Linear-scaling density-functional simulations of charged point defects in Al_2O_3 using hierarchical sparse matrix algebra

N. D. M. Hine,[1,a)] P. D. Haynes,[1] A. A. Mostofi,[1] and M. C. Payne[2]
[1]*Department of Physics and Department of Materials, Imperial College London, Exhibition Road, SW7 2AZ London, United Kingdom*
[2]*Theory of Condensed Matter Group, Cavendish Laboratory, J. J. Thomson Avenue, CB3 0HE Cambridge, United Kingdom*

(Received 21 July 2010; accepted 2 September 2010; published online 20 September 2010)

We present calculations of formation energies of defects in an ionic solid (Al_2O_3) extrapolated to the dilute limit, corresponding to a simulation cell of infinite size. The large-scale calculations required for this extrapolation are enabled by developments in the approach to parallel sparse matrix algebra operations, which are central to linear-scaling density-functional theory calculations. The computational cost of manipulating sparse matrices, whose sizes are determined by the large number of basis functions present, is greatly improved with this new approach. We present details of the sparse algebra scheme implemented in the ONETEP code using hierarchical sparsity patterns, and demonstrate its use in calculations on a wide range of systems, involving thousands of atoms on hundreds to thousands of parallel processes. © *2010 American Institute of Physics*. [doi:10.1063/1.3492379]

I. INTRODUCTION

It is well established that accurate calculations of the properties of point defects in crystalline materials require the use of very large simulation cells containing large numbers of atoms.[1] In particular, the formation energies of charged point defects are strongly affected by finite size errors up to very large systems, due to the very slow decay of the elastic and electrostatic fields resulting from the localized perturbation. Crystalline alumina (Al_2O_3) is an example of a system where the high ionic charges (formally Al^{3+} and O^{2-}) mean that the predominant defects are charged (V_O^{2+} and V_{Al}^{3-}) and thus challenging to simulate accurately.[2]

For these types of calculation, density-functional[3] theory (DFT) is well established as a mainstay of computational methods. However, traditional approaches to DFT encounter a cubic-scaling "wall" with increasing system size since the total computational effort of a calculation involving N atoms increases as $O(N^3)$. Furthermore, this effort is nontrivial to parallelize to large numbers of cores, rendering calculations of much beyond 1000 atoms rather impractical. Alternative approaches, known as linear-scaling DFT (LS-DFT),[4,5] reformulate traditional DFT to avoid the computation of orthonormal eigenstates of the Hamiltonian since manipulation of these is inherently cubic-scaling with system size.

LS-DFT has been the focus of considerable development effort, with several mature codes now available.[6–12] Rather than working with extended eigenstates, linear-scaling approaches work in a basis of localized functions, which are, in general, nonorthogonal. Each of these functions has significant overlap only with a small number of other functions on nearby atoms and this number remains constant with increasing system size N in the limit of large N. This means that a matrix representing a local operator expressed in this localized basis contains a total number of nonzero elements which scales only as $O(N)$ rather than $O(N^2)$. The whole matrix can then be calculated in $O(N)$ computational effort if each individual element can be calculated with effort independent of system size. Furthermore, in an insulator, the single-electron density matrix is itself exponentially localized. Expressed in a separable form in terms of this localized basis, the density matrix can be truncated to $O(N)$ nonzero elements. Finally, multiplication of sparse matrices whose columns each contain only $O(1)$ elements is possible in $O(N)$ effort. It is these three crucial points which enable an overall linear-scaling approach to total energy calculations within DFT in insulating systems.

The efficiency and feasibility of linear-scaling methods depends strongly on the methods used for manipulation of sparse matrices. Performance depends not only on the prefactor relating total computational time to system size, but also on the degree of parallelizability of the method. The computing power available in a single parallel process (one core of one processor in a parallel computer) has not increased significantly in recent years, so if feasible system sizes are to increase, the scaling of algorithms with number of parallel processes P is just as important as scaling with N. Put another way, as N grows, algorithms for linear-scaling DFT must remain able to perform simulations within feasible wall-clock time by allowing P to increase. To ensure this, if N and P are increased proportionally, the total time must remain roughly constant. This is not simply a computer science issue, but requires parallel algorithms designed around the physics of the problem and the systems being studied.

In this paper, we describe and apply a novel approach to matrix algebra in the context of electronic structure methods, optimized for linear-scaling calculations on parallel computers. In this "hybrid" scheme, designed for speed and parallel

[a)] Electronic mail: nicholas.hine@imperial.ac.uk.

efficiency, we mix elements of both sparse and dense matrix algebra. We divide up matrices according to the parallel process on which the data associated with a given atom is located, then define segments of the matrix to be stored (and communicated) in sparse-indexed or dense format according to the density of nonzero elements. It becomes possible to determine for a given operation a fixed subset of parallel processes with which each process will require communication and how best to transmit the communicated data on the basis of the physical distribution of the atoms within the simulation cell. At a constant ratio of atoms per core N/P, the number of other cores with which communication is required is shown to remain constant. Therefore, for larger and larger equivalent calculations on more and more parallel cores, total time does not significantly increase, resulting in a constant "time-to-science" for any system size, given adequate computational hardware.

We have implemented our approach within the ONETEP LS-DFT code,[12] although the methods described are generally applicable in any LS-DFT code that employs a localized basis. Its implementation in ONETEP (Sec. II) has led to significant performance improvements which we describe in Sec. III. In Sec. IV, we present an application of the improved methods to the calculation of the formation energies of charged defects in α-Al_2O_3 (corundum). Extrapolating to the limit of infinite dilution the formation energies of the defects, which vary with the size of the simulation cell due to finite size effects, we are able to systematically identify and eliminate the finite size errors present and analyze their effects.

II. SPARSE MATRIX ALGEBRA FOR ELECTRONIC STRUCTURE

A. Sparse matrices in electronic structure

ONETEP is a linear-scaling DFT code for total energy and force calculations on systems of hundreds to hundreds of thousands of atoms.[13] It uses a set of optimizable localized functions, referred to as nonorthogonal generalized Wannier functions (NGWFs),[14] expressed in terms of periodic sinc functions (psincs),[15] to represent the density matrix. The basis of psinc functions has very similar favorable properties to the plane-wave basis frequently used in traditional DFT. ONETEP therefore combines the benefits of linear-scaling with system size with the variational bounds and systematic convergence with respect to basis size provided by a plane-wave basis. Recent development work[13] on the ONETEP code resulted in considerable speedup to the performance of LS-DFT simulations. However, it was also demonstrated that sparse algebra operations remained the limiting factor on parallel efficiency when scaling to large numbers of processes and that scaling of the calculation wall-clock time as $O(N/P)$ was not obtained beyond around $P\sim 100$.

The matrix algebra required for electronic structure calculations using nonorthogonal localized orbitals mostly involves combinations of three basic matrices: the overlap matrix $S_{\alpha\beta}=\langle\phi_\alpha|\phi_\beta\rangle$ between pairs of orbitals $\phi_\alpha(\mathbf{r})$ and $\phi_\beta(\mathbf{r})$, the Kohn–Sham Hamiltonian $H_{\alpha\beta}=\langle\phi_\alpha|\hat{H}|\phi_\beta\rangle$ in the basis of these functions, and the density kernel $K^{\alpha\beta}$, which expresses the single-particle density matrix

$$\rho(\mathbf{r},\mathbf{r}') = \phi_\alpha(\mathbf{r})K^{\alpha\beta}\phi_\beta(\mathbf{r}'). \tag{1}$$

We are using the convention of summation over repeated Greek indices, subscripts to denote covariant indices, and superscripts to denote contravariant ones. The generalization of these and all the following expressions to spin dependent forms is straightforward, so spin-labels will be omitted.

The NGWFs $\{\phi_\alpha(\mathbf{r})\}$ are strictly localized with a cutoff radius R_ϕ (typically around 3–4 Å) so $S_{\alpha\beta}$ is only nonzero if ϕ_α and ϕ_β overlap. $H_{\alpha\beta}$ is nonzero either if ϕ_α and ϕ_β overlap directly or if they both overlap a common nonlocal projector. $K^{\alpha\beta}$, meanwhile, is defined to be nonzero only for elements α,β on atoms at $\mathbf{R}_\alpha$ and $\mathbf{R}_\beta$ for which $|\mathbf{R}_\alpha-\mathbf{R}_\beta| < R_K$, where the kernel cutoff R_K is typically chosen to be of order 10–25 Å in an insulator. All quantities are therefore expressed in terms of matrices containing only $O(N)$ nonzero elements in the limit of large N and for which the matrix structure is known *a priori*. Note that this is in contrast to linear-scaling methods which apply a thresholding approach to sparsity patterns by discarding matrix elements whose magnitude is below some threshold, in which case the matrix structure may change dynamically as the calculation progresses.

Minimization of the total energy in the above formalism corresponds to minimizing

$$E[\{K^{\alpha\beta}\},\{\phi_\alpha\}] = K^{\alpha\beta}H_{\beta\alpha} + E_{\mathrm{DC}}[n], \tag{2}$$

with respect to the kernel $K^{\alpha\beta}$ and the set of functions $\{\phi_\alpha\}$ simultaneously. $E_{\mathrm{DC}}[n]$ is a double-counting term written in terms of the electron density $n(\mathbf{r})=\rho(\mathbf{r},\mathbf{r})$, which subtracts off density-density interactions accounted for twice in the $\mathrm{Tr}[KH]$ term. The total energy E is variational with respect to each of the plane-wave cutoff E_{cut} of the grid on which $\phi_\alpha(\mathbf{r})$ is evaluated, the NGWF radii $\{R_\phi\}$ and the kernel cutoff R_K. Details of the methods used to optimize the kernel[16] and the NGWFs (Ref. 17) can be found elsewhere. The important point is that all aspects of the minimization process require extensive use of sparse matrix algebra, in particular the evaluation of the product of pairs of sparse matrices.

For the optimization of the density matrix, manipulation of matrices of considerably lower sparsity than the density kernel itself is required, so as not to discard contributions to the kernel gradient during conjugate gradients optimization. There is therefore a need for a sparse algebra system capable of high performance and excellent scaling across a range spanning from highly sparse matrices (such as 0.01% nonzero elements) to fully dense matrices (100% nonzero elements) within the same framework. The usual approach to sparse matrix algebra problems is to store only the nonzero elements of the matrix, either determined by the geometry of the problem, or indexed element by element, in an index stored separately from the data itself. However, in large systems, there are very large numbers of nonzero elements and the computational overhead of indexing them can be enormous.

For this reason, for a given matrix algebra operation, there will exist a threshold of element density above which the dense matrix algebra is more efficient than sparse algebra. This is often around 10% or less for a matrix product, though precise details depend on the specific software and hardware implementation. Previous versions of ONETEP (and to our knowledge, most other linear-scaling electronic structure codes) have supported only either all sparse-indexed matrices or all dense matrices within one calculation. However, neither of these extremes is able to obtain good performance and scaling for typical realistic systems. We thus describe a hybrid hierarchical system able to handle the cases of both highly sparse and fully dense matrices efficiently within the same framework, building the distribution of the matrix over the processes of a parallel computer into the framework in a natural way. Hierarchical approaches to sparse matrix algebra, such as combining atom-blocks into larger groups, have been described previously.[18–21] However, this has generally been applied to electronic structure methods applying thresholding to determine sparsity patterns. Here we will discuss the specifics of application to the case of fixed matrix sparsity, in particular the extra benefits that can be obtained in terms of reducing the volume of parallel communication between processes.

B. Parallel distribution of sparse matrix algebra

The first step is to distribute the atoms over the parallel processes in such a way that each process only has atoms whose functions overlap those of as small as possible a number of other processes. This is achieved by ordering the atoms according to a space-filling curve (SFC).[21] Atoms are assigned a Gray code formed by interleaving the binary digits of their position expressed in terms of a coarsened grid along each of the three axes. The use of a separate grid for each axis ensures that in simulation cells with high aspect ratio, the absolute distances along the three axes are given equal weight. The atoms are then sorted according to their Gray code and distributed to processes in such a way as to balance the number of NGWFs per atom evenly. This ensures each process holds a spatially localized group of atoms, with adjoining spatially localized groups on "nearby" processes numerically.

The distribution of matrix data over processes then follows the distribution of atoms: columns corresponding to functions on a given atom are held by the process to which that atom belongs. For reasons of efficient parallelization, these process-columns are further divided into "segments," corresponding to row-functions associated with a given process. The result is a grid of $P \times P$ segments each of size $M_j \times N_i$, where N_i is the number of column elements on process i and M_j is the number of row elements on process j (Fig. 1).

For a set of atom-centered functions, such as NGWFs or nonlocal pseudopotential projectors, if any function on a given atom I overlaps a function on a different atom J, then all the functions on both atoms overlap, giving rise to a block of $m_J \times n_I$ matrix elements which are all nonzero, where n_I is the number of (column-) functions on atom I and m_J is the number of (row-) functions on atom J. Therefore, rather than indexing individual nonzero elements, a large saving in both memory and CPU time is obtained by indexing nonzero atom-blocks. This form of sparse blocked matrix is common to many linear-scaling electronic structure implementations.[21–23] Note that the number of row elements may differ from the number of column elements for a given atom to allow treatment of nonsquare matrices such as the overlap matrix between NGWFs and projectors.

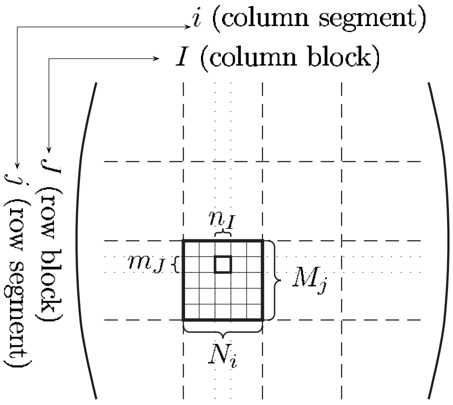

FIG. 1. Parallel distribution of a sparse matrix, first into column-segments (distributed over processes) then into row-segments within each column, according to the same pattern. Each segment is either blank, sparse-indexed by atom-blocks, or fully dense. In the example shown, a matrix with 20 atoms is divided columnwise over four processes, with four row-segments per process, each containing 5×5 atom-blocks, each of which may or may not be present in the sparse index. One individual segment j on process i of size $M_j \times N_i$ is highlighted, as is one block of size $m_J \times n_I$ within this segment, associated with the overlap of atoms I and J.

The typical cutoffs required, especially when dealing with the density kernel in systems with a small energy gap, are quite large on the scale of typical interatomic distances. Therefore, each atom may be associated with nonzero elements in blocks associated with a large number of other atoms. As seen in Fig. 2, the SFC ordering ensures that the majority of these nonzero elements will belong to either the same process or one of a small number of nearby processes. Therefore, a typical matrix will contain a broad band of nonzero elements centered on the diagonal but extending some way off it. Figure 2 shows examples of the segment filling fractions for the product $(KS)^{\alpha}{}_{\beta}$ of the density kernel and the overlap matrix for various typical systems. Because of this banding of nonzero elements near the diagonal, it is often the case that for a given segment near the diagonal, most or in many cases, all of the elements in such a segment are nonzero, especially in the matrices representing products such as KS, KSK, and $KSKS$. Similarly, many of the segments far from the diagonal will contain few nonzero elements or none at all.

We therefore add a second level of indexing of the sparsity pattern, corresponding to the aforementioned segments of the matrix. A count is made of the number of nonzero elements within each segment. This is then used to determine whether each segment either (a) contains a fraction of nonzero elements greater than or equal to some threshold η, where $0 \leq \eta \leq 1$, and is thus declared "dense" and stored in full (including zero elements); (b) contains a fraction of non-

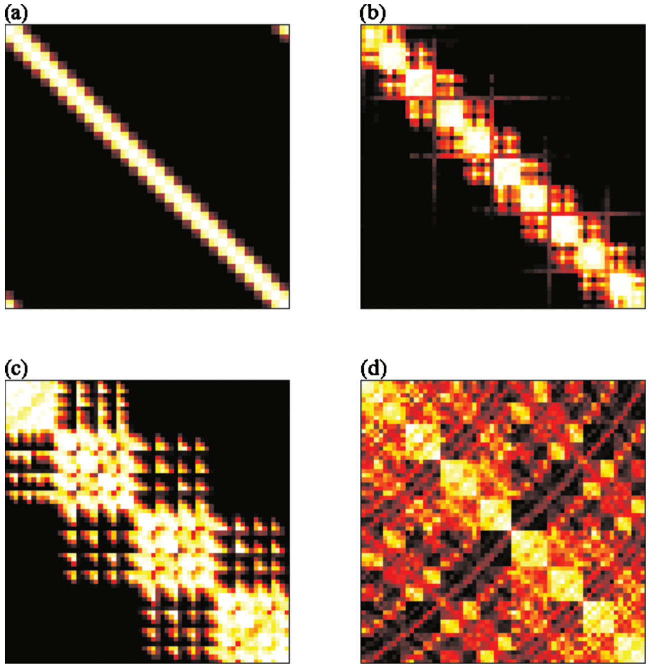

FIG. 2. Segment-by-segment filling factors of sparse matrices in typical large systems divided over $P=64$ processes. Matrices of the sparsity pattern $(KS)^{\alpha}{}_{\beta}$ (the product of the density kernel and overlap matrices) are shown for (a) a (10,0) zigzag nanotube (4000 atoms), (b) a 64 base-pair sequence of *B*-DNA (4182 atoms), (c) a H-terminated wurtzite-structure GaAs nanorod (4296 atoms), and (d) $8\times8\times8$ supercell of eight-atom cubic unit cells of crystalline Si (4096 atoms). Each pixel represents a segment, whose color shows the fraction of matrix elements in that segment which are nonzero: black segments contain no nonzero elements, through red, then yellow, to white segments containing all nonzero elements. The nonzero elements are seen to be clustered near the diagonal of the matrix (though less so with increasing periodicity and complexity of the structure). The space-filling curve ensures that in a given column, there are nonzero overlaps only with rows of atoms on nearby processes, so the nonzero elements form a broad band centered on the diagonal. This is clearest for the simple structure of the nanotube, but even for the crystalline solid, there are segments in which there are few or no nonzero elements.

zero elements less than η and is thus "sparse," and the block-indexing is retained; or (c) contains no nonzero elements and is thus "blank." This segmentation has numerous advantages in terms of reducing both the computation and communications requirements of matrix algebra, particularly matrix multiplication.

Consider the matrix product operation $C^{\alpha}{}_{\beta}=A^{\alpha\gamma}B_{\gamma\beta}$. Each segment can be thought of as a submatrix (in general these are not square). Denoting by $\mathbf{X}_{ki}$ the row-segment k of the matrix $\mathbf{X}$ in column-segment i (thus, stored on process i), we can write the segments of the product matrix as

$$\mathbf{C}_{ki}=\sum_{j}\mathbf{A}_{kj}\cdot\mathbf{B}_{ji}.$$

The individual contributions j to $\mathbf{C}_{ki}$ can be evaluated in several different ways according to the density η in the various segments involved. Note that if either $\mathbf{A}_{kj}$ or $\mathbf{B}_{ji}$ is blank, then no calculation is required for that index value j, while if $\mathbf{C}_{ki}$ is blank, no calculation is required at all.

At the other extreme, if $\mathbf{A}_{kj}$, $\mathbf{B}_{ji}$, and $\mathbf{C}_{ki}$ are all dense segments, then the multiplication operation can be performed very efficiently through a dense linear algebra library call, without reference to the indexing. If $\mathbf{A}_{kj}$ and $\mathbf{B}_{ji}$ are dense, but due to truncation $\mathbf{C}_{ki}$ is not, then the small number of nonzero elements of $\mathbf{C}_{ki}$ can be calculated individually by multiplying the relevant row of $\mathbf{A}_{kj}$ by the column of $\mathbf{B}_{ji}$ and summing the result.

These simplifications greatly reduce the indexing overhead, as they bypass the need for indexing entirely within the most time consuming part of the operation. This alone results in very considerable speedup when the nonzero elements are well-localized near the matrix diagonal. The total effort of this approach does not scale up with P: there are $O(P^3)$ pairs of segments to consider in general (loops over j and k on each process i), but the size of each segment scales down as $1/P$, so the total computational effort is constant with P.

Furthermore, if one imagines a given system and then increases it in size, by scaling up N and P together, the total number of such pairs of segments where both are within some range of the diagonal, and thus dense, will only scale as $O(N)$. This is simply a discretization of the idea that allows sparse algebra to be $O(N)$ in the first place. Therefore, the overall computational effort required for the matrix product remains $O(N)$ even though a large part of it is being performed using dense matrix algebra on fairly large matrices.

C. Communications patterns

There are also significant advantages to be gained from division into segments in terms of reducing communications load. Recall that each process holds a fraction approximately equal to $1/P$ of the columns of the matrix. Taking the simplest approach, the whole index and all the nonzero elements of $\mathbf{A}$ local to process j would be sent to process i, in order that process i could calculate all terms contributing to the local data of $\mathbf{C}$. In that case, a total volume of data $O(N/P)$ is sent by process i to P different processes. The total communications load per process would therefore grow as $O(N)$, but would not scale down at all with increasing P. Asymptotically, the wall-clock time could not behave as $O(1/P)$ with increasing P: a limit would inevitably be reached where communication of the matrix data to all other processes would become the limiting factor on performance.

This undesirable situation can be avoided though the use of shared information about the segments stored on other process. Alongside the creation of the index for a matrix $\mathbf{A}$, process j sends the number of nonzero elements in the segment numbered i stored on process j to process i for future use. When performing a matrix product, process i then only needs to receive the index and data of $\mathbf{A}$ from processes j, for which the segment $\mathbf{B}_{ji}$ on process i is not blank. If many off-diagonal segments of $\mathbf{B}$ are blank, this results in a huge saving in the amount of data sent.

With this approach, if P is increased at fixed N, the total communications load remains constant. In fact, as P and N increase proportionally, the communications load per process does not have to grow, as there remains only a small, constant number of other processes with which any communication is required. Note that this is, in effect, the same principle which allows for $O(N)$ scaling of the total computational effort in the sparse product operation, applied now to the total communications volume of scaling the calculation up to P processes.

The segment-based approach mandates a further improvement in the parallel communications algorithm in order to work effectively. Reference 13 described a communications pattern whereby "blocking" operations, in which all processes received all data from a given process simultaneously, were replaced by a "round-robin" system in which each received data first from its adjacent neighbor, then its next-nearest-neighbor, and so on in synchrony. This represented an excellent improvement in efficiency over the previous system. However, for this algorithm to scale up perfectly to large P requires an idealized distribution of nonzero elements in the matrices involved: the number of overlaps, and thus the amount of computation involved in calculating the contribution to the matrix product on process i of the segments of process j, needs to remain roughly constant for processes of a given numerical separation $|i-j|$. In that case, the algorithm could remain near-ideal even in the presence of division of the matrix into segments and avoidance of communications for noncontributing segments.

However, an ideal distribution of elements is rarely encountered in practice, as illustrated by the nonuniform banding of Fig. 2. The simple algorithm just described thus begins to scale poorly with P at around $P=200$ and very little further increase in speed is obtained beyond about $P=500$.[13] Note that this was nevertheless a significant improvement over the effective $P\sim 64$ limit of the blocking communications approach. Given this uneven distribution, it becomes necessary, for large numbers of processes P, to implement an "on-demand" communications system, whereby in order to receive the index and data of matrix $\mathbf{A}$, process i must first send a request message to process j, and process j then replies by sending the required index and data. Despite the fact that this method incurs a latency penalty twice, this algorithm is almost invariably faster than forcing all processes to work in synchrony, as in the round-robin communications system. In particular, with simultaneously increasing N and P, this approach allows the communications load to scale as $O(N/P)$ as long as each process only requires communications with a small fraction of the other processes.

A further large reduction in the amount of data that must be sent from process j to process i can be achieved even for cases where there are nonzero elements in segment $\mathbf{B}_{ji}$. For each segment k, only those nonzero blocks of $\mathbf{A}_{kj}$ which actually contribute to $\mathbf{C}_{ki}$ on i need to be sent from process j to process i. For this to be the case, there must be nonzero blocks in $\mathbf{B}_{ji}$ which will multiply nonzero blocks in $\mathbf{A}_{kj}$ such that they contribute to nonzero blocks of $\mathbf{C}_{ki}$. A list of such contributing blocks can be formed by having process i request and receive first the indices of each the segments $\mathbf{A}_{kj}$ on j. Looping over the nonzero blocks in the indices of $\mathbf{B}_{ji}$ and $\mathbf{C}_{ki}$ on process i, a list can be made of those nonzero blocks in $\mathbf{A}_{kj}$ which contribute to the result. From this, a "cropped" version of the index of $\mathbf{A}_{kj}$ is constructed which contains only the nonzero blocks contributing to the result. This list is sent back from i to j; upon receiving it, process j extracts the required blocks from the local segments $\mathbf{A}_{kj}$ and sends them to process i.

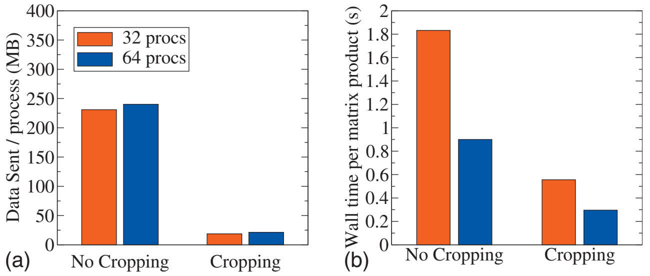

FIG. 3. (a) Total data sent per process and (b) total time per operation for a matrix product operation $K\cdot S\rightarrow KS$ performed for a 4000-atom carbon nanotube on 32 and 64 processes. This system has four NGWFs per atom, hence K, S, and KS are all $16\,000\times 16\,000$ matrices, with roughly 2%, 7%, and 10% nonzero elements, respectively. Results are shown with and without cropping of the data of K before it is sent, showing the decrease in communications volume and the resulting improvement in both overall time and scaling. See Fig. 2(a) for a representation of the sparsity pattern of the matrix KS.

Using this "cropping" method, any given nonzero block in the data of each $\mathbf{A}_{kj}$ is only sent to those processes which need it, which may be very much less than the total number of processes with which there are any overlaps at all. This results in a very considerable reduction in total communications volume. Figure 3(a) shows the total amount of data sent per process while performing a typical matrix multiplication ($K\cdot S\rightarrow KS$), with and without cropping the data of $\mathbf{A}$, for a system comprising 100 unit cells of a (10,0) zigzag carbon nanotube (4000 atoms). Figure 3(b) shows the wall time for this product operations with and without the cropping.

III. PERFORMANCE

The hierarchical sparsity scheme described here obtains significant performance benefits over previous methods in real simulations. To demonstrate this, we compare the wall-clock time for systems representing five common uses of linear-scaling DFT. These are (i) a section of a (10,0) zigzag nanotube, (ii) a 64 base-pair strand of a molecule of DNA with a random sequence of base-pairs, (iii) a large orthorhombic supercell of α-alumina ($6\times 6\times 3$ copies of the 30-atom hexagonal cell), (iv) a gallium arsenide nanorod, and (v) a large supercell ($8\times 8\times 8$ copies of the eight-atom cubic cell) of crystalline silicon (a small-gap semiconductor). These systems are all of comparable size in terms of numbers of atoms (around 4000), but display a wide range of levels of matrix sparsity, plane-wave cutoffs, NGWF and kernel cutoffs, and numbers of nonlocal projectors.

Table I summarizes the details of these test systems. Note that we have chosen in each case the most accurate available norm-conserving potential available; in some cases, this results in a high number of projectors. The Appendix describes the application of these sparse algebra techniques to the nonlocal contribution to the Hamiltonian and the speed up thus obtained in the routines treating nonlocal pseudopotentials. This has the result that the nonlocal potential calculation is a comparatively small fraction of the total time (less than 20% in all cases). The cutoff energies are chosen so as

TABLE I. Key to the abbreviations used for the five different test systems, chosen to represent a cross-section of common uses of LS-DFT (carbon nanostructures, organic molecules, ceramic oxides, semiconductor nanostructures, and crystals). Choices also represent different extremes of cutoff energy E_{cut} and kernel and NGWF cutoffs R_K and R_ϕ. The number of atoms N_{at}, number of NGWFs N_ϕ, and number of nonlocal pseudopotential projectors N_{proj} are also shown.

Abbreviation	System	E_{cut}/eV	R_K/a_0	R_ϕ/a_0	N_{at}	N_ϕ	N_{proj}
C4000 NT	(10,0) carbon nanotube	400	20	6.7	4000	16 000	36 000
DNA 64bp	64 base-pairs of DNA (Na^+-neutralized)	700	30	7.0	4182	9776	2526
$6\times6\times3$ Al_2O_3	α-alumina $6\times6\times3\times$hexagonal cell	1200	24	8.0	3240	12 960	18 792
GaAs NR	H-terminated wurtzite GaAs nanorod	400	40	10.0	4296	14 376	13 440
Si4096	Si crystal, $8\times8\times8\times$cubic cell	600	24	6.7	4096	16 384	32 768

to ensure convergence of the total energy in ONETEP (which has slightly more stringent requirements on the spacing of the underlying psinc grid compared to the real space grid corresponding to a given plane-wave cutoff). In some cases, such as the 600 eV cutoff used for the silicon system, this is higher than the corresponding value required for a typical plane-wave calculation. In all these tests, we have used 16–64 quad-core Intel Core i7 processors with 12 GB of memory per processor, i.e., 64–256 individual cores. These processors are linked by a dual-infiniband network with very low latency.

As explained in Sec. II, segments of each matrix structure are determined to be either blank, sparse, or dense depending on the fraction of nonzero elements they contain. The dividing line between sparse and dense storage and use is determined by a parameter η, the fractional filling above which a given segment is stored as a full matrix rather than indexed. For serial matrix algebra, (where the whole matrix is effectively one segment) the optimal value for this is often quoted as of order 0.1, but in the case of parallel matrix algebra, where there is a large contribution to the total time from communications overhead, it may differ significantly from this value. To find a suitable default choice, we compare in Fig. 4 the total time spent on sparse matrix product operations during a single NGWF optimization iteration with ONETEP for these typical systems. As η is varied from $\eta=0$, at which point all segments are stored in dense format, to $\eta>1$, at which point all segments are stored in sparse format, the sparse algebra becomes at first more then subsequently less efficient and a minimum is observed for most systems at around $\eta=0.3-0.4$. The exceptions are the solid systems, where due to the 3D periodicity, there is a less uniform distribution of overlaps [see Fig. 2(d)], so less benefit is obtained through matrix sparsity. Nevertheless, $\eta=0.4$ appears to represent a good compromise which works well for almost all systems.

To show the effect of the new approach to sparse matrix algebra on total execution times, Fig. 5(a) shows the total time for one NGWF optimization iteration on 64 parallel cores, with and without the application of the aforementioned algorithm to sparse algebra. Typically 10–25 iterations are required for full energy minimization, independent of system size, so overall this represents approximately 7%–10% of the time for a full calculation (given the setup and initialization time). Execution times are somewhat reduced at fixed $P=64$ with the new system, in particular the fraction devoted to sparse algebra tasks.

More significant, though, is the improved scaling to larger numbers of processors, as shown in Fig. 5(b), which shows the speedup over 64 cores achievable on 128, 192, and 256 cores for both approaches. The scaling is seen to be significantly improved with the current approach, increasing the scale of calculations that can feasibly be performed. Note that the speedup is normalized to the time on 64 processors for that approach (old or new), so any improvement in the speedup is gained on top of the improvement at fixed $P=64$ shown in Fig. 5(a).

To illustrate more clearly the scaling with increasing system size N and process count P, we focus on one particular test system: the DNA strand. This is an isolated nonperiodic system, typical of problems studied in computational biochemistry, and an ideal application of linear-scaling methods. The systems simulated comprise straight strands of deoxyribonucleic acid, consisting of variable numbers of base-pairs AT, TA, CG and GC. The sequences were generated ran-

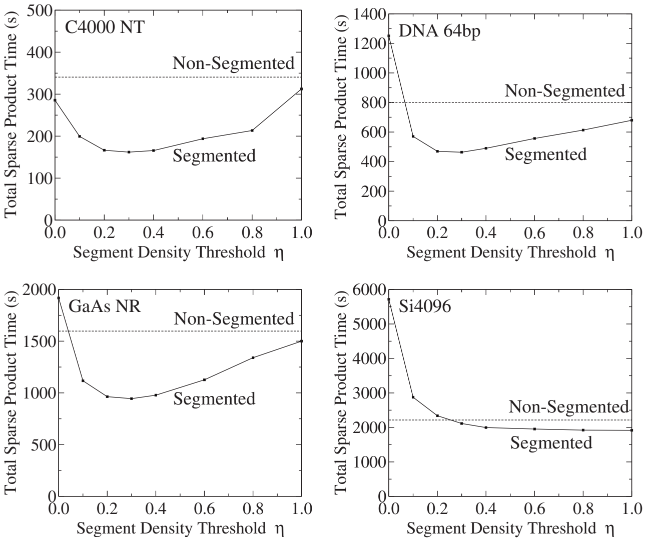

FIG. 4. Timings for the sparse algebra operations of a typical set of runs as a function of η. A minimum is observed in most cases for around $\eta=0.3-0.4$. The dotted line shows the time taken to run with the same parallel communications algorithm, but without the division into segments, so describing the whole matrix in sparse block-indexed form. In all cases, notable speedup can be obtained, with best performance for the more linear systems (nanotube, DNA, and nanorod).

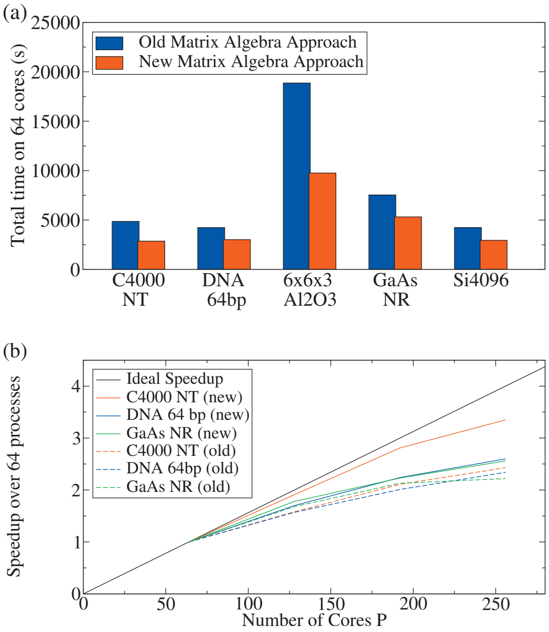

FIG. 5. (a) Total timings for one NGWF iteration for a range of typical systems, on $P=64$ parallel processes. (b) Speedup $T(P)/T(64)$ achieved on P processes relative to time on $P=64$ processes with the new (solid lines) and old (dashed lines) approaches to sparse algebra.

domly, with atom positions created with the NUCLEIC ACID BUILDER[24] code, then phosphate groups neutralized by adding Na^+ ions to the system with the AMBER code.[25] Finally, the positions were relaxed within an empirical potential framework, again using AMBER. This generated a starting point for DFT where the forces on the atoms were tolerably low.

In Fig. 6, we show the wall-clock time for a full single-point energy calculation of strands of increasing length, all run on the same number of cores ($P=256$). Clear linear-scaling behavior is seen over the range from 2091 to 16 775 atoms.

Figure 7, on the other hand, shows timings for a system of fixed size (64 base-pairs, 4184 atoms) as the number of processes is increased. One can see that as P increases, T initially falls as $1/P$ but eventually the parallel efficiency decreases as the number of atoms per core falls to below around 50. For this system, it is possible to go up to around 256 cores before notable inefficiency is observed. For larger systems, this maximum would increase.

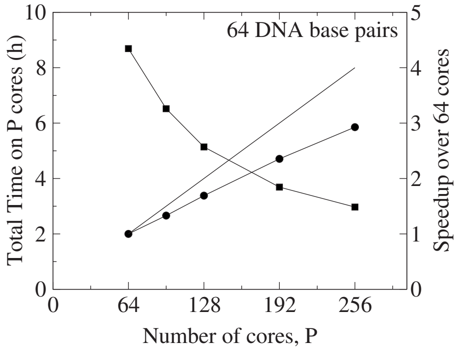

FIG. 7. Wall-clock time (left scale, squares) and speedup over 64 cores (right scale, circles) for a full total energy calculation of a 64 base-pair strand of DNA (4182 atoms) on varying numbers of parallel cores ($P=64$, 96, 128, 192, 256).

Finally, Fig. 8 represents the most insightful test of parallel efficiency for real-world usage: here we compare total execution times for an N-atom simulation on P processes where the ratio N/P is kept constant, in that as the size of the system increases, the number of cores the simulation is run on increases proportionally. This corresponds to demanding a constant, feasible time-to-science, given larger and larger problems on correspondingly larger parallel computers. If one wishes to obtain completed simulations in feasible wall-clock time by increasing the number of processes without limit, then this execution time must be able to remain roughly constant.

As the nearly flat line of Fig. 8 shows, this goal is very nearly achieved with the ONETEP approach in the 64–256 processor regime. To take advantage of scaling to larger systems through the regime of so-called "terascale" computing

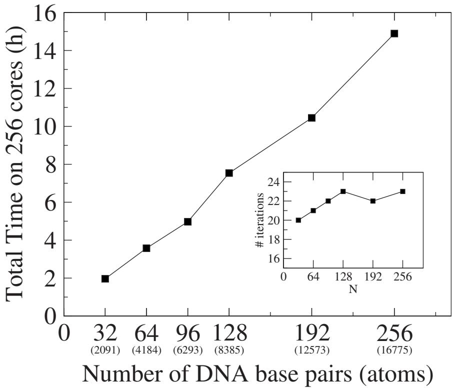

FIG. 6. Wall-clock time for strands of random DNA of increasing numbers of base-pairs (12, 24, 48, and 96), hence increasing numbers of atoms, with $P=256$. The inset shows the number of iterations required for the convergence of the NGWF optimization. There is minor random variation with size, including an apparent upward trend at low N, but over a wider range of sizes, there does not appear to be a systematic increase in number of iterations with system size.

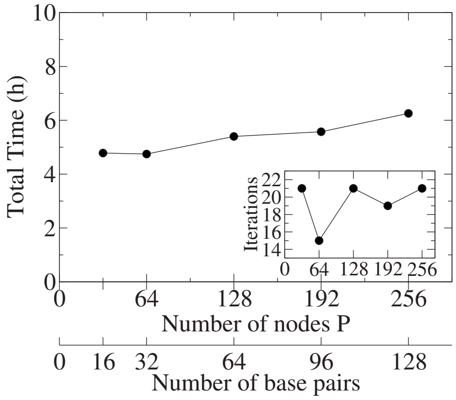

FIG. 8. Wall-clock time for 16, 32, 64, and 128 base-pair strands of DNA executed on 32, 64, 128, and 256 cores, respectively, so as to keep the ratio N/P constant (the number of atoms per core). Wall-clock time does not increase very greatly—less than 30% over nearly an order of magnitude in N. Inset: number of NGWF iterations required for convergence of the total energy, which varies slightly between different size systems due to random variation in the difficulty of attaining convergence for different base-pair sequences.

TABLE II. Supercell size in terms of number of atoms N, Madelung energy v_M, total energies of perfect and defect supercells, and defect formation energies (in electron volts) for V_O^{2+} and V_{Al}^{3-} calculated with the ONETEP approach.

N	v_M (eV)	E_{perf} (eV)	$E_{def}(V_O^{2+})$ (eV)	$E_{def}(V_{Al}^{3-})$ (eV)	$\Delta E_f(V_O^{2+})$ (eV)	$\Delta E_f(V_{Al}^{3-})$ (eV)
120	3.80	−34 356.18	−33 937.82	−34 259.99	4.39	1.35
960	1.90	−274 861.26	−274 442.68	−274 763.72	4.60	2.70
3240	1.27	−927 660.84	−927 243.13	−927 563.83	4.65	3.08

(10^{12} floating point operations/s) and toward the petascale regime (10^{15} flop/s) efficiently, codes must be able to maintain this scaling from to tens of thousands to hundreds of thousands of cores. This remains a significant challenge for high-accuracy LS-DFT approaches.

IV. APPLICATION TO DEFECTS IN α-Al_2O_3

The increased capabilities of this approach to sparse algebra enable us to calculate the formation energy of charged vacancies in alumina in simulation cells of significantly increased size. Formation energies of charged defects in metal oxides are strongly affected by finite size effects on the electrostatic and elastic energy, when simulated in periodic supercells (see Ref. 2 and references therein). The large cell sizes accessible to calculations with ONETEP allow us to address these finite size effects directly by calculating formation energies using a series of larger simulation cells and extrapolating to infinite size.[26]

The primitive cell of α-Al_2O_3 is trigonal, containing two formula units (ten atoms), but this is a poor choice for defect calculations as the spacing between periodic images of the defect falls slowly with increasing simulation cell size due to the elongated nature of the cell. A better approach is to form the equivalent 30-atom hexagonal cell, repeat it $2\times2\times1$ times, and thus form the 120-atom orthorhombic cell. One can then construct even larger simulation cells by repeating the 120-atom cell: doubling it along each lattice vector creates a 960-atom cell and tripling it creates a 3240-atom cell. Traditional plane-wave DFT would operate efficiently at the scale of the 120-atom cell. However, since the memory and computational time requirements scale as $O(N^2)$ and $O(N^3)$, respectively, the method would struggle with the 960-atom cell, and the 3240-atom cell would be unfeasible even on state-of-the-art hardware. Note that when using high-accuracy norm-conserving pseudopotentials for oxygen, the G-vector grid required to converge the total energy and forces must be fairly fine ($E_{cut}=1200$ eV was used here), hence the large memory requirements.

Limited to one or two points, it is impossible, or at best highly unreliable, to attempt an extrapolation to infinite size. Alternate approaches to this type of extrapolation have been proposed[2] which can work with independent scaling of the cell along different lattice vectors. However, the validity of this latter approach is yet to be verified directly and, in any case, the method is only fully reliable in systems where the defect is strongly localized and has negligible multipole moments beyond the monopole term.

Using the new approach to sparse algebra and nonlocal projectors described here, it becomes feasible to simulate the 960-atom and 3240-atom systems directly, running full geometry optimizations within comparatively modest computational requirements. We can then perform a direct extrapolation to infinite cell size of the formation energy as a function of $1/L$ based on the first three achievable system sizes. Full geometry relaxation is performed for each defect, starting from perfect cell positions with an atom removed near the center of the cell.

Defect formation energies are calculated using the approach referred to as the Zhang–Northrup formalism,[27] following Ref. 2. We take representative values (specific to the choice of pseudopotential and functional) of chemical potentials for oxygen atoms and aluminum atoms from Ref. 2. These are $\mu_O=433.13$ eV an $\mu_{Al}=-66.11$ eV. The specific choices of these chemical potentials do not affect the convergence of the formation energy with system size since they are simply bulk properties depending on the material and chosen formation conditions, namely, the partial pressure p_{O_2} and temperature T of the oxygen atmosphere with which the material is contact during annealing. These choices represent $T=1750$ K and $p_{O_2}=0.2$ atm. The electron chemical potential is determined by the requirement of overall charge neutrality; for the sake of displaying representative values during the extrapolation to infinite size, we shall simply arbitrarily set μ_e at 1.5 eV above the valence band edge at $E_{VBM}=7.84$ eV.

Table II shows the total energies of the bulk supercell at different cell sizes, the total energies of equivalent supercells containing V_O^{2+} defects V_{Al}^{3-} defects, and the defect formation energies of each. The formation energy of the V_O^{2+} defect is

TABLE III. Supercell size, Madelung energy v_M, total energies of perfect and defect supercells, and defect formation energies (in electron volts) for V_O^{2+} and V_{Al}^{3-} calculated using the CASTEP plane-wave DFT package.

N	v_M (eV)	E_{perf} (eV)	$E_{def}(V_O^{2+})$ (eV)	$E_{def}(V_{Al}^{3-})$ (eV)	$\Delta E_f(V_O^{2+})$ (eV)	$\Delta E_f(V_{Al}^{3-})$ (eV)
120	3.80	−34 357.04	−33 938.69	−34 260.82	4.38	1.38
960	1.90	−274 857.27	−274 438.71	−274 759.79	4.58	2.64

$$E_f(V_{\rm O}^{2+},L) = E_{\rm def}(L) - E_{\rm perf}(L) + \mu_{\rm O} + 2\mu_e, \tag{3}$$

while that of the aluminum vacancy is

$$E_f(V_{\rm Al}^{3-},L) = E_{\rm def}(L) - E_{\rm perf}(L) + \mu_{\rm Al} - 3\mu_e. \tag{4}$$

For comparison, Table III shows the same calculations repeated using the CASTEP code[28] for the smaller two cells. CASTEP is a traditional cubic-scaling plane-wave pseudopotential DFT code. Within CASTEP, it is possible to choose exactly the same pseudopotential, exchange-correlation functional, and grid spacings so as to as near as possible match those used in the ONETEP calculation. The CASTEP calculation remains faster than the ONETEP calculation at 120 and 960 atoms (though for the latter, the two are becoming comparable) but becomes unfeasible in terms of both memory and computational time well before 3240 atoms, whereas ONETEP is into the linear-scaling regime by this point. As discussed previously,[29] the fact that in plane-wave codes such as CASTEP, orbitals are expanded in a plane-wave basis cutoff on a sphere in reciprocal space, whereas ONETEP is effectively using a full rectangular grid, means that the ONETEP results are effectively obtained at a higher cutoff and hence return a slightly lower total energy. Despite this, the formation energies—which are energy differences—agree well between the two approaches: agreement to better than 0.1 eV is seen in all cases.

The sheer scale of each supercell total energy (nearing 10^6 eV in the largest cases) emphasizes the level of accuracy required to obtain the defect formation energies accurately at large system sizes. The systematic behavior of the defect formation energy with system size L demonstrates that the energies are sufficiently well-converged to be used reliably in an energy difference such as Eqs. (3) and (4) and that the cancellation of error between large similar systems is operating beneficially.

Figure 9(a) shows $\Delta E_{\rm f}(V_{\rm O}^{2+})$ as a function of $v_M \propto 1/L$, while Fig. 9(b) shows the formation energy $E_f(V_{\rm Al}^{3-})$ of the $V_{\rm Al}^{3-}$ defect. Also shown is a linear fit to v_M, under the assumption that the dominant term in the finite size error is a monopole-monopole term. Such an assumption can be shown to give[30,31]

$$\Delta E_{\rm FS} = \frac{q^2 v_M}{2\epsilon_{\rm fit}}, \tag{5}$$

where $\epsilon_{\rm fit}$ is a fitting parameter dependent on the defect. This fitting parameter is loosely related to the static dielectric constant of the material, but varies significantly between different defects even in the same material, motivating the need for a fit and extrapolation rather than a single-shot correction of a result for a small supercell.

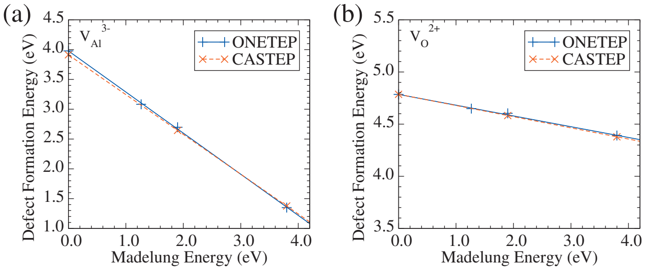

FIG. 9. Scaling of defect formation energy against Madelung energy for (a) aluminum vacancy $V_{\rm Al}^{3-}$ and (b) oxygen vacancy $V_{\rm O}^{2+}$. The Madelung energy scales as $1/L$, so the extrapolation to $v_M \rightarrow 0$ represents extrapolation to the infinite dilution limit.

TABLE IV. Mean distances from the vacancy site to the shells of nearest-neighbors of $V_{\rm O}^{2+}$ as system size increases. Nearest-neighbors (1NN) are four Al ions, second nearest-neighbors (2NN) are 12 O ions, and 3NN are six Al ions. In the smallest cell, the 3NN ions are constrained by the artificial periodicity to hardly move from their bulk positions. Increasing the cell size allows them to relax, but the relaxations are slow to converge to their infinite cell-size limit.

System	$r_{\rm 1NN}(a_0)$	$r_{\rm 2NN}(a_0)$	$r_{\rm 3NN}(a_0)$
Perfect crystal	3.5582	5.0666	6.3415
$V_{\rm O}^{2+}$ in 120-atom cell	4.0030	4.9653	6.3444
$V_{\rm O}^{2+}$ in 960-atom cell	4.0006	4.9572	6.3483
$V_{\rm O}^{2+}$ in 3240-atom cell	4.0116	4.9773	6.3652

While there are not enough data to perform statistical analysis, the combination of a close agreement between the linear fit with the data, plus the good agreement between the ONETEP results and the available CASTEP results, suggests both that the model is accurate and that the ONETEP results are sufficiently well-converged to give accurate results in these very large systems. The resulting extrapolated defect formation energies, at these chosen values of chemical potential, give $\Delta E_f(V_{\rm O}^{2+}) = 4.78$ eV and $E_f(V_{\rm Al}^{3-}) = 3.97$ eV.

Notably, if the uncorrected value from the $N=120$ supercell had been used, the resulting finite size error on the formation energies would have been 0.4 eV and 2.6 eV, respectively. Similarly, as shown in Table IV, while the bond lengths in the region immediately around the defect site [the first nearest-neighbor (1NN) shell] are close to convergence at $N=120$, those far away are barely different from their bulk values since the artificial symmetry imposed by the periodic boundary conditions prevents the relaxation the defect would normally induce. It can be seen that only by going to the largest system size do the third nearest-neighbor (3NN) distances begin to change significantly from their bulk values.

By assuming the simplest form of disorder involving these two defect species, namely, Schottky equilibrium, we can make a very rough estimate of the dependence of defect concentrations on the simulation cell size through the finite size errors on the formation energy. We make the crude approximation of replacing the temperature-dependent free energy with the 0 K DFT total energy. We then relate the vacancy concentrations per formula unit of Al_2O_3 at temperature T to the formation energies through

$$[V_{\rm O}^{2+}] = 3{\rm e}^{-\Delta E_f(V_{\rm O}^{2+})/(kT)},$$

$$[V_{\rm Al}^{3-}] = 2{\rm e}^{-\Delta E_f(V_{\rm Al}^{3-})/(kT)}.$$

Assuming perfect Schottky equilibrium and overall charge neutrality, we then have

TABLE V. Dependence of the estimated concentrations of oxygen and aluminum vacancies per formula unit on system size through finite size errors on formation energies.

System	μ_e (eV)	$[V_{\mathrm{O}}^{2+}]$ per f.u.	$[V_{\mathrm{Al}}^{3-}]$ per f.u.
120-atom cell	$E_{\mathrm{VBM}}+1.15$	1.56×10^{-9}	2.34×10^{-9}
960-atom cell	$E_{\mathrm{VBM}}+1.39$	2.03×10^{-11}	3.05×10^{-11}
3240-atom cell	$E_{\mathrm{VBM}}+1.45$	5.61×10^{-12}	8.41×10^{-12}
Infinite-size extrapolation	$E_{\mathrm{VBM}}+1.60$	3.12×10^{-13}	4.68×10^{-13}

$$3[V_{\mathrm{O}}^{2+}]=2[V_{\mathrm{Al}}^{3-}]$$

and hence we can estimate the position of the electron chemical potential which gives charge neutrality as

$$\mu_e=\tfrac{1}{5}\left(E_{\mathrm{def}}(V_{\mathrm{Al}}^{3-})-E_{\mathrm{def}}(V_{\mathrm{O}}^{2+})+\mu_{\mathrm{Al}}-\mu_{\mathrm{O}}+kT\ln\tfrac{4}{9}\right).$$

Finally, therefore, we can estimate the effect of the finite size errors on the concentrations one would calculate using the uncorrected results as a function of cell size. This is shown in Table V for a typical annealing temperature ($T=1750$ K). An error of four orders of magnitude is seen to result from the finite size effect at the smallest cell size compared to its infinite-size extrapolation. Note that these concentrations likely represent quite a significant underestimate of the real concentrations under similar conditions due to the neglect of the vibrational contribution to the free energy.

The slow convergence of all these properties, and the very large finite size errors on small supercells, more than justify the need for the combination of large supercells and extrapolation to the infinite limit. Furthermore, more complex defects such as clusters of intrinsic defects and substitutional dopants do not behave as point charges and must be treated in even larger simulation cells to accurately remove the effect of finite size errors. The need for a linear-scaling formalism in the study of defects in such materials is therefore clear.

V. CONCLUSION

We have described a number of advances to the methodology underlying linear-scaling density-functional theory, implemented here in the ONETEP code. We have presented a unified approach to sparse algebra suited to the calculation of the sparse matrix product operations typically encountered in linear-scaling electronic structure theory, suitable across a very wide range of system sizes and types, and which scales well over a wide range of numbers of parallel processes. The segments described, which are the segments of the rows associated with a given parallel process belonging to the columns of a second given process, form a natural second level of hierarchy over and above the atom-blocks that emerge naturally from a basis consisting of atom-centered localized functions. These segments allow use of optimally sized dense algebra to increase performance which, along with a framework for significant reduction in the communication, demands of sparse matrix algebra.

We have applied this new scheme to a challenging calculation of the size-convergence of the formation energy of charged point defects in alumina (α-Al_2O_3). Alumina is a challenging material for first-principles simulation since its combination of a low-symmetry structure with high ionic charges means that large simulation cells must be used when calculating properties of systems with localized charge, such as point defects. The combination provided by the ONETEP formalism, of high accuracy, equivalent to that of the plane-wave approach, with linear-scaling computational effort, allowing access to system sizes in the thousands of atoms, has been shown to be sufficient to provide converged results for the formation energies of intrinsic vacancies in this material.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Engineering and Physical Sciences Research Council (EPSRC Grant No. EP/G055882/1) for funding through the HPC Software Development program. P.D.H. acknowledges the support of a University Research Fellowship from the Royal Society. A.A.M. acknowledges the support of the RCUK fellowship program. The authors are grateful for the computing resources provided by Imperial College's High Performance Computing service (CX2), which has enabled all the simulations presented here.

APPENDIX: NONLOCAL PSEUDOPOTENTIAL MATRICES

The sparse algebra algorithms presented in this work have been designed to treat matrices with different blocking schemes (in terms of numbers of elements per block) for rows and columns on equal footing and even mix the two. This enables additional parts of the calculation to be treated with distributed sparse matrix algorithms.

In standard Local Density Approximation (LDA) or Generalized Gradient Approximation (GGA) calculations, the Hamiltonian is composed of the contributions $\hat{H}=\hat{T}+\hat{V}_{\mathrm{Hlxc}}+\hat{V}_{\mathrm{nl}}$, where $\hat{T}$ is kinetic energy, $\hat{V}_{\mathrm{Hlxc}}$ is the total local potential (Hartree, exchange-correlation and local ionic pseudopotential contributions), and $\hat{V}_{\mathrm{nl}}$ is the nonlocal ionic pseudopotential expressed in standard Kleinman–Bylander form[32] as a sum over nonlocal projectors. These three operators correspond to sparse matrices $T_{\alpha\beta}$, $V_{\alpha\beta}^{\mathrm{Hlxc}}$, and $V_{\alpha\beta}^{\mathrm{nl}}$ in the NGWF basis, given by

$$T_{\alpha\beta}=\langle\phi_\alpha|-\tfrac{1}{2}\nabla^2|\phi_\beta\rangle,$$

$$V_{\alpha\beta}^{\mathrm{Hlxc}}=\langle\phi_\alpha|\hat{V}_{\mathrm{Hlxc}}|\phi_\beta\rangle,$$

$$V_{\alpha\beta}^{\mathrm{nl}}=\sum_{i=1}^{N_{\mathrm{proj}}}\frac{\langle\phi_\alpha|\chi_i\rangle\langle\chi_i|\phi_\beta\rangle}{D_i},$$

where $|\chi_i\rangle$ are the nonlocal pseudopotential projectors, D_i are the Kleinman-Bylander denominators, and N_{proj} is the total number of projectors in the system.

Calculation of the overlaps $\langle\phi_\alpha|\chi_i\rangle$ between NGWFs and projectors is performed in ONETEP using the FFT-box approach.[33] The FFT-box is a localized box surrounding the atom on which NGWF ϕ_α is located, of size typically of side length $6R_\phi$, where R_ϕ is the largest NGWF radius required in

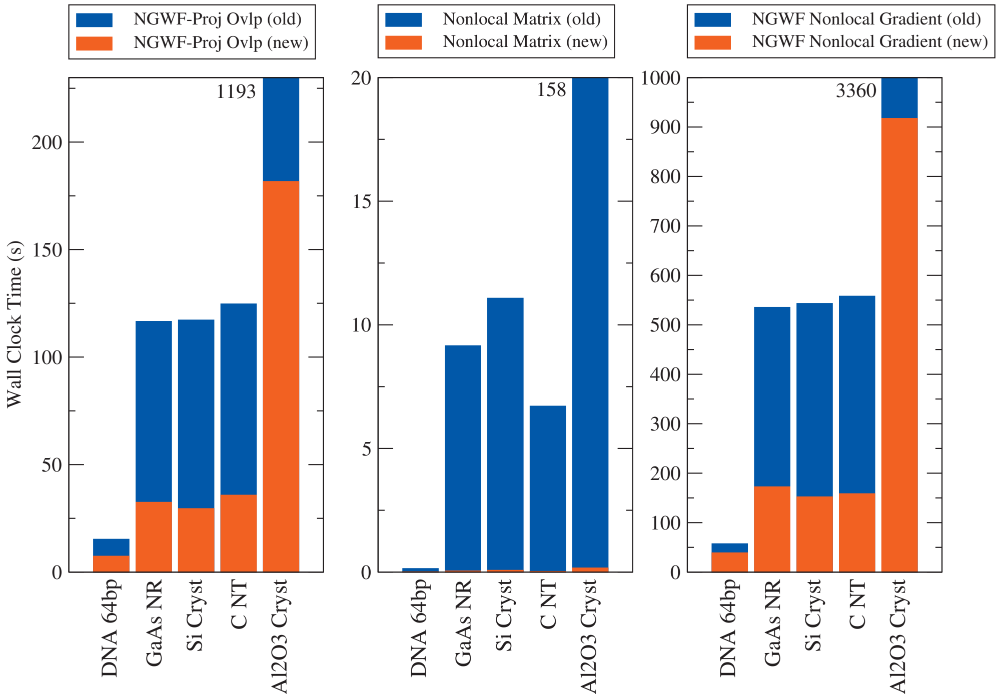

FIG. 10. Timings for routines evaluating (a) overlap matrices of nonlocal pseudopotential projectors and NGWFs, (b) nonlocal pseudopotential contribution to Hamiltonian, and (c) nonlocal pseudopotential contribution to NGWF gradient. The systems tested are described in Table I and all runs are on all 64 cores of 16 Intel Core i7 processors. Blue bars show the times without parallelization of the matrices required and red bars show the times using the new scheme. Significant reduction in wall-clock time is obtained for all systems on all routines, especially for the nonlocal matrix, which is rendered negligible (all timings are under 1 s using the new scheme). The most demanding system is the densely overlapping oxide Al_2O_3, as this combines a high cutoff energy with large spheres and many projectors per atom, so these results are off the displayed scale.

the system. For strictly localized functions such as NGWFs and nonlocal pseudopotential projectors, the FFT-box allows the advantages of the use of Fourier transforms in plane-wave DFT to be carried over to the linear-scaling formalism.

To calculate $\langle\phi_\alpha|\chi_i\rangle$ one evaluates the projector in reciprocal space by interpolating $\chi_i(q)$ on to the grid points $\mathbf{G}_{\mathrm{FFT}}$ of the reciprocal-space FFT-box; then one performs the Fourier transform on the FFT-box to obtain $\chi_i(\mathbf{r})$; finally, the points of $\chi_i(\mathbf{r})$ which overlap points of $\phi_\alpha(\mathbf{r})$ are extracted from the FFT-box and used to find the overlap $\langle\phi_\alpha|\chi_i\rangle$.

Given that the relevant NGWF data for ϕ_α will generally not reside on the same process as the projector data, there are two possible approaches to the communications required for this algorithm to work in parallel: either one could recreate each projector on every parallel process holding a NGWF overlapping that projector, or one could generate each projector once (on the process which holds its atom) and then communicate each NGWF overlapping that projector from the processes which hold them. In practice, the latter allows a large saving in computational effort as long as the communications overhead of NGWF communication is less than the computational time of generating the projectors many times over on different processes.

We therefore use the latter approach to generate the block-indexed sparse matrix $P_{\alpha i}=\langle\phi_\alpha|\chi_i\rangle$, whose columns i correspond to projector kets $|\chi_i\rangle$ and whose rows α correspond to NGWF bras $\langle\phi_\alpha|$. From this matrix, it is trivial to also form $R_{i\beta}=\langle\chi_i|\phi_\beta\rangle$, the transpose of $P_{\alpha i}$. Then, using the aforementioned sparse product algorithm, one can calculate the nonlocal matrix as

$$V^{\mathrm{nl}}_{\alpha\beta}=\sum_i P_{\alpha i}D_i^{-1}R_{i\beta}.$$

A similar representation can be used to generate the nonlocal pseudopotential contribution to the NGWF gradient efficiently. The nonlocal pseudopotential contribution to the band structure energy E_{nl} can be written in terms of the nonlocal matrix as a trace of its product with the density kernel, as $E_{\mathrm{nl}}=V^{\mathrm{nl}}_{\beta\alpha}K^{\alpha\beta}$. To optimize the NGWFs, we require the gradient of this quantity with respect to changes in the value of the NGWF ϕ_α at position $\mathbf{r}$, so we have

$$\frac{\delta E_{\mathrm{nl}}}{\delta\phi_\alpha(\mathbf{r})}=2\sum_{i=1}^{N_{\mathrm{proj}}}\chi_i(\mathbf{r})\sum_\beta\frac{\langle\chi_i|\phi_\beta\rangle K^{\beta\alpha}}{D_i}=2\sum_{i=1}^{N_{\mathrm{proj}}}\chi_i(\mathbf{r})Q_i^\alpha. \tag{A1}$$

Again, $Q_i^\alpha=D_i^{-1}R_{i\beta}K^{\beta\alpha}$ can be constructed efficiently through the use of the above sparse product algorithm.

Furthermore, since all the projectors χ_i which contribute to the gradient for a given ϕ_α need to be constructed in the same FFT-box centered on the atom of ϕ_α, another improvement is possible. Note that in any system, there are generally only a small number of different types of projector N_{sp}, since for each atom type, there is a set of nonlocal channels with angular momentum values $l=0,\ldots,l_{\max}$ and for each channel, azimuthal angular momentum values $m=-l,\ldots,l$.

For each projector species s, we denote by $\chi_s(\mathbf{G})$ the projector evaluated in reciprocal space for a projector at the origin in real space. To translate it to its correct position within the FFT-box of ϕ_α, a phase shift of $e^{i\mathbf{G}\cdot(\mathbf{R}_i-\mathbf{R}_\alpha)}$ is applied in reciprocal space, where $\mathbf{R}_i$ is the position of the atom of projector i and $\mathbf{R}_\alpha$ is the position of the atom of ϕ_α.

We can write Eq. (A1) in terms of a sum over species and over projectors of that species, making the Fourier transform required to construct $\chi_i(\mathbf{r})$ explicit,

$$\frac{\partial E_{\mathrm{nl}}}{\partial \phi_\alpha(\mathbf{r})} = 2\mathcal{F}\left[\sum_{s=1}^{N_{\mathrm{sp}}}\left(\sum_{i=1}^{N_{\mathrm{proj}}(s)} Q_i^\alpha e^{i\mathbf{G}\cdot\mathbf{t}_{i\alpha}}\right)\chi_s(\mathbf{G})\right]. \tag{A2}$$

The term inside the inner brackets is in effect a reciprocal-space structure factor for each projector species, evaluated on the reciprocal-space grid of the FFT-box. Constructing this structure factor first and then multiplying it by the projector in reciprocal space allows one to avoid the work of the large number of multiply-add operations of the whole projector in real space.

Figure 10 shows the computational time required for evaluation of $\langle\phi_\alpha|\chi_i\rangle$, $V_{\alpha\beta}^{\mathrm{nl}}$, and $\partial E_{\mathrm{nl}}/\partial\phi_\alpha(\mathbf{r})$ in the test systems. Figure 10(a) shows the timings with the previous unparallelized approach, while Fig. 10(b) shows timings with the system just described. The computational effort saved is dramatic, particularly in large solid systems with large numbers of densely overlapping NGWFs and projectors. The approach is also very much more parallelizable. Figure 10(c) shows the total time for evaluation of each of the above quantities for the 960-atom alumina system on 32, 64, 128, and 256 cores, demonstrating near-ideal scaling with $1/P$.

[1] M. I. J. Probert and M. C. Payne, Phys. Rev. B **67**, 075204 (2003).

[2] N. D. M. Hine, K. Frensch, W. M. C. Foulkes, and M. W. Finnis, Phys. Rev. B **79**, 024112 (2009).

[3] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Rev. Mod. Phys. **64**, 1045 (1992).

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

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

[6] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, J. Phys.: Condens. Matter **14**, 2745 (2002).

[7] D. R. Bowler, R. Choudhury, M. J. Gillan, and T. Miyazaki, Phys. Status Solidi B **243**, 989 (2006).

[8] J. VandeVondele, M. Krack, F. Mohammed, M. Parrinello, T. Chassaing, and J. Hutter, Comput. Phys. Commun. **167**, 103 (2005).

[9] J.-L. Fattebert and J. Bernholc, Phys. Rev. B **62**, 1713 (2000); J.-L. Fattebert and F. Gygi, *ibid.* **73**, 115124 (2006).

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

[11] T. Ozaki and H. Kino, Phys. Rev. B **72**, 045121 (2005); See http://www.openmx-square.org for details of the *OPENMX* code.

[12] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, J. Chem. Phys. **122**, 084119 (2005).

[13] N. D. M. Hine, P. D. Haynes, A. A. Mostofi, C.-K. Skylaris, and M. C. Payne, Comput. Phys. Commun. **180**, 1041 (2009).

[14] C.-K. Skylaris, A. A. Mostofi, P. D. Haynes, O. Diéguez, and M. C. Payne, Phys. Rev. B **66**, 035119 (2002).

[15] A. A. Mostofi, P. D. Haynes, C.-K. Skylaris, and M. C. Payne, J. Chem. Phys. **119**, 8842 (2003).

[16] P. D. Haynes, C.-K. Skylaris, A. A. Mostofi, and M. C. Payne, J. Phys.: Condens. Matter **20**, 294207 (2008).

[17] C.-K. Skylaris and P. D. Haynes, J. Chem. Phys. **127**, 164712 (2007).

[18] E. H. Rubensson, E. Rudberg, and P. Salek, J. Comput. Chem. **28**, 2531 (2007).

[19] C. Saravanan, Y. Shao, R. Baer, P. N. Ross, and M. Head-Gordon, J. Comput. Chem. **24**, 618 (2003).

[20] D. R. Bowler, T. Miyazaki, and M. J. Gillan, Comput. Phys. Commun. **137**, 255 (2001).

[21] M. Challacombe, Comput. Phys. Commun. **128**, 93 (2000).

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

[23] S. Itoh, P. Ordejón, and R. M. Martin, Comput. Phys. Commun. **88**, 173 (1995).

[24] T. Macke and D. A. Case, "Modeling unusual nucleic acid structures," in *Molecular Modeling of Nucleic Acids*, edited by J. N. B. Leontes and J. Santa Lucia (American Chemical Society, Washington, DC, 1998).

[25] D. A. Case, T. A. Darden, T. E. Cheatham III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, R. C. Walker, W. Zhang, K. M. Merz, B. P. Roberts, B. Wang, S. Hayik, A. Roitberg, G. Seabra, I. Kolossváry, K. F. Wong, F. Paesani, J. Vanicek, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M.-J. Hsieh, G. Cui, D. R. Roe, D. H. Mathews, M. G. Seetin, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, and P. A. Kollman, AMBER 11, University of California, San Francisco, 2010.

[26] C. W. M. Castleton, A. Höglund, and S. Mirbt, Phys. Rev. B **73**, 035215 (2006).

[27] S. B. Zhang and J. E. Northrup, Phys. Rev. Lett. **67**, 2339 (1991).

[28] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. J. Probert, K. Refson, and M. C. Payne, Z. Kristallogr. **220**, 567 (2005).

[29] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, J. Phys.: Condens. Matter **17**, 5757 (2005).

[30] M. Leslie and M. J. Gillan, J. Phys. C **18**, 973 (1985).

[31] G. Makov and M. C. Payne, Phys. Rev. B **51**, 4014 (1995).

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

[33] C.-K. Skylaris, A. A. Mostofi, P. D. Haynes, C. J. Pickard, and M. C. Payne, Comput. Phys. Commun. **140**, 315 (2001).