AIP | The Journal of Chemical Physics

Graph-based linear scaling electronic structure theory

Anders M. N. Niklasson, Susan M. Mniszewski, Christian F. A. Negre, Marc J. Cawkwell, Pieter J. Swart, Jamal Mohd-Yusof, Timothy C. Germann, Michael E. Wall, Nicolas Bock, Emanuel H. Rubensson, and Hristo Djidjev

Citation: The Journal of Chemical Physics **144**, 234101 (2016); doi: 10.1063/1.4952650
View online: http://dx.doi.org/10.1063/1.4952650
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/144/23?ver=pdfcov
Published by the AIP Publishing

Graph-based linear scaling electronic structure theory

Anders M. N. Niklasson,[1,a)] Susan M. Mniszewski,[2] Christian F. A. Negre,[1]
Marc J. Cawkwell,[1] Pieter J. Swart,[1] Jamal Mohd-Yusof,[2] Timothy C. Germann,[1]
Michael E. Wall,[2] Nicolas Bock,[1] Emanuel H. Rubensson,[3] and Hristo Djidjev[2]
[1]*Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[2]*Computer, Computational, and Statistical Sciences Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA*
[3]*Division of Scientific Computing, Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden*

(Received 24 March 2016; accepted 5 May 2016; published online 15 June 2016)

We show how graph theory can be combined with quantum theory to calculate the electronic structure of large complex systems. The graph formalism is general and applicable to a broad range of electronic structure methods and materials, including challenging systems such as biomolecules. The methodology combines well-controlled accuracy, low computational cost, and natural low-communication parallelism. This combination addresses substantial shortcomings of linear scaling electronic structure theory, in particular with respect to quantum-based molecular dynamics simulations. [http://dx.doi.org/10.1063/1.4952650]

I. INTRODUCTION

The importance of electronic structure theory in materials science, chemistry, and molecular biology relies on the development of theoretical methods that provide sufficient accuracy at a reasonable computational cost. Currently, the field is dominated by Kohn-Sham density functional theory,[1–4] which often combines good theoretical fidelity with a modest computational workload that is constrained mainly by the diagonalization of the Kohn-Sham Hamiltonian—an operation that scales cubically with the system size. However, for systems beyond a few hundred atoms, the diagonalization becomes prohibitively expensive. This bottleneck was removed with the development of linear scaling electronic structure theory,[5,6] which allows calculations of systems with millions of atoms.[7,8] Unfortunately, the immense promise of linear scaling electronic structure theory has never been fully realized because of some significant shortcomings, in particular, (a) the accuracy is reduced to a level that is often difficult, if not impossible, to control; (b) the computational pre-factor is high and the linear scaling benefit occurs only for very large systems that in practice often are beyond acceptable time limits or available computer resources; and (c) the parallel performance is generally challenged by a significant overhead and the wall-clock time remains high even with massive parallelism. In quantum-based molecular dynamics simulations,[9] all these problems coalesce and we are constrained either to small system sizes or short simulation times.

In this paper we propose to overcome these shortcomings by introducing a formalism based on graph theory[10,11] that allows practical and easily parallelizable electronic structure calculations of large complex systems with well-controlled accuracy. The graph-based electronic structure theory combines the natural parallelism of a divide and conquer approach[12–17] with the automatically adaptive and tunable accuracy of a thresholded sparse matrix algebra,[18–31] which can be combined with fast, low pre-factor, recursive Fermi operator expansion methods[32–41] and can be applied to modern formulations of Born-Oppenheimer molecular dynamics.[42–50]

The article is outlined as follows: first we introduce the graph-based formalism for general sparse matrix polynomials expanded over separate subgraphs, thereafter we apply the methodology to the Fermi-operator expansion in electronic structure theory with demonstrations for a protein-like structure of polyalanine solvated in water, before analyzing applications in molecular dynamics simulations. At the end we give our conclusions.

II. GRAPH-BASED ELECTRONIC STRUCTURE THEORY

A. Expansions of thresholded sparse matrix polynomials

Our graph-based electronic structure theory relies on the equivalence between the calculation of thresholded sparse matrix polynomials and a graph partitioning approach. Let $P(X)$ be a Mth-order polynomial of a $N \times N$ symmetric square matrix X that is given as a linear combination of some basis polynomials $T^{(n)}(X)$,

$$P(X) = \sum_{n=0}^{M} c_n T^{(n)}(X). \tag{1}$$

We define an approximation $P_\tau(X)$ of $P(X)$ using a *globally* thresholded sparse matrix algebra, where matrix elements

a)amn@lanl.gov

with a magnitude below a numerical threshold τ *in all terms*, $T^{(n)}(X)$, are ignored. The pattern of the remaining matrix entries, which at any point of the expansion have been (or are expected to be) greater than τ, can be described by a *data dependency graph* $\mathcal{S}_\tau$ that represents all possible data dependencies between the matrix elements in the polynomial expansion. Formally, we define the graph S_τ with a vertex for each row of X and an edge (i,j) between vertices i and j if

$$\{T^{(n)}(X)\}_{i,j} \geq \tau \text{ for any } n \leq M. \tag{2}$$

For a matrix A, we denote by $\lfloor A \rfloor_{\mathcal{S}_\tau}$ the thresholded version of A, where

$$\{\lfloor A \rfloor_{\mathcal{S}_\tau}\}_{i,j} = \begin{cases} A_{i,j} & \text{if } (i,j) \text{ is an edge of } S_\tau \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The thresholded polynomial $P_\tau(X)$ of $P(X)$ with respect to $\mathcal{S}_\tau$ is given by

$$P_\tau(X) = \sum_{n=0}^{M} c_n T^{(n)}_{\mathcal{S}_\tau}(X), \tag{4}$$

where the thresholded $T^{(n)}_{\mathcal{S}_\tau}(X)$ can be calculated from a linear recurrence

$$T^{(n)}_{\mathcal{S}_\tau}(X) = \alpha_n \lfloor X T^{(n-1)}_{\mathcal{S}_\tau}(X) \rfloor_{\mathcal{S}_\tau} + \sum_{m=0}^{n-1} \alpha_m T^{(m)}_{\mathcal{S}_\tau}(X), \tag{5}$$

with $T^{(0)}_{\mathcal{S}_\tau}(X) = I$. A key observation of this paper is that the calculation of $P_\tau(X)$ in Eqs. (4) and (5) is equivalent to a partitioned subgraph expansion on $\mathcal{S}_\tau$. This approach is illustrated in Fig. 1. For any vertex i of $\mathcal{S}_\tau$, let s^i_τ be the subgraph of $\mathcal{S}_\tau$ induced by the *core* (meaning belonging to a single subgraph) vertex i and all *halo* (shared) vertices that are directly connected to i in $\mathcal{S}_\tau$. Then the ith matrix column of $P_\tau(X)$ is given by the thresholded expansion determined by s^i_τ only, i.e.,

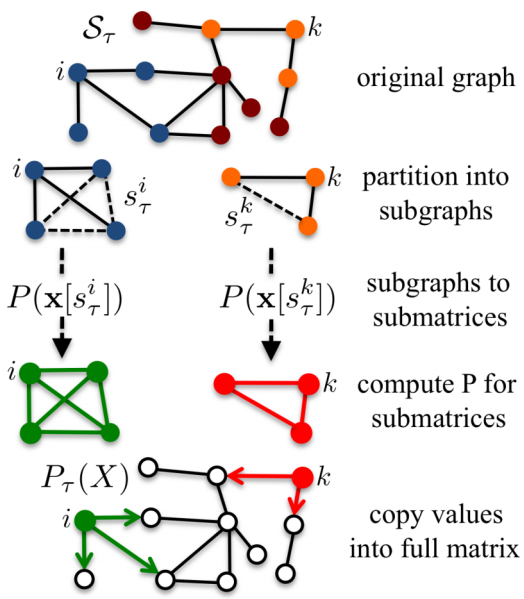

FIG. 1. The data dependency graph $\mathcal{S}_\tau$ and the subgraphs (s^i_τ or s^k_τ), one for each *core* vertex (i or k) including all directly connected *halo* vertices in $\mathcal{S}_\tau$. The full matrix polynomial $P_\tau(X)$ is given by an assembly from $P(\mathbf{x}[s^i_\tau])$ of the separate dense subgraph contractions $\mathbf{x}[s^i_\tau]$.

$$\{P_\tau(X)\}_{:,i} = \{P(\mathbf{x}[s^i_\tau])\}_{:,j}. \tag{6}$$

Here j is the column (or row) of the polynomial for the subgraph s^i_τ containing all edges from the core vertex i that corresponds to column i of the complete matrix polynomial on the left-hand side. $\mathbf{x}[s^i_\tau]$ is the small dense principal submatrix that contains only the entries of X corresponding to s^i_τ. The full matrix $P_\tau(X)$ can then be assembled, column by column, from the set of smaller dense matrix polynomials $P(\mathbf{x}[s^i_\tau])$ for each vertex i. The calculation of a numerically thresholded matrix polynomial $P_\tau(X)$ thus can be replaced by a sequence of fully independent small dense matrix polynomial expansions determined by a graph partitioning.

Equation (6) represents an exact relation between a globally thresholded sparse matrix algebra and a graph partitioning approach, which is valid for a general matrix polynomial $P(X)$, including all terms to any order. An explicit code example illustrating the equivalence is given in the supplementary material[76] and a more rigorous graph-theoretical proof will be published elsewhere.[61] Several observations can be made about this equivalence: (i) $P_\tau(X)$ is not symmetric and with the order of the matrix product for the threshold in Eq. (5) we collect $P_\tau(X)$ column by column in Eq. (6) as illustrated by the *directed* graph at the bottom of Fig. 1; (ii) the accuracy of the matrix polynomial increases (decreases) as the threshold τ is reduced (increased) and the number of edges of $\mathcal{S}_\tau$ increases (decreases); (iii) we may thus include additional edges in $\mathcal{S}_\tau$ without loss of accuracy; (iv) the polynomial $P_\tau(X)$ is zero at all entries outside of $\mathcal{S}_\tau$; (v) apart from spurious cancellations, the non-zero pattern of $P_\tau(X)$ is therefore the same as $\mathcal{S}_\tau$ and we can expect a numerically thresholded exact matrix polynomial, $\lfloor P(X) \rfloor_\tau$, to have a non-zero structure similar to $\mathcal{S}_\tau$; (vi) the graph partitioning can be generalized such that each vertex corresponds to a combined set of vertices, i.e., a community, without loss of accuracy; (vii) we may reduce the computational cost by identifying such communities using highly efficient off-the-shelf graph partitioning schemes that can be tailored for optimal platform-dependent performance; (viii) the exact relation given by Eqs. (4)–(6) holds for any structure of $\mathcal{S}_\tau$ and is not limited to the threshold in Eq. (2); (ix) the particular sequence of matrix operations in the calculation of $P_\tau(X)$ is of importance because of the thresholding in Eq. (5), whereas the order (or grouping) of the matrix multiplications is arbitrary for the contracted matrix polynomials $P(\mathbf{x}[s^i_\tau])$ in Eq. (6); and (x) the computational cost of each polynomial expansion is dominated by separate sequences of dense matrix-matrix multiplication that can be performed independently and in parallel.

B. Graph-based Fermi-operator expansion

A main point of this paper is that the equivalence between the calculation of the thresholded sparse matrix polynomial and the graph partitioned expansion in Eq. (6) provides a natural framework for a graph-based formulation of linear scaling electronic structure theory. In Kohn-Sham density functional theory, the matrix polynomial in Eq. (1) is replaced by the Fermi-operator expansion[3,51,52] where

$$P(H) = D = \left[e^{\beta(H-\mu)} + 1\right]^{-1} \approx \sum_{n=0}^{M} c_n T^{(n)}(H). \qquad (7)$$

Here D is the density matrix, H the Hamiltonian, μ the chemical potential, and β the inverse temperature. The matrix functions, $T^{(n)}(X)$, are typically Chebyshev polynomials constructed by a recurrence equation as in Eq. (5). With a local basis set, H and $P(H)$ have sparse matrix representations above some numerical threshold for sufficiently large non-metallic systems.[5,6] The graph-based construction of sparse matrix polynomials in Eq. (6) can then be applied to the calculation of the density matrix with the data dependency graph $\mathcal{S}_\tau$ estimated from an approximate prior density matrix that is available in an iterative self-consistent field (SCF) optimization or from previous time steps in a molecular dynamics simulation. The computation can be accelerated with a recursive Fermi-operator expansion.[32–37,39–41] In the zero temperature limit the Fermi function equals the Heaviside step function θ and a recursive expansion is then given by $D = \theta(\mu I - H) = \lim_{n\to\infty} f_n(f_{n-1}(\ldots f_0(H)\ldots))$, which reaches a high expansion order much more rapidly compared to the serial form in Eq. (1). With $f_n(X)$ being 2nd-order polynomials[35] we reach an expansion order of over a billion in only 30 iterations. The ability to use a fast recursive expansion is motivated from (ix) above, and since any recursive expansion also can be written in the general form of Eq. (1). Once the density matrix D is known, the expectation value of any operator A is given by $\langle A\rangle = Tr[DA]$. Generalizations to quantum perturbation theory are straightforward.[53,54]

The Fermi-operator expansion in Eq. (7) is based on an orthogonal representation of H and P(H). A generalization for a non-orthogonal expansion, $D' = P'(H')$, where the prime indicates a non-orthogonal basis set representation, is in principle straightforward. If Z is the inverse factor of the basis-set overlap matrix S such that $Z^T S Z = I$, then $D' = ZP(Z^T H Z)Z^T$. In our numerical test and analysis below, only orthogonal formulations are considered.

III. NUMERICAL TESTS AND ANALYSIS

A. Macromolecular test system

Figure 2 shows the error per atom in the density matrix of the band energy, $E_{\text{band}} = Tr[DH]$, calculated with the graph-based formulation above for a 19 945-atom macromolecular system of polyalanine solvated in water, Fig. 3 (see Appendix B). The calculations were performed using self-consistent charge density functional tight-binding theory[55–57] as implemented in the electronic structure program LATTE[58] in combination with the recursive second-order spectral projection (SP2) zero-temperature Fermi-operator expansion scheme.[35] The data dependency graphs, $\mathcal{S}_\tau$, were estimated by thresholding an "exact" density matrix with varying thresholds, τ. Different numbers of subgraph communities (512, 1024, or 2048) were chosen and optimized with the METIS heuristic multilevel graph partitioning package[59] for the different data dependency graphs (one for each threshold) using the multilevel recursive bisection method. The errors were determined in comparison to the "exact" density matrix,

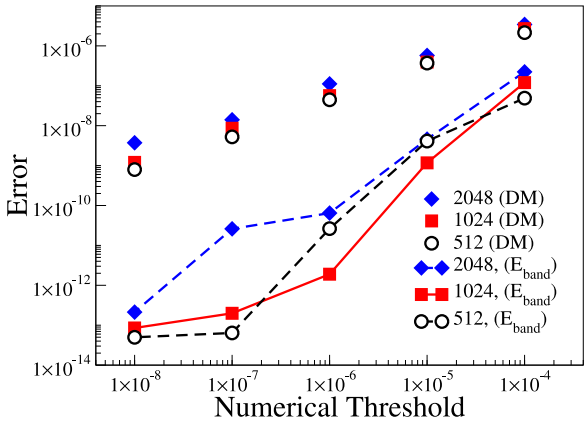

FIG. 2. The error in the calculated density matrix (DM) for polyalanine (2593 atoms) in water with a total of 19 945 atoms (in Fig. 3) as measured by the Frobenius norm (normalized per atom) for partitions with 512, 1024, and 2048 separate communities based on graphs, $\mathcal{S}_\tau$, from varying numerical thresholds τ. The connected symbols (lower part) show the error in band energy, $E_{\text{band}} = Tr[HD]$, in units of eV per atom.

which was calculated using regular sparse matrix algebra with a tight threshold of 10^{-12}. The error is fairly insensitive to the number of graph partitions and is instead controlled by the value of the threshold that is used to estimate the data dependency graphs. In contrast, the computational cost varies significantly with the size of the graph partitions. The cost in the limit of only one large community, containing the whole system, or in the opposite limit, with one partition for each orbital, scales as $O(N^3)$ or $O(Nm^3)$, respectively, where m is the average number of edges per vertex in $\mathcal{S}_\tau$ and $N \times N$ is the size of H. A straightforward graph partitioning may thus lead to a significant overhead compared to a Fermi-operator expansion using thresholded sparse matrix algebra,[5] which scales as $O(Nm^2)$. However, with an optimized graph partitioning the total cost can be reduced to scale as $O(Nm^2)$ (see Appendix A). A similar optimization can be performed for divide and conquer methods, but may not be applicable to

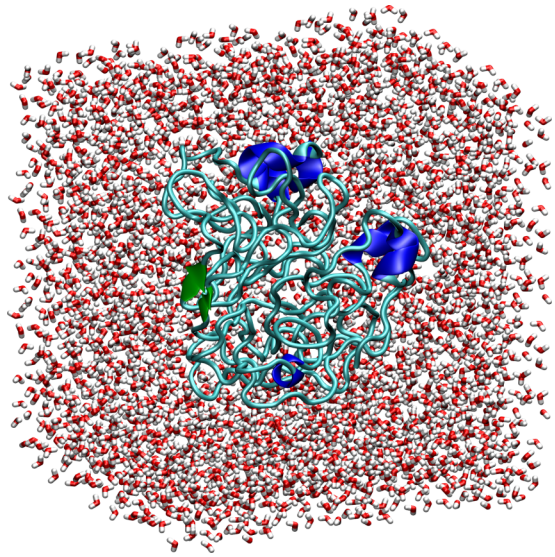

FIG. 3. Polyalanine (2593 atoms) solvated in water with a total of 19 945 atoms.

inhomogeneous systems.[17] Figure 4 shows the timing (12 s, red dashed line) for a thresholded sparse matrix algebra (SpM Alg) Fermi-operator expansion with Intel's MKL sparse matrix library[30] running in parallel on a dual eight-core CPU. With the graph-based approach (filled circles) using the METIS graph partitioning (Graph Part.) program for varying numbers of communities, it is possible to significantly reduce the run time on the same platform (23 s) compared to, for example, a single atom-based decomposition. The graph-based formalism also has the additional advantage of an almost trivial and highly scalable parallelism as is demonstrated by the run times on 1, 16, or 32 graphics processing units (GPUs) on separate nodes (open symbols).[60] The parallel performance is close to ideal, reaching a performance of about 25 μs/atom and a subsecond wall-clock time (0.5 s) on the 32 node GPU platform.

As is demonstrated here, the off-the-shelf graph partitioning scheme works very well and drastically reduces the overhead compared to a straightforward implementation. However, by adjusting the graph partitioning to the particular requirements of the electronic structure calculation as well as the computational platform, further optimizations are possible.[61]

B. Molecular dynamics simulation

Linear scaling divide and conquer methods[12–17] rely on an estimated finite range of direct electron interaction, which can be motivated by the localized character of the Wannier functions.[62–64] This allows a system to be partitioned into smaller overlapping regions that are solved separately (apart from long-range electrostatic interactions), within predetermined local interaction zones, and then reassembled. Divide and conquer schemes are naturally parallel and in spirit similar to our graph-based approach. However, their numerical accuracy can be difficult to control without careful prior testing and convergence analysis.[6,65,66] An automatic, adjustable error control is particularly challenging in molecular dynamics simulations of inhomogeneous materials, where reacting or floppy molecules and atoms can move across predetermined local interaction zones and where transitions between localized and itinerant electronic states may occur. Molecular dynamics simulations of inhomogeneous molecular systems with significant changes in the electronic overlap are therefore of particular interest when we evaluate our framework. Furthermore, the precision can be gauged very sensitively by the accuracy and long-term stability of the total energy, which is affected by the accuracy in the calculation of the potential energy surface in each time step and by the accumulated and integrated error in the forces.

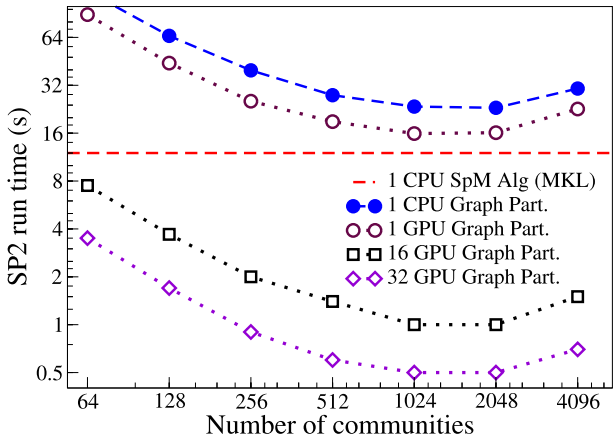

FIG. 4. The time to calculate the density matrix using the SP2 expansion (with threshold $\tau = 10^{-5}$) partitioned over different sets of subgraphs for the solvated polyalanine system (19 945 atoms). The time to calculate the graph partitioning (about 0.4 s in a serial single node calculation with METIS) is not included in the run time. In a molecular dynamics simulation the computational overhead from the graph partitioning can be reduced significantly since, in practice, only in-frequent partial updates are needed.

The data dependency graph $\mathcal{S}_\tau(t)$ can be estimated from the numerically thresholded density matrix in the previous molecular dynamics time step, $\lfloor D(t-\delta t)\rfloor_\tau$, and new Hamiltonian matrix elements, $H(t)$, as the atoms move, for example, from

$$\mathcal{S}_\tau(t) \leftarrow \lfloor(\lfloor D(t-\delta t)\rfloor_\tau + H(t))^2\rfloor_\epsilon. \tag{8}$$

In our molecular dynamics simulation below, we use the symbolic representation of $\mathcal{S}_\tau(t)$ in Eq. (8), which is given from the non-zero pattern of the thresholded density matrix (with $\tau = 10^{-4}$) combined with the non-zero pattern of $H(t)$, and instead of the matrix square we use paths of length two, corresponding to the symbolic operation ($\epsilon = 0$). This approach that adapts $\mathcal{S}_\tau(t)$ to each new molecular dynamics time step by including additional redundant edges works surprisingly well (see Appendix C), though with the estimate above, $\mathcal{S}_\tau(t)$ cannot increase by more than paths of length two between two molecular dynamics steps. However, generalizations including longer paths are straightforward and the similar estimates can also be applied in the iterative SCF optimization.

Figure 5 shows the fluctuations of the total energy during a microcanonical molecular dynamics simulation of liquid water that was performed using LATTE[58] and the extended Lagrangian formulation of Born-Oppenheimer molecular dynamics.[50,67–70] The density matrix was calculated from a partitioning over separate subgraphs of $\mathcal{S}_\tau(t)$, with one water molecule per core. For the Fermi-operator expansion (at zero temperature) we used the recursive SP2 algorithm.[35] In each time step the complete SP2 sequence (the same for each subgraph expansion) for the correct total occupation is pre-determined from the HOMO-LUMO gap that is estimated from the previous time step as in Ref. 41. In this way each full expansion can be performed independently, without exchange of information during or between each matrix multiplication as otherwise would be required.[8,28] Communication is reduced to a minimum and no additional adjustments of the electronic occupation, as in divide and conquer calculations,[14] is required. The inset of Fig. 5 shows the number of water molecules of a single subgraph (core + halo) along the trajectory of an individual molecule, which oscillates as $\mathcal{S}_\tau(t)$ adaptively follows the fluctuations in the electronic overlap. Despite the large oscillations, including between 1 and 25 molecules, the total energy is both accurate and stable. The "exact" calculation with fully converged density matrices (≥4 SCFs per step) using dense matrix algebra based on full $O(N^3)$ diagonalization, is virtually indistinguishable for the first 0.5 ps (or 1000 time steps).

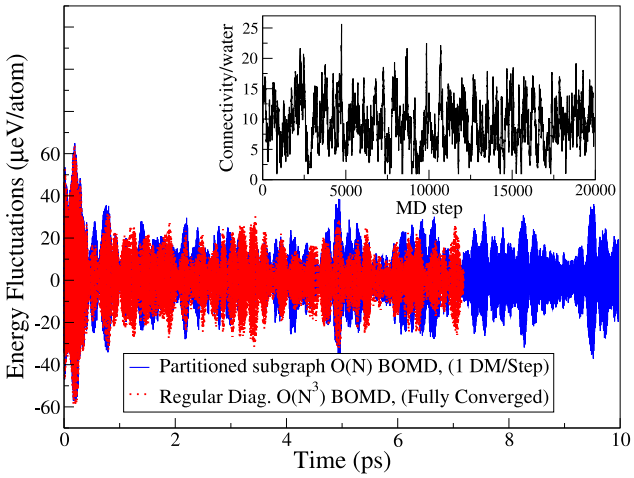

FIG. 5. The total energy fluctuations in a microcanonical Born-Oppenheimer molecular dynamics (BOMD) simulation of liquid water (100 molecules, $T \sim 300$ K, $\delta t = 0.5$ fs), using graph partitioning and one density matrix (DM) construction per step vs. SCF optimized BOMD with diagonalization (Diag.). The inset shows the number of water molecules associated with the subgraph of an individual molecule. Energy drift is less than ~0.2 μeV/atom per ps.

Linear scaling molecular dynamics simulations using divide and conquer or radial truncation approaches often show systematic energy drifts[71–73] that are significantly higher than regular $O(N^3)$ methods[9,42,43] and multiple orders of magnitude larger than the graph-based molecular dynamics simulation in Fig. 5. Such problems may occur because of difficulties controlling the error in the force evaluations[6,74] as atoms move across the local zone boundaries and as the electronic overlap fluctuates, or because of incomplete SCF optimization causing a broken time-reversal symmetry.[42,75] The problem is illustrated in Fig. 6, which shows a comparison between a divide and conquer approach and our graph-based calculation of the density matrix for a snapshot from a molecular dynamics simulation of the water system in Fig. 5. Without the adaptivity of the graph-based method, the divide and conquer approach needs a large cutoff radius, R_{cut}, to reach sufficient convergence in the calculation of the density matrix for the water system, which leads to a significant overhead. With the graph-based framework as demonstrated here in combination with a modern formulation of Born-Oppenheimer molecular dynamics,[42–50] these problems can be avoided.

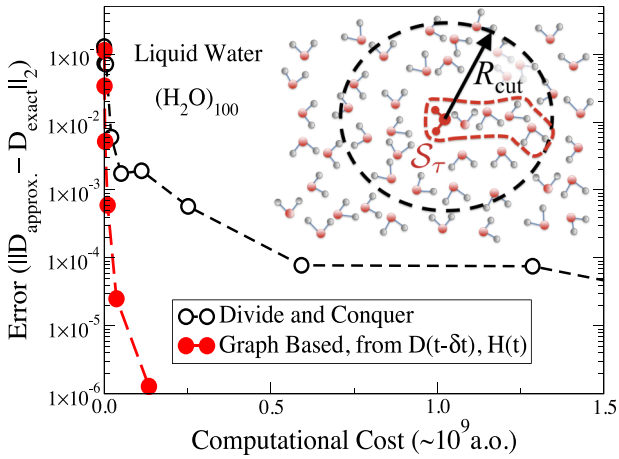

FIG. 6. The convergence of the density matrix error for a snapshot during a molecular dynamics simulation of the water system in Fig. 5 (100 molecules, $T \sim 300$ K, $\delta t = 0.5$ fs) as a function of the computational cost for various numerical thresholds ($\tau = 10^{-1}, 10^{-2}, \ldots, 10^{-6}$) in the symbolic estimate of the data-dependency graph in Eq. (8) for the graph-based method, and for different sizes of the cutoff radius, R_{cut}, in a divide and conquer approach. To capture a hypothetical electronic overlap within the red dashed border in the inset (associated with the data-dependency graph $\mathcal{S}_\tau$ for the large red molecule at the center), the cutoff radius needs to be large, which leads to a significant overhead for the divide and conquer approach. The efficiency would be similar only for a homogeneous system. The computational cost was estimated from the sum of the number arithmetic operations (a.o.) required to calculate the density matrices ($\sim m^3$ a.o.) from all the separate subgraph partitions or divide and conquer regions (given by $m \times m$ matrices)—one for each water molecule.

IV. CONCLUSIONS

In this article we have shown how graph theory can be combined with quantum theory to calculate the electronic structure of large complex systems with well-controlled accuracy. The graph formalism is general and applicable to a broad range of electronic structure methods and materials, for which sparse matrix representations can be used, including molecular dynamics simulations, overcoming significant gaps in linear scaling electronic structure theory.

ACKNOWLEDGMENTS

We acknowledge support from the Department of Energy Offices of Basic Energy Sciences (Grant No. LANL2014E8AN) and the Laboratory Directed Research and Development program of Los Alamos National Laboratory (LANL). Generous support and discussions with T. Peery at the T-division International Java Group are acknowledged. The research used resources provided by the LANL Institutional Computing Program. LANL, an affirmative action/equal opportunity employer, is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. DOE under Contract No. DE-AC52-06NA25396.

APPENDIX A: $O(Nm^2)$ SCALING ESTIMATE WITH AN OPTIMIZED GRAPH PARTITIONING FOR THE FERMI-OPERATOR EXPANSION

Figure 7 shows the set of the vertices associated with one part of the data dependency graph that forms each contracted dense submatrix in the graph-based Fermi operator expansion. The inner set of this subgraph belongs to the *core* part and the outer set, called *halo*, contains the vertices not in the core, but adjacent to at least one core vertex. Each vertex from the core belongs to exactly one part whereas the halo will overlap with other subgraphs. We assume a uniform data dependency graph with m edges connected to each vertex. The total cost (C_{Gr}) of the graph-based Fermi operator expansion of a full Hamiltonian matrix of dimension $N \times N$, i.e., with a data dependency graph with a total of N vertices, as measured by the number of arithmetic operations (one arithmetic operation = 1 multiplication + 1 addition), can then be estimated by

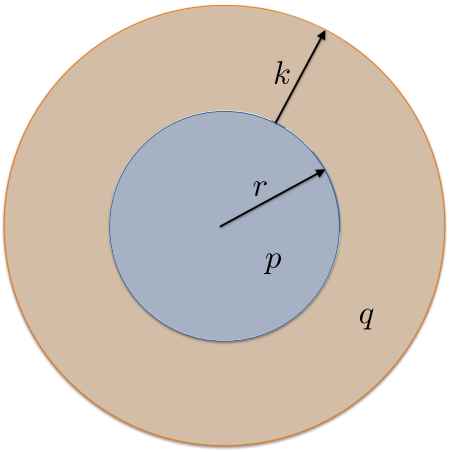

FIG. 7. Illustration of the geometry of a single graph partition. For simplicity, each part is assumed to have the same parameters p, q, r, and k, where p is the number of vertices in the *core*, q is the number of vertices in the *halo*, r is the radius of the core, and $r+k$ is the radius of the whole part.

$$\mathrm{C_{Gr}} = M\frac{N}{p}(p+q)^3, \tag{A1}$$

where M is the number of matrix-matrix multiplications in the Fermi operator expansion (typically between 20 and 40 multiplications are required). In dimension d (1, 2, or 3) the relation between the total number of vertices $p+q$ included within the radius $r+k$, assuming a uniform distribution of nodes, is given by

$$p - 1 + q = c_d(r+k)^d, \tag{A2}$$

for some dimensional dependent constant c_d, and for the inner halo we have that

$$p - 1 = c_d r^d. \tag{A3}$$

The 1 is subtracted assuming that a single vertex has no extension alone with a radius $r = 0$. In the limit $r \to 0$ the number of vertices q in the halo is equal to the number of edges m of each vertex, i.e.,

$$m = c_d k^d. \tag{A4}$$

This means that $r = c_d^{-1/d}(p-1)^{1/d}$ and $k = c_d^{-1/d}m^{1/d}$ and

$$\begin{aligned}\mathrm{C_{Gr}} &= M\frac{N}{p}\left(c_d(r+k)^d\right)^3 \\ &= M\frac{N}{p}\left(c_d(c_d^{-1/d}(p-1)^{1/d} + c_d^{-1/d}m^{1/d})^d\right)^3 \\ &= M\frac{N}{p}\left(c_d^{1/d}(c_d^{-1/d}(p-1)^{1/d} + c_d^{-1/d}m^{1/d})\right)^{3d} \\ &= M\frac{N}{p}\left((p-1)^{1/d} + m^{1/d}\right)^{3d}.\end{aligned} \tag{A5}$$

We can now determine the optimal size of the core partitioning from the minima of the arithmetic cost, i.e., when $d\mathrm{C_{Gr}}/dp = 0$. This leads to the equation

$$(2p+1)(p-1)^{1/d-1} = m^{1/d}, \tag{A6}$$

from which we get

$$\begin{aligned}m &= \frac{(2p+1)^d}{(p-1)^{d-1}} = (2p+1)\left(\frac{2p+1}{p-1}\right)^{d-1} \\ &= (2p+1)\left(2 + \frac{3}{p-1}\right)^{d-1} \\ &= (2p+1)\left(2^{d-1} + \frac{3(d-1)2^{d-2}}{p-1} + O\left(\frac{1}{(p-1)^2}\right)\right) \\ &= 2^d p + 2^{d-1} + 3(d-1)2^{d-1}\frac{p}{p-1} + O(p^{-1}).\end{aligned} \tag{A7}$$

Hence, for $m \gg 1$, the cost is minimized for $p = 2^{-d}m - (3d-2)/2 + O(m^{-1})$, or, approximately, $p \approx 2^{-d}m$. Inserting this approximate value of p we find that

$$\begin{aligned}\mathrm{C_{Gr}} &\approx 2^d M\frac{N}{m}\left(\frac{1}{2}m^{1/d} + m^{1/d}\right)^{3d} \\ &= 2^d M\frac{N}{m}\left(\frac{3}{2}m^{1/d}\right)^{3d} \\ &= 2^d MNm^2\left(\frac{3}{2}\right)^{3d} \\ &= MNm^2\left(\frac{27}{4}\right)^d.\end{aligned} \tag{A8}$$

This optimized cost should be compared to the cost of using sparse matrix-matrix multiplication (SpM) in the Fermi operator expansion, which has the estimated cost in terms of arithmetic operations

$$\mathrm{C_{SpM}} = MNm^2. \tag{A9}$$

The ratio between these two costs is thus given by

$$\frac{\mathrm{C_{Gr}}}{\mathrm{C_{SpM}}} \approx \left(\frac{27}{4}\right)^d. \tag{A10}$$

The computational overhead of the graph-based expansion in terms of the number of arithmetic operations with respect to a Fermi operator expansion using sparse matrix-matrix multiplications is thus a factor of about 7, 46, and 308 ($d = 1,2,3$). The overhead is system size independent and is governed by the dimensionality of the data dependency graph as given by Eqs. (A2) and (A3) and the figure. Our estimate is based on a number of idealized assumptions but illustrates that the general $O(Nm^2)$ scaling behavior of a thresholded sparse matrix algebra is achievable also with the graph-based approach. It also highlights an improved efficiency for quasi low-dimensional problems such as molecular liquids, polymers, and protein structures. In addition, the ability to reach close to peak performance using the dense matrix algebra for the subgraph partitions, combined with an almost trivial parallelism requiring only a minimal amount of data transfer, provides a significant advantage and simplification compared to a sparse matrix algebra techniques.

APPENDIX B: CONSTRUCTION OF POLYALANINE IN WATER

The test system we used for the analysis is based on a 19 945 atoms system of polyalanine (2593 atoms) in liquid water as illustrated in Fig. 3. We have chosen alanine because it is possibly the simplest chiral amino acid which allows for

the formation of stable secondary structures. In consequence, with this simple peptide, we can build models which will include linear, α-helix, and β-sheet polyalanine secondary structures introducing extra complexity to the system which is ultimately desired for testing the graph-based electronic structure framework. The construction of the model is done following four systematic steps: (1) Construction of a linear helix chain; (2) application of an artificial compression along the principal axis (z axis); (3) an NPT equilibration of 100 ps in vacuum followed by solvation with water molecules; and (4) a geometry optimization of the full system. In the first two steps we used GROMACS version 5.0.4 with the OPLS force field and in the last two steps we used the self-consistent charge density functional based tight-binding code LATTE. The density of the final globular structure is around 0.7 g/ml, which is a reasonable value for globular proteins.

APPENDIX C: ADAPTIVE ESTIMATE OF THE DATA CONNECTIVITY GRAPH

The adaptivity of the estimate for the data connectivity graph in Eq. (8) can be understood from the illustration in Fig. 8 as two separate subsystems, $D_a(t-\delta t)$ and $D_b(t-\delta t)$, move closer together and get connected through a Hamiltonian overlap term, $H_{ab}(t)$. The estimated data dependency graph, $S_{ab}(t)$, includes paths of length two, i.e., the "double jumps" indicated by the dashed lines. The connectivity graph, $S_{ab}(t)$, can then be partitioned into a subgraph from which we can collect a new density matrix, $D(t)$, which after a numerical threshold, $\lfloor D(r)\rfloor_\tau$, gives a new starting point for the next time step. This process allows new connections to form and vanish as the system evolves, which is illustrated by the hypothetical electronic overlap of $\lfloor D(r)\rfloor_\tau$ at the bottom of the figure, with two new connections and one removed.

APPENDIX D: EXPERIMENT AND ARCHITECTURE DETAILS

All the runs shown in Figs. 2 and 4 used the Moonlight cluster at LANL (with each node comprised of 2 eight-core Intel Xeon E5-2670 CPUs running at 2.6 GHz) and 2 Nvidia Tesla M2090 GPUs per node. Only 1 GPU per node was used for the distributed runs shown in Fig. 3 in the main paper. The software environment included the GNU 4.8.2 C compiler with OpenMP, the MKL 11.2 matrix algebra library, and OpenMPI 1.6.5 (for distributed runs). 16 OpenMP threads were used in all cases. CUDA and the CuBLAS matrix algebra library were used for the GPU SP2 implementation.

The experimental setup for Fig. 2 was as follows. Initially, the sparse matrix recursive SP2 Fermi expansion was run on the polyalanine in water system using threshold, $\tau = 10^{-12}$. The resulting density matrix was thresholded with $\tau = 10^{-3}$, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, and 10^{-8}. Those thresholded graphs were used to generate the METIS graph partitionings for 512, 1024, and 2048 partitions using the multilevel recursive bisection scheme (gpmetis-ptype = rb). Runs were made for each partitioning (512, 1024, 2048) at each threshold level (10^{-3} to 10^{-8}). The resulting density matrix in each case was compared to the density matrix from the SP2 run with threshold, $\tau = 10^{-12}$. The error in the new calculated density matrices was measured by the Frobenius norm (normalized per atom), as well as the error in band energy, $E_{\text{band}} = Tr[HD]$, per atom. These runs were made on a single node of the Moonlight cluster.

The experimental setup for Fig. 4 was as follows. Initially, SP2 Fermi-operator expansion was run on the polyalanine in water system using threshold, $\tau = 10^{-5}$ using sparse matrix algebra. The resulting density matrix was used as an estimate of the data dependency graph $\mathcal{S}_\tau$ for the generation of METIS graph partitionings of size 64, 128, 256, 512, 1024, 2048, and 4096. Graph-based SP2 runs were performed for each partitioning with dense matrix algebra, i.e. with threshold, $\tau = 0$. The distributed graph-based runs took advantage of hybrid parallelism combining the use of MPI, OpenMP, and GPU parallelism on 1, 16, and 32 CPU-GPU nodes. The SP2 algorithm using the threshold $\tau = 10^{-5}$ and the MKL compressed sparse row (CSR) format run on a single node of the Moonlight cluster is shown for comparison.

The wall-clock time required to calculate the density matrix using regular sparse matrix algebra with an optimized shared memory parallelism running on a single CPU node is reduced by a factor of 133 with the optimized graph partitioning approach on the 32 node GPU platform. The (strong-scaling) ability to reach subsecond wall-clock times in the calculation of the density matrix is critical for many molecular dynamics simulations that often require hundreds of thousands of time steps.

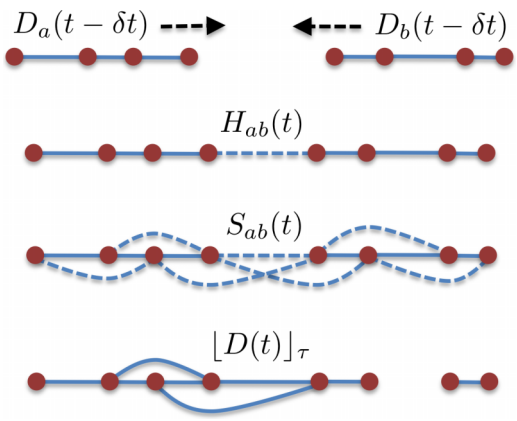

FIG. 8. Illustration of the adaptive evolution of the data dependency graph, $S_{ab}(t)$, between two time steps in a molecular dynamics simulation.

[1]P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).
[2]W. Kohn and L. J. Sham, Phys. Rev. B **140**, A1133 (1965).
[3]R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules* (Oxford University Press, Oxford, 1989).
[4]R. Dreizler and K. Gross, *Density-Functional Theory* (Springer Verlag, Berlin Heidelberg, 1990).
[5]S. Goedecker, Rev. Mod. Phys. **71**, 1085 (1999).
[6]D. R. Bowler and T. Miyazaki, Rep. Prog. Phys. **75**, 036503 (2012).
[7]D. R. Bowler and T. Miyazaki, J. Phys.: Condens. Matter **22**, 074207 (2010).
[8]J. VandeVondele, U. Borstnik, and J. Hutter, J. Chem. Theory Comput. **8**, 3565 (2012).
[9]D. Marx and J. Hutter, in *Modern Methods and Algorithms of Quantum Chemistry*, 2nd ed., edited by J. Grotendorst (John von Neumann Institute for Computing, Jülich, Germany, 2000).

[10]G. Chartrand, *Introductory Graph Theory* (Dover Publications, New York, 1985).
[11]J. A. Bondy, *Graph Theory* (Springer-Verlag, London, 2008).
[12]W. Yang, Phys. Rev. Lett. **66**, 1438 (1991).
[13]P. D. Walker and P. G. Mezey, J. Am. Chem. Soc. **115**, 12423 (1993).
[14]W. T. Yang and T. S. Lee, J. Chem. Phys. **103**, 5674 (1995).
[15]I. A. Abrikosov, A. M. N. Niklasson, S. I. Simak, B. Johansson, A. V. Ruban, and H. L. Skriver, Phys. Rev. Lett. **76**, 4203 (1996).
[16]K. Kitaura, E. Ikeo, T. Nakano, and M. Uebayasi, Chem. Phys. Lett. **313**, 701 (1999).
[17]T. Ozaki, Phys. Rev. B **74**, 245101 (2006).
[18]F. G. Gustavson, ACM Trans. Math. Software **4**, 250 (1978).
[19]S. Pissanetzky, *Sparse Matrix Technology* (Academic Press, London, 1984).
[20]W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipies in FORTRAN* (Cambridge University Press, Port Chester, NY, 1992).
[21]Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS Publishing, Boston, 1996).
[22]M. Challacombe, Comput. Phys. Commun. **128**, 93 (2000).
[23]E. H. Rubensson, E. Rudberg, and P. Salek, J. Comput. Chem. **28**, 2531 (2007).
[24]E. H. Rubensson, E. Rudberg, and P. Salek, J. Chem. Phys. **128**, 74109 (2008).
[25]A. Buluc and J. R. Gilbert, SIAM J. Sci. Comput. **34**, 170 (2012).
[26]U. Borstnik, J. VandeVondele, V. Weber, and J. Hutter, Parallel Comput. **40**, 47 (2014).
[27]N. Bock, M. Challacombe, and L. V. Kale, SIAM J. Sci. Comput. **38**, C1–C21 (2016).
[28]V. Weber, T. Latino, A. Pozdeev, I. Feduova, and A. Curioni, J. Chem. Theory Comput. **11**, 3145 (2015).
[29]S. M. Mniszewski, M. J. Cawkwell, M. E. Wall, J. Mohd-Yusof, N. Bock, T. C. Germann, and A. M. N. Niklasson, J. Chem. Theory Comput. **11**, 4644 (2015).
[30]Intel MKL, Intel Math Kernel Library, 2015, https://software.intel.com/en-us/intel-mkl.
[31]NVIDIA cuSPARSE, 2014, https://developer.nvidia.com/cusparse.
[32]R. McWeeny, Proc. R. Soc. London, Ser. A **235**, 496 (1956).
[33]A. H. R. Palser and D. E. Manolopoulos, Phys. Rev. B **58**, 12704 (1998).
[34]A. Holas, Chem. Phys. Lett. **340**, 552 (2001).
[35]A. M. N. Niklasson, Phys. Rev. B **66**, 155115 (2002).
[36]A. M. N. Niklasson, Phys. Rev. B **68**, 233104 (2003).
[37]W. Z. Liang, C. Saravanan, Y. Shao, R. Baer, A. T. Bell, and M. Head-Gordon, J. Chem. Phys. **119**, 4117 (2003).
[38]E. Rudberg and E. H. Rubensson, J. Phys.: Condens. Matter **23**, 075502 (2011).
[39]E. H. Rubensson, J. Chem. Theory Comput. **7**, 1233 (2011).
[40]P. Suryanarayana, Chem. Phys. Lett. **555**, 291 (2013).
[41]E. H. Rubensson and A. M. N. Niklasson, SIAM J. Sci. Comput. **36**, 148 (2014).
[42]P. Pulay and G. Fogarasi, Chem. Phys. Lett. **386**, 272 (2004).
[43]J. Herbert and M. Head-Gordon, Phys. Chem. Chem. Phys. **7**, 3269 (2005).
[44]A. M. N. Niklasson, C. J. Tymczak, and M. Challacombe, Phys. Rev. Lett. **97**, 123001 (2006).
[45]T. D. Kühne, M. Krack, F. R. Mohamed, and M. Parrinello, Phys. Rev. Lett. **98**, 066401 (2007).
[46]G. Zheng, A. M. N. Niklasson, and M. Karplus, J. Chem. Phys. **135**, 044122 (2011).
[47]J. Hutter, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **2**, 604 (2012).
[48]L. Lin, J. Lu, and S. Shao, Entropy **16**, 110 (2014).
[49]M. Arita, D. R. Bowler, and T. Miyazaki, J. Chem. Theory Comput. **10**, 5419 (2014).
[50]A. M. N. Niklasson and M. Cawkwell, J. Chem. Phys. **141**, 164123 (2014).
[51]S. Goedecker and L. Colombo, Phys. Rev. Lett. **73**, 122 (1994).
[52]R. N. Silver and H. Roder, Int. J. Mod. Phys. C **5**, 735 (1994).
[53]A. M. N. Niklasson and M. Challacombe, Phys. Rev. Lett. **92**, 193001 (2004).
[54]V. Weber, A. M. N. Niklasson, and M. Challacombe, Phys. Rev. Lett. **92**, 193002 (2004).
[55]M. Elstner, D. Poresag, G. Jungnickel, J. Elsner, M. Haugk, T. Frauenheim, S. Suhai, and G. Seifert, Phys. Rev. B **58**, 7260 (1998).
[56]M. W. Finnis, A. T. Paxton, M. Methfessel, and M. van Schilfgarde, Phys. Rev. Lett. **81**, 5149 (1998).
[57]T. Frauenheim, G. Seifert, M. E. Z. Hajnal, G. Jungnickel, D. Poresag, S. Suhai, and R. Scholz, Phys. Status Solidi **217**, 41 (2000).
[58]M. J. Cawkwell and A. M. N. Niklasson, J. Chem. Phys. **137**, 134105 (2012).
[59]G. Karypis and V. Kumar, SIAM J. Sci. Comput. **20**, 359 (1999).
[60]NVIDIA cuBLAS, 2014, https://developer.nvidia.com/cuBLAS.
[61]H. N. Djidjev, G. Hahn, S. M. N. Mniszewski, C. F. A. Negre, A. M. N. Niklasson, and V. B. Sardeshmukh, "Graph partitioning methods for fast parallel quantum molecular dynamics," e-print arXiv:1605.01118 [quant-ph] (2016).
[62]W. Kohn, Phys. Rev. Lett. **76**, 3168 (1996).
[63]W. Kohn, Phys. Rev. A **133**, A171 (1964).
[64]N. F. Mott, Philos. Mag. **6**, 278 (1961).
[65]T. S. Lee, D. M. York, and W. Yang, J. Chem. Phys. **105**, 2744 (1996).
[66]D. M. York, T. S. Lee, and W. Yang, Phys. Rev. Lett. **80**, 5011 (1998).
[67]A. M. N. Niklasson, Phys. Rev. Lett. **100**, 123004 (2008).
[68]P. Steneteg, I. A. Abrikosov, V. Weber, and A. M. N. Niklasson, Phys. Rev. B **82**, 075110 (2010).
[69]P. Souvatzis and A. M. N. Niklasson, J. Chem. Phys. **140**, 044117 (2014).
[70]B. Aradi, A. M. N. Niklasson, and T. Frauenheim, J. Chem. Theory Comput. **11**, 3357 (2015).
[71]F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashista, Phys. Rev. B **77**, 085103 (2008).
[72]E. Tsuchida, J. Phys.: Condens. Matter **20**, 294212 (2008).
[73]F. Shimojo, S. Hattori, R. K. Kalia, M. Kusaneth, W. W. Mou, A. Nakano, K. Nomura, S. Ohmura, P. Rajak, K. Shimamura, and P. Vashista, J. Chem. Phys. **140**, 18529 (2014).
[74]M. Kobayashi, T. Kunisada, T. Akama, D. Sakura, and H. Nakai, J. Chem. Phys. **134**, 034105 (2011).
[75]D. K. Remler and P. A. Madden, Mol. Phys. **70**, 921 (1990).
[76]See supplementary material at http://dx.doi.org/10.1063/1.4952650 for pseudo code that demonstrates the exact relation between a globally thresholded sparse matrix algebra and a graph partitioning approach.