculations can be repeated twice in an almost independent way: only to calculate $V^{xc}(\boldsymbol{r})$ need they be combined. However, in the noncollinear spin case [39–42], the density at every point is represented not only by the up and down values, but also by a vector giving the spin direction. Equivalently, it may be represented by a local spin density matrix

$$\rho^{\alpha\beta}(\boldsymbol{r}) = \sum_i n_i \psi_i^{\beta *}(\boldsymbol{r})\psi_i^\alpha(\boldsymbol{r}) = \sum_{\mu\nu} \rho_{\mu\nu}^{\alpha\beta}\phi_\nu^*(\boldsymbol{r})\phi_\mu(\boldsymbol{r}) \tag{36}$$

$$\psi_i^\alpha(\boldsymbol{r}) = \sum_\mu \phi_\mu(\boldsymbol{r})c_{\mu i}^\alpha \tag{37}$$

$$\rho_{\mu\nu}^{\alpha\beta} = \sum_i c_{\mu i}^\alpha n_i c_{i\nu}^\beta \tag{38}$$

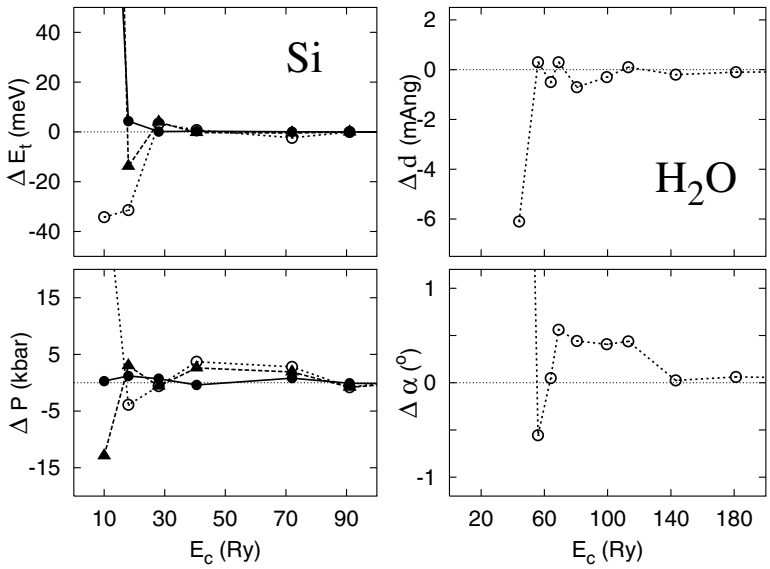

Figure 5. (*a*) Convergence of the total energy and pressure in bulk silicon as a function of the energy cutoff E_{cut} of the real-space integration mesh. Circles and continuous line: using a grid-cell sampling of eight refinement points per original grid point. The refinement points are used only in the final calculation, not during the selfconsistency iteration (see text). Triangles: two refinement points per original grid point. White circles: no grid-cell sampling. (*b*) Bond length and angle of the water molecule as a function of E_{cut}.

where α, β are spin indices, with up or down values. The coefficients $c^{\alpha}_{\mu i}$ are obtained by solving the generalized eigenvalue problem

$$\sum_{\nu\beta}(H^{\alpha\beta}_{\mu\nu} - E_i S_{\mu\nu}\delta^{\alpha\beta})c^{\beta}_{\nu i} = 0 \tag{39}$$

where $H^{\alpha\beta}_{\mu\nu}$, like $\rho^{\alpha\beta}_{\mu\nu}$, is a $(2N \times 2N)$ matrix, with N the number of basis functions:

$$H^{\alpha\beta}_{\mu\nu} = \langle\phi_\mu|\hat{T} + \hat{V}^{KB} + V^{NA}(\boldsymbol{r}) + \delta V^{H}(\boldsymbol{r}) + V^{\alpha\beta}_{XC}(\boldsymbol{r})|\phi_\nu\rangle. \tag{40}$$

This is in contrast to the collinear spin case, in which the Hamiltonian and density matrices can be factorized into two $N \times N$ matrices, one for each spin direction. To calculate $V^{\alpha\beta}_{XC}(\boldsymbol{r})$ we first diagonalize the 2×2 matrix $\rho^{\alpha\beta}(\boldsymbol{r})$ at every point, in order to find the up and down spin densities $\rho^{\uparrow}(\boldsymbol{r})$, $\rho^{\downarrow}(\boldsymbol{r})$ in the direction of the local spin vector. We then find $V^{\uparrow}_{XC}(\boldsymbol{r})$, $V^{\downarrow}_{XC}(\boldsymbol{r})$ in that direction, with the usual local spin density functional [15], and we rotate $V^{\alpha\beta}_{XC}(\boldsymbol{r})$ back to the original direction. Thus, the grid operations are still basically the same, except that they need now to be repeated three times, for the $\uparrow\uparrow$, $\downarrow\downarrow$ and $\uparrow\downarrow$ components. Notice that $\rho^{\alpha\beta}(\boldsymbol{r})$ and $V^{\alpha\beta}_{XC}(\boldsymbol{r})$ are locally Hermitian, while $H^{\alpha\beta}_{\mu\nu}$ and $\rho^{\alpha\beta}_{\mu\nu}$ are globally Hermitian ($H^{\beta\alpha}_{\nu\mu} = H^{\alpha\beta *}_{\mu\nu}$), so their $\downarrow\uparrow$ components can be obtained from the $\uparrow\downarrow$ ones.

8. Brillouin zone sampling

Integration of all magnitudes over the Brillouin zone (BZ) is essential for small and moderately large unit cells, especially of metals. Although SIESTA is designed for large unit cells, in practice it is very useful, especially for comparisons and checks, to be able to also perform calculations efficiently on smaller systems without using expensive superlattices. On the other

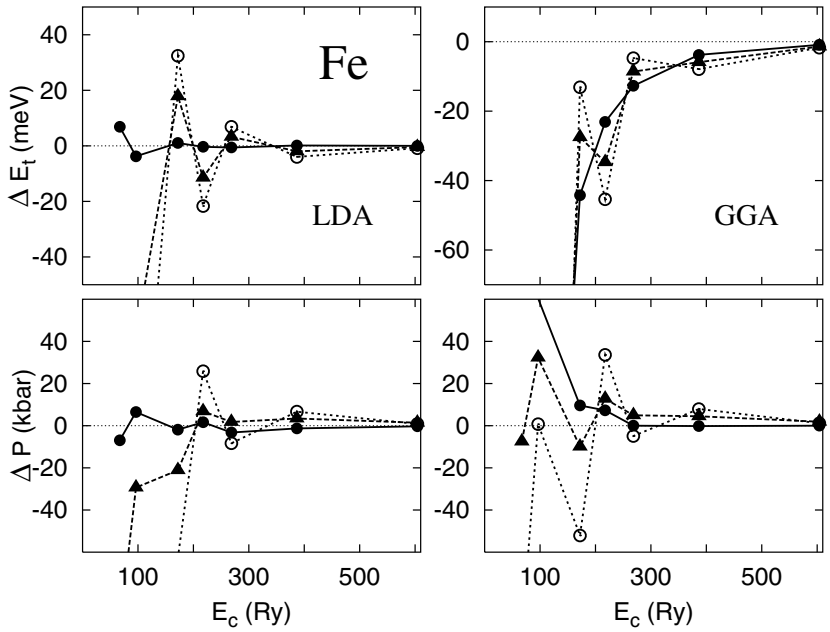

Figure 6. The same as figure 5 for the total energy and pressure of bulk iron. This is presented as an especially difficult case because of the very hard partial core correction ($r_m = 0.7$ au) required for a correct description of XC.

hand, an efficient k-sampling implementation should not penalize, because of the required complex arithmetic, the Γ-point calculations used in large cells. A solution used in some programs is to have two different versions of all or part of the code, but this poses extra maintenance requirements. We have dealt with this problem in the following way: around the unit cell (and comprising the unit cell itself) we define an auxiliary supercell large enough to contain all the atoms whose basis orbitals are nonzero at any of the grid points of the unit cell, or which overlap with any of the basis orbitals in it. We calculate all the nonzero two-centre integrals between the unit-cell basis orbitals and the supercell orbitals, without any complex phase factors. We also calculate the grid integrals between *all* the supercell basis orbitals $\phi_{\mu'}$ and $\phi_{\nu''}$ (primed indices run over all the supercell), but *within the unit cell only*. We accumulate these integrals in the corresponding matrix elements, thus making use of the relation

$$\langle\phi_\mu|V(\boldsymbol{r})|\phi_{\nu'}\rangle = \sum_{(\mu'\nu'')\equiv(\mu\nu')} \langle\phi_{\mu'}|V(\boldsymbol{r})f(\boldsymbol{r})|\phi_{\nu''}\rangle. \tag{41}$$

$f(\boldsymbol{r}) = 1$ for $\boldsymbol{r}$ within the unit cell and is zero otherwise. ϕ_μ is within the unit cell. The notation $\mu' \equiv \mu$ indicates that $\phi_{\mu'}$ and ϕ_μ are equivalent orbitals, related by a lattice vector translation. $(\mu'\nu'') \equiv (\mu\nu')$ means that the sum extends over all pairs of supercell orbitals $\phi_{\mu'}$ and $\phi_{\nu''}$ such that $\mu' \equiv \mu$, $\nu'' \equiv \nu'$, and $\boldsymbol{R}_\mu - \boldsymbol{R}_{\nu'} = \boldsymbol{R}_{\mu'} - \boldsymbol{R}_{\nu''}$. Once all the real overlap and Hamiltonian matrix elements are calculated, we multiply them, at every k-point, by the corresponding phase factors and accumulate them by folding the supercell orbital to its unit-cell counterpart. Thus

$$H_{\mu\nu}(\boldsymbol{k}) = \sum_{\nu'\equiv\nu} H_{\mu\nu'} \mathrm{e}^{\mathrm{i}\boldsymbol{k}(\boldsymbol{R}_{\nu'}-\boldsymbol{R}_\mu)} \tag{42}$$

where ϕ_μ and ϕ_ν are within the unit cell. The resulting $N \times N$ complex eigenvalue problem, with N the number of orbitals in the unit cell, is then solved at every sampled k point, finding

the Bloch-state expansion coefficients $c_{\mu i}(\boldsymbol{k})$:

$$\psi_i(\boldsymbol{k}, \boldsymbol{r}) = \sum_{\mu'} \mathrm{e}^{\mathrm{i}\boldsymbol{k}\boldsymbol{R}_{\mu'}} \phi_{\mu'}(\boldsymbol{r}) c_{\mu' i}(\boldsymbol{k}) \tag{43}$$

where the sum in μ' extends to all basis orbitals in space, i labels the different bands, $c_{\mu' i} = c_{\mu i}$ if $\mu' \equiv \mu$ and $\psi_i(\boldsymbol{k}, \boldsymbol{r})$ is normalized in the unit cell.

The electron density is then

$$\rho(\boldsymbol{r}) = \sum_i \int_{BZ} n_i(\boldsymbol{k}) |\psi_i(\boldsymbol{k}, \boldsymbol{r})|^2 \, \mathrm{d}\boldsymbol{k} = \sum_{\mu'\nu'} \rho_{\mu'\nu'} \phi^*_{\nu'}(\boldsymbol{r}) \phi_{\mu'}(\boldsymbol{r}) \tag{44}$$

where the sum is again over all basis orbitals in space, and the density matrix

$$\rho_{\mu\nu} = \sum_i \int_{BZ} c_{\mu i}(\boldsymbol{k}) n_i(\boldsymbol{k}) c_{i\nu}(\boldsymbol{k}) \mathrm{e}^{\mathrm{i}\boldsymbol{k}(\boldsymbol{R}_\nu - \boldsymbol{R}_\mu)} \, \mathrm{d}\boldsymbol{k} \tag{45}$$

is real (for real ϕ_μ) and periodic, i.e. $\rho_{\mu\nu} = \rho_{\mu'\nu'}$ if $(\nu, \mu) \equiv (\nu', \mu')$ (with '$\equiv$' meaning again 'equivalent by translation'). Thus, to calculate the density at a grid point of the unit cell, we simply find the sum (44) over all the pairs of orbitals ϕ_μ, ϕ_ν in the supercell that are nonzero at that point.

In practice, the integral in (45) is performed in a finite, uniform grid of the BZ. The fineness of this grid is controlled by a k-grid cutoff l_{cut}, a real-space radius which plays a role equivalent to the plane-wave cutoff of the real-space grid [43]. The origin of the k-grid may be displaced from $\boldsymbol{k} = 0$ in order to decrease the number of inequivalent k-points [44].

If the unit cell is large enough to allow a Γ-point-only calculation, the multiplication by phase factors is skipped and a single real-matrix eigenvalue problem is solved (in this case, the real matrix elements are accumulated directly in the first stage, if multiple overlaps occur). In this way, no complex arithmetic penalty occurs, and the differences between Γ-point and k-sampling are limited to a very small section of the code, while all the two-centre and grid integrals always use the same real-arithmetic code.

9. Total energy

The Kohn–Sham [14] total energy can be written as a sum of a band-structure (BS) energy plus some correction terms, sometimes called 'double-count' corrections. The BS term is the sum of the energies of the occupied states ψ_i:

$$E^{BS} = \sum_i n_i \langle \psi_i | \hat{H} | \psi_i \rangle = \sum_{\mu\nu} H_{\mu\nu} \rho_{\nu\mu} = \mathrm{Tr}(H\rho) \tag{46}$$

where spin and k-sampling notations are omitted here for simplicity. At convergence, the ψ_i are simply the eigenvectors of the Hamiltonian, but it is important to realize that the Kohn–Sham functional is also perfectly well defined outside this so-called 'Born–Oppenheimer surface', i.e. it is defined for any set of orthonormal ψ_i. The correction terms are simple functionals of the electron density, which can be obtained from equation (35), and the atomic positions. The Kohn–Sham total energy can then be written as

$$E^{KS} = \sum_{\mu\nu} H_{\mu\nu} \rho_{\nu\mu} - \frac{1}{2} \int V^H(\boldsymbol{r}) \rho(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r} + \int (\epsilon^{xc}(\boldsymbol{r}) - V^{xc}(\boldsymbol{r})) \rho(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r} + \sum_{I<J} \frac{Z_I Z_J}{R_{IJ}} \tag{47}$$

where I, J are atomic indices, $R_{IJ} \equiv |\boldsymbol{R}_J - \boldsymbol{R}_I|$, Z_I, Z_J are the valence ion pseudoatom charges and $\epsilon^{xc}(\boldsymbol{r})\rho(\boldsymbol{r})$ is the exchange–correlation energy density. In order to avoid the long-range interactions of the last term, we construct from the local pseudopotential V_I^{local}, which

has an asymptotic behaviour of $-Z_I/r$, a diffuse ion charge, $\rho_I^{local}(r)$, whose electrostatic potential is equal to $V_I^{local}(r)$:

$$\rho_I^{local}(\boldsymbol{r}) = -\frac{1}{4\pi}\nabla^2 V_I^{local}(\boldsymbol{r}). \tag{48}$$

Notice that we define the electron density as positive, and therefore $\rho_I^{local} \leqslant 0$. Then, we write the last term in (47) as

$$\sum_{I<J}\frac{Z_I Z_J}{R_{IJ}} = \tfrac{1}{2}\sum_{IJ} U_{IJ}^{local}(R_{IJ}) + \sum_{I<J}\delta U_{IJ}^{local}(R_{IJ}) - \sum_I U_I^{local} \tag{49}$$

where U_{IJ}^{local} is the electrostatic interaction between the diffuse ion charges in atoms I and J,

$$U_{IJ}^{local}(|\boldsymbol{R}|) = \int V_I^{local}(\boldsymbol{r})\rho_J^{local}(\boldsymbol{r}-\boldsymbol{R})\,\mathrm{d}^3\boldsymbol{r}, \tag{50}$$

δU_{IJ}^{local} is a small short-range interaction term to correct for a possible overlap between the soft-ion charges, which appears when the core densities are very extended,

$$\delta U_{IJ}^{local}(R) = \frac{Z_I Z_J}{R} - U_{IJ}^{local}(R), \tag{51}$$

and U_I^{local} is the fictitious selfinteraction of an ion charge (notice that the first right-hand sum in (49) includes the $I = J$ terms):

$$U_I^{local} = \tfrac{1}{2}U_{II}^{local}(0) = \tfrac{1}{2}\int V_I^{local}(r)\rho_I^{local}(r)4\pi r^2\,\mathrm{d}r. \tag{52}$$

Defining ρ_I^{NA} from V_I^{NA}, analogously to ρ_I^{local}, we have that $\rho_I^{NA} = \rho_I^{local} + \rho_I^{atom}$, and equation (47) can be transformed, after some rearrangement of terms, into

$$E^{KS} = \sum_{\mu\nu}(T_{\mu\nu} + V_{\mu\nu}^{KB})\rho_{\nu\mu} + \tfrac{1}{2}\sum_{IJ} U_{IJ}^{NA}(R_{IJ}) + \sum_{I<J}\delta U_{IJ}^{local}(R_{IJ}) - \sum_I U_I^{local}$$
$$+ \int V^{NA}(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} + \tfrac{1}{2}\int \delta V^H(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} + \int \epsilon^{xc}(\boldsymbol{r})\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} \tag{53}$$

where $V^{NA} = \sum_I V_I^{NA}$ and $\delta\rho = \rho - \sum_I \rho_I^{atom}$.

$$U_{IJ}^{NA}(R) = \int V_I^{NA}(\boldsymbol{r})\rho_J^{NA}(\boldsymbol{r}-\boldsymbol{R})\,\mathrm{d}^3\boldsymbol{r} = -\frac{1}{4\pi}\int V_I^{NA}(\boldsymbol{r})\nabla^2 V_J^{NA}(\boldsymbol{r}-\boldsymbol{R})\,\mathrm{d}^3\boldsymbol{r} \tag{54}$$

is a radial pairwise potential that can be obtained from $V_I^{NA}(r)$ as a two-centre integral, by the same method as described previously for the kinetic matrix elements:

$$T_{\mu\nu} = \langle\phi_\mu| -\tfrac{1}{2}\nabla^2|\phi_\nu\rangle = -\tfrac{1}{2}\int \phi_\mu^*(\boldsymbol{r})\nabla^2\phi_\nu(\boldsymbol{r}-\boldsymbol{R}_{\mu\nu})\,\mathrm{d}^3\boldsymbol{r}. \tag{55}$$

$V_{\mu\nu}^{KB}$ is also obtained by two-centre integrals:

$$V_{\mu\nu}^{KB} = \sum_\alpha \langle\phi_\mu|\chi_\alpha\rangle v_\alpha^{KB}\langle\chi_\alpha|\phi_\nu\rangle \tag{56}$$

where the sum is over all the KB projectors χ_α that overlap simultaneously with ϕ_μ and ϕ_ν.

Although (53) is the total-energy equation actually used by SIESTA, its meaning may be further clarified if the $I = J$ terms of $\frac{1}{2}\sum_{IJ} U_{IJ}^{NA}(R_{IJ})$ are combined with $\sum_I U_I^{local}$ to yield

$$E^{KS} = \sum_{\mu\nu}(T_{\mu\nu} + V_{\mu\nu}^{KB})\rho_{\nu\mu} + \sum_{I<J} U_{IJ}^{NA}(R_{IJ}) + \sum_{I<J}\delta U_{IJ}^{local}(R_{IJ}) + \sum_I U_I^{atom}$$
$$+ \int V^{NA}(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} + \tfrac{1}{2}\int \delta V^H(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} + \int \epsilon^{xc}(\boldsymbol{r})\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} \tag{57}$$

where

$$U_I^{atom} = \int_0^\infty (V_I^{local}(r) + \tfrac{1}{2} V_I^{atom}(r))\rho_I^{atom}(r) 4\pi r^2 \, \mathrm{d}r \tag{58}$$

is the electrostatic energy of an isolated atom.

The last three terms in equation (53) are calculated using the real-space grid. In addition to getting rid of all long-range potentials (except that implicit in $\delta V^H(\boldsymbol{r})$), the advantage of (53) is that, apart from the relatively slowly varying exchange–correlation energy density, the grid integrals involve $\delta\rho(\boldsymbol{r})$, which is generally much smaller than $\rho(\boldsymbol{r})$. Thus, the errors associated with the finite grid spacing are drastically reduced. Critically, the kinetic energy matrix elements can be calculated almost exactly, without any grid integrations.

It is frequently desirable to introduce a finite electronic temperature T and/or a fixed chemical potential μ, either because of true physical conditions or to accelerate the selfconsistency iteration. Then, the functional that must be minimized is the free energy [45]

$$F(\boldsymbol{R}_I, \psi_i(\boldsymbol{r}), n_i) = E^{KS}(\boldsymbol{R}_I, \psi_i(\boldsymbol{r}), n_i) - \mu \sum_i n_i$$
$$- k_B T \sum_i (n_i \log n_i + (1 - n_i)\log(1 - n_i)). \tag{59}$$

Minimization with respect to n_i yields the usual Fermi–Dirac distribution $n_i = 1/(1 + \mathrm{e}^{(\epsilon_i - \mu)/k_B T})$.

10. Harris functional

We shall mention here a special use of the Harris energy functional, that is generally defined as [46, 47]

$$E^{Harris}[\rho^{in}] = \sum_i n_i^{out} \langle \psi_i^{out} | \hat{H}^{in} | \psi_i^{out} \rangle - \frac{1}{2} \iint \frac{\rho^{in}(\boldsymbol{r})\rho^{in}(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}^3\boldsymbol{r} \, \mathrm{d}^3\boldsymbol{r}'$$
$$+ \int (\epsilon_{xc}^{in}(\boldsymbol{r}) - v_{xc}^{in}(\boldsymbol{r}))\rho^{in}(\boldsymbol{r}) \, \mathrm{d}^3\boldsymbol{r} + \sum_{I<J} \frac{Z_I Z_J}{R_{IJ}} \tag{60}$$

where $\hat{H}^{in}$ is the KS Hamiltonian produced by a trial density ρ^{in} and ψ_i^{out} are its eigenvectors (which in general are different from those whose density is ρ^{in}). As in equation (46), the first term in (60) can be written as $\mathrm{Tr}(H^{in}\rho^{out})$, and the rest are the so-called 'double-count corrections'. An important advantage of equation (60) is that it does not require ρ_i^{in} to be obtained from a set of orthogonal electron states ψ_i^{in}, and in fact ρ^{in} is frequently taken as a simple superposition of atomic densities. However, we shall assume here that the states ψ_i^{in} are indeed known. In this case, the Kohn–Sham energy $E^{KS}[\rho^{in}]$, equation (47), obeys exactly the same expression (60), except that ψ_i^{out} and n_i^{out} must be replaced by ψ_i^{in} and n_i^{in}. Thus, a simple subtraction gives

$$E^{Harris}[\rho^{in}] = E^{KS}[\rho^{in}] + \sum_{\mu\nu} H_{\nu\mu}^{in}(\rho_{\mu\nu}^{out} - \rho_{\mu\nu}^{in}). \tag{61}$$

Generally the Harris functional is used nonselfconsistently, with a trial density given by the sum of atomic densities, but here we want to comment on its usefulness to improve dramatically the estimate of the converged total energy, by taking $\rho_{\mu\nu}^{in}$ as the density matrix of the $(n-1)$th selfconsistency iteration and $\rho_{\mu\nu}^{out}$ of the nth iteration. In fact, E^{Harris} frequently gives, after just two or three iterations, a better estimate than E^{KS} after tens of iterations. Unfortunately, we have found that there is hardly any improvement in the convergence of the atomic forces thus estimated, and therefore the selfconsistent Harris functional is less useful for geometry relaxations or molecular dynamics.

11. Atomic forces

Atomic forces and stresses are obtained by direct differentiation of (53) with respect to atomic positions. They are obtained simultaneously with the total energy, mostly in the same places in the code, under the general paradigm 'a piece of energy ⇒ a piece of force/stress' (except that some pieces are calculated only in the last selfconsistency step). This ensures that all force contributions, including Pulay corrections, are automatically included. The force contribution from the first term in (53) is

$$\frac{\partial}{\partial \boldsymbol{R}_I}\sum_{\mu\nu}(T_{\mu\nu}+V^{KB}_{\mu\nu})\rho_{\nu\mu}=\sum_{\mu\nu}(T_{\mu\nu}+V^{KB}_{\mu\nu})\frac{\partial\rho_{\nu\mu}}{\partial \boldsymbol{R}_I}+2\sum_{\mu}\sum_{\nu\in I}\frac{\mathrm{d}T_{\mu\nu}}{\mathrm{d}\boldsymbol{R}_{\mu\nu}}\rho_{\nu\mu}$$
$$+2\sum_{\mu}\sum_{\nu\in I}\sum_{\alpha}S_{\mu\alpha}v^{KB}_{\alpha}\frac{\mathrm{d}S_{\alpha\nu}}{\mathrm{d}\boldsymbol{R}_{\alpha\nu}}\rho_{\nu\mu}-2\sum_{\mu\nu}\sum_{\alpha\in I}S_{\mu\alpha}v^{KB}_{\alpha}\frac{\mathrm{d}S_{\alpha\nu}}{\mathrm{d}\boldsymbol{R}_{\alpha\nu}}\rho_{\nu\mu} \tag{62}$$

where α are KB projector indices, $\in I$ indicates orbitals or KB projectors belonging to atom I, and we have considered that

$$\frac{\partial S_{\mu\nu}}{\partial \boldsymbol{R}_{I_\nu}}=-\frac{\partial S_{\mu\nu}}{\partial \boldsymbol{R}_{I_\mu}}=\frac{\mathrm{d}S_{\mu\nu}}{\mathrm{d}\boldsymbol{R}_{\mu\nu}}, \tag{63}$$

where $\boldsymbol{R}_{I_\mu}$ is the position of atom I_μ, to which orbital ϕ_μ belongs and $\boldsymbol{R}_{\mu\nu}=\boldsymbol{R}_{I_\nu}-\boldsymbol{R}_{I_\mu}$.

Leaving aside for appendix A the terms containing $\partial\rho_{\nu\mu}/\partial\boldsymbol{R}_I$, the other derivatives can be obtained by straightforward differentiation of their expansion in spherical harmonics (equation (25)). However, instead of using the spherical harmonics $Y_{lm}(\hat{\boldsymbol{r}})$ themselves, it is convenient to multiply them by r^l, in order to make them analytic at the origin. Thus

$$\frac{\mathrm{d}S_{\mu\nu}(\boldsymbol{R})}{\mathrm{d}\boldsymbol{R}}=\sum_{lm}\nabla\left(\frac{S^{\mu\nu}_{lm}(R)}{R^l}R^lY_{lm}(\hat{\boldsymbol{R}})\right)=\sum_{lm}\frac{\mathrm{d}}{\mathrm{d}R}\left(\frac{S^{\mu\nu}_{lm}(R)}{R^l}\right)R^lY_{lm}(\hat{\boldsymbol{R}})\hat{\boldsymbol{R}}$$
$$+\sum_{lm}\frac{S^{\mu\nu}_{lm}(R)}{R^l}\nabla(R^lY_{lm}(\hat{\boldsymbol{R}})). \tag{64}$$

In fact, it is $S^{\mu\nu}_{lm}(R)/R^l$, rather than $S^{\mu\nu}_{lm}(R)$, that is stored as a function of R on a radial grid. Its derivative, $\mathrm{d}(S^{\mu\nu}_{lm}(R)/R^l)/\mathrm{d}R$, is then obtained from the same cubic spline interpolation as used for the value itself. The value and gradient of $R^lY_{lm}(\hat{\boldsymbol{R}})$ are calculated analytically from explicit formulae (up to $l=2$) or recurrence relations [28]. Entirely analogous equations apply to $\mathrm{d}T_{\mu\nu}/\mathrm{d}\boldsymbol{R}_{\mu\nu}$.

The second and third terms in equation (53) are simple interatomic pair potentials whose force contributions are calculated trivially from their radial spline interpolations. The fourth term is a constant which does not depend on the atomic positions. Taking into account that $V^{NA}(\boldsymbol{r})=\sum_I V^{NA}_I(\boldsymbol{r}-\boldsymbol{R}_I)$, and therefore $\partial V^{NA}(\boldsymbol{r})/\partial\boldsymbol{R}_I=-\nabla V^{NA}_I(\boldsymbol{r}-\boldsymbol{R}_I)$, the force contribution from the fifth term is

$$\frac{\partial}{\partial \boldsymbol{R}_I}\int V^{NA}(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r}=-\int\nabla V^{NA}_I(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r}+\int V^{NA}(\boldsymbol{r})\frac{\partial\delta\rho(\boldsymbol{r})}{\partial\boldsymbol{R}_I}\,\mathrm{d}^3\boldsymbol{r}. \tag{65}$$

The sixth term is the electrostatic selfenergy of the charge distribution $\delta\rho(\boldsymbol{r})$:

$$\frac{\partial}{\partial\boldsymbol{R}_I}\frac{1}{2}\int\delta V^H(\boldsymbol{r})\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r}=\int\delta V^H(\boldsymbol{r})\frac{\partial\delta\rho(\boldsymbol{r})}{\partial\boldsymbol{R}_I}\,\mathrm{d}^3\boldsymbol{r}. \tag{66}$$

In the last term, we take into account that $\mathrm{d}(\rho\epsilon^{xc})/\mathrm{d}\rho=v^{xc}$ to obtain

$$\frac{\partial}{\partial\boldsymbol{R}_I}\int\epsilon^{xc}(\boldsymbol{r})\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r}=\int V^{xc}(\boldsymbol{r})\frac{\partial\rho(\boldsymbol{r})}{\partial\boldsymbol{R}_I}\,\mathrm{d}^3\boldsymbol{r}. \tag{67}$$

Now, using equation (35) and that, for $\nu \in I$, $\partial\phi_\nu(\boldsymbol{r})/\partial\boldsymbol{R}_I = -\nabla\phi_\nu$, the changes of the selfconsistent and atomic densities are

$$\frac{\partial\rho(\boldsymbol{r})}{\partial\boldsymbol{R}_I} = \mathrm{Re}\sum_{\mu\nu}\frac{\partial\rho_{\nu\mu}}{\partial\boldsymbol{R}_I}\phi^*_\mu(\boldsymbol{r})\phi_\nu(\boldsymbol{r}) - 2\mathrm{Re}\sum_\mu\sum_{\nu\in I}\rho_{\nu\mu}\phi^*_\mu(\boldsymbol{r})\nabla\phi_\nu(\boldsymbol{r}) \tag{68}$$

$$\frac{\partial\rho^{atom}(\boldsymbol{r})}{\partial\boldsymbol{R}_I} = -2\mathrm{Re}\sum_{\mu\in I}\rho^{atom}_{\mu\mu}\phi^*_\mu(\boldsymbol{r})\nabla\phi_\mu(\boldsymbol{r}) \tag{69}$$

where we have taken into account that the density matrix of the separated atoms is diagonal. Thus, still leaving aside the terms with $\partial\rho_{\nu\mu}/\partial\boldsymbol{R}_I$, the last term in equation (65), as well as those in (66) and (67), has the general form

$$\mathrm{Re}\sum_\mu\sum_{\nu\in I}\rho_{\nu\mu}\int V(\boldsymbol{r})\phi^*_\mu(\boldsymbol{r})\nabla\phi_\nu(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} = \mathrm{Re}\sum_\mu\sum_{\nu\in I}\rho_{\nu\mu}\langle\phi_\mu|V(\boldsymbol{r})|\nabla\phi_\nu\rangle. \tag{70}$$

These integrals are calculated on the grid, in the same way as those for the total energy (i.e. $\langle\phi_\mu|V(\boldsymbol{r})|\phi_\nu\rangle$). The gradients $\nabla\phi_\nu(\boldsymbol{r})$ at the grid points are obtained analytically, like those of $\phi_\nu(\boldsymbol{r})$ from their radial grid interpolations of $\phi(r)/r^l$:

$$\nabla\phi_{Ilmn}(\boldsymbol{r}) = \frac{\mathrm{d}}{\mathrm{d}r}\left(\frac{\phi_{Iln}(r)}{r^l}\right)r^l Y_{lm}(\hat{\boldsymbol{r}})\hat{\boldsymbol{r}} + \frac{\phi_{Iln}(r)}{r^l}\nabla(r^l Y_{lm}(\hat{\boldsymbol{r}})). \tag{71}$$

In some special cases, with elements that require hard-partial-core corrections or explicit inclusion of the semicore, the grid integrals may pose a problem for geometry relaxations, because they make the energy dependent on the position of the atoms relative to the grid. This 'eggbox effect' is small for the energy itself, and it decreases fast with the grid spacing, but the effect is larger and the convergence slower for the forces, as they are proportional to the amplitude of the energy oscillation, but inversely proportional to its period. These force oscillations complicate the force landscape, especially when the true atomic forces become small, making the convergence of the geometry optimization more difficult. Of course, the problem can be avoided by decreasing the grid spacing but this has an additional cost in computer time and memory. Therefore, we have found it useful to minimize this problem by recalculating the forces, at a set of positions, determined by translating the whole system by a set of points in a finer mesh. This procedure, which we call 'grid-cell sampling', has no extra cost in memory, and since it is done only at the end of the selfconsistency iteration, for fixed $\rho_{\mu\nu}$, it has only a moderate cost in CPU time.

At finite temperature, the forces are really the derivatives of the *free* energy with respect to atomic displacements since

$$\frac{\mathrm{d}F(\boldsymbol{R}_I,\psi_i(\boldsymbol{r}),n_i)}{\mathrm{d}\boldsymbol{R}_I} = \frac{\partial F}{\partial\boldsymbol{R}_I} + \sum_i\frac{\partial F}{\partial n_i}\frac{\partial n_i}{\partial\boldsymbol{R}_I} + \sum_i\int\frac{\partial F}{\partial\psi^*_i(\boldsymbol{r})}\frac{\partial\psi_i(\boldsymbol{r})}{\partial\boldsymbol{R}_I}\,\mathrm{d}^3\boldsymbol{r} = \frac{\partial E}{\partial\boldsymbol{R}_I}. \tag{72}$$

In this particular equation we have used the notation $\mathrm{d}/\mathrm{d}\boldsymbol{R}_I$, as opposed to $\partial/\partial\boldsymbol{R}_I$, to indicate the inclusion of the change in $\psi_i(\boldsymbol{r})$ and n_i when we move the atom, in calculating the derivative, but we have used also that $\partial F/\partial n_i = \partial F/\partial\psi_i(\boldsymbol{r}) = 0$ and that the last two terms in (59) do not depend on $\boldsymbol{R}_I$, so that $\partial F/\partial\boldsymbol{R}_I = \partial E/\partial\boldsymbol{R}_I$. The latter are the atomic forces actually calculated. Notice, however, that $\mathrm{d}E/\mathrm{d}\boldsymbol{R}_I \neq \partial E/\partial\boldsymbol{R}_I$, so the calculated forces are indeed the total derivatives of the free, not the internal energy.

We would like to also mention the calculation of forces using the nonselfconsistent Harris functional, in which the 'in' density is a superposition of atomic densities. We have implemented this as an option for 'quick and dirty' calculations because, used with a minimal basis set, it makes SIESTA competitive with tight-binding methods, which are much faster than density functional calculations. The problem that we address here is that, although E^{Harris} is

stationary with respect to ρ^{out}, it is not so with respect to ρ^{in}. In particular, there appears a force term

$$\int \frac{\partial V_{xc}^{in}(\boldsymbol{r})}{\partial \boldsymbol{R}_I} \rho^{out}(\boldsymbol{r})\, \mathrm{d}^3\boldsymbol{r}. \tag{73}$$

A similar term appears for the electrostatic interaction between the input and output density, but it presents no special problems because of the linear character of the Hartree potential. However, evaluation of (73) requires the change of the exchange–correlation potential with density, a quantity also required to evaluate the linear response of the electron gas, but not in normal energy and force calculations. Finally, notice that, apart from this minor difficulty, the Harris-functional forces are perfectly well defined *at the first iteration only*. For later iterations (but still not converged) there is no practical way to calculate $\partial\rho^{in}/\partial\boldsymbol{R_I}$ and, without the help of the Hellman–Feynmann theorem (which applies only at convergence), the forces are not well defined. Of course, the omission of the terms depending on this quantity produces an estimate of the forces, but we have found that their convergence is not appreciably faster than those estimated from the Kohn–Sham functional.

12. Stress tensor

We define the stress tensor as the positive derivative of the total energy with respect to the strain tensor

$$\sigma_{\alpha\beta} = \frac{\partial E^{KS}}{\partial \epsilon_{\alpha\beta}} \tag{74}$$

where α, β are Cartesian coordinate indices. To translate to standard units of pressure, we must simply divide by the unit-cell volume and change sign. During the deformation, all vector positions, including those of atoms and grid points (and of course lattice vectors), change according to

$$r'_\alpha = \sum_{\beta=1}^{3} (\delta_{\alpha\beta} + \epsilon_{\alpha\beta}) r_\beta. \tag{75}$$

The shapes of the basis functions, KB projectors and atomic densities and potentials do not change, but their origin is displaced according to (75). From this equation, we find that

$$\frac{\partial r_\gamma}{\partial \epsilon_{\alpha\beta}} = \delta_{\gamma\alpha} r_\beta. \tag{76}$$

The change in E^{KS} is essentially due to these position displacements, and therefore the calculation of the stress is almost perfectly parallel to that of the atomic forces, thus being performed in the same sections of the code. For example,

$$\frac{\partial T_{\mu\nu}}{\partial \epsilon_{\alpha\beta}} = \sum_{\gamma=1}^{3} \frac{\partial T_{\mu\nu}}{\partial r^\gamma_{\mu\nu}} \frac{\partial r^\gamma_{\mu\nu}}{\partial \epsilon_{\alpha\beta}} = \frac{\partial T_{\mu\nu}}{\partial r^\alpha_{\mu\nu}} r^\beta_{\mu\nu}. \tag{77}$$

Since $\partial T_{\mu\nu}/\partial r^\alpha_{\mu\nu}$ is evaluated to calculate the forces, it takes very little extra effort to also multiply it by $r^\beta_{\mu\nu}$ for the stress. Equally, force contributions such as (70) have their obvious stress counterpart

$$\sum_{\mu\nu} \rho_{\nu\mu} \langle \phi_\mu | V(\boldsymbol{r}) | (\nabla_\alpha \phi_\nu) r_\beta \rangle. \tag{78}$$

However, there are three exceptions to this parallelism. The first concerns the change of the volume per grid point or, in other words, the Jacobian of the transformation (75) in the integrals over the unit cell. This Jacobian is simply $\delta_{\alpha\beta}$, and it leads to a stress contribution

$$\left[\int (V^{NA}(\boldsymbol{r}) + \tfrac{1}{2}\delta V^{H}(\boldsymbol{r}))\delta\rho(\boldsymbol{r})\,\mathrm{d}^3\boldsymbol{r} + E^{xc}\right]\delta_{\alpha\beta}. \tag{79}$$

Notice that the renormalization of the density, required to conserve the charge when the volume changes, enters through the orthonormality constraints, to be discussed in appendix A. The second special contribution to the stress lies in the fact that, as we deform the lattice, there is a change in the factor $1/|\boldsymbol{r}-\boldsymbol{r}'|$ of the electrostatic energy integrals. We deal with this contribution in reciprocal space, when we calculate the Hartree potential by FFTs, by evaluating the derivative of the reciprocal-space vectors with respect to $\epsilon_{\alpha\beta}$. Since $G'_\alpha = \sum_\beta G_\beta(\delta_{\beta\alpha} - \epsilon_{\beta\alpha})$,

$$\frac{\partial}{\partial\epsilon_{\alpha\beta}}\frac{1}{G^2} = \frac{2G_\alpha G_\beta}{G^4}. \tag{80}$$

Finally, the third special stress contribution arises in GGA XC, from the change of the gradient of the deformed density $\rho(\boldsymbol{r}) \to \rho(\boldsymbol{r}')$. The treatment of this contribution is explained in detail in [?].

13. Electric polarization

The calculation of the electric polarization, as an integral in the grid across the unit cell, is standard and almost free for molecules, chains and slabs (in the directions perpendicular to the chain axis, or to the surface). For bulk systems, the electric polarization cannot be found from the charge distribution in the unit cell alone. In this case, we need the so-called Berry-phase theory of polarization [48,49], which allows us to compute quantities such as the dynamical charges [48] and piezoeletric constants [50,51]. Here we comment some details of our implementation [52].

If $\boldsymbol{R}_\alpha$ are the lattice vectors and $\boldsymbol{P}^e = \sum_{\alpha=1}^{3} P^e_\alpha \boldsymbol{R}_\alpha$ is the electronic contribution to the macroscopic polarization, then we have

$$2\pi\ P^e_\alpha = \boldsymbol{G}_\alpha \cdot \boldsymbol{P}^e = -\frac{2e}{(2\pi)^3}\int_{BZ} \mathrm{d}\boldsymbol{k}\,\boldsymbol{G}_\alpha \cdot \frac{\partial}{\partial \boldsymbol{k}'}\Phi(\boldsymbol{k},\boldsymbol{k}')\bigg|_{\boldsymbol{k}'=\boldsymbol{k}} \tag{81}$$

where $\boldsymbol{G}_\alpha$ is the corresponding reciprocal lattice vector, e is the electron charge, $u_i(\boldsymbol{k},\boldsymbol{r}) = \mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}\psi_i(\boldsymbol{k},\boldsymbol{r})$ is the periodic part of the Bloch function and the factor of two originates from the spin degeneracy. The quantum phase $\Phi(\boldsymbol{k},\boldsymbol{k}')$ is defined as

$$\Phi(\boldsymbol{k},\boldsymbol{k}') = \mathrm{Im}\,[\ln(\det\langle u_i(\boldsymbol{k},\boldsymbol{r})|u_j(\boldsymbol{k}',\boldsymbol{r})\rangle)]. \tag{82}$$

The derivative in (81) depends on a gauge that must be chosen such that $u(\boldsymbol{k}+\boldsymbol{G},\boldsymbol{r}) = \mathrm{e}^{-\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}}u(\boldsymbol{k},\boldsymbol{r})$. In practice, the integral is replaced by a discrete summation, and a finite-difference approximation is made for the derivative [48]: $\Delta\boldsymbol{k}_\alpha \frac{\partial}{\partial\boldsymbol{k}'_\alpha}\Phi(\boldsymbol{k},\boldsymbol{k}')|_{\boldsymbol{k}'=\boldsymbol{k}} \approx \frac{1}{2}[\Phi(\boldsymbol{k},\boldsymbol{k}+\Delta\boldsymbol{k}_\alpha) - \Phi(\boldsymbol{k},\boldsymbol{k}-\Delta\boldsymbol{k}_\alpha)]$, where $\Delta\boldsymbol{k}_\alpha = \boldsymbol{G}_\alpha/N_\alpha$. Then (81) becomes, for $\alpha = 1$,

$$\boldsymbol{G}_1 \cdot \boldsymbol{P}_e \approx -\frac{2e}{\Omega N_2 N_3}\sum_{i_2=0,i_3=0}^{N_2-1,N_3-1}\ \sum_{i_1=0}^{N_1-1}\Phi(\boldsymbol{k}_{i_1 i_2 i_3}, \boldsymbol{k}_{i_1+1 i_2 i_3}), \tag{83}$$

where we have split the sum to stress the fact that we have a two-dimensional integral in the plane defined by $\boldsymbol{G}_2$ and $\boldsymbol{G}_3$, and a linear integral along $\boldsymbol{G}_1$. Due to the approximation in

the derivative, the linear integral usually requires a finer mesh than the surface integral. To evaluate $\Phi(\boldsymbol{k}, \boldsymbol{k}+\Delta\boldsymbol{k})$ we use our LCAO basis:

$$\langle u_i(\boldsymbol{k})|u_j(\boldsymbol{k}+\Delta\boldsymbol{k})\rangle = \langle\psi_i(\boldsymbol{k})|\mathrm{e}^{-\mathrm{i}\Delta\boldsymbol{k}\cdot\boldsymbol{r}}|\psi_j(\boldsymbol{k}+\Delta\boldsymbol{k})\rangle$$
$$= \sum_{\nu}\sum_{\mu'} c_{i\nu}(\boldsymbol{k})c_{\mu' j}(\boldsymbol{k}+\Delta\boldsymbol{k})\mathrm{e}^{-\mathrm{i}\boldsymbol{k}\cdot(\boldsymbol{R}_\nu-\boldsymbol{R}_{\mu'})}\langle\phi_\nu|\mathrm{e}^{-\mathrm{i}\Delta\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{R}_{\mu'})}|\phi_{\mu'}\rangle. \quad (84)$$

Formulae similar to (84) have been implemented by several authors [53, 54], mainly in the context of Hartree–Fock calculations, in which the basis orbitals are expanded in Gaussians whose matrix elements can be found analytically [53]. Our numerical, localized pseudo-atomic basis orbitals are not well suited for a Gaussian expansion. Instead, we expand the plane waves appearing in equation (84) to first order in $\Delta\boldsymbol{k}$, $\mathrm{e}^{-\mathrm{i}\Delta\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{R}_\mu)} \approx 1 - i\,\Delta\boldsymbol{k}\cdot(\boldsymbol{r}-\boldsymbol{R}_\mu) + \mathcal{O}(\Delta k^2)$, and then we calculate the matrix elements of the position operator as explained in section 5. It is interesting to note that, since the discretized formula (83) only holds to $\mathcal{O}(\Delta k^2)$, the approximation of the matrix elements in (84) does not introduce any further errors in the calculation of the polarization. In a symmetrized version, we approximate equation (84) as

$$\sum_{\nu}\sum_{\mu'} c_{i\nu}(\boldsymbol{k})c_{\mu' j}(\boldsymbol{k}+\Delta\boldsymbol{k})\mathrm{e}^{-\mathrm{i}(\boldsymbol{k}+\frac{\Delta\boldsymbol{k}}{2})\cdot(\boldsymbol{R}_\nu-\boldsymbol{R}_{\mu'})}$$
$$\times\left[\langle\phi_\nu|\phi_{\mu'}\rangle - \mathrm{i}\frac{\Delta\boldsymbol{k}}{2}\cdot(\langle\phi_\nu|(\boldsymbol{r}-\boldsymbol{R}_\nu)|\phi_{\mu'}\rangle + \langle\phi_\nu|(\boldsymbol{r}-\boldsymbol{R}_{\mu'})|\phi_{\mu'}\rangle)\right]. \quad (85)$$

14. Order-N functional

The basic problem for solving the Kohn–Sham equations in $\mathcal{O}(N)$ operations is that the solutions (the Hamiltonian eigenvectors) are extended over the whole system and overlap with each other. Just to check the orthogonality of N trial solutions, by performing integrals over the whole system, involves $\sim N^3$ operations. Among the different methods proposed to solve this problem [5, 9], we have chosen the localized-orbital approach [6, 55, 56] because of its superior efficiency for nonorthogonal basis sets. The initially proposed functional [6, 55] used a fixed number of occupied states, equal to the number of electron pairs, and it was found to have numerous local minima in which the electron configuration was easily trapped. A revised functional form [56], which uses a larger number of states than electron pairs, with variable occupations, has been found empirically to avoid the local-minimum problem. This is the functional that we use and recommend.

Each of the localized, Wannier-like states is constrained to its own localization region. Each atom I is assigned a number of states equal to $\mathrm{int}(Z_I^{val}/2 + 1)$ so that, if doubly occupied, they can contain at least one excess electron (they can also become empty during the minimization of the energy functional). These states are confined to a sphere of radius R_c (common to all states) centred at $\boldsymbol{R}_I$. More precisely, the expansion (equation (32)) of a state ψ_i centred at $\boldsymbol{R}_I$ may contain only basis orbitals ϕ_μ centred on atoms J such that $|\boldsymbol{R}_{IJ}| < R_c$. This implies that $\psi_i(\boldsymbol{r})$ may extend to a maximum range $R_c + r_c^{max}$, where r_c^{max} is the maximum range of the basis orbitals. For covalent systems, a localization region centred on bonds rather than atoms is more efficient [57] (it leads to a lower energy for the same R_c), but it is less suitable for a general algorithm, especially in the case of ambiguous bonds. Therefore, we generally use the atom-centred localization regions.

In the method of Kim, Mauri and Galli (KMG) [56], the BS energy is rewritten as

$$E^{KMG} = 2\sum_{ij}(2\delta_{ji} - S_{ji})(H_{ij} - \eta S_{ij}) = 4\sum_{i}\sum_{\mu\nu} c_{i\mu}\delta H_{\mu\nu}c_{\nu i}$$
$$- 2\sum_{ij}\sum_{\alpha\beta\mu\nu} c_{i\alpha}S_{\alpha\beta}c_{\beta j}c_{j\mu}\delta H_{\mu\nu}c_{\nu i} \quad (86)$$

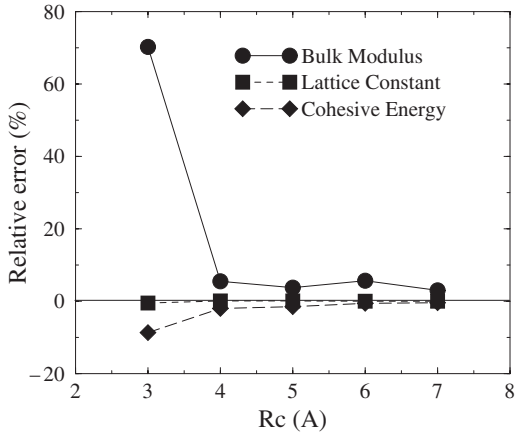

Figure 7. Convergence of the lattice constant, bulk modulus and cohesive energy as a function of the localization radius R_c of the Wannier-like electron states in silicon. We used a supercell of 512 atoms and a minimal basis set with a cutoff radius $r_c = 5$ au for both s and p orbitals.

where $S_{ij} = \langle\psi_i|\psi_j\rangle$, $H_{ij} = \langle\psi_i|H|\psi_j\rangle$, $\delta H_{\mu\nu} = H_{\mu\nu} - \eta S_{\mu\nu}$ and we have assumed a nonmagnetic solution with doubly occupied states. The 'double-count' correction terms of equation (47) remain unchanged and the electron density is still defined by (35), but the density matrix is re-defined as

$$\rho_{\mu\nu} = 2\sum_{ij} c_{\mu i}(2\delta_{ij} - S_{ij})c_{j\nu} = 4\sum_{i} c_{\mu i}c_{i\nu} - 2\sum_{ij}\sum_{\alpha\beta} c_{\mu i}c_{i\alpha}S_{\alpha\beta}c_{\beta j}c_{j\nu}. \tag{87}$$

The parameter η in equation (86) plays the role of a chemical potential, and must be chosen to lie within the bandgap between the occupied and empty states. This may be tricky sometimes, since the electron bands can shift during the selfconsistency process or when the atoms move. In general, the number of electrons will not be exactly the desired one, even if η is within the bandgap, because the minimization of (86) implies a tradeoff in which the localized states become fractionally occupied. To avoid an infinite Hartree energy in periodic systems, we simply renormalize the density matrix so that the total electron charge $\sum_{\mu\nu} S_{\mu\nu}\rho_{\nu\mu}$ is equal to the required value.

For a given potential, the functional (86) is minimized by the conjugate-gradients method, using its derivatives with respect to the expansion coefficients

$$\frac{\partial E^{KMG}}{\partial c_{i\mu}} = 4\sum_{\nu}\delta H_{\mu\nu}c_{\nu i} - 2\sum_{j}\sum_{\alpha\beta\nu}(S_{\mu\nu}c_{\nu j}c_{j\alpha}\delta H_{\alpha\beta}c_{\beta i} + \delta H_{\mu\nu}c_{\nu j}c_{j\alpha}S_{\alpha\beta}c_{\beta i}). \tag{88}$$

The minimization proceeds without any need to orthonormalize the electron states ψ_i. Instead, the orthogonality, as well as the correct normalization (unity below η and zero above it), results from the minimization of E^{KMG}. This is because, in contrast to the KS functional, E^{KMG} is designed to penalize any nonorthogonality [56]. The KS ground state, with all the occupied ψ_i orthonormal, is also the minimum of (86), at which $E^{KMG} = E^{KS}$. If the variational freedom is constrained by the localization of the ψ_i, the orthogonality cannot be exact, and the resulting energy is slightly larger than for unconstrained wavefunctions. In insulators and semiconductors, the Wannier functions are exponentially localized [58], and the energy excess due to their strict localization decreases rapidly as a function of the localization radius R_c, as can be seen in figure 7.

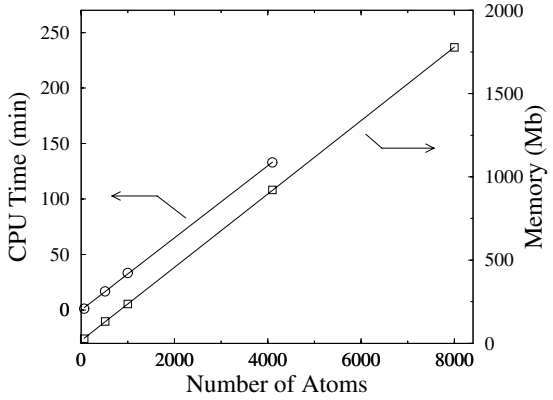

Figure 8. CPU time and memory for silicon supercells of 64, 512, 1000, 4096 and 8000 atoms. Times are for one average molecular dynamics step at 300 K. This includes ten SCF steps, each with ten conjugate gradient minimization steps of the $\mathcal{O}(N)$ energy functional. Memories are peak ones. Although the memory requirement for 8000 atoms was determined accurately, the run could not be performed because of insufficient memory in the PC used.

If the system is metallic, or if the chemical potential is not within the bandgap (for example because of the presence of defects), the KMG functional cannot be used in practice. In fact, although some $\mathcal{O}(N)$ methods can handle metallic systems in principle [9], we are not aware of any practical calculations at a DFT level. In such cases we copy the Hamiltonian and overlap matrices to standard expanded arrays and solve the generalized eigenvalue problem by conventional order-N^3 diagonalization techniques [59]. However, even in this case, most of the operations, and particularly those to find the density and potential, and to set up the Hamiltonian, are still performed in $\mathcal{O}(N)$ operations.

Irrespective of whether the $\mathcal{O}(N)$ functional or the standard diagonalization is used, an outer selfconsistency iteration is required, in which the density matrix is updated using Pulay's residual metric minimization by direct inversion of the iterative subspace (RMM-DIIS) method [60, 61]. Even when the code is strictly $\mathcal{O}(N)$, the CPU time may increase faster if the number of iterations required to achieve the solution increases with N. In fact, it is a common experience that the required number of selfconsistency iterations increases with the size of the system. This is mainly because of the 'charge sloshing' effect, in which small displacements of charge from one side of the system to the other give rise to larger changes of the potential as the size increases. Fortunately, the localized character of the Wannier-like wavefunctions used in the $\mathcal{O}(N)$ method helps to solve also this problem, by limiting the charge sloshing. Table 2 presents the average number of iterations required to minimize the $\mathcal{O}(N)$ functional and the average number of selfconsistency iterations, during a molecular dynamics simulation of bulk silicon at room temperature. It can be seen that these numbers are quite small and that they increase very moderately with system size. As might be expected, the number of minimization iterations increases with the localization radius, i.e. with the number of degrees of freedom ($c_{\mu i}$ coefficients) of the wavefunctions, but this increase is also rather moderate.

Figure 8 shows the essentially perfect $\mathcal{O}(N)$ behaviour of the overall CPU time and memory. This is not surprising in view of the completely strict enforcement of $\mathcal{O}(N)$ algorithms everywhere in the code (except the marginal $N \log N$ factor in the FFT used to solve Poisson's equation, which represents a very small fraction of CPU time even for 4000 atoms).

Table 2. Average number of selfconsistency (SCF) iterations (per molecular dynamics step) and average number of conjugate-gradient (CG) iterations (per SCF iteration) required to minimize the $\mathcal{O}(N)$ functional, during a simulation of bulk silicon at ∼300 K. We used the Verlet method [62] at constant energy, with a time step of 1.5 fs, and a minimal basis set with a cutoff radius $r_c = 5$ au. R_c is the localization radius of the Wannier-like wavefunctions used in the $\mathcal{O}(N)$ functional (see text). N is the number of atoms in the system.

	$R_c = 4$ Å		$R_c = 5$ Å	
N	CG	SCF	CG	SCF
64	5.8	9.3	8.4	8.4
512	4.9	11.4	8.8	10.1
1000	4.3	11.5	9.9	11.5

15. Other features

Here we shall simply mention some of the possibilities and features of the SIESTA implementation of DFT.

- A general-purpose package [63], the flexible data format (fdf), initially developped for the SIESTA project, allows the introduction of all the data and precision parameters in a simple tag-oriented, order-independent format which accepts different physical units. The data can then be accessed from anywhere in the program, using simple subroutine calls in which a default value is specified for the case in which the data are not present. A simple call also allows the read pointer to be positioned in order to read complex data 'blocks' also marked with tags.
- The systematic calculation of atomic forces and stress tensor allows the simultaneous relaxation of atomic coordinates and cell shape and size, using a conjugate gradients minimization or several other minimization/annealing algorithms.
- It is possible to perform a variety of molecular dynamics simulations, at constant energy or temperature, and at constant volume or pressure, also including Parrinello–Rahman dynamics with variable cell shape [62]. The geometry relaxation may be restricted, to impose certain positions or coordinates, or more complex constraints.
- The auxiliary program VIBRA processes systematically the atomic forces for sets of displaced atomic positions, and from them computes the Hessian matrix and the phonon spectrum. An interface to the PHONON program [64] is also provided within SIESTA.
- A linear response program (LINRES) to calculate phonon frequencies has also been developed [65]. The code reads the SCF solution obtained by SIESTA, and calculates the linear response to the atomic displacements, using first-order perturbation theory. It then calculates the dynamical matrix, from which the phonon frequencies are obtained.
- A number of auxiliary programs allows various representations of the total density, the total and local density of states and the electrostatic or total potentials. The representations include both two-dimensional cuts and three-dimensional views, which may be coloured to simultaneously represent the density and potential.
- Thanks to an interface with the TRANSIESTA program, it is posible to calculate transport properties across a nanocontact, finding selfconsistently the effective potential across a finite voltage drop, at a DFT level, using the Keldysh Green function formalism [66].
- The optical response can be studied with SIESTA using different approaches. An approximate dielectric function can be calculated from the dipolar transition matrix elements between occupied and unoccupied single-electron eigenstates using first-order time-dependent pertubation theory [67]. For finite systems, these are easily calculated

from the matrix elements of the position operator between the basis orbitals. For infinite periodic systems, we use the matrix elements of the momentum operator. It is important to notice, however, that the use of nonlocal pseudopotentials requires some correction terms [68].

We have also implemented a more sophisticated approach to compute the optical response of finite systems, using the adiabatic approximation to time-dependent DFT [69, 70]. The idea is to integrate the time-dependent Schrödinger equation when a time-dependent perturbation is applied to the system [71]. From the time evolution, it is then possible to extract the optical adsorption and dipole strength functions, including some genuinely many-body effects, such as plasmons. Using this approach we have succesfully calculated the electronic response of systems such as fullerenes and small metallic clusters [72].

16. Summary

We have presented one of the first fully operational $\mathcal{O}(N)$ implementations of DFT. This implementation has required many scientific and technical breakthroughs to calculate all the terms of the DFT Hamiltonian and to solve Schrödinger equation in strictly $\mathcal{O}(N)$ operations. Some of these innovations are the following.

- A flexible, numerical atomic basis set, which allows extremely fast calculations, using a minimal-basis, as well as highly converged ones, using multiple-ζ + polarization bases. New methods have been devised to generate these numerical basis sets, adapting well known principles of quantum chemistry, such as the split-valence concept.
- Norm-conserving pseudopotentials optimized for smoothness of the local potential, while the application of KB nonlocal projectors to our atomic basis orbitals is nearly free using two-centre integrals.
- A flexible, efficient and general method to calculate two-centre integrals of arbitrary numerical radial functions, using convolutions and a new FFT method for radial functions.
- Evaluation of the matrix elements of the selfconsistent potential using a regular real-space grid. The density gradient is evaluated by finite differences in the grid, to calculate the XC potential of the GGA.
- New expressions for the total energy and forces, in which long-range interactions are handled efficiently by using the difference between the Hartree potentials of the selfconsistent density and of the sum of atomic densities. This also considerably decreases the errors due to the finite integration grid.
- Minimization of an $\mathcal{O}(N)$ functional [56] with localized Wannier-like orbitals allows us to find the electronic ground state without any need to orthogonalize the one-electron states, which instead become orthonormal as a result of the minimization. The ground-state information is 'coded' into the one-electron density matrix, which is then used to find the electron density and total energy, without any further knowledge of the individual electron states. This allows a unified treatment of the ground states obtained by $\mathcal{O}(N)$ methods or by conventional Hamiltonian diagonalization, as well as the inclusion of k-sampling and finite-temperature effects.
- It has been found that the Harris-functional energy converges much faster than the Kohn–Sham energy, even if it is the latter (or the $\mathcal{O}(N)$ functional) that is minimized. As a single-iteration scheme, with a minimal basis set, the Harris functional provides a nonselfconsistent, but reasonable and extremely fast, method for initial relaxations and exploratory molecular dynamics.

In conclusion, the SIESTA method provides a very general scheme to perform a range of calculations from very fast to very accurate, depending on the needs and stage of the simulation, of all kinds of molecule, material and surface. It allows DFT simulations of more than a thousand atoms in modest PC workstations, and over a hundred thousand atoms in parallel platforms [73].

Acknowledgments

We are deeply indebted to Otto Sankey and David Drabold for allowing us to use their code as an initial seed for this project, and to Richard Martin for continuous ideas and support. We thank Jose Luis Martins for numerous discussions and ideas, and Jürgen Kübler for helping us implement the noncollinear spin. The exchange–correlation methods and routines were developed in collaboration with Carlos Balb'as and Jose L Martins. We also thank In-Ho Lee, Maider Machado, Juana Moreno and Art R Williams for some routines, and Eduardo Anglada and Oscar Paz for their computational help. This work was supported by the Fundación Ramón Areces and by Spain's MCyT grant BFM2000-1312. JDG would like to thank the Royal Society for a University Research Fellowship and EPSRC for the provision of computer facilities. DSP acknowledges support from the Basque Government (Programa de Formación de Investigadores).

Appendix A. Orthogonality force and stress

We have yet to comment on the force and stress terms containing $\partial\rho_{\mu\nu}/\partial\boldsymbol{R}_I$. Substituting the first term of equation (68) into equations (65)–(67) and adding the first term of equation (62) we obtain a simple expression: $\sum_{\mu\nu} H_{\nu\mu}\partial\rho_{\mu\nu}/\partial\boldsymbol{R}_I$. Now, $\rho_{\mu\nu}$ is a function of the Hamiltonian eigenvector coefficients and occupations only (equation (34)). On the Born–Oppenheimer surface (BOS), E^{KS} is stationary with respect to these coefficients and occupations, and the Hellman–Feynmann theorem guarantees that any change of them will not modify the total energy to first order, and therefore will not affect the forces. In other words, the atomic forces are the partial derivatives $\partial E^{KS}/\partial\boldsymbol{R}_I$ *at constant* $c_{\mu i}$ *and* n_i. Even in the Car–Parrinello scheme, in which the system moves out of the BOS, making the Hellman–Feynmann theorem invalid, the atomic forces are nevertheless *defined* as derivatives at constant $c_{\mu i}$ and n_i. Thus, it may seem that the terms $\partial\rho_{\mu\nu}/\partial\boldsymbol{R}_I$ are irrelevant for the calculation of the forces. However, in the previous discussion we have omitted to say that the KS energy must be minimized under the constraint of orthonormality of the occupied states and that, therefore, at the BOS the energy is stationary *only* with respect to changes of ψ_i which do not violate the orthonormality. With an atomic basis set, the displacement of atoms (and the deformation of the unit cell) modifies the basis, and therefore the occupied states $\psi_i = \sum_\mu \phi_\mu c_{\mu i}$, even at constant $c_{\mu i}$. Moreover, the change of the states affects their orthonormality. Thus, in order to calculate the new total energy, we need to re-orthonormalize the occupied states, by changing their coefficients $c_{\mu i}$. Schematically, we must solve

$$\langle\psi_i|\delta S|\psi_j\rangle + \langle\delta\psi_i|S|\psi_j\rangle + \langle\psi_i|S|\delta\psi_j\rangle = 0 \tag{A.1}$$

where δS represents the change of $S_{\mu\nu}$ due to the atomic displacements, and $\delta\psi_i$ the modification of ψ_i due to the change of $c_{\mu i}$. Without lack of generality, we can expand $\delta\psi_i$ in the basis of the eigenvectors ψ_i as $\delta\psi_i = \sum_j \psi_j\lambda_{ji}$. Substituting this expansion into (A.1) and using that $\langle\psi_i|S|\psi_j\rangle = \delta_{ij}$ we obtain $\lambda_{ji} = -\frac{1}{2}\langle\psi_j|\delta S|\psi_i\rangle$. Thus

$$|\delta\psi_i\rangle = -\tfrac{1}{2}\sum_j |\psi_j\rangle\langle\psi_j|\delta S|\psi_i\rangle. \tag{A.2}$$

In terms of the coefficients $c_{\mu i}$, we have $\langle\psi_j|\delta S|\psi_i\rangle = \sum_{\mu\nu} c_{j\mu}\delta S_{\mu\nu}c_{\nu i}$ and

$$\delta c_{\mu i} = -\tfrac{1}{2}\sum_j \sum_{\eta\nu} c_{\mu j}c_{j\eta}\delta S_{\eta\nu}c_{\nu i} = -\tfrac{1}{2}\sum_{\eta\nu} S^{-1}_{\mu\eta}\delta S_{\eta\nu}c_{\nu i} \tag{A.3}$$

where we have used that $c_{\mu i} = \langle\tilde{\phi}_\mu|\psi_i\rangle$, and

$$\sum_i c_{\mu i}c_{i\nu} = \langle\tilde{\phi}_\mu|\tilde{\phi}_\nu\rangle = S^{-1}_{\mu\nu}. \tag{A.4}$$

Differentiating now equation (34) and using (A.3) we obtain

$$\delta\rho_{\mu\nu} = -\tfrac{1}{2}\sum_{\eta\zeta}(\rho_{\mu\eta}\delta S_{\eta\zeta}S^{-1}_{\zeta\nu} + S^{-1}_{\mu\eta}\delta S_{\eta\zeta}\rho_{\zeta\nu}) \tag{A.5}$$

and

$$\sum_{\mu\nu} H_{\mu\nu}\delta\rho_{\nu\mu} = -\sum_{\mu\nu} E_{\mu\nu}\delta S_{\nu\mu} \tag{A.6}$$

where $E_{\mu\nu}$ is the so-called energy–density matrix:

$$E_{\mu\nu} = \tfrac{1}{2}\sum_{\eta\zeta}(S^{-1}_{\mu\eta}H_{\eta\zeta}\rho_{\zeta\nu} + \rho_{\mu\eta}H_{\eta\zeta}S^{-1}_{\zeta\nu}) = \sum_i c_{\mu i}n_i\epsilon_i c_{i\nu} \tag{A.7}$$

where ϵ_i are the eigenstate energies. To calculate the orthogonalization force or stress, $\delta S_{\mu\nu}$ must be substituted by the appropriate derivative:

$$\boldsymbol{F}^{orthog}_I = 2\sum_\mu \sum_{\nu\in I} E_{\nu\mu}\frac{\partial S_{\mu\nu}}{\partial \boldsymbol{R}_{\mu\nu}}. \tag{A.8}$$

This equation has been derived before in different ways, and Ordejón *et al* [74] found it also for the $\mathcal{O}(N)$ functional, even though it does not require the occupied states to be orthogonal. In this case, equation (A.7) must be substituted by a more complicated expression [74].

Similarly, the stress contribution is

$$\sigma^{orthog}_{\alpha\beta} = -\sum_{\mu\nu} E_{\nu\mu}\frac{\partial S_{\mu\nu}}{\partial R^\alpha_{\mu\nu}}R^\beta_{\mu\nu}. \tag{A.9}$$

Appendix B. Radial fast Fourier transform

We consider here how to perform fast integrals of the form

$$\psi_l(k) = \int_0^\infty r^2\,\mathrm{d}r\; j_l(kr)\psi_l(r) \tag{B.1}$$

where $j_l(kr)$ is a spherical Bessel function and $\psi_l(r)$ is a radial function which behaves as $\psi_l(r) \sim r^l$ for $r \to 0$. Although methods to perform fast Bessel and Hänkel transforms have been described previously in different fields [75–77] we have developed a simple method adapted to our needs. It is based on the fact that $j_l(x)$ has the general form $(P^s_l(x)\sin(x) + P^c_l(x)\cos(x))/x^{l+1}$, where $P^s_l(x)$, $P^c_l(x)$ are simple polynomials $P^{s,c}_l(x) = \sum_{n=0}^l c^{s,c}_{ln}x^n$. Thus, the method involves computing $l+1$ fast sine and cosine transforms [28] and adding the different terms:

$$\psi_l(k) = \sum_{n=0}^l \frac{c^s_{ln}}{2k^{l+1-n}}\int_{-\infty}^{+\infty}\frac{\psi_l(r)}{r^{l-1-n}}\sin(kr)\,\mathrm{d}r + \sum_{n=0}^l \frac{c^c_{ln}}{2k^{l+1-n}}\int_{-\infty}^{+\infty}\frac{\psi_l(r)}{r^{l-1-n}}\cos(kr)\,\mathrm{d}r. \tag{B.2}$$

Notice that we have extended the integral to the whole real axis, defining $\psi_l(-r) \equiv (-1)^l\psi_l(r)$, in accordance with the behaviour $\psi_l(r) \sim r^l$, $r \to 0$. The coefficients $c^{s,c}_{ln}$ can be obtained

by defining a complex polynomial $P_l(x) = P_l^c(x) + \mathrm{i}P_l^s(x)$, which obeys the recurrence relations [78]

$$\begin{aligned} P_0(x) &= \mathrm{i} \equiv \sqrt{-1} \\ P_1(x) &= \mathrm{i} - x \\ P_{l+1}(x) &= (2l+1)P_l(x) - x^2 P_{l-1}(x). \end{aligned} \tag{B.3}$$

In order to perform the integrals in (B.2) using discrete FFTs, we need to calculate $\psi(r)$ on a regular radial grid, up to a maximum radius r_{max}, beyond which $\psi(r)$ is assumed to be strictly zero. The separation Δr between grid points determines a cutoff $k_{max} = \pi/\Delta r$ in reciprocal space, and vice versa, $\Delta k = \pi/r_{max}$. For convolutions, such as those involved in equation (28), we need $r_{max} = r_1^c + r_2^c$ and $k_{max} = \max(k_1^c, k_2^c)$, where $r_{1,2}^c$, $k_{1,2}^c$ are the cutoff radii and maximum wavevectors of $\psi_{1,2}$, respectively. We must then pad with zeros the intervals $[r_{1,2}^c, r_{max}]$ for the forward transforms $\psi_{1,2}(r) \to \psi_{1,2}(k)$. In practice, we set $r_{max} = 2\max_\mu(r_\mu^c)$, $k_{max} = \max_\mu(k_\mu^c)$, where μ labels all the basis orbitals and KB projectors, and we use the same real and reciprocal grids for all orbital pairs. In this way, we need to perform the forward transform only once for each radial function $\psi_\mu(r)$. Finally, notice that in equation (28) $\psi^*_{1,l_1m_1}(k)\psi_{2,l_2m_2}(k) \sim k^{l_1+l_2}$ for $k \to 0$, while l_1+l_2-l is even and non-negative, so the integrands of equation (B.2) for the backward transform are all even and well behaved at the origin.

Appendix C. Extended-mesh algorithm

We describe here a simple and efficient algorithm to handle mesh indices in three-dimensional periodic systems. Its versatility makes it suitable for several different tasks in SIESTA such as neighbour-list constructions, basis orbital evaluation in the real-space integration grid, density-gradient calculations in the GGA etc. It would be also very apropriate for other problems, such as the solution of partial differential equations by real-space discretization or the calculation of the interaction energy in lattice models. For clarity of the exposition, we shall describe the algorithm for a particularly simple application, namely the evaluation of the Laplacian of a function $f(\boldsymbol{r})$ using finite differences, even though the algorithm is not used in SIESTA for this purpose. In three dimensions, one generally discretizes space in all three periodic directions, using an index for each direction. For simplicity, let us consider an orthorhombic unit cell, with mesh steps Δx, Δy, Δz. Then the simplest formula for the Laplacian is

$$\begin{aligned} \nabla^2 f_{i_x,i_y,i_z} = {} & (f_{i_x+1,i_y,i_z} - 2f_{i_x,i_y,i_z} + f_{i_x-1,i_y,i_z})/\Delta x^2 + (f_{i_x,i_y+1,i_z} - 2f_{i_x,i_y,i_z} + f_{i_x,i_y-1,i_z})/\Delta y^2 \\ & + (f_{i_x,i_y,i_z+1} - 2f_{i_x,i_y,i_z} + f_{i_x,i_y,i_z-1})/\Delta z^2. \end{aligned}$$

A direct translation of this expression into Fortran90 code might read

```
Lf(ix,iy,iz) =                              &
     ( f(modulo(ix+1,nx),iy,iz) +           &
       f(modulo(ix-1,nx),iy,iz) )/dx2       &
   + ( f(ix,modulo(iy+1,ny),iz) +           &
       f(ix,modulo(iy-1,ny),iz) )/dy2       &
   + ( f(ix,iy,modulo(iz+1,nz)) +           &
       f(ix,iy,modulo(iz-1,nz)) )/dz2       &
   - f(ix,iy,iz) * (2/dx2+2/dy2+2/dz2)
```

where the indices i_α ($\alpha = \{x, y, z\}$) of the arrays `f` and `Lf` run from 0 to $n_\alpha - 1$, as in C. There are two problems with this construction. First, the `modulo` operations are required to bring the indices back to the allowed range $[0, n_\alpha - 1]$, and second, the use of three indices to refer to a mesh point implies implicit arithmetic operations, generated by the compiler, to translate them into a single index giving its position in memory.

A straightforward solution to these inefficiencies would be to create a neighbour-point list `j_neighb(i,neighb)`, of the size of the number of mesh points multiplied by the number of neighbour points. However, although the latter is only six in our simple example, it may frequently be as high as several hundred, which generally makes this approach unfeasible. A partial solution, addressing only the first problem, is to create six (or more for longer ranges) one-dimensional tables $j_\alpha^{\pm 1}(i_\alpha) = \mathrm{mod}(i_\alpha \pm 1, n_\alpha)$ to avoid the modulo computations [79]. Here, we describe a multidimensional generalization of this method, which solves both problems at the expense of a very reasonable amount of extra storage.

The method is based on an *extended mesh*, which extends beyond the periodic unit cell, by as much as required to cover all the space that can be reached from the unit cell by the range of the interactions or the finite-difference operator. The extended mesh range is $i_\alpha^{min} = -\Delta n_\alpha$ and $i_\alpha^{max} = n_\alpha - 1 + \Delta n_\alpha$, where $\Delta n_\alpha = 1$ in our particular example, in which the Laplacian formula extends just to first-neighbour mesh points. In principle, in cases with a small unit cell and a long range, the mesh extension may be larger than the unit cell itself, extending over several neighbour cells. However, in the more relevant case of a large system, we shall expect the extension region to be small compared with the unit cell. We then consider two combined indices, one associated with the normal unit-cell mesh, and another one associated with the extended mesh

$$i = i_x + n_x i_y + n_x n_y i_z, \qquad i_{ext} = (i_x - i_x^{min}) + n_x^{ext}(i_y - i_y^{min}) + n_x^{ext} n_y^{ext}(i_z - i_z^{min}),$$

where $n_\alpha^{ext} = i_\alpha^{max} - i_\alpha^{min} + 1 = n_\alpha + 2\Delta n_\alpha$. The key observation is that, if i_{ext} is a mesh point *within* the unit cell ($0 \leqslant i_\alpha \leqslant n_\alpha - 1$), and if j_{ext} is a neighbour mesh point (within its interaction range, i.e. $|j_\alpha - i_\alpha| \leqslant \Delta n_\alpha$), then the arithmetic difference $j_{ext} - i_{ext}$ depends only on the relative positions of i_{ext} and j_{ext} (i.e. on $j_\alpha - i_\alpha$), and not on the position of i_{ext} within the unit cell. We can then create a list of neighbour strides $\Delta i j_{ext}$, and two arrays to translate back and forth between i and i_{ext}. One of the arrays maps the unit-cell points to the central region of the extended mesh, while the other one folds back the extended mesh points to their periodically equivalent points within the unit cell. Then, to access the neighbours of a point i, we (a) translate $i \to i_{ext}$, (b) find $j_{ext} = i_{ext} + \Delta i j_{ext}$ and (c) translate $j_{ext} \to j$. Notice that several points of the extended mesh will map to the same point within the unit cell and that, in principle, a unit-cell point j may be a neighbour of i through different values of j_{ext}. In our example, the innermost loop would then read

```
Lf(i) = 0
do neighb = 1,n_neighb
   j_ext = i_extended(i) + ij_delta(neighb)
   j = i_cell(j_ext)
   Lf(i) = Lf(i) + L(neighb) * f(j)
end do
```

where the number of neighbour points would be `n_neighb=7`, including the central point itself, and

```
ij_delta(1) =  1             ; L(1) = 1/dx2
ij_delta(2) = -1             ; L(1) = 1/dx2
ij_delta(3) =  nx_ext        ; L(3) = 1/dy2
ij_delta(4) = -nx_ext        ; L(4) = 1/dy2
ij_delta(5) =  nx_ext*ny_ext ; L(5) = 1/dz2
ij_delta(6) = -nx_ext*ny_ext ; L(6) = 1/dz2
ij_delta(7) =  0  ; L(7)=-2/dx2-2/dy2-2/dz2
```

Notice that the above loop is completely general, for any linear operator, using an arbitrary number of neighbour points for its finite-difference representation. In fact, it is even independent of the space dimensionality. Furthermore, the index operations required are just one addition and three memory calls to arrays of range one[13]. This inner-loop simplicity comes at the expense of the two extra arrays `i_extended` and `i_cell` (of the size of the normal and extended meshes, respectively) which are generally an acceptable memory overhead. Notice, however, that the neighbour-point list `ij_delta` is independent of the mesh index i, which makes this array quite small in most problems of interest.

Appendix D. Sparse matrix techniques

We shall describe here some of the sparse-matrix multiplication techniques used in evaluating equations (35) and (86)–(88). There are a large variety of sparse-matrix representations and algorithms, each one optimized for a different kind of sparsity. The main constraint for choosing our representation and algorithms is that they must be $\mathcal{O}(N)$ in both memory and CPU time. We enforce this condition strictly by requiring, for example, that a vector of size $\sim N$ will not be reset to zero a number $\sim N$ of times. In our sparse matrices, such as $S_{\mu\nu}$, $H_{\mu\nu}$, $c_{\mu i}$, $\rho_{\mu\nu}$ and $\phi_\mu(\boldsymbol{r})$, the number p of nonzero elements in a row is typically much larger than unity (but still of order $\sim N^0$) and much smaller than the row size $m \sim N^1$. Such matrix rows are efficiently stored as a real vector of size p, containing the nonzero elements, and an integer vector of the same size containing the column index of each nonzero element. The whole matrix A of n rows is then represented by two arrays `A` and `jcol`, of size $n \times p$, such that[14] $A_{ij} =$ `A(i,k)`, where $j =$ `jcol(i,k)`. The problem with this representation is that, given a value j of the column index, there is no simple way to access the element A_{ij} without scanning the whole row, which is frequently too costly. One solution is to unpack a row i, that will be repeatedly used, into 'expanded form', i.e. to transfer it to a vector `Arow` of the full row size m (containing also all the zeros), so that $A_{ij} =$ `Arow(j)`. Since $p \gg 1$, the size of `Arow` is negligible compared with that of `A` and `jcol`.

To find the matrix product C of two sparse matrices A and B

$$C_{ik} = \sum_j A_{ij} B_{jk}$$

we proceed iteratively for each row i of A (which will generate the same row of C): each nonzero element j of the row is multiplied by every nonzero element of the jth row of B (whose column index is, say k) and the result is accumulated in the kth position of an auxiliary 'expanded' vector. After finishing with that row of A we pack the vector in sparse format into

[13] In the present example, further savings can be achieved by extending the arrays $f(i)$ and $Lf(i)$ themselves, eliminating the index translations and facilitating the paralelization. Another efficient possibility is provided in Fortran90 by the intrinsic cshift operation. However, these approaches are more complicated in other cases, such as atomic-neighbour list constructions, while the double-index algorithm is quite general.

[14] In Fortran, we alternate the order of i and k, to store the row elements consecutively in memory. If the number of nonzero elements fluctuates widely for different rows, we also store all the rows consecutively as a large single vector.

the ith row of C and restore the auxiliary vector to zero. In fact, the packing can be performed simultaneously with the product, using an auxiliary index vector instead:

```
C = 0.
ncolC = 0
jcolC = 0
pos = 0  ! Auxiliary index vector
do i = 1, nA
  do jA = 1,ncolA(i)
    j = jcolA(i,jA)
    do jB = 1,ncolB(j)
      k = jcolB(j,jB)
      jC = pos(k)
      if (jC==0) then ! New nonzero col
        ncolC(i) = ncolC(i) + 1
        jC = ncolC(i)
        jcolC(i,jC) = k
        pos(k) = jC
      endif
      C(i,jC) = C(i,jC) + A(i,jA)*B(j,jB)
    enddo
  enddo
  do jA = 1,ncolA(i) ! Restore pos to zero
    j = jcolA(i,jA)
    do jB = 1,ncolB(j)
      k = jcolB(j,jB)
      pos(k) = 0
    enddo
  enddo
enddo
```

Notice that the auxiliary vector `pos`, which keeps the position in 'packed format' of the nonzero elements of one row of C, is initiallized in full only once. Notice also that this algorithm, unlike those of [28], does not require the matrix elements to be stored in ascending column order.

The previous algorithm generates all the nonzero elements of C but in many cases we need only some of them. For example, to calculate the electron density (equation (35)), we need only the density matrix elements $\rho_{\mu\nu}$ for which ϕ_μ and ϕ_ν overlap. Also the expression (88) needs to be evaluated only for the coefficients $c_{\mu i}$ which are allowed to be nonzero by the localization constraints. In these cases, in which the array `jcolC` is already known, another algorithm is more effective. We start by finding the sparse representation of B in *column* order or, in other words, the transpose of B:

```
Bt = 0 ! B transpose
jcolBt = 0
ncolBt = 0
do i = 1,nB
  do jB = 1,ncolB(i)
    j = jcolB(i,jB)
    ncolBt(j) = ncolBt(j) + 1
    jBt = ncolBt(j)
    jcolBt(j,jBt) = i
```

```
    Bt(j,jBt) = B(i,jB)
  enddo
enddo
```

We then unpack a row i of A and multiply it by a column j of B (a row of its transpose) for each required matrix element C_{ij} of their product:

```
C = 0.
Arow = 0.  ! Auxiliary vector
do i = 1,nC
  do jA = 1,ncolA(i) ! Copy one row of A
    j = jcolA(i,jA)
    Arow(j) = A(i,jA)
  enddo
  do jC = 1,ncolC(i) ! Calculate Cij
    j = jcolC(i,jC)
    do jBt = 1,ncolBt(j)
      k = jcolBt(j,jBt)
      C(i,jC) = C(i,jC) + Arow(k)*Bt(j,jBt)
    enddo
  enddo
  do jA = 1,ncolA(i) ! Restore Arow to zero
    j = jcolA(i,jA)
    Arow(j) = 0.
  enddo
enddo
```

The combination of these two matrix multiplication algorithms allows an efficient evaluation of equations (86)–(88). Since these equations involve a trace or a relatively small subset of a final matrix, it is important to control the order and sparsity of the intermediate products, in order to keep them as sparse as possible. Notice that, once a row of $A \times B$ has been evaluated, it may be multiplied by a third matrix, to obtain a row of the final product, without any need to store the whole intermediate matrix.

To calculate the density at a grid point using equation (35) we need to access the matrix elements $\rho_{\mu\nu}$, and this is inefficient if they are stored in sparse format. Thus, we first copy the matrix elements, between the n_r basis orbitals which are nonzero at the grid point r, into an auxiliary matrix array, of size $n_{aux} \times n_{aux}$, with $n_{aux} \geqslant n_r$. We also create a lookup table `pos`, of size equal to the total number of basis orbitals, such that `pos(mu)` is the position, in the auxiliary matrix, of the matrix elements of orbital `mu` (or zero if they have not been copied to it). If there are new nonzero orbitals at the next grid points, we keep copying them into the auxiliary matrix, until all its n_{aux} slots are full, at which point we erase it and restart the process. Since succesive grid points tend to contain the same nonzero basis orbitals, these copies and erasures are not frequent.

References

[1] Greengard L 1994 *Science* **265** 909
[2] Hockney R W and Eastwood J W 1988 *Computer Simulation Using Particles* (Bristol: Institute of Physics Publishing)
[3] Car R and Parrinello M 1985 *Phys. Rev. Lett.* **55** 2471
[4] Goringe C M, Bowler D R and Hernandez E 1997 *Rep. Prog. Phys.* **60** 1447
[5] Ordejon P 1998 *Comput. Mater. Sci.* **12** 157

[6] Ordejon P, Drabold D A, Grumbach M P and Martin R M 1993 *Phys. Rev.* B **48** 14 646
[7] Sankey O F and Niklewski D J 1989 *Phys. Rev.* B **40** 3979
[8] Payne M C, Teter M P, Allan D C, Arias T A and Joannopoulos J D 1992 *Rev. Mod. Phys.* **64** 1045
[9] Goedecker S 1999 *Rev. Mod. Phys.* **71** 1085
[10] Hernandez E and Gillan M J 1995 *Phys. Rev.* B **51** 10 157
[11] Ordejon P, Artacho E and Soler J M 1996 *Phys. Rev.* B **53** R10 441
[12] Sanchez-Portal D, Ordejon P, Artacho E and Soler J M 1997 *Int. J. Quantum Chem.* **65** 453
[13] Ordejon P 2000 *Phys. Status Solidi* b **217** 335 see also http://www.uam.es/siesta
[14] Kohn W and Sham L J 1965 *Phys. Rev.* **140** A1133
[15] Perdew J P and Zunger A 1981 *Phys. Rev.* B **23** 5048
[16] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[17] Hamann D R, Schlüter M and Chiang C 1979 *Phys. Rev. Lett.* **43** 1494
[18] Bachelet G B, Hamann D R and Schlüter M 1982 *Phys. Rev.* B **26** 4199
[19] Kleinman L and Bylander D M 1982 *Phys. Rev. Lett.* **48** 1425
[20] Louie S G, Froyen S and Cohen M L 1982 *Phys. Rev.* B **26** 1738
[21] Kleinman L 1980 *Phys. Rev.* B **21** 2630
[22] Bachelet G B and Schlüter M 1982 *Phys. Rev.* B **25** 2103
[23] Troullier N and Martins J L 1991 *Phys. Rev.* B **43** 1993
[24] Blöchl P E 1990 *Phys. Rev.* B **41** 5414
[25] Ramer N J and Rappe A M 1999 *Phys. Rev.* B **59** 12 471
[26] Vanderbilt D 1985 *Phys. Rev.* B **32** 8412
[27] Vanderbilt D 1990 *Phys. Rev.* B **41** 7892
[28] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 1992 *Numerical Recipes* (Cambridge: Cambridge University Press)
[29] Sanchez-Portal D, Soler J M and Artacho E 1996 *J. Phys.: Condens. Matter* **8** 3859
[30] Lippert G, Hutter J, Ballone P and Parrinello M 1996 *J. Phys. Chem.* **100** 6231
[31] Artacho E, Sanchez-Portal D, Ordejon P, Garcia A and Soler J M 1999 *Phys. Status Solidi* b **215** 809
[32] Huzinaga S *et al* 1984 *Gaussian Basis Sets for Molecular Calculations* (Berlin: Elsevier)
[33] Junquera J, Paz O, Sanchez-Portal D and Artacho E 2001 *Phys. Rev.* B **64** 235111
[34] Filippi C, Singh D J and Umrigar C J 1994 *Phys. Rev.* B **50** 14 947
[35] Kittel C 1986 *Introduction to Solid State Physics* (New York: Wiley)
[36] Jackson J D 1962 *Classical Electrodynamics* (New York: Wiley)
[37] Balb'as L C, Martins J L and Soler J M 2001 *Phys. Rev.* B **64** 165110
[38] Makov G and Payne M C 1995 *Phys. Rev.* B **51** 4014
[39] Sandratskii L M and Guletskii P G 1986 *J. Phys. F: Met. Phys.* **16** L43
[40] Kübler J, Höck K H, Sticht J and Williams A R 1988 *J. Appl. Phys.* **63** 3482
[41] Oda T, Pasquarello A and Car R 1998 *Phys. Rev. Lett.* **80** 3622
[42] Postnikov A V, Engel P and Soler J M 2001 *Preprint* cond-mat/0109540
[43] Moreno J and Soler J M 1992 *Phys. Rev.* B **45** 13 891
[44] Monkhorst H J and Pack J D 1976 *Phys. Rev.* B **13** 5188
[45] Mermin N D 1965 *Phys. Rev.* B **137** A1441
[46] Harris J 1985 *Phys. Rev.* B **31** 1770
[47] Foulkes W M C and Haydock R 1989 *Phys. Rev.* B **39** 12 520
[48] King-Smith R D and Vanderbilt D 1993 *Phys. Rev.* B **47** 1651
[49] Resta R 1994 *Rev. Mod. Phys.* **66** 899
[50] Saghi-Szabo G, Cohen R E and Krakauer H 1998 *Phys. Rev. Lett.* **80** 4321
[51] Vanderbilt D 2000 *J. Phys. Chem. Solids* **61** 147
[52] Sánchez-Portal D, Souza I and Martin R M 2000 *Fundamental Physics of Ferroelectrics (AIP Conf. Proc. Vol. 535)* ed R Cohen (Melville: AIP) pp 111–20
[53] Dall'Olio S and Dovesi R 1997 *Phys. Rev.* B **56** 10 105
[54] Yaschenko E, Fu L, Resca L and Resta R 1998 *Phys. Rev.* B **58** 1222
[55] Mauri F, Galli G and Car R 1993 *Phys. Rev.* B **47** 9973
[56] Kim J, Mauri F and Galli G 1995 *Phys. Rev.* B **52** 1640
[57] Stephan U, Drabold D A and Martin R M 1998 *Phys. Rev.* B **58** 13 472
[58] Kohn W 1959 *Phys. Rev.* **115** 809
[59] Anderson E *et al* 1999 *LAPACK Users' Guide* (Philadelphia, PA: SIAM)
[60] Pulay P 1980 *Chem. Phys. Lett.* **73** 393
[61] Pulay P 1982 *J. Comput. Chem.* **13** 556

[62] Allen M P and Tildesley D J 1987 *Computer Simulation of Liquids* (Oxford: Oxford University Press)
[63] Garcia A and Soler J M 2001 unpublished
[64] Parlinski K 2001 unpublished
[65] Pruneda J M, Estreicher S, Junquera J, Ferrer J and Ordejon P 2002 *Phys. Rev.* B **65** -75210
[66] Brandbyge M, Stokbro K, Taylor J, Mozos J-L and Ordejon P 2001 *Mater. Res. Soc. Symp. Proc.* **636** D9.25.1
[67] Economou E N 1983 *Green's Functions in Quantum Physics* (Berlin: Springer)
[68] Read A J and Needs R J 1991 *Phys. Rev.* B **44** 13 071
[69] Gross E K U, Ullrich C A and Gossmann U J 1995 *Density Functional Theory* ed E K U Gross and R M Dreizler (New York: Plenum) pp 149–71
[70] Gross E K U, Dobson J F and Petersilka M 1996 *Density Functional Theory II: Relativistic and Time Dependent Extensions (Topics in Current Chemistry Vol. 181)* ed R F Nalewajski (Berlin: Springer) pp 81–172
[71] Yabana K and Bertsch G F 1996 *Phys. Rev.* B **54** 4484
[72] Tsolakidis A, Sanchez-Portal D and Martin R M 2001 *Preprint* cond-mat/0109488
[73] Gale J D 2002 unpublished
[74] Ordejon P, Drabold D A, Martin R M and Grumbach M P 1995 *Phys. Rev.* B **51** 1456
[75] Suter B W 1991 *IEEE Trans. Signal Process.* **39** 532
[76] Mohsen A A and Hashish E A 1994 *Geophys. Prospect.* **42** 131
[77] Ferrari T, Perciante D and Dubra A 1999 *J. Opt. Soc. Am.* A **16** 2581
[78] Abramowitz M and Stegun I A 1964 *Handbook of Mathemathical Functions* (New York: Dover)
[79] Binder K and Heermann D W 1992 *Monte Carlo Simulation in Statistical Physics* (Berlin: Springer)
[80] Gonze X, Stumpf R and Scheffler M 1991 *Phys. Rev.* B **44** 8503