8 SCIENTIFIC HIGHLIGHT OF THE MONTH

Linear scaling *ab initio* calculations: Recent Progress with the Conquest code

David Bowler[1], Tsuyoshi Miyazaki[2] and Mike Gillan[1]

[1] Department of Physics and Astronomy, University College London
Gower Street, London, WC1E 6BT, U.K.
[2] National Institute for Materials Science,
1-2-1 Sengen, Tsukuba, Ibaraki 305-0047, Japan

Abstract

We describe recent progress in the practical implementation of linear scaling, *ab initio* calculations, referring in particular to our highly parallel code CONQUEST. After reviewing the state of the field, we present the basic ideas underlying almost all linear scaling methods, and discuss specific practical details of the implementation. We also note the connection between linear scaling methods and embedding techniques.

1. Introduction

The last ten years have seen an upsurge of interest in $\mathcal{O}(N)$ electronic-structure methods[1] for treating condensed matter both within tight-binding theory and within density functional theory [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. In these methods, the number of computer operations needed to determine the electronic ground state is proportional to the number of atoms N in the system, instead of showing the N^2 or N^3 dependence characteristic of traditional methods. $\mathcal{O}(N)$ methods are possible because electronic phase coherence is localised [14, 19, 26, 27, 28, 29]. This localisation property can be expressed by saying that the density matrix between two points, $\rho(\mathbf{r}, \mathbf{r}')$, decays to zero with increasing distance between points $\mathbf{r}$ and $\mathbf{r}'$.

Locality is the unifying theme between almost all $\mathcal{O}(N)$ methods, and it requires a local formulation of quantum mechanics (or, at least, electronic structure theory). The density matrix $\rho(\mathbf{r}, \mathbf{r}')$ mentioned above is a key quantity in this local formulation, and in terms of the Kohn-Sham orbitals, can be written as:

$$\rho(\mathbf{r}, \mathbf{r}') = \sum_i f_i \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}'), \tag{1}$$

where f_i are the occupation numbers. The Kohn-Sham energy can be easily rewritten in terms of the density matrix: the Hartree and exchange-correlation energies, which are written in terms of the charge density ($n(\mathbf{r}) = \rho(\mathbf{r}, \mathbf{r})$) do not change; the kinetic and pseudopotential energies become:

$$E_{\mathrm{KE}} = -\frac{\hbar^2}{2m} \int \mathrm{d}\mathbf{r} (\nabla_r^2 \rho(\mathbf{r}, \mathbf{r}'))_{\mathbf{r}=\mathbf{r}'} \tag{2}$$

$$E_{\mathrm{ps}} = 2 \int \mathrm{d}\mathbf{r}\ \mathrm{d}\mathbf{r}' V_{\mathrm{ps}}(\mathbf{r}, \mathbf{r}') \rho(\mathbf{r}, \mathbf{r}') \tag{3}$$

It can be shown that $\rho(\mathbf{r}, \mathbf{r}')$ decreases as the separation between $\mathbf{r}$ and $\mathbf{r}'$ increase (either exponentially, for insulators, or algebraically, for metals[27, 28, 29, 30, 31]): $\rho(\mathbf{r}, \mathbf{r}') \to 0$ as $\mid \mathbf{r} - \mathbf{r}' \mid \to \infty$. This locality implies that the amount of information scales linearly with the size of the system; an $\mathcal{O}(N)$ method can be created by making an approximation, and *enforcing* locality: $\rho(\mathbf{r}, \mathbf{r}') = 0, \mid \mathbf{r} - \mathbf{r}' \mid > R_c$. While the scaling of both memory and computational effort will allow large systems to be simulated on a workstation, for very large systems containing thousands or tens of thousands of atoms, the codes need to run efficiently on parallel computers; this aspect is discussed later in the article.

However, the six-dimensional quantity $\rho(\mathbf{r}, \mathbf{r}')$ is not the ideal variable to work with, so the assumption is made (with only the restriction that the original quantity had a finite number of non-zero eigenvalues) that it can be written in *separable* form:

$$\rho(\mathbf{r}, \mathbf{r}') = \sum_{i\alpha, j\beta} \phi_{i\alpha}(\mathbf{r}) K_{i\alpha j\beta} \phi_{j\beta}(\mathbf{r}'), \tag{4}$$

where $\phi_{i\alpha}(\mathbf{r})$ is a *support function* (or a *localised orbital*) centred on atom i, and $K_{i\alpha j\beta}$ is the density matrix in the basis of the support functions. Then locality can be enforced by applying separate cutoffs to both $K_{i\alpha j\beta}$ and the set of *support functions* $\{\phi_{i\alpha}(\mathbf{r})\}$. The minimisation of the energy with respect to each of these quantities then drives the system towards the ground state.

The support functions are *non-orthogonal*, and forced to be confined within a *localisation region*, of radius R_c. They are themselves represented in terms of other basis functions (which will be described in detail in Section 3), and it is important that they be freely varied in the search for the electronic ground state. By increasing the cutoffs and improving the completeness of the basis set representing the support functions, an $\mathcal{O}(N)$ method can be made to reproduce traditional methods with plane wave accuracy.

The minimisation itself involves three different variables: the elements of the density matrix, K; the support functions, $\phi_{i\alpha}$; and the charge density, $n(\mathbf{r})$, which must be consistent with the potential $V(\mathbf{r})$. We have chosen to decouple these variables, fixing first the support functions and the charge density, and minimising with respect to the density matrix, then achieving self-consistency (while minimising with respect to the density matrix every time that the potential changes) and finally varying the support functions. This scheme has various advantages: first, it decouples the degrees of freedom associated with the support functions and the density matrix; second, it allows us to treat regions of the system with fixed support functions (in an *ab initio* tight binding manner) while freely varying others; third, on a practical level, it allows us to optimise the procedures for the different minimisations independently.

The rest of the article is arranged as follows: in the next section, we describe minimisation of the energy with respect to the density matrix, K; then we consider possible basis sets for the support functions, and describe our choice; we then describe some practical details relating to the implementation of our scheme and conclude the article.

2. Finding the density matrix

Within the atom-centred basis of support functions, the density matrix $K_{i\alpha j\beta}$ (also called the *kernel*) is clearly equivalent to the density matrix in a non-orthogonal tight binding formulation. There has been a great deal of work investigating effective $\mathcal{O}(N)$ methods for finding the density matrix in tight binding[2, 4, 5, 6, 7, 8, 9, 10, 17, 18, 21, 22, 24, 1] which gives us a strong position to start from. However, there is an important issue, which will be addressed fully below: in the formulation described up to this point, the basis in which the density matrix, $K_{i\alpha j\beta}$, is written is *non-orthogonal*, while most tight binding methods have been primarily formulated with an orthogonal basis set. We shall first describe the methods, and then address this important question of non-orthogonality.

Tight binding techniques which have been extended to *ab initio* techniques fall broadly into four categories: recursion[10]; density matrix minimisation[5, 6, 12, 24]; orbital minimisation[7, 9]; and penalty functions[23]. These have been described in some detail by Goedecker[1], so we shall only outline the methods.

The Fermi Operator Expansion technique[10, 32, 33, 34] is a conceptually and computationally simple way of obtaining the density matrix, at the expense of introducing a finite electronic temperature and losing a variational principle. The Fermi matrix (a finite temperature density matrix) can be defined as:

$$\mathbf{F}_{\mu,\mathrm{T}} = f\left(\frac{\mathbf{H}-\mu}{\mathrm{kT}}\right), \tag{5}$$

where $f(x) = 1/(1 + exp(x))$, the Fermi function.

Now, the Fermi function only has to cover a finite width, that is the width of the density of states for the system in question (or the difference between the minimum and maximum eigenvalues of the Hamiltonian). Within this range, it can be represented by a polynomial in the energy,

$$f(x) = \sum_{p=0}^{n_{pl}} C_p E^p, \tag{6}$$

which means that the Fermi matrix $F_{\mu,T}$ can be represented as a polynomial in the Hamiltonian,

$$\mathbf{F}_{\mu,T} = \sum_{p=0}^{n_{pl}} C_p \hat{H}^p. \tag{7}$$

This then gives the expression for one element of the Fermi matrix as:

$$\langle i\alpha \mid F_{\mu,T} \mid j\beta \rangle = \sum_{p=0}^{p=n_{pl}} C_p \langle i\alpha \mid \hat{H}^p \mid j\beta \rangle, \tag{8}$$

Conceptually, then, the method works by fitting a polynomial to the Fermi function over the range of the eigenvalues. Then, using the coefficients of this polynomial and moments of the Hamiltonian, elements of the finite temperature density matrix, or the Fermi matrix, are constructed. To make the method $\mathcal{O}(N)$, the Fermi matrix can be truncated beyond a certain cutoff radius. In practice, for stability, a Chebyshev polynomial is used[32], which leads to a recursion relation for the coefficients:

$$p_{\mu,T}(H) = \frac{c_0}{2} + \sum_{j=1}^{n_{pl}} c_j T_j(H), \text{and} \tag{9}$$

$$\begin{aligned} T_0(H) &= I \\ T_1(H) &= H \\ T_{j+1} &= 2HT_j(H) - T_{j-1}(H) \end{aligned} \tag{10}$$

Once the Fermi matrix has been truncated, the forces are not exactly equal to the derivative of energy; there is, however, a formalism which gives a force which is the exact derivative of the energy[33]. In some cases the error in energy due to the high electronic temperature may be significant; a scheme is available[35] for extrapolating the T=0 energy from a high temperature which can correct this.

Density matrix minimisation (DMM)[5, 12, 37, 36, 24] seeks to find the density matrix by minimising the energy with respect to the density matrix elements (since $E_{\text{band}} = 2\text{Tr}[\rho H]$). However, two constraints must be applied to the minimisation: (i) either constant electron number or constant fermi energy; (ii) idempotency of ρ. The first is relatively easy to address; maintaining constant fermi energy is as simple as minimising $2\text{Tr}[(\rho - \mu I)H]$, with μ the fermi energy, while various schemes exist for maintaining electron number constant[38, 39].

Idempotency of ρ (which is equivalent to requiring that the eigenvalues of ρ all be either zero or one, or that $\rho^2 = \rho$) is a much harder constraint to impose during a variational minimisation, and instead use is made of McWeeny's purification transformation[40]:

$$\rho = 3\tilde{\rho}^2 - 2\tilde{\rho}^3. \tag{11}$$

Provided that the eigenvalues of $\tilde{\rho}$ lie between -$\frac{1}{2}$ and $\frac{3}{2}$, the eigenvalues of ρ will lie between zero and one. McWeeny first proposed this as an iterative procedure (so that if $\rho_n = \tilde{\rho}$ then $\rho_{n+1} = \rho$), but another technique[5] is to minimise $E_{\rm band} = 2Tr[\rho H]$ with respect to the elements of $\tilde{\rho}$. If this approach is taken, then we must assume that $\tilde{\rho}$ is also separable (as ρ is in eq. 4): $\tilde{\rho} = \sum_{i\alpha j\beta} \phi_{i\alpha}(\mathbf{r}) L_{i\alpha j\beta} \phi_{j\beta}$; then we can write $K = 3L^2 - 2L^3$. Approaches to DMM are varied: pure McWeeny iteration can be used[41] (also with a modified cubic form which preserves electron number); pure minimisation can be used[5, 6, 38]; minimisation followed by[36] or interspersed with[37] McWeeny purification is also pursued; finally, McWeeny purification (which is not variational) followed by minimisation (which is variational)[24]. It can be shown[24] that the McWeeny purification approaches a manifold of idempotent density matrices perpendicularly in its final stages, while the minimisation yields a gradient tangential to this surface (and preserves idempotency to first order).

Another method for finding the ground state density matrix is orbital minimisation, which was proposed from two different routes, leading to essentially the same formalism[9, 7]. Consider a system of N electrons, described by $N/2$ non-interacting states $\{\mid \psi_i\rangle\}$. In order to avoid an *explicit* orthonormalisation step (which scales with N^2 in a localised basis set and N^3 in a plane wave basis), the following functional is defined:

$$E = 2\mathrm{Tr}[QH] - \eta\left[2\mathrm{Tr}[QS] - N\right] \tag{12}$$

$$Q = \sum_{n=0}^{\mathcal{N}} (I - S)^n, \tag{13}$$

where $S_{ij} = \langle \psi_i \mid \psi_j \rangle$ is the overlap matrix. The orbitals $\{\mid \psi_i\rangle\}$ are represented by a basis $\{\mid \phi_\mu\rangle\}$, so that:

$$\mid \psi_i\rangle = \sum_\mu C_{i\mu} \mid \phi_\mu\rangle. \tag{14}$$

Then as the energy is minimised with respect to the coefficients $C_{i\mu}$, the overlap matrix will tend to the identity. The method can be made $\mathcal{O}(N)$ by localising the orbitals ψ_i, and only allowing contributions from ϕ_μ within a specified volume.

The drawback with the method is that it is subject to many local minima if the minimal set of orbitals ($N/2$) is used; it has been extended to more orbitals, which to some extent corrects this problem[42].

Penalty functionals apply a similar idea, but instead seek to penalise the deviation away from idempotency[19]. The original technique defined the following functional:

$$Q[\rho;\mu,\alpha] = E_{NI}[\rho] - \mu N[\rho] + \alpha P[\rho] \tag{15}$$

$$P[\rho] = \left\{\rho^2(1-\rho^2)\right\}^{\frac{1}{2}}, \tag{16}$$

where E_{NI} is the non-interacting energy. However, it was found[43] that the branch point introduced by the square root prevented minimisation using techniques such as conjugate gradients. Instead, a general functional was introduced[23]:

$$Q[\rho] = E[\rho] + \alpha P^2[\rho], \tag{17}$$

which allows use of minimisation techniques.

All of the above methods have been described using orthogonal bases, but the formalism above relies on non-orthogonal bases. Each of the methods can be reformulated in a non-orthogonal basis, but this raises various issues:

1. Matrix expressions become more complex (e.g. we now have $K = 3LSL - 2LSLSL$, where S is the overlap matrix)
2. Compound matrices (such as LSL) are longer ranged
3. When using variational methods, the inverse of the overlap, S^{-1}, is required to correct the gradients[44]

Some choose to work within the non-orthogonal basis, and derive an approximate S^{-1}[34, 25], while others convert to an orthogonal basis, using a variety of methods (incomplete inverse Cholesky factorisation[36], Cholesky decomposition[45]). All of these approaches require an approximation (either in the S^{-1} or in the Cholesky factorisation), with the former having the advantage that no basis set changes are being used, and the latter that the formalism is much simpler and shorter ranged.

We finish by noting that the localisation inherent in $\mathcal{O}(N)$ schemes makes them ideal for use in embedding[46]. By this, we mean embedding of one system within another (e.g. a point defect into perfect bulk), not embedding of one technique within another. If the system is divided into region I (the region of interest – e.g. the point defect and its surroundings) and region II (the embedding matrix – e.g. the perfect bulk) then the amount of region II required is equivalent to the range of the density matrix, and we expect the energy convergence with size of region I to scale just as the energy convergence scales with density matrix range. This is illustrated in Figure 1, which shows the convergence of the energy for a Ge substitutional defect in diamond Si with radius of region I.

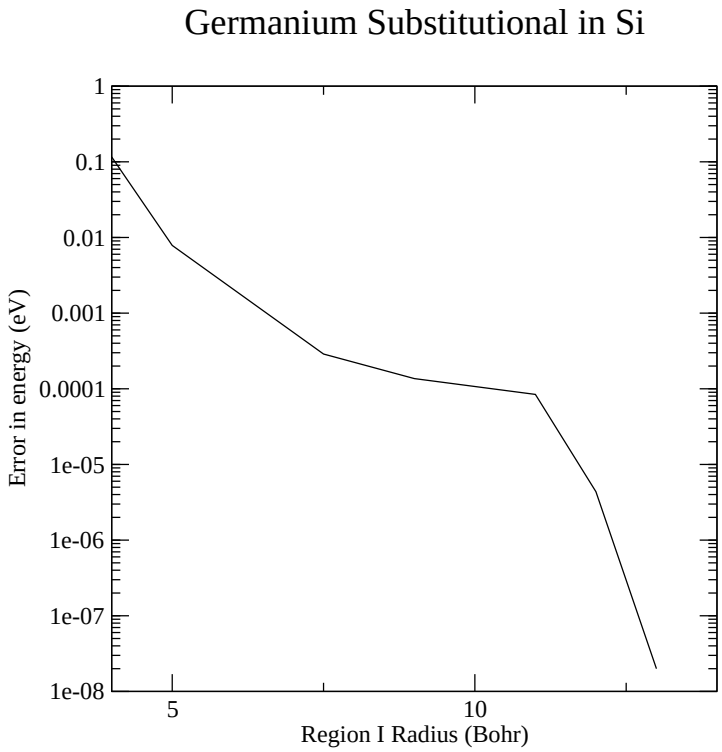

Figure 1. Convergence of energy with region I radius for a Ge substitutional impurity in Si expressed relative to result for infinite region I radius.

3. Support Functions

3.1 Representing Support Functions

As emphasised in Sec. 1, the localised orbitals $\phi_{i\alpha}(\mathbf{r})$ have to be freely varied in the search for the DFT ground state. This raises the technical problem common to all quantum calculations – the representation of the orbitals, i.e. basis sets. A peculiar feature of the basis sets needed in the scheme we have outlined is that the orbitals to be represented vanish outside the localisation region, and the basis functions clearly need to have the same property. Before discussing the approaches that have been taken to this, it is useful to consider briefly the conflicting requirements that basis functions have to satisfy.

First, they should ideally be well adapted to the function to be represented, which means that only a *few* basis functions should be needed to represent the orbitals. Second, the representation should be systematically improvable; this means not just that the basis set must be complete, but that the convergence should be rapid as the size of the basis set is increased. As a rider to this, the computational effort should not increase too fast as one increases the number of basis functions, and ideally no faster than linearly. Third, the operations to be done with the basis functions should be mathematically simple, so that the number of computer operations is small. Fourth, since some parts of a total-enegy calculation have to be done on spatial grid, which generally causes problems like breaking of translational symmetry, the basis must be designed so that as little as possible has to be done on the grid.

Several types of basis sets have been used or proposed for linear-scaling DFT calculations. When CONQUEST was first written, the support functions were represented as numerical values on a grid. A grid basis set gives a natural way of representing orbitals that vanish outside specified regions. Nevertheless, it is the ultimate in maladaption, and needs fine grids and large amounts of memory in order to achieve good precision. The kinetic energy is particularly troublesome. In more conventional DFT/pseudopotential schemes based on grid basis sets, high-order finite-difference methods are used to give an accurate representation of the Laplacian operator. Because of this, grid basis sets were replaced in CONQUEST by a scheme akin to finite elements.

This finite-element scheme represents the $\phi_{i\alpha}(\mathbf{r})$ in terms of piecewise continuous polynomials, using a technique sometimes referred to as B-splines. Full details of the scheme, with demonstrations of its effectiveness, are presented in a published report[47], so here we give only a brief summary. In one dimension, the B-spline basis consists of localised functions $\theta_s(x)$, centred on the points of a grid, s, with spacing a. The basis functions are all images of each other, displaced by an integer number of grid spacings, so that $\theta_s(x) = \theta_0(x - X_s)$. The basis function $\theta_0(x)$ vanishes identically outside the range $-2a < x < 2a$. Inside this range, it is put together out of

cubic polynomials:

$$\theta_0(x) = \begin{cases} 1 - \frac{3}{2}x^2 + \frac{3}{4}|x|^3 & \text{if} \quad 0 < |x| < a \\ \frac{1}{4}(2 - |x|)^3 & \text{if} \quad a < |x| < 2a \\ 0 & \text{if} \quad 2a < |x| \end{cases} \tag{18}$$

and has the property that it and its first two derivatives are continuous everywhere. In fact, the only discontinuities are in the third derivative at the points $|x| = 0$, a and $2a$. The representation of a continuous function

$$f(x) \simeq \sum_s b_s \theta_s(x) \tag{19}$$

can be made arbitrarily precise by systematically reducing the grid spacing a. This is exactly analogous to increasing the plane-wave cut-off $G_{\max}$ when taking a plane-wave calculations to convergence.

In practice, of course, we work in three dimensions, and the three-dimensional B-splines $\Theta_s(\mathbf{r})$ are defined as Cartesian products:

$$\Theta(\mathbf{r} - \mathbf{R}_s) = \theta(x - X_s)\theta(y - Y_s)\theta(z - Z_s) \ , \tag{20}$$

where (X_s, Y_s, Z_s) are the Cartesian components of $\mathbf{R}_s$, and the support functions are represented as:

$$\phi_{i\alpha} = \sum_s b_{i\alpha s} \Theta_s(\mathbf{r}) \ . \tag{21}$$

In the current scheme, the blip-grid on which the $\Theta_s(\mathbf{r})$ are sited is defined separately for each atom, and moves with that atom. To enforce the vanishing of $\phi_{i\alpha}(\mathbf{r})$ outside the support region, we include in eqn (21) only those $\Theta_s(\mathbf{r})$ that are non-zero only for points within the region. The reason for making the blip-grid move with the atom is that this ensures that each $\phi_{i\alpha}(\mathbf{r})$ is represented always in terms of the same set of basis functions.

Blip functions therefore give us a scheme that is closely related to plane waves, but at the same time respects the strict localisation of the support functions. It also shares another feature with plane waves, and that is that as the blip spacing is decreased, the computational effort grows linearly only with the number of blip functions. This is because the number of blip functions that are non-zero at each point in space does not increase as a decreases.

But this is certainly not the only spatially localised basis set that is closely related to plane waves. An alternative is the spherical-wave basis proposed by Haynes and Payne[48]. Spherical waves are the energy eigenfunctions of a particle confined to a spherical box, where the radius of the box is the support region radius R_{reg}. The properties of this basis set have been explored in detail in Ref. [48, 49], where tests on molecules (H_2, HCl, Cl_2, SiH_4) and bulk silicon show reasonable convergence properties relative to plane wave results.

This scheme has the nice feature that a total-energy calculation can be converged in exactly the same way as a plane-wave calculation, by systematically increasing the plane-wave cut-off wavevector $G_{\max}$. However, it has serious objection, which may make it difficult to use in practical calculations. This is that as the size of the basis set is increased, the computational effort grows as the square of the number of basis functions. This is because the number of basis functions that are non-zero at any point in space is proportional to the total number of basis

functions, so that matrix elements $\langle\phi_{i\alpha}|O|\phi_{j\beta}\rangle$ of any operator O for any two atoms i and j grows very rapidly. Since we know in advance that at least 100 spherical waves will be needed per atom, the number of operations needed to calculate each such matrix element is likely to be at least 10^4, and this may make the calculations too slow.

Finally, we mention the pseudo-atomic basis sets used in SIESTA[50]. The basic philosophy is similar to that originally developed by Sankey and Niklewski [51]. The basis functions are the atomic orbitals obtained from a self-consistent DFT calculation on the free atom, except that, in order to ensure that the funcitons vanish identically outside the region radius, the free-atom calculation is done in the presence of a confining potential. In the original Sankey-Niklewski scheme, the confining potential is simply an infinite potential beyond the radius $R_{\rm reg}$. This makes the localised orbitals go to zero linearly as $r \to R_{\rm reg}$, so that there is a discontinuity in the first derivative, which may well exacerbate the breaking of translational symmetry in the parts of the calculation done on a spatial grid. This gives a motivation for making the confining potential go to infinity more continuously as $r \to R_{\rm reg}$, and this freedom is being exploited in the latest SIESTA basis sets[52]. In addition, in order to obtain satisfactory precision, it is essential to go beyond minimal basis sets, and to include at least two basis functions for each angular momentum (so-called 'double-zeta' basis sets), and to include also polarisation functions.

The different approaches to the problem of basis sets taken with linear-scaling DFT schemes thus reflect the tension between the four criteria outlined at the start of this Section, and particularly the tension between good adaptation of the basis set to the form of the orbitals, and resulting economy in memory use, on the one hand, and systematic improvability onthe other hand. As in more conventional DFT methods, there cannot be a 'best' approach to basis sets, since the physical problem being addressed and the resources available will place different weights on the criteria. Our view is therefore that there is great merit in a flexible approach, in which different types of basis set are employed for different problems, or at different stages of a given problem. We also believe it may be possible to combine different schemes, for example pseudo-atomic orbitals and B-splines, in the same way as mixed basis sets have long been used in conventional DFT/pseudopotential calculations.

3.2 Blip Operations

As described above, the support functions are represented in a basis of blip functions (or B-splines), defined on a grid that moves with each atom. We need to to perform integrals involving the support functions to generate matrix elements (such as $S_{i\alpha j\beta} = \int {\rm d}\mathbf{r}\phi_{i\alpha}(\mathbf{r})\phi_{j\beta}(\mathbf{r})$). For the overlap matrix and the kinetic energy part of the Hamiltonian, the integration can be performed analytically in terms of the $b_{i\alpha s}$ coefficients. For example, $S_{i\alpha,j\beta}$ can be expressed as:

$$S_{i\alpha,j\beta} = \sum_{m,n} b_{i\alpha m} b_{j\beta n} s_{im,jn} \ , \tag{22}$$

where:

$$s_{im,jn} = \int d\mathbf{r}\, \Theta_{im}\Theta_{jn} \ , \tag{23}$$

However, some parts of the Hamiltonian matrix cannot be calculated analytically, and integration must be approximated by summation on a grid. This 'integration grid' is completely distinct from the blip grid, and is a single fixed grid covering the whole simulation cell. For a

uniform cubic grid of spacing $h_{\rm int}$, a matrix element such $S_{i\alpha,j\beta}$ would be approximated as:

$$S_{i\alpha,j\beta} \simeq \delta\omega_{\rm int} \sum_{\ell} \phi_{i\alpha}(\mathbf{r}_\ell)\phi_{j\beta}(\mathbf{r}_\ell) \ , \tag{24}$$

where $\delta\omega_{\rm int} = h_{\rm int}^3$ is the volume per grid point, and $\mathbf{r}_\ell$ is the position of the ℓth grid point. As shown elsewhere [20], $h_{\rm int}$ should generally be about half of $h_{\rm blip}$.

Analytic evaluation, if possible, is preferable, but the double summation required (see eq. 22) brings a computational cost. Our strategy is to evaluate analytically the on-site ($i = j$) matrix elements of overlap and kinetic energy, and to use grid summation for all others. The thinking here is that the on-site terms are large, so that accuracy is important; but there are few of them, so that the cost of analytic evaluation is small.

The transformation from the coefficients $b_{i\alpha s}$ to the values of $\phi_{i\alpha}(\mathbf{r}_l)$, where $\mathbf{r}_l$ is a grid point, is called a blip-to-grid transform[47, 21] – using the separable form of blips shown above in eq. 20, this is extremely similar in concept to an FFT, and can be evaluated extremely efficiently. The integration can be performed efficiently by creating small blocks of integration grid points, and making partial contributions to matrix elements between all atoms touching the block with a single BLAS call, as discussed below.

4. Practical Details

In this section, we address the practical implementation of CONQUEST on massively parallel machines, as well as some of the practical problems which we have encountered in the course of writing CONQUEST, related both to efficiency on parallel computers and to robustness[21, 25, 53].

4.1 Implementation

The division of workload between processors is an important part of all parallel codes – indeed, load balancing is a large subject in its own right. Nominally, the computational effort and storage requirements of CONQUEST are divided up as follows:

1. Every processor has responsibility for a group of atoms (storing the blip coefficients for each atom and transforming their values onto the integration grid) – the primary set.
2. Every processor has responsibility for rows of matrices of these atoms (storing values and performing multiplications for these rows).
3. Every processor has responsibility for an area of the integration grid (storing data on this area and performing integrations) – the domain.

Practically, the assignment of groups of atoms and areas of the integration grid will have a large effect on the efficiency of the code – typically, we want the groups of atoms and grid points to be compact and local and we want the two groups to overlap as much as possible (to restrict communication). Below, we describe a key technique in achieving flexibility in load balancing as well as efficiency in computation – small groups.

4.2 The Use of Small Groups

In CONQUEST, there are two key areas of effort: matrix multiplication, and grid operations (integration, blip-to-grid transforms etc). In both of these areas, there are two natural levels of organisation: individual (e.g. grid point or matrix element); and global (all atoms or grid points). We have found that it is vital for efficiency to create an intermediate level of organisation – small groups of the entities (we call a small group of atoms a *partition* and a small group of integration grid points a *block*).

To understand the use of small groups, let us take an example of matrix multiplication, where we perform:

$$C_{ij} = \sum_k A_{ik} B_{kj}. \tag{25}$$

Here, each processor is responsible for calculating all elements C_{ij} for atoms i for which it is responsible – its primary set. It already has the values A_{ik} stored locally, but will have to fetch the values B_{kj} for the atoms k outside its primary set. Following the two natural levels mentioned above, there are two extremes we can consider for fetching the elements B_{kj} (and hence interleaving communication and calculation): individually (fetch a single element, compute the partial contribution to C_{ij}, repeat – fine interleaving); and globally (fetch all elements B_{kj} which will be required, then do all computations – coarse interleaving). The first of these will be overly expensive in communications (all communication involves a latency or start-up cost, so that a significant gain is made by transferring long messages) while the second will potentially be expensive in memory. Somewhere between these two extremes there will be a good balance between memory required on-processor, and the latency of short communications.

Partitions of atoms (and hence matrix elements) are extremely useful as they simplify the task of finding the compromise between the two extremes given above. For example, if a processor is responsible for several partitions, then transferring the B_{kj} according to the partitions will split up the calculation in a natural way; this will be explored more in the next section, and has been extensively discussed in a recent paper[53]. Another area where these groups are helpful is in integration: we use the highly optimised BLAS routines on all integration grid points in a block to yield extremely efficient integration routines; this is touched on later.

4.3 Matrix Multiplication

Recently, we have studied the efficiency of sparse matrix multiplication on highly parallel machines[53]. Each processor takes responsibility for several partitions of atoms, formed into its primary set. For a given matrix multiplication, we form a *halo* consisting of all atoms (or partitions) within range of *any* primary set atom, and loop over these atoms during the multiplication. It is also helpful to form a *covering set*: a super set of different matrix haloes. This simplifies searches and indexing for various different matrix multiplications. We have also identified a multiplication *kernel*: a piece of code that is repeatedly called, and which can be optimised on different machines. We have achieved 10% of peak speed on a Cray T3E, which is respectable given the pattern of matrices.

We illustrate the practical performance of the CONQUEST code in Figure 2. Here we show the time taken for various aspects of the calculation for increasing system sizes on a Cray T3E-

1200. As the technique is variational, the time to self-consistency will improve after the slow, initial search. We see that for fixed number of atoms per processor, the time taken is almost constant, showing that the matrix multiplication (which forms the bulk of the workload for these calculations) scales extremely well in parallel.

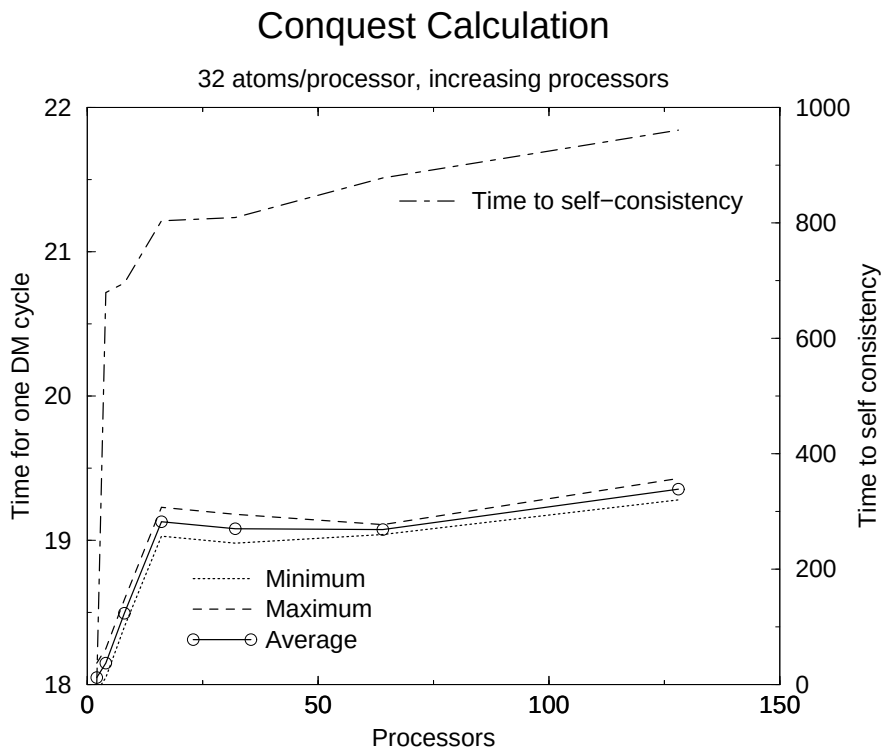

Figure 2. Time taken for CONQUEST calculations for a single density matrix minimisation (left axis, bottom three lines) and for a complete search for self-consistency, starting from scratch, fixed support functions.

4.4 Integration and Grid Operations

In the light of the recent developments with matrix multiplication, we have been considering the efficiency of blip-to-grid transforms and the indexing associated with this and with integration. Let us first consider what kind of operations this requires. The matrix elements made from support functions, projector functions, or their derivatives such as $S_{i\alpha,j\beta} = \langle \phi_{i\alpha} | \phi_{j\beta} \rangle$, $P_{i\alpha,j\beta} = \langle \phi_{i\alpha} | \chi_{j\beta} \rangle$, and $\langle \nabla \phi_{i\alpha} | \nabla \phi_{j\beta} \rangle$, are calculated by a summation over integration grid points. For example, $S_{i\alpha,j\beta}$ is calculated by

$$S_{i\alpha,j\beta} = \omega_{\rm int} \sum_{\boldsymbol{r}_l} \phi_{i\alpha}(\boldsymbol{r}_l) \phi_{j\beta}(\boldsymbol{r}_l) \tag{26}$$

Here, $\omega_{\rm int}$ is the volume per grid point and $\boldsymbol{r}_l$ is integration grid which is common to the support region of i and that of j. The above summation for a given set of indices (i,α) and (j,β) can be regarded as a matrix multiplication, where $\boldsymbol{r}_l$ serves as the column index of the left matrix and as the row index of the right matrix. As explained in Ref. [21], we use a BLAS-3 routine `dgemm` to do these matrix multiplications and we have introduced the term 'blocks' for the effective use of this routine. A block is an assembly of integration grid points and the above summation is

divided into the summation of partial contributions from all blocks and the calculation of the partial contribution by `dgemm` for each block. As we will see later again, we refer to an atom whose support region contains at least one integration grid point in a block b as a neighbour atom of a block b for support functions. For these calculations, we must have the list of the pairs of atoms (i,j) for each block b, both of which are neighbour atoms of the block. In practice, a block is a cuboid[3] containing $n_x \times n_y \times n_z$ points and its size should be determined by comparing the gain in the speed of `dgemm` by increasing the size of matrices and the loss by unnecessary operations from zero values in $\phi_{i\alpha}(\boldsymbol{r}_l)$.

To perform the integration, we also must know the values of support or projector functions on integration grid. For support functions, we have to calculate the values from a set of coefficients $\{b_{i\alpha s}\}$, where s is the index of blip grid.

$$\phi_{i\alpha}(\boldsymbol{r}_l) = \sum_s b_{i\alpha s}\Theta(\boldsymbol{r}_l - \boldsymbol{R}_{is}). \tag{27}$$

Here, $\boldsymbol{R}_{is}$ is the position of blip grid and we call this type of operations blip-grid transforms. Blip-grid transforms are performed only for $\boldsymbol{r}_l$ in the blocks which include one or more integration grid points in the support region the atom i. We refer to these blocks as neighbour blocks of the atom i.

In short, we need to consider the optimal way of performing the following tasks:

1. Making lists and tables to perform blip-grid transforms and the calculation of matrix elements: making lists of neighbour atoms of blocks; making lists of neighbour blocks of atoms; and so on.
2. Blip-grid transform
3. Calculation of matrix elements

It should be noted that the matrices calculated in this way are used to calculate other matrices, LSL, $LSLSL$, $SLSLH$ and so on.

In the new method, we use the same data structure shown in the previous sections. In the method for performing matrix multiplications, each node has a set of small groups called partitions, and each partition has its members, i.e. atoms. Obviously, we can regard a block as a small group of integration grid points. Each processor is responsible for a set of blocks which we call domain.

Table 1. How small groups are made up for different members.

members	small groups	primary sets
atoms	partitions	bundles
integration grid points	blocks	domains

Each processor has one domain and one bundle. Hereafter, we refer to a node which performs operations with respect to atoms as a bundle-responsible node, and the node doing operations

[3]This is not necessary in principle, and non-orthorhombic cells and hence blocks will be addressed in the future

regarding integration grid as a domain-responsible node. In performing the above three tasks, we have a lot of communications between bundle-responsible nodes and domain-responsible nodes. In blip-grid transforms, for example, bundle-responsible nodes first calculate $\phi_{i\alpha}(\boldsymbol{r}_l)$ for integration grid points $\boldsymbol{r}_l$ in the support region of the atoms i, because bundle-responsible nodes have a set of blip coefficients $\{b_{ian}\}$. Then, they must send these values to domain-responsible nodes which have integration grid $\boldsymbol{r}_l$. In the calculation of matrix elements, each domain-responsible node accumulates the partial contributions to matrix elements from all blocks in the domain, and these contributions are sent to bundle-responsible nodes, which accumulate the contributions sent from their halo nodes to make their matrix elements. How to organise these communications is the key point for performing the operations in this section, efficiently.

In searching neighbour atoms of blocks or neighbour blocks of atoms, and in the way of labelling, we can completely follow the scheme used in matrix multiplications. For each block b in the primary set, there are atoms i in the system whose distance from b is less than the cutoff radius of support functions or projector functions. We refer to these atoms as neighbour atoms of b. The atoms i which are neighbours of at least one block in a domain form a set which we call halo atoms of the domain. The set of partitions containing at least one halo atoms are refered to as halo partitions, and the set of nodes having at least one halo partitions as halo nodes. Further, we can define a covering set made of partitions as the one which includes all neighbour atoms of one or more blocks in a domain. We call this a domain covering set (DCS) of partitions. Following this way of naming, a grand covering set used in matrix multiplications can be refered to as a bundle covering set (BCS) of partitions. Similary, as we need a list of neighbour blocks of atoms in a primary set, we define the terms, such as neighbour blocks of the atom i, halo blocks of a bundle, and a BCS of blocks. Even in increasing the size of a simulation cell, if we increase the number of processors and keep the form of each domain, the number of members in covering sets is obviously constant. Thus, with the covering sets, the cost of searching neighbour atoms or blocks is proportional to N, not to N^2. This advantage of the new code is important for the calculations on very large systems.

5. Future Prospects

In many ways, linear-scaling DFT is now established as a viable technique. Within the CONQUEST project, we have shown how practical linear-scaling performance can be achieved on systems of many thousands of atoms. Most of the practical problems presented by the search for the self-consistent ground state for such large systems are now solved, or are close to being solved. The practical challenges of implementing the algorithms on large parallel computers, including PC clusters, have also been addressed in detail. However, one issue that is not yet solved to our satisfaction is that of the basis sets for representing support functions, and this is the main reason why CONQUEST has not yet been applied to major scientific problems.

But the ideas embodied in CONQUEST can be seen as part of a larger current of thought that is being followed by many condensed-matter groups - a move away from extended orbitals

and extended basis sets and towards a formulation in terms of localized orbitals and localized basis sets. The key issue of how to make effective localized basis sets, which is so crucial to CONQUEST, was explored in depth in the very recent CECAM workshop on 'Localized orbitals and linear-scaling calculations', which was part-funded by Psi-k. A clear message from this workshop was that the new atomic-like basis functions being developed by several groups can often compete very effectively with plane waves, while demanding far less memory and often far fewer cpu cycles. By contrast, CONQUEST is currently based on finite-element methods that are deliberately related to the plane-wave approach. The exciting recent progress with atomic-like basis sets is already making an important impact in the SIESTA[52] and PLATO[54] codes and in other codes, and will undoubtedly be important for the future of CONQUEST and for linear-scaling DFT in general. We have high hopes that the SIESTA-CONQUEST collaboration, now in its early stages, will accelerate progress in this area.

Finally, we want to emphasize the very broad importance of linear-scaling ideas. One reason for this importance is the close relation between linear scaling and the 'embedding' problem, i.e. the problem of performing quantum calculations on a limited region which is 'embedded' in a much larger surrounding region. Since embedding also demands a localized formulation of quantum mechanics, and can be seen as the problem of embedding the density matrix in one region into that of a surrounding region, it is pretty well guaranteed that progress in linear scaling can be exploited for the embedding problem. Another completely different reason for the broad importance of linear scaling is its implications for quantum Monte Carlo. It is already clear that many of the current linear-scaling ideas can be directly transferred to improve the system-size scaling of QMC. By the same token, we expect that the long-standing problem of QMC embedding will also be helped forward by some of the new ideas. The future looks exciting!

Acknowledgments

We are happy to acknowledge useful discussions with D. Manolopoulos, A. Horsfield and S. Goedecker, and assistance with optimisation and parallelisation from EPCC (Ian Bush) and CSAR (Martyn Foster and Stephen Pickles).

References

[1] S.Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).

[2] D. G. Pettifor, Phys. Rev. Lett. **63**, 2480 (1989).

[3] W. Yang, Phys. Rev. Lett. **66**, 1438 (1991).

[4] G. Galli and M. Parrinello, Phys. Rev. Lett. **69**, 3547 (1992).

[5] X.-P. Li, R. W. Nunes and D. Vanderbilt, Phys. Rev. B **47**, 10891 (1993).

[6] M. S. Daw, Phys. Rev. B **47**, 10895 (1993).

[7] P. Ordejón, D. Drabold, M. Grumbach and R. Martin, Phys. Rev. B **48**, 14646 (1993).

[8] M. Aoki, Phys. Rev. Lett. **71**, 3842 (1993).

[9] F. Mauri, G. Galli and R. Car, Phys. Rev. B **47**, 9973 (1993).

[10] S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).

[11] E. B. Stechel, A. R. Williams and P. J. Feibelman, Phys. Rev. B **49**, 10088 (1994).

[12] R. W. Nunes and D. Vanderbilt, Phys. Rev. B **50**, 17611 (1994).

[13] W. Hierse and E. B. Stechel, Phys. Rev. B **50**, 17811 (1994).

[14] W. Kohn, Int. J. Quant. Chem. **56**, 229 (1995).

[15] E. Hernández and M. J. Gillan, Phys. Rev. B **51**, 10157 (1995).

[16] J. D. Kress and A. F. Voter, Phys. Rev. B **53**, 12733 (1996).

[17] A. P. Horsfield, Mat. Sci. Engin. B **37**, 219 (1996).

[18] A. P. Horsfield, A. M. Bratkovsky, M. Fearn, D. G. Pettifor and M. Aoki, Phys. Rev. B **53**, 12964 (1996).

[19] W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).

[20] E. Hernández, M. J. Gillan and C. M. Goringe, Phys. Rev. B **53**, 7147 (1996).

[21] C. M. Goringe, E. Hernández, M. J. Gillan and I. J. Bush, Comput. Phys. Commun. **102**, 1 (1997).

[22] R. Baer and M. Head-Gordon, Phys. Rev. Lett. **79**, 3962 (1997).

[23] P. D. Haynes and M. C. Payne, Phys. Rev. B **59**, 12173 (1999).

[24] D. R. Bowler and M. J. Gillan, Comp. Phys. Commun. **120**, 95 (1999).

[25] D. R. Bowler, I. J. Bush and M. J. Gillan, Int. J. Quant. Chem. **77**, 831 (2000).

[26] D. R. Bowler, M. Aoki, C. M. Goringe, A. P. Horsfield and D. G. Pettifor, Modell. Simul. Mat. Sci. Eng. **5**, 199 (1997).

[27] S. Ismail-Beigi and T. Arias, Phys. Rev. Lett. bf 82, 2127 (1999).

[28] S. Goedecker, Phys. Rev. B **58**, 3501 (1998).

[29] L. He and D. Vanderbilt, Phys. Rev. Lett. **86**, 5341 (2001).

[30] W. Kohn, Phys. Rev. **115**, 809 (1959).

[31] J. des Cloiseaux, Phys. Rev. **135** A658 (1964).

[32] S. Goedecker and M. Teter, Phys. Rev. B **51**, 9455 (1995).

[33] A. F. Voter, J. D. Kress and R. N. Silver, Phys. Rev. B **53**, 12733 (1996).

[34] U. Stephan and D. Drabold, Phys. Rev. B **57**, 6391 (1998).

[35] M. J. Gillan, J. Phys.: Condens. Matter **1**, 689 (1989).

[36] M. Challacombe, J. Chem. Phys **110**, 2332 (1999).

[37] A. D. Daniels, J. M. Millam and G. E. Scuseria, J. Chem. Phys. **107**, 425 (1997).

[38] S.-Y. Qiu, C. Z. Wang, K. M. Ho and C. T. Chan J. Phys: Condens. Matter **6**, 9153 (1994).

[39] C. M. Goringe, D.Phil Thesis, Oxford University (1995).

[40] R. McWeeny, Rev. Mod. Phys. **32**, 335 (1960).

[41] A. H. R. Palser and D. E. Manolopoulos, Phys. Rev. B, **58**, 12704 (1998).

[42] J. Kim, F. Mauri and G. Galli, Phys. Rev. B **52**, 1640 (1995).

[43] P. D. Haynes and M. C. Payne, Sol. Stat. Commun. **108**, 737 (1998).

[44] C. A. White, P. Maslen, M. S. Lee and M. Head-Gordon, Chem. Phys. Lett. **276**, 133 (1997).

[45] J. M. Millam and G. E. Scuseria, J. Chem. Phys **106**, 5569 (1997).

[46] D. R. Bowler and M. J. Gillan, submitted to Chem. Phys. Lett. (2001).

[47] E. Hernández, M. J. Gillan and C. M. Goringe, Phys. Rev. B **55**, 13485 (1997).

[48] P. D. Haynes and M. C. Payne, Comput. Phys. Commun. **102**, 17 (1997).

[49] C. K. Gan, P. D. Haynes and M. C. Payne, Phys. Rev. B **63**, 205109 (2001).

[50] D. Sánchez-Portal, P. Ordejón, E. Artacho and J. M. Soler, Int. J. Quant. Chem. **65**, 453 (1997).

[51] O. F. Sankey and D. J. Niklewski, Phys. Rev. B **40**, 3979 (1989).

[52] E. Artacho, D. Sánchez-Portal, P. Ordejón, A. García and J. M. Soler, phys. stat. sol. b **215**, 809 (1999).

[53] D. R. Bowler, T. Miyazaki and M. J. Gillan, Comput. Phys. Commun. **321**, 1000 (2001).

[54] S. D. Kenny, A. P. Horsfield and H. Fujitani, Phys. Rev. B **62**, 4899 (2000).