REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 29-10-2007	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 15-Jun-2006 - 14-Jun-2007

4. TITLE AND SUBTITLE SEPARATED REPRESENTATIONS AND FAST ALGORITHMS FOR MATERIALS SCIENCE	5a. CONTRACT NUMBER W911NF-06-1-0254
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 6D10S7
6. AUTHORS Gregory Beylkin, Lucas Monzon and Fernando Perez	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Colorado - Boulder Office of Contracts and Grants Campus Box 572, 3100 Marine Street Rm 481 Boulder, CO 80309 -0572	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 50966-PH-DRP.1

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; Federal purpose rights

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

Our goal (within the time-frame of the grant) was to finish the development of algorithms and software for applying Green's functions (and other operators) and to develop and test algorithms for computing multiparticle wave functions, both based based on representing operators and functions of many variables as short sums of separable functions, the so-called separated representations. Our approach is different from the Fast

15. SUBJECT TERMS

Separated representations, Green's function, fast algorithms, Schrodinger equation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Gregory Beylkin
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR		19b. TELEPHONE NUMBER 303-492-6935

Report Title

SEPARATED REPRESENTATIONS AND FAST ALGORITHMS FOR MATERIALS SCIENCE

ABSTRACT

Our goal (within the time-frame of the grant) was to finish the development of algorithms and software for applying Green's functions (and other operators) and to develop and test algorithms for computing multiparticle wave functions, both based based on representing operators and functions of many variables as short sums of separable functions, the so-called separated representations. Our approach is different from the Fast Multipole Method and allows us to develop and use fast algorithms in high dimensions.

We have accomplished these tasks with the results described in two papers attached to this report. We started the process of comparing our approach to solving multiparticle Schrodinger equation to that currently used in Quantum Chemistry.

List of papers submitted or published that acknowledge ARO support during this reporting period. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

G. Beylkin, V. Cheruvu, and F. Perez
Fast adaptive algorithms in the non-standard form for multidimensional problems, Appl. Comput. Harmon. Anal. Accepted for publication.
(2007) APPM Preprint 550, available electronically from Appl. Comput. Harmon. Anal. website

G. Beylkin, M.J. Mohlenkamp, and F. Perez, Approximating a wavefunction as an unconstrained sum of Slater determinants, Submitted to J. Math. Phys., (2007). APPM Preprint 554.

Number of Papers published in peer-reviewed journals: 2.00

(b) Papers published in non-peer-reviewed journals or in conference proceedings (N/A for none)

Number of Papers published in non peer-reviewed journals: 0.00

(c) Presentations

Gregory Beylkin, Martin J. Mohlenkamp, and Fernando Perez,
"Toward Solving the Multiparticle Schrodinger Equation
via an Unconstrained Sum of Slater Determinants",
Pacific Northwest National Lab, Richland, WA, January 25-26, 2007

Gregory Beylkin, "Fast algorithms for adaptive application of integral operators in high dimensions", The 2007 John H. Barrett Memorial Lectures, University of Tennessee, April 28, 2007

Gregory Beylkin, Martin J. Mohlenkamp, and Fernando Perez, "Preliminary results on approximating a wavefunction as an unconstrained sum of Slater determinants", ICIAM, Zurich, Switzerland, July 17, 2007

Number of Presentations: 3.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts): 0

(d) Manuscripts

Number of Manuscripts: 0.00

Number of Inventions:

Graduate Students

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Lucas Monzon	0.25
Fernando Perez	0.25
FTE Equivalent:	**0.50**
Total Number:	**2**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Gregory Beylkin	0.08	No
FTE Equivalent:	**0.08**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: ......

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:......

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:......

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale): ......

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering: ......

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense ......

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields: ......

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME	PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

FINAL REPORT "SEPARATED REPRESENTATIONS AND FAST ALGORITHMS FOR MATERIALS SCIENCE" AWARD W911NF-06-1-0254 PERIOD: JUNE 15, 2006 TO JUNE 14, 2007

GREGORY BEYLKIN * (PI), LUCAS A. MONZÓN, AND FERNANDO PÉREZ

1. Abstract

Our goal (within the time-frame of the grant) was to finish the development of algorithms and software for applying Green's functions (and other operators) and to develop and test algorithms for computing multiparticle wave functions, both based based on representing operators and functions of many variables as short sums of separable functions, the so-called separated representations. Our approach is different from the Fast Multipole Method and allows us to develop and use fast algorithms in high dimensions.

We have accomplished these tasks with the results described in two papers attached to this report. We started the process of comparing our approach to solving multiparticle Schrodinger equation to that currently used in Quantum Chemistry.

2. Tasks

Our work addresses two tasks in using separated representations [5, 4]:

(1) The first class of algorithms addresses problems that currently require either the Fast Multipole Method (FMM) and/or Particle in cell methods and/or Ewald summation techniques. Our approach is simpler, more flexible and is as effective as FMM. It is based on separated representations and its main advantage is that it works for operators in higher dimensions (much greater than three) provided that the functions are maintained in separated representation as well.

(2) The second class of algorithms addresses multiparticle computations and confronts the problem of dimensionality directly, by providing a representation of operators and functions that, on one hand, achieves a finite but arbitrary accuracy and, on the other, limits the growth of the computational cost to a linear function of the number of particles (at least asymptotically). With this type of algorithms, we compete with the Configuration Interaction (CI) and of Coupled Cluster (CC) methods. Our approach is different from both CI and CC, and holds the promise of a breakthrough.

Department of Applied Mathematics, University of Colorado at Boulder; 526 UCB, Boulder, CO 80309-0526; beylkin@colorado.edu; phone 303-492-6935, fax 303-492-4066.

3. Results for Task 1: Multiresolution separated representations and fast algorithms

Our goal has been to obtain multiresolution representations of Green's functions in a form that facilitates solving integral equations. We note that computations involving multidimensional free space Green's functions are greatly simplified by using separated representations [5, 4, 18, 19, 29, 30, 13, 3]. Our approach is based on representing spherically symmetric functions by sums of products of Gaussians.

3.1. **Separated representations for Poisson-type kernels.** We construct a separated approximation of the function $1/r^{\alpha}$, where $r = ||\mathbf{x}||$, $\mathbf{x} \in \mathbb{R}^3$ using a collection of Gaussians. The approximation is obtained by discretizing the integral

$$\frac{1}{r^{\alpha}} = \frac{2}{\Gamma(\alpha/2)} \int_{-\infty}^{\infty} e^{-r^2 e^{2s} + \alpha s}\, ds\,. \tag{1}$$

For $\alpha = 1$ it is the same integral as used in the Ewald summation (up to a change of variables, see e.g., [16]). We have

Proposition 3.1. *For any $\alpha > 0$, $0 < \delta \le 1$, and $0 < \epsilon \le \min\left\{\frac{1}{2}, \frac{8}{\alpha}\right\}$, there exist positive numbers p_m and w_m such that*

$$\left|\frac{1}{r^{\alpha}} - \sum_{m=1}^{M} w_m e^{-p_m r^2}\right| \le \frac{\epsilon}{r^{\alpha}}, \tag{2}$$

where

$$M = \log \epsilon^{-1} [c_0 + c_1 \log \epsilon^{-1} + c_2 \log \delta^{-1}], \tag{3}$$

where c_1, c_2 and c_3 are constants that only depend on α. For fixed power α and accuracy ϵ, we have $M = \mathcal{O}(\log \delta^{-1})$.

A proof of Proposition 3.1 can be found in [7].

Using $r = ||x||$, where $x = (x_1, x_2, x_3)$, and $\alpha = 1$ in (2), we arrive at a separated representation for the Poisson kernel. Although in this paper we compute the lattice sums corresponding to the Poisson kernel, the same approach will work for any $\alpha > 0$ as well as other spherically symmetric potentials, e.g. Yukawa potential $e^{-\mu r}/r$.

As in [18, 19], the approximation in (2) is obtained using trapezoidal rule. First, we discretize the integral (1), namely, set $p_m = e^{2s_m}$ and $w_m = 2\Delta s\, e^{\alpha s_m}/\Gamma(\alpha/2)$, where $s_m = s_0 + (m-1)\Delta s$, $m = 1 \ldots, M$. For a given accuracy ϵ and range $0 < \delta \le r \le 1$, we select s_0 and $s_M = s_0 + (M-1)\Delta s$, the end points of the interval of integration replacing the real line in (1), so that at these points the function $f(s) = e^{-r^2 e^{2s} + \alpha s}$ and a sufficient number of its derivatives are close to zero to within the desired accuracy. We also select M, the number of points in the quadrature, so that the accuracy requirement is satisfied.

Based on these representations, we have implemented a fast, adaptive multiresolution algorithm for applying integral operators with a wide class of radially symmetric kernels in dimensions one, two and three. In particular, we consider operators of the class $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \ge 0$ and $0 < \alpha < 3/2$, and illustrate performance of the algorithm for the Poisson and Schrödinger equations in dimension three. The same algorithm may be used for all operators with radially symmetric kernels approximated as a weighted sum of Gaussians, making it applicable across multiple fields by reusing a single implementation. This fast algorithm provides controllable accuracy at a reasonable cost, comparable to that of the Fast Multipole Method (FMM). It differs from the FMM by the type of approximation used to represent kernels and has the advantage of being easily extendable to higher dimensions.

The details of the algorithm are described in the paper by Gregory Beylkin, Vani Cheruvu and Fernando Pérez "Fast Adaptive Algorithms in the Non-Standard Form for Multidimensional Problems" [2] accepted for publication in Applied and Computational Harmonic Analysis. The paper is attached to this report.

4. Results for Task 2: Multiparticle Schrödinger Equation

This part of the project is the beginning of the program to develop accurate methods for solving equations of multiparticle quantum mechanics. We have verified correctness of our approach by computing electron structure of a few elements. We started the process of comparing performance of our algorithm with other methods in Quantum Chemistry.

4.1. Approximating the Wavefunction with an unconstrained sum of Slater Determinants.

The multiparticle Schrödinger equation is the basic governing equation in quantum mechanics. We consider the time-independent case, and fix the nuclei according to the Born-Oppenheimer approximation, so the equation describes the steady state of an interacting system of electrons. For each of the N electrons in the system there are three spatial variables $\mathbf{r} = (x, y, z)$, and a discrete spin variable σ taking the values $\{-\frac{1}{2}, \frac{1}{2}\}$, which we combine and denote $(\mathbf{r}, \sigma)$ by γ. The Hamiltonian operator $\mathcal{H}$ is a sum of a kinetic energy operator $\mathcal{T}$, a nuclear potential operator $\mathcal{V}$, and an electron-electron interaction operator $\mathcal{W}$, defined by

$$\mathcal{H} = \mathcal{T} + \mathcal{V} + \mathcal{W} = -\frac{1}{2}\sum_{i=1}^{N} \Delta_i + \sum_{i=1}^{N} v(\mathbf{r}_i) + \frac{-1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|}\,, \tag{4}$$

where Δ_i is the three-dimensional Laplacian acting in the variable $\mathbf{r}_i$ and $v(\mathbf{r})$ is a sum of terms of the form $z_a/\|\mathbf{r} - \mathbf{R}_a\|$ from a nucleus at position $\mathbf{R}_a$ with charge z_a. The antisymmetric eigenfunctions of $\mathcal{H}$ represent electronic states of the system and are called wavefunctions. Antisymmetric means that under the exchange of any two coordinates, the wavefunction is odd, e.g. $\psi(\gamma_2, \gamma_1, \ldots) = -\psi(\gamma_1, \gamma_2, \ldots)$. The bound-state wavefunctions have negative eigenvalues, and are of greatest interest, so we will focus on the wavefunction with the most negative eigenvalue. In summary, our goal is to find the most negative (discrete) eigenvalue

$$\mathcal{H}\psi = \lambda\psi\,, \tag{5}$$

subject to the antisymmetry condition on ψ.

Analytic methods can give qualitative results about its solutions, and determine limiting cases, but most quantitative results must be obtained numerically. Although the equation is a 'simple' eigenvalue problem, its numerical solution presents several serious difficulties, among them the large number of variables and the antisymmetry condition on the solution. The simplest method that addresses these two difficulties is Hartree-Fock (HF), which uses the anti-symmetrization of a single product to approximate the N-particle wavefunction, i.e.,

$$\psi_{\mathrm{HF}} = \mathcal{A} \prod_{i=1}^{N} \phi_i(\gamma_i)\,. \tag{6}$$

Any approximation $\tilde{\psi}$ to the wavefunction ψ can be substituted into

$$\frac{\langle \mathcal{H}\tilde{\psi}, \tilde{\psi}\rangle}{\|\tilde{\psi}\|} \tag{7}$$

to obtain an upper bound on the lowest value of λ that solves (5). Substituting (6) into (7), one can derive a system of equations for ϕ_i to minimize (7). The resulting ψ_{HF} will best approximate ψ, in the sense of providing the best estimate (7).

To improve upon HF, it is natural to consider a sum of products

$$\psi_{(r)} = \mathcal{A} \sum_{l=1}^{r} s_l \prod_{i=1}^{N} \phi_i^l(\gamma_i)\,. \tag{8}$$

The coefficients s_l are not strictly necessary, but they allow us to assume $\|\phi_i^l\| = 1$. Many methods are based on this form, and the distinction is in how they use it. The Configuration Interaction (CI) method (see e.g. [27]) chooses the functions ϕ_i^l from a preselected master set of orthogonal functions and decides on a large number r of combinations to consider, based on excitation level. Substituting (8) into (7) leads to a matrix eigenvalue problem that can be solved for the scalar coefficients s_l. The Multi-Configuration Self-Consistent Field (MCSCF) method (e.g. [15, 9]), chooses a pattern of excitations similar to CI, but then solves for the master set of orthogonal functions as well as the scalar coefficients. Many variations and combinations of these methods have been developed, and indeed there is a whole industry in producing them.

We demonstrate a method that also uses a wavefunction of the form (8) but without constraints such as orthogonality on the ϕ_i^l. By removing these constraints we produce much better approximations at much smaller r than existing methods allow. In another context [5] we have given examples where removing constraints produces expansions that are *exponentially* more efficient, i.e. $r = N$ instead of 2^N or $r = \log N$ instead of N. For example, in our approach we can have a two-term representation

$$\begin{aligned} &\phi_1(\gamma_1)\phi_2(\gamma_2)\cdots\phi_N(\gamma_N) \\ +c\quad &[\phi_1(\gamma_1) + \phi_{N+1}(\gamma_1)][\phi_2(\gamma_2) + \phi_{N+2}(\gamma_2)]\cdots[\phi_N(\gamma_N) + \phi_{2N}(\gamma_N)]\,, \end{aligned} \tag{9}$$

where $\{\phi_j\}_{j=1}^{2N}$ form an orthonormal set. To represent (9) while requiring all factors to come from a master orthogonal set would force one to multiply out the second term and thus obtain a representation with 2^N terms. It is common sense that removal of constraints could produce better results, and steps in that direction have been taken (e.g.[26, 1, 14, 11, 12, 31, 25]). These works, however, were only able to partially remove the constraints, and so, we claim, did not achieve the full potential.

We will use a Green's function iteration to move a trial wavefunction toward the minimum of (7) without using (7) directly. This iteration was introduced in [22, 21] and used in e.g.[19]. Define the Green's function

$$\mathcal{G}_\mu = (\mathcal{T} - \mu\mathcal{I})^{-1}\,, \tag{10}$$

for $\mu < 0$. The Green's function iteration is

$$\begin{aligned} g_n &= -\mathcal{G}_{\mu_n}[(\mathcal{V}+\mathcal{W})f_n] \\ \mu_{n+1} &= \mu_n - \langle(\mathcal{V}+\mathcal{W})f_n, f_n - g_n\rangle/\|g_n\|^2 \\ f_{n+1} &= g_n/\|g_n\|. \end{aligned} \tag{11}$$

The Green's function iteration is essentially an inverse power method. The convergence rate is only linear, but if the initial μ can be chosen near to but less than the lowest eigenvalue, then the error will decrease by a substantial fraction at each iteration, and not many iterations will be needed. We use I to denote the number of Green's function iterations needed.

We use approximate wavefunctions of the form (8), with r fixed. The iteration (11) does not directly produce an approximation of the same form, so we modify it by defining g_n to be the function of the form (8) that minimizes

$$\|g_n - (-\mathcal{G}_{\mu_n}[(\mathcal{V}+\mathcal{W})f_n])\|. \tag{12}$$

In order to assure convergence to an antisymmetric solution, we use the pseudo-norm induced by the pseudo inner product $\langle\cdot,\cdot\rangle_{\mathcal{A}} = \langle\mathcal{A}(\cdot),\mathcal{A}(\cdot)\rangle$, as we did in [5]. Constructing g_n is the most challenging part of the method, and requires the bulk of our effort. To simplify notation, we now suppress the iteration index n and set $\psi = f_n$ and $\tilde{\psi} = g_n$.

We begin with some approximation $\tilde{\psi}$ (such as ψ itself) and will iteratively improve it. The outermost loop of our iteration is simply to repeat our refinement until it appears that $\tilde{\psi}$ has converged. For the computational cost estimates we denote the number of repetitions by K. To refine our representation we loop through the variables (electrons). The functions in variables other than the current variable are fixed, and the functions in the current variable will be modified to minimize the overall error $\|\tilde{\psi} - \psi\|_{\mathcal{A}}$. This Alternating Least-Squares (ALS) approach is well-known (see e.g. [20, 23, 24, 8, 10, 28]). We will alternate through the directions, but for ease of exposition we describe the $k = 1$ case. So, $\tilde{\phi}_k^l$ is fixed for $k > 1$, and we will solve for the values of $\tilde{\phi}_1^l$ for all l.

To refine in the current variable, we set up and solve a linear least-squares problem. The normal equations for a least-squares problem are derived by taking a gradient with respect to the free parameters and setting this equal to zero. As long as $\tilde{\psi}$ is linear and not degenerate in these parameters, the resulting equations are linear and have a unique solution. Usually these free parameters are coefficients of the representation in some basis. We instead take the parameters to be the point values of our functions ϕ_1^l, or, formally, as the coefficients of the point evaluation functional $\langle\gamma\rangle$. The formulas that we derive can be used with a fixed basis, but are stated independent of the basis. We still obtain linear normal equations

$$\mathbb{A}\mathbf{x} = \mathbf{b}\,, \tag{13}$$

but now $b(l)$ is a function of γ, $x(l')$ is a function of γ', and $A(l,l')$ is an integral operator mapping functions of γ' to functions of γ. The kernel of $\mathbb{A}$ is defined by

$$A(l,l')(\gamma,\gamma') = \left\langle \langle\gamma'\rangle \prod_{i=2}^{N} \tilde{\phi}_i^{l'}, \langle\gamma\rangle \prod_{i=2}^{N} \tilde{\phi}_i^{l} \right\rangle_{\mathcal{A}}\,, \tag{14}$$

where the point evaluation functionals are acting in the $i = 1$ direction. The functions in $\mathbf{b}$ are defined by

$$b(l)(\gamma) = \sum_{m}^{r} s_m \left\langle -\mathcal{G}_\mu[\mathcal{V} + \mathcal{W}] \prod_{i=1}^{N} \phi_i^m, \langle\gamma\rangle \prod_{i=2}^{N} \tilde{\phi}_i^l \right\rangle_{\mathcal{A}} . \tag{15}$$

Once $\mathbb{A}$ and $\mathbf{b}$ have been constructed, we will apply the Conjugate Gradient iterative method (see e.g. [17]) to solve (13). We initialize with $\mathbf{r} = \mathbf{b} - \mathbb{A}\mathbf{x}$, $\mathbf{v} = \mathbf{r}$, and $c = \langle \mathbf{r}, \mathbf{r} \rangle$, and then the core of the method is the sequence of assignments $\mathbf{z} \leftarrow \mathbb{A}\mathbf{v}$, $t \leftarrow c/\langle \mathbf{v}, \mathbf{z} \rangle$, $\mathbf{x} \leftarrow \mathbf{x} + t\mathbf{v}$, $\mathbf{r} \leftarrow \mathbf{r} - t\mathbf{z}$, $d \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$, $\mathbf{v} \leftarrow \mathbf{r} + (d/c)\mathbf{v}$, and $c \leftarrow d$, applied iteratively. We use S to denote the number of conjugate gradient iterations needed. Thus $\mathbf{x}$ is constructed using only matrix-vector products and vector additions, all which are compatible with our formulation with integral operators. The conjugate gradient method applies only to positive-definite operators. Our operator $\mathbb{A}$ is only semi-definite due to the null-space in the antisymmetric pseudo-norm. Fortunately, $\mathbf{b}$ was computed with the same pseudo-norm and has no component in the null-space of $\mathbb{A}$.

One advantage of using this iterative method with integral operators is that our algorithm is "basis-free". The representation of $\mathbf{x}$ can naturally be adaptive in γ, for example refining near the nuclei as indicated by the refinement in $\mathbf{b}$. For the estimates of computational cost, we use M to denote the cost to represent a function of γ, or integrate such a function. The antisymmetry constraint requires $N \leq M$, and in general we expect M to be much larger than N. For our numerical results, we use adaptive polynomial multiwavelets, following [18, 19]. In those works it was shown that this basis effectively eliminates basis-set error within HF.

We are left with the problem of how to construct $\mathbb{A}$ in (14) and $\mathbf{b}$ in (15). We have develop the machinery and algorithms for computing these antisymmetric inner products. Our formulation uses low-rank perturbations of matrices, thus avoiding co-factor expansions.

The computational cost for the whole method is acceptable. As noted above, the cost depends on N, r, M, I, K, and S. Although S in theory could be as many as rM, we have a very good starting point, and so expect only a very small constant number to be needed. We use $M \log M$ to denote the cost to convolve a function of γ with $1/\|\mathbf{r}\|$. Some Poisson solvers achieve this complexity, but this cost may vary with the choice of basis. We use L to denote the number of terms used to approximate the Green's function with Gaussians. The final computational cost is then

$$\mathcal{O}(KIr^2N^2[L(N + M \log M) + S(N + M)]). \tag{16}$$

For comparison, the cost to evaluate a single instance of Löwdin's rules is $\mathcal{O}(N^2(N + M))$. The size r needed in practice, and how it depends on the various parameters in the problem, is still an open question.

We have verified our approach by computing electron structure of a few elements and are in the process of accelerating the algorithm and verifying its performance against methods currently used in quantum chemistry. Further details may be found in the paper by Gregory Beylkin, Martin J. Mohlenkamp and Fernando Pérez "Approximating a Wavefunction as an Unconstrained Sum of Slater Determinants" [6] submitted for publication to Journal of Mathematical Physics. The paper is attached to this report.

References

[1] H. Agren, A. Flores-Riveros, and H. Jensen, *Evaluation of first- and second-order nonadiabatic coupling elements from large multiconfigurational self-consistent-field wave functions*, Physical Review A, 34 (1986), pp. 4606–4614.

[2] G. Beylkin, V. Cheruvu, and F. Pérez, *Fast adaptive algorithms in the non-standard form for multidimensional problems*, Appl. Comput. Harmon. Anal., (2007). Accepted for publication. APPM Preprint #550.

[3] G. Beylkin, R. Cramer, G. Fann, and R. Harrison, *Multiresolution separated representations of singular and weakly singular operators*, Appl. Comput. Harmon. Anal., 23 (2007), pp. 235–253.

[4] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 10246–10251. http://www.pnas.org/cgi/content/abstract/112329799v1.

[5] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159. http://amath.colorado.edu/pub/wavelets/papers/BEY-MOH2005.pdf.

[6] G. Beylkin, M. J. Mohlenkamp, and F. Pérez, *Approximating a wavefunction as an unconstrained sum of Slater determinants*, J. Math. Phys., (2007). Submitted. APPM Preprint #554.

[7] G. Beylkin and L. Monzón, *On approximation of functions by exponential sums*, Appl. Comput. Harmon. Anal., 19 (2005), pp. 17–48. http://amath.colorado.edu/pub/wavelets/papers/afes.pdf.

[8] R. Bro, *Parafac. tutorial & applications*, in Chemom. Intell. Lab. Syst., Special Issue 2nd Internet Conf. in Chemometrics (INCINC'96), vol. 38, 1997, pp. 149–171. http://www.models.kvl.dk/users/rasmus/presentations/parafac_tutorial/paraf.htm.

[9] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. Le Bris, and Y. Maday, *Computational quantum chemistry: a primer*, in Handbook of Numerical Analysis, Vol. X, North-Holland, Amsterdam, 2003, pp. 3–270.

[10] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \cdots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.

[11] F. Dijkstra and J. H. V. Lenthe, *On the rapid evaluation of cofactors in the calculation of nonorthogonal matrix elements*, International Journal of Quantum Chemistry, 67 (1998), pp. 77–83.

[12] ———, *Gradients in valence bond theory*, Journal of Chemical Physics, 113 (2000), pp. 2100–2108.

[13] G. Fann, G. Beylkin, R. Harrison, and K. Jordan, *Singular operators in multiwavelet bases*, IBM Journal of Research and Development, 48 (2004), pp. 161–171.

[14] P. Gilbert, *The reconstruction of a three-dimensional structure from projections and its applications to electron microscopy II. Direct methods*, Proc. R. Soc. Lond. B., (1972), pp. 89–102.

[15] T. L. Gilbert, *Multiconfiguration self-consistent-field theory for localized orbitals. I. The orbital equations*, Phys. Rev. A, 6 (1972).

[16] M. Glasser and I. Zucker, *Lattice sums*, in Theoretical Chemistry: Advances and Perspectives, H. Eyring and D. Henderson, eds., vol. 5, Academic Press, 1980, pp. 67–139.

[17] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, 3rd ed., 1996.

[18] R. Harrison, G. Fann, T. Yanai, and G. Beylkin, *Multiresolution quantum chemistry in multiwavelet bases*, in Lecture Notes in Computer Science. Computational Science-ICCS 2003, P.M.A. Sloot et. al., ed., vol. 2660, Springer, 2003, pp. 103–110.

[19] R. Harrison, G. Fann, T. Yanai, Z. Gan, and G. Beylkin, *Multiresolution quantum chemistry: basic theory and initial applications*, J. Chem. Phys., 121 (2004), pp. 11587–11598. http://amath.colorado.edu/pub/wavelets/papers/mrqc.pdf.

[20] R. A. Harshman, *Foundations of the parafac procedure: Model and conditions for an "explanatory" multi-mode factor analysis*, Working Papers in Phonetics 16, UCLA, 1970. http://publish.uwo.ca/∼harshman/wpppfac0.pdf.

[21] M. H. Kalos, *Monte Carlo calculations of the ground state of three- and four-body nuclei*, Phys. Rev. (2), 128 (1962), pp. 1791–1795.

[22] ———, *Monte Carlo integration of the Schrödinger equation*, Trans. New York Acad. Sci. (2), 26 (1963/1964), pp. 497–504.

[23] P. M. Kroonenberg and J. de Leeuw, *Principal component analysis of three-mode data by means of alternating least squares algorithms*, Psychometrika, 45 (1980), pp. 69–97.

[24] S. E. Leurgans, R. A. Moyeed, and B. W. Silverman, *Canonical correlation analysis when the data are curves*, J. Roy. Statist. Soc. Ser. B, 55 (1993), pp. 725–740.

[25] A. Lüchow and R. Fink, *On the systematic improvement of fixed-node diffusion quantum Monte Carlo energies using natural orbital CI guide functions*, J. Chem. Phys., 113 (2000), pp. 8457–8463.

[26] S. P. Rudin, *Configuration Interaction with Non-orthogonal Slater Determinants Applied to the Hubbard Model, Atoms, and Small Molecules*, PhD thesis, The Ohio State University, 1997.

[27] C. D. Sherrill and H. F. Schaefer III, *The configuration interaction method: Advances in highly correlated approaches*, Advances in Quantum Chemisty, 127 (1999), pp. 143–269.

[28] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis. Applications in the Chemical Sciences*, John Wiley & Sons, 2004.

[29] T. Yanai, G. Fann, Z. Gan, R. Harrison, and G. Beylkin, *Multiresolution quantum chemistry: Analytic derivatives for Hartree-Fock and density functional theory*, J. Chem. Phys., 121 (2004), pp. 2866–2876.

[30] ———, *Multiresolution quantum chemistry: Hartree-Fock exchange*, J. Chem. Phys., 121 (2004), pp. 6680–6688.

[31] J. Zanghellini, *Multi-Electron Dynamics in the Ionization of Molecules by Strong Laser Pulses*, PhD thesis, Vienna University of Technology, Vienna, Austria, 2004.

Fast Adaptive Algorithms in the Non-Standard Form for Multidimensional Problems

Gregory Beylkin, Vani Cheruvu and Fernando Pérez

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526, United States

Abstract

We present a fast, adaptive multiresolution algorithm for applying integral operators with a wide class of radially symmetric kernels in dimensions one, two and three. This algorithm is made efficient by the use of separated representations of the kernel. We discuss operators of the class $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \geq 0$ and $0 < \alpha < 3/2$, and illustrate the algorithm for the Poisson and Schrödinger equations in dimension three. The same algorithm may be used for all operators with radially symmetric kernels approximated as a weighted sum of Gaussians, making it applicable across multiple fields by reusing a single implementation.

This fast algorithm provides controllable accuracy at a reasonable cost, comparable to that of the Fast Multipole Method (FMM). It differs from the FMM by the type of approximation used to represent kernels and has an advantage of being easily extendable to higher dimensions.

Key words: Separated representation; multiwavelets; adaptive algorithms; integral operators.

1 Introduction

For a number of years, the Fast Multipole Method (FMM) [1,2,3] has been the method of choice for applying integral operators to functions in dimensions $d \leq 3$. On the other hand, multiresolution algorithms in wavelet and multiwavelet bases introduced in [4] for the same purpose were not efficient

[1] This research was partially supported by DOE grant DE-FG02-03ER25583, DOE/ORNL grant 4000038129 and DARPA/ARO grant W911NF-04-1-0281.

enough to be practical in more than one dimension. Recently, with the introduction of separated representations [5,6,7], practical multiresolution algorithms in higher dimensions [8,9,10,11] became available as well. In this paper we present a new fast, adaptive algorithm for applying a class of integral operators with radial kernels in dimensions $d = 1, 2, 3$, and we briefly discuss its extension to higher dimensions.

In physics, chemistry and other applied fields, many important problems may be formulated using integral equations, typically involving Green's functions as their kernels. Often such formulations are preferable to those via partial differential equations (PDEs). For example, evaluating the integral expressing the solution of the Poisson equation in free space (the convolution of the Green's function with the mass or charge density) avoids issues associated with the high condition number of a PDE formulation. Integral operators appear in fields as diverse as electrostatics, quantum chemistry, fluid dynamics and geodesy; in all such applications fast and accurate methods for evaluating operators on functions are needed.

The FMM and our approach both employ approximate representations of operators to yield fast algorithms. The main difference lies in the type of approximations that are used. For example, for the Poisson kernel $1/r$ in dimension $d = 3$, the FMM [3] uses a plane wave approximation starting from the integral

$$\frac{1}{r} = \frac{1}{2\pi} \int_0^\infty e^{-\lambda(z-z_0)} \int_0^{2\pi} e^{i\lambda((x-x_0)\cos\alpha + (y-y_0)\sin\alpha)} d\alpha d\lambda, \tag{1}$$

where $r = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2}$. The elegant approximation derived from this integral in [3] is valid within a solid angle, and thus requires splitting the application of an operator into directional regions; the number of such regions grows exponentially with dimension. For the same kernel, our approach starts with the integral

$$\frac{1}{r} = \frac{2}{\sqrt{\pi}} \int_{-\infty}^\infty e^{-r^2 e^{2s} + s} ds, \tag{2}$$

and its discretization with finite accuracy ϵ yields a spherically symmetric approximation as a weighted sum of gaussians. Other radial kernels can be similarly treated by a suitable choice of integrals. The result is a separated representation of kernels and, therefore, an immediate reduction in the cost of their application. This difference in the choice of approximation dictates the differences in the corresponding algorithms. In dimension $d \leq 3$ both approaches are practical and yield comparable performance. The key advantage of our approach is its straightforward extensibility to higher dimensions [6,7].

Given an arbitrary accuracy ϵ, we effectively represent kernels by a set of exponents and weights describing the terms of the gaussian approximation of integrals like in (2). The number of terms in such sum is roughly proportional to $\log(\epsilon^{-1})$, or a low power of $\log(\epsilon^{-1})$, depending on the operator. Since operators are fully described up to an accuracy ϵ by the exponents and weights of the sum of gaussians, a single algorithm applies all such operators. These include operators such as $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \geq 0$ and $0 < \alpha < 3/2$, and certain singular operators such as the projector on divergence-free functions. Since many physically significant operators depend only on the distance between interacting objects, our approach is directly applicable to problems involving a wide class of operators with radial kernels.

We combine separated and multiresolution representations of kernels and use multiwavelet bases [12] that provide *inter alia* a method for discretizing integral equations, as is the case in quantum chemistry [8,9,11,10]. This choice of multiresolution bases accommodates integral and differential operators as well as a wide variety of boundary conditions, without degrading the order of the method [13,14]. Multiwavelet bases retain the key desired properties of wavelet bases, such as vanishing moments, orthogonality, and compact support. Due to the vanishing moments, wide classes of integro-differential operators have an *effectively sparse* matrix representation, i.e., they differ from a sparse matrix by an operator with small norm. Some of the basis functions of multiwavelet bases are discontinuous, similar to those of the Haar basis and in contrast to wavelets with regularity (see e.g. [15,16]). The usual choices of scaling functions for multiwavelet bases are either the scaled Legendre or interpolating polynomials. Since these are also used in the discontinuous Galerkin and discontinuous spectral elements methods, our approach may also be seen as an adaptive extension of these methods.

The algorithm for applying an operator to a function starts with computing its adaptive representation in a multiwavelet basis, resulting in a 2^d-tree with blocks of coefficients at the leaves. Then the algorithm adaptively applies the (modified) separated non-standard form [4] of the operator to the function by using only the necessary blocks as dictated by the function's tree representation. We note that in higher dimensions, $d \gg 3$, functions need to be in a separated representation as well, since the usual constructions via bases or grids are prohibitive (see [6,7]).

We start in Section 2 by recalling the basic notions of multiresolution analysis, non-standard operator form and adaptive representation of functions underlying our development. We then consider the separated representation for radially symmetric kernels in Section 3, and use it to efficiently extend the modified ns-form to multiple spatial dimensions in Section 4. We pay particular attention to computing the band structure of the operator based on one dimensional information. We use this construction in Section 5 to introduce the adaptive algorithm for application of multidimensional operators in the modified ns-form, and illustrate its performance in Section 6. We consider two examples: the Poisson equation in free space and the ground state of the Hydrogen atom. We conclude with a brief discussion in Section 7.

2 Preliminary considerations

This section and Appendix are provided for the convenience of the reader in order to keep this paper reasonably self-contained. We provide background material and introduce necessary notation.

The essence of our approach is to decompose the operator using projectors on a Multiresolution Analysis (MRA), and to efficiently apply its projections using a separated representation. We use the decomposition of the operator into the ns-form [4], but we organize it differently (thus, *modified* ns-form) to achieve greater efficiency. This modification becomes important as we extend this algorithm to higher dimensions.

In this section we introduce notation for MRA, describe the adaptive representation of functions and associated data structures, introduce the modified ns-form and an algorithm for its adaptive application in dimension $d = 1$ as background material for the multidimensional case.

2.1 Multiresolution analysis

Let us consider the multiresolution analysis as a decomposition of $L^2([0,1]^d)$ into a chain of subspaces

$$\mathbf{V}_0 \subset \mathbf{V}_1 \subset \mathbf{V}_2 \subset \dots \subset \mathbf{V}_n \subset \dots,$$

so that $L^2([0,1]^d) = \overline{\cup_{j=0}^{\infty} \mathbf{V}_j}$. We note that our indexing of subspaces (increasing towards finer scales) follows that in [13], and is the reverse of that in [4,15]. On each subspace $\mathbf{V}_j$, we use the tensor product basis of scaling functions obtained using the functions $\phi^j_{kl}(x)$ $(k = 0, \dots, p-1)$ which we briefly describe in Appendix.

The wavelet subspaces $\mathbf{W}_j$ are defined as the orthogonal complements of $\mathbf{V}_j$ in $\mathbf{V}_{j-1}$, thus

$$\mathbf{V}_n = \mathbf{V}_0 \oplus_{j=0}^{n} \mathbf{W}_j \,.$$

Introducing the orthogonal projector on $\mathbf{V}_j$, $\mathbf{P}_j : L^2([0,1]^d) \to \mathbf{V}_j$ and considering an operator $\mathbf{T} : L^2([0,1]^d) \to L^2([0,1]^d)$, we define its projection $\mathbf{T}_j : \mathbf{V}_j \to \mathbf{V}_j$ as $\mathbf{T}_j = \mathbf{P}_j \mathbf{T} \mathbf{P}_j$. We also consider the orthogonal projector $\mathbf{Q}_j : L^2([0,1]^d) \to \mathbf{W}_j$, defined as $\mathbf{Q}_j = \mathbf{P}_{j+1} - \mathbf{P}_j$.

2.2 Adaptive representation of functions

Let us describe an adaptive refinement strategy for construction multiresolution representations of functions $f : B \to B$, where $B = [0,1]^d$. We proceed by recursive binary subdivision of the box B, so the basic structure representing our functions is a 2^d-tree with arrays of coefficients stored at the leaves (terminal nodes) and no data stored on internal nodes. On each box obtained via this subdivision, our basis is a tensor product of orthogonal polynomials of degree $k = 0, \ldots, p-1$ in each variable, as described in Appendix 8. Therefore, the leaves carry d-dimensional arrays of p^d coefficients which may be used to approximate function values anywhere in the box corresponding to the spatial region covered by it, via (30) or its equivalent for higher values of d. For conciseness, we will often refer to these d-dimensional arrays of coefficients stored at tree nodes as *function blocks*.

This adaptive function decomposition algorithm is similar to that used in [17]. Such construction formally works in any dimension d. However, since its complexity scales exponentially with d, its practical use is restricted to fairly low dimension, e.g. $d \lesssim 4$. In higher dimensions, alternate representation strategies for functions such as [6,7] should be considered. In high dimensions, these strategies deal with the exponential growth of complexity by using controlled approximations that have linear cost in d.

For simplicity, we will describe the procedure for the one-dimensional case since the extension to dimensions $d = 2, 3$ is straightforward. Since we can not afford to construct our representation by starting from a fine scale (especially in $d = 3$), we proceed by successive refinements of an initial coarse sampling. This approach may result in a situation where the initial sampling is insufficient to resolve a rapid change in a small volume; however, in practical applications such situations are rare and may be avoided by an appropriate choice of the initial sampling scale.

Let $B_l^j = [2^{-j}l, 2^{-j}(l+1)]$, $l = 0, \ldots, 2^j - 1$, represent a binary subinterval on scale j. We denote by $f_l^j = \left\{ f_l^j(x_k) \right\}_{k=0}^{p-1}$ the vector of values of the function f on the Gaussian nodes in B_l^j. From these values we compute the coefficients $\left\{ s_{kl}^j \right\}_{k=0}^{p-1}$ (see (32) in Appendix) and interpolate $f(x)$ for any $x \in B_l^j$ by using (30). We then subdivide B_l^j into two child intervals, B_{2l}^{j+1} and B_{2l+1}^{j+1}, and evaluate the function f on the Gaussian nodes in B_{2l}^{j+1} and B_{2l+1}^{j+1}. We then interpolate f by using the coefficients $\left\{ s_{kl}^j \right\}_{k=0}^{p-1}$ from their parent interval and denote by $\widetilde{f}_{2l}^{j+1}$ and $\widetilde{f}_{2l+1}^{j+1}$ the vectors of interpolated values on the two subintervals. Now, if for a given tolerance ϵ either $\left\| f_{2l}^{j+1} - \widetilde{f}_{2l}^{j+1} \right\| > \epsilon$ or $\left\| f_{2l+1}^{j+1} - \widetilde{f}_{2l+1}^{j+1} \right\| > \epsilon$, we repeat the process recursively for both subintervals,

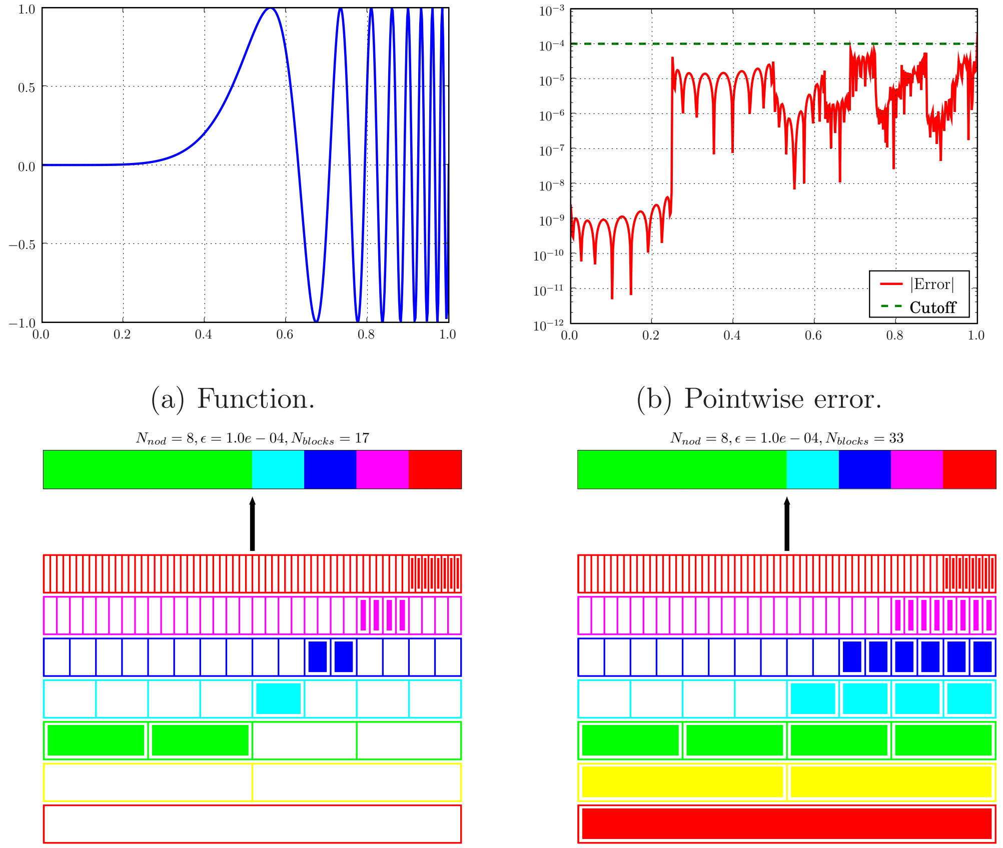
B_{2l}^{j+1} and B_{2l+1}^{j+1}; otherwise, we keep the coefficients $\left\{ s_{kl}^{j} \right\}_{k=0}^{p-1}$ to represent the function on the entire interval B_l^j. This interval then becomes a leaf in our tree.

At this stage we use the ℓ^∞ norm, thus constructing an approximation $\tilde{f}$ to the original function f such that $\left\| f - \widetilde{f} \right\|_\infty < \epsilon$, which immediately implies that $\left\| f - \widetilde{f} \right\|_2 < \epsilon$. This estimate clearly extends to any dimension. Once the approximation with ℓ^∞ norm is constructed, the corresponding tree may be pruned if an application only requires the approximation to be valid in the ℓ^2 norm. We start this process on the finest scale and simply remove all blocks whose cumulative contribution is below ϵ. Other norms, such as H_1, can be accommodated by appropriately weighing the error tolerance with a scale-dependent factor in the initial (coarse to fine) decomposition process.

The complete decomposition algorithm proceeds by following the above recipe, starting with an initial coarse scale (typically $j = 0$) and continuing recursively until the stopping criterion is met for all subintervals. In practice, we choose a stopping scale $j_{\max}$, beyond which the algorithm will not attempt to subdivide any further. Reaching $j_{\max}$ means that the function has significant variations which are not accurately resolved over an interval of width $2^{-j_{\max}}$ using a basis of order p. A pseudo-code listing of this process is presented as Algorithm 1.

Algorithm 1 Adaptive Function Decomposition.

```
Start at a coarse scale, typically $j = 0$.
Recursively, for all boxes $b^j$ on scale $j$, proceed as follows:
Construct the list $C$ of $2^d$ child boxes $b^{j+1}$ on scale $j + 1$.
Compute the values of the function $f(b^j)$ at the $p^d$ Gauss-Legendre quadrature nodes in $b^j$.
for all boxes $b^{j+1} \in C$ do
    From $f(b^j)$, interpolate to the Gauss-Legendre quadrature nodes in $b^{j+1}$, producing values $\widetilde{f}(b^{j+1})$.
    Compute the values of $f(b^{j+1})$ at the Gauss-Legendre nodes of $b^{j+1}$, by direct evaluation.
    if $\left\| f(b^{j+1}) - \tilde{f}(b^{j+1}) \right\|_\infty > \epsilon$ then
        Recursively repeat the entire process for all boxes $b^{j+1} \in C$.
    end if
end for
# Geting here means that the interpolation from the parent was successful for all child boxes. We store the parent's coefficients from $b^j$ in the function tree.
```



(a) Function.

(b) Pointwise error.

(c) Adaptive tree

(d) Redundant tree

Figure 1. The function $f(x) = \sin(16\pi x^6)$ shown in (a), is decomposed with $p = 8$ and $\epsilon = 10^{-4}$; (b) shows the pointwise approximation error. (c) is the resulting adaptive tree, where smaller subdivisions are required in regions with higher frequency content. (d) is the redundant tree associated with this adaptive decomposition, where all internal nodes have been filled with data.

2.2.1 Tree structures for representing functions

The decomposition Algorithm 1 naturally leads to a tree data structure to represent functions, with the leaves of the tree corresponding to the spatial intervals over which the multiwavelet basis provides a sufficiently accurate local approximation. By using (30) or its higher-dimensional extensions, the only data needed to approximate $f(x)$ anywhere in B is the array of basis coefficients on these leaves. Thus we use a tree structure where the leaves store these coefficients and the internal nodes do not contain any data (and are effectively removed since we use hash tables for storage). We will refer to this structure as an *adaptive tree.* Each level in the tree corresponds to a scale in the MRA, with the root node corresponding to the coarsest projection $f_0 \in \mathbf{V}_0$.

Now, in order to apply the modified non-standard form of an operator to a function, we will show in the next section that we also need the basis coefficients corresponding to internal nodes of the tree. Hence, from the adaptive tree data structure we will compute a similar tree but where we do keep the coefficients of scaling functions on *all* nodes (leaves and internal).

The coefficients on the internal nodes are redundant since they are computed from the function blocks stored in the leaves. We will thus refer to the tree containing coefficients on all nodes as a *redundant tree*. It is constructed recursively starting from the leaves, by projecting the scaling coefficients from all sibling nodes onto their parent node, using the decomposition (41).

Figure 1 shows both the adaptive and the redundant trees for a sample function. This figure displays the coarsest scales at the bottom and progressively finer ones further up, with filled boxes representing nodes where scaling coefficients are stored and empty boxes indicating nodes with no data in them (these do not need to be actually stored in the implementation).

2.3 Modified ns-form

The non-standard form [4] (see also [13] for the version specialized for multiwavelets) of the operator $\mathbf{T}$ is the collection of components of the telescopic expansion

$$\mathbf{T}_n = (\mathbf{T}_n - \mathbf{T}_{n-1}) + (\mathbf{T}_{n-1} - \mathbf{T}_{n-2}) + \cdots + \mathbf{T}_0 = \mathbf{T}_0 + \sum_{j=0}^{n-1} (\mathbf{A}_j + \mathbf{B}_j + \mathbf{C}_j), \quad (3)$$

where $\mathbf{A}_j = \mathbf{Q}_j\mathbf{T}\mathbf{Q}_j$, $\mathbf{B}_j = \mathbf{Q}_j\mathbf{T}\mathbf{P}_j$, and $\mathbf{C}_j = \mathbf{P}_j\mathbf{T}\mathbf{Q}_j$. The main property of this expansion is that the rate of decay of the matrix elements of the operators $\mathbf{A}_j$, $\mathbf{B}_j$ and $\mathbf{C}_j$ away from the diagonal is controlled by the number of vanishing moments of the basis and, for a finite but arbitrary accuracy ϵ, the matrix elements outside a certain band can be set to zero resulting in an error of the norm less than ϵ. Such behavior of the matrix elements becomes clear if we observe that the derivatives of kernels of Calderón-Zygmund and pseudo-differential operators decay faster than the kernel itself. If we use the Taylor expansion of the kernel to estimate the matrix elements away from the diagonal, then the size of these elements is controlled by a high derivative of the kernel since the vanishing moments eliminate the lower order terms [4]. We note that for periodic kernels the band is measured as a periodic distance from the diagonal, resulting in filled-in ‘corners’ of a matrix representation.

Let us introduce notation to show how the telescopic expansion (3) is used when applying an operator to a function. If we apply the projection of the operator $\mathbf{T}_{j-1}$ not on its “natural scale” $j-1$, but on the finer scale j, we denote its upsampled version as $\uparrow (\mathbf{T}_{j-1})$. In the matrix representation of

$\mathbf{T}_{j-1}$, this operation results in the doubling of the matrix size in each direction. This upsampling $\uparrow(\cdot)$ and downsampling $\downarrow(\cdot)$ notation will also be used for projections of functions.

With this notation, computing $g = \mathbf{T}f$ via (3) splits across scales,

$$
\begin{aligned}
\hat{g}_0 &= \mathbf{T}_0 f_0 \\
\hat{g}_1 &= [\mathbf{T}_1 - \uparrow(\mathbf{T}_0)] f_1 \\
\hat{g}_2 &= [\mathbf{T}_2 - \uparrow(\mathbf{T}_1)] f_2 \\
\cdots&\cdots\cdots \\
\hat{g}_j &= [\mathbf{T}_j - \uparrow(\mathbf{T}_{j-1})] f_j \\
\cdots&\cdots\cdots
\end{aligned}
\tag{4}
$$

where $f_j = \mathbf{P}_j f$.

As in the application of the usual ns-form in [4], to obtain g_n after building the set $\{\hat{g}_0, \hat{g}_1, \ldots, \hat{g}_n\}$, we have to compute

$$g_n = \hat{g}_n + \uparrow(\hat{g}_{n-1} + \uparrow(\hat{g}_{n-2} + \uparrow(\hat{g}_{n-3} + \ldots + (\uparrow \hat{g}_0)\ldots))) \,. \tag{5}$$

The order of the parentheses in this expression is essential, as it indicates the order of the actual operations which are performed starting on the coarsest subspace $\mathbf{V}_0$. For example, if the number of scales $n = 4$, then (5) yields $g_4 = \hat{g}_4 + \uparrow(\hat{g}_3 + \uparrow(\hat{g}_2 + \uparrow(\hat{g}_1 + (\uparrow \hat{g}_0))))$, describing the sequence of necessary operations.

Unfortunately, the sparsity of the non-standard form induced by the vanishing moments of bases is not sufficient for fast practical algorithms in dimensions other than $d = 1$. For algorithms in higher dimensions, we need an additional structure for the remaining non-zero coefficients of the representation. We will use separated representations (see Section 3) introduced in [6,18] and first applied in a multiresolution setting in [8,9,10,11]. Within the retained bands, the components of the non-standard form are stored and applied in a separated representation and, as a result, the numerical application of operators becomes efficient in higher dimensions.

2.4 Modified ns-form in 1D

Let us describe a one-dimensional construction for operators on $L^2([0,1])$ to introduce all the features necessary for a multidimensional algorithm. Since we use banded versions of operators, we need to introduce the necessary bookkeeping.

The "template" for the band structure on scale j comes from the band on the previous scale $j-1$. For each block on scale $j-1$, the upsampling operation $\uparrow(\mathbf{T}_{j-1})$ creates four blocks (all combinations of even/odd row and column indices). We insist on maintaining the strict correspondence between these four blocks and those of $\mathbf{T}_j$. For this reason the description of the retained blocks of $\mathbf{T}_j$ involves the parity of their row and column indices. Let us denote the blocks in the matrix representing $\mathbf{T}_j$ by $t^{j;ll'}$, where $l, l' = 0 \dots 2^{j-1}$. Individual elements within these blocks are indexed as $t^{j;ll'}_{ii'}$, where $i, i' = 0, \dots, p-1$, and p is the order of the multiwavelet basis. For a given width of the band b_j, we keep the operator blocks $t^{j;ll'}$ with indices satisfying

$$\begin{aligned} l - b_j + 1 \le l' \le & \quad l + b_j, \quad \text{for even } l, \\ l - b_j \qquad \le l' \le & \; l + b_j - 1, \text{ for odd } l. \end{aligned} \tag{6}$$

We denote the banded operators where we keep only blocks satisfying (6) as $\mathbf{T}_j^{b_j}$ and $\uparrow(\mathbf{T}_{j-1})^{b_j}$. If we downsample the operator $\uparrow(\mathbf{T}_{j-1})^{b_j}$ back to its original scale $j-1$, then (6) leads to the band described by the condition

$$l - \lfloor b_j/2 \rfloor \le l' \le l + \lfloor b_j/2 \rfloor\,, \tag{7}$$

where $\lfloor b_j/2 \rfloor$ denotes the integer part of $b_j/2$. We denote the banded operator on scale $j-1$ as $\mathbf{T}_{j-1}^{\lfloor b_j/2 \rfloor}$, where we retain blocks satisfying (7).

If we now rewrite (4) *keeping only blocks within the bands* on each scale, we obtain

$$\begin{aligned} \hat{g}_0 &= \mathbf{T}_0 f_0 \\ \hat{g}_1 &= [\mathbf{T}_1^{b_1} - \uparrow(\mathbf{T}_0)^{b_1}] f_1 = \mathbf{T}_1^{b_1} f_1 - \uparrow(\mathbf{T}_0)^{b_1} f_1 \\ \hat{g}_2 &= [\mathbf{T}_2^{b_2} - \uparrow(\mathbf{T}_1)^{b_2}] f_2 = \mathbf{T}_2^{b_2} f_2 - \uparrow(\mathbf{T}_1)^{b_2} f_2 \\ &\cdots\cdots\cdots \\ \hat{g}_j &= [\mathbf{T}_j^{b_j} - \uparrow(\mathbf{T}_{j-1})^{b_j}] f_j = \mathbf{T}_j^{b_j} f_j - \uparrow(\mathbf{T}_{j-1})^{b_j} f_j \\ &\cdots\cdots\cdots \end{aligned} \tag{8}$$

For any arbitrary but finite accuracy, instead of applying the full $[\mathbf{T}_j - \uparrow(\mathbf{T}_{j-1})]$, we will only apply its banded approximation.

A simple but important observation is that

$$\downarrow([\uparrow(\mathbf{T}_{j-1})] f_j) = \mathbf{T}_{j-1} f_{j-1}\,, \tag{9}$$

which follows from the fact that $\mathbf{Q}_j \mathbf{P}_j = \mathbf{P}_j \mathbf{Q}_j = 0$, since these are orthogonal projections. Thus, we observe that $\downarrow(\uparrow(\mathbf{T}_{j-1})^{b_j}) = \mathbf{T}_{j-1}^{\lfloor b_j/2 \rfloor}$; so instead of applying $\uparrow(\mathbf{T}_{j-1})^{b_j} f_j$ on scale j, we can obtain the same result using (9), so that

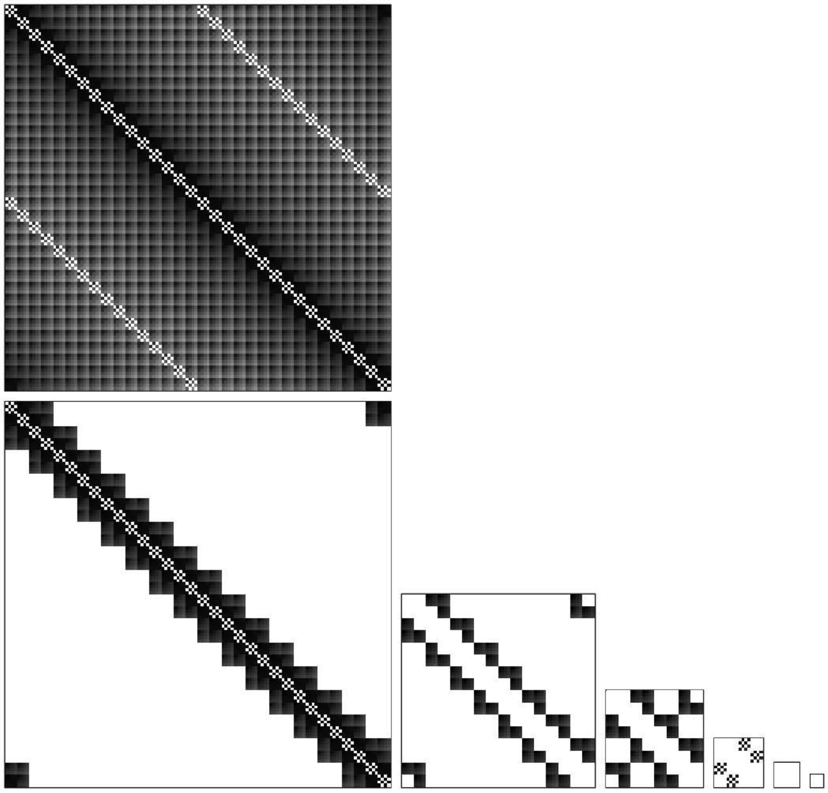


Figure 2. Modified non-standard form of the convolution operator in (11) in a multiwavelet basis, with white representing 0 and black representing large values. The top matrix is the projection of this operator on $\mathbf{V}_5$, resulting in a dense matrix. The lower half depicts the multiresolution representation in (10) with only the blocks that are actually retained for a given accuracy. We will call the leftmost matrix in this series a *whole band matrix* and all others *outer band matrices.* The two empty outer band matrices on scales $j = 0, 1$ are explained in the main text.

$\uparrow \left(\mathbf{T}_{j-1}^{\lfloor b_j/2 \rfloor} f_{j-1}\right) =\uparrow (\mathbf{T}_{j-1})^{b_j} f_j$. Therefore, we will compute only $\mathbf{T}_{j-1}^{\lfloor b_j/2 \rfloor} f_{j-1}$ on scale $j-1$ and combine it with computing $\mathbf{T}_{j-1}^{b_{j-1}} f_{j-1}$. Incorporating this into (5), we arrive at

$$g_n = \mathbf{T}_n^{b_n} f_n + \uparrow \left[\left(\mathbf{T}_{n-1}^{b_{n-1}} - \mathbf{T}_{n-1}^{\lfloor b_n/2 \rfloor}\right) f_{n-1} + \right. \tag{10}$$
$$\left. \uparrow \left[\left(\mathbf{T}_{n-2}^{b_{n-2}} - \mathbf{T}_{n-2}^{\lfloor b_{n-1}/2 \rfloor}\right) f_{n-2} + \ldots + \left[\uparrow \left[\left(\mathbf{T}_0 - \mathbf{T}_0^{\lfloor b_1/2 \rfloor}\right) f_0\right]\right] \ldots\right]\right].$$

Using this expression yields an efficient algorithm for applying an operator, as on each scale j, $\mathbf{T}_j^{b_j} - \mathbf{T}_j^{[b_{j+1}/2]}$ is a sparse object, due to the cancellation which occurs for most of the blocks. In particular, $\mathbf{T}_j^{b_j} - \mathbf{T}_j^{[b_{j+1}/2]}$ is missing the blocks near the diagonal, and we will refer to it as an *outer band matrix.* We will call $\mathbf{T}_j^{b_j}$ a *whole band matrix* as it contains both the inner and outer bands.

The structure of these matrices is illustrated in Figure 2. The two empty scales $j = 0, 1$ arise due to the complete cancellation of blocks on these scales. We note that the modified non-standard form is constructed adaptively in the number of scales necessary for a given function. For just two scales, this construction will have the scale $j = 1$ non-empty. Given an adaptive decomposition of a function on n scales, we precompute the modified non-standard form (depicted in Figure 2 for five scales) on all scales $n, n-1, \ldots, 1$. For matrices requiring $2^n \cdot 2^n$ blocks on the finest scale n, we need to keep and apply only $\mathcal{O}(2^n)$ blocks, as with the original non-standard form in [4].

2.4.1 Adaptive application

Let us show how to use the multiscale representation in (10) to apply the operator T to a function f with controlled accuracy ϵ. We describe an adaptive application of the operator to a function, where we assume (as is often the case) that the tree structure of the input is sufficient to adequately describe the output with accuracy ϵ. This assumption will be removed later.

Our algorithm uses the structure shown in Figure 2 in an adaptive fashion. We copy the structure of the redundant tree for the input function, and use that as a template to be filled for the output g. For each node of the output tree we determine whether it is a leaf or an internal node: for leaves, we must apply a whole band matrix on the scale of that node, such as the leftmost matrix for $j = 5$ in the example shown in Figure 2. For internal nodes, we apply an outer band matrix (for that scale). We note that our construction of the operator produces both whole and outer band matrices for all scales, and we simply choose the appropriate kind for each node of output as needed. Upon completion of this process, we apply the projection (5) to construct the final adaptive tree representing the output.

Algorithm 2 returns an adaptive tree representing the function g. This tree contains sufficient information to evaluate g at arbitrary points by interpolation and may be used as an input in further computations.

We note that Step 2 in Algorithm 2 naturally resolves the problem that is usually addressed by mortar methods, see e.g. [19,20,21,22]. Since adaptive representations have neighboring blocks of different sizes, they encounter difficulties when applying non-diagonal operators, as they require blocks which do not exist on that scale. Our approach simply constructs these as needed and caches them for reuse, without requiring any additional consideration on the part of the user.

Algorithm 2 Adaptive non-standard form operator application in $d = 1$, $g = \mathbf{T}f$

```
**Initialization:** Construct the redundant tree for $f$ and copy it as skeleton tree for $g$ (see Section 2.2.1).
for all scales $j = 0, \dots, n-1$ do
  for all function blocks $g_l^j$ in the tree for $g$ at scale $j$ do
    # Step 1. Determine the list of all contributing blocks of the modified ns-form $\mathbf{T}_{ll'}^j$ (see Section 2.3):
    if $g_l^j$ belongs to a leaf then
      Read operator blocks $\mathbf{T}_{ll'}^j$ from row $l$ of whole band matrix $\mathbf{T}_j^{b_j}$.
    else
      Read operator blocks $\mathbf{T}_{ll'}^j$ from row $l$ of outer band matrix $\mathbf{T}_j^{b_j} - \mathbf{T}_j^{[b_{j+1}/2]}$.
    end if
    # Step 2. Find the required blocks $f_{l'}^j$ of the input function $f$:
    if function block $f_{l'}^j$ exists in the redundant tree for $f$ then
      Retrieve it.
    else
      Create it by interpolating from a coarser scale and cache for reuse.
    end if
    # Step 3. Output function block computation:
    Compute the resulting output function block according to $\hat{g}_l^j = \sum_{l'} \mathbf{T}_{ll'}^j f_{l'}^j$, where the operation $\mathbf{T}_{ll'}^j f_{l'}^j$ indicates a regular matrix-vector multiplication.
  end for
end for
# Step 4. Adaptive projection:
Project resulting output function blocks $\hat{g}_l^j$ on all scales into a proper adaptive tree by using Eq. (5).
Discard from the resulting tree unnecessary function blocks at the requested accuracy.
Return: the function $g$ represented by its adaptive tree.
```

2.4.2 Numerical example

Let us briefly illustrate the application of the modified ns-form with an example of a singular convolution on the unit circle, the operator with the kernel $K(x) = \cot(\pi x)$,

$$(Cf)(y) = \text{p.v.} \int_0^1 \cot(\pi(y-x))\, f(x)\, dx, \tag{11}$$

a periodic analogue of the Hilbert transform. In order to find its representation in multiwavelet bases, we compute

$$r_{ii'}^{j;l} \;=\; 2^{-j} \int_{-1}^{1} K(2^{-j}(x+l))\, \Phi_{ii'}(x)\, dx \;=\; 2^{-j} \int_{-1}^{1} \cot(\pi\, 2^{-j}(x+l))\, \Phi_{ii'}(x)\, dx\,, \tag{12}$$

where $\Phi_{ii'}(x)$, $i, i' = 0, \ldots, k-1$ are cross-correlation functions described in Appendix 8.4 and $l = 0, \pm 1, \pm 2, \ldots 2^j - 1$. We compute $r_{ii'}^{j;l}$ using the convergent integrals

$$r_{ii'}^{j;l} \;=\; 2^{-j} \sum_{k=i'-i}^{i'+i} c_{ii'}^{k} \int_{0}^{1} \Phi_{k,0}^{+}(x)\, \Big(\cot(\pi\, 2^{-j}(x+l)) + (-1)^{i+i'} \cot(\pi\, 2^{-j}(-x+l))\Big)\, dx,$$

where $\Phi_{k,0}^{+}$ is a polynomial described in Appendix 8.4. In our numerical experiment, we apply (11) to the periodic function on $[0,1]$,

$$f(x) = \sum_{k \in \mathbb{Z}} e^{-a(x+k-1/2)^2},$$

which yields

$$(Cf)(y) = -\sum_{k \in \mathbb{Z}} e^{-a(y+k-1/2)^2} \mathrm{Erfi}[\sqrt{a}(y+k-1/2)] = i\sqrt{\frac{\pi}{a}} \sum_{n \in \mathbb{Z}} \mathrm{sign}(n) e^{-n^2\pi^2/a} e^{2\pi i n y}, \tag{13}$$

where $e^{-y^2}\mathrm{Erfi}(y) = \frac{2}{\sqrt{\pi}} \int_0^y e^{s^2-y^2} ds$. Expression (13) is obtained by first observing that the Hilbert transform of e^{-ax^2} is $-e^{-ay^2}\mathrm{Erfi}(\sqrt{a}y)$, and then evaluating the lattice sum, noting that (see [23, formula 4.3.91])

$$\cot(\pi x) = \frac{1}{\pi}\Big(\frac{1}{x} + \sum_{k=1}^{\infty} \frac{2x}{x^2-k^2}\Big).$$

Table 1 summarizes the numerical construction of this solution for $a = 300$, at various requested precisions. Optimal performance is obtained by adjusting the order of the basis p as a function of the requested precision, to ensure that the operator remains a banded matrix with small band, and that the adaptive representation of the input function requires a moderate number of scales. The resulting numerical error (as compared to the exact analytical solution), measured in the ℓ^2 norm, is shown in the last column. Figure 3 shows the input and results for this example, as well as the point-wise error for the case where $\epsilon_{\mathrm{req}} = 10^{-12}$ and $p = 14$ (the last row in the table).

3 Separated representations of integral kernels

The approach we've discussed so far does not efficiently generalize to the application of non-separable multidimensional integral kernels. Since several

p	Scales	N_{blocks}	ϵ	E_2
5	[2,3,4]	8	10^{-3}	$1.5 \cdot 10^{-4}$
8	[2,4,5]	12	10^{-6}	$1.3 \cdot 10^{-7}$
11	[2,4,5]	14	10^{-9}	$1.1 \cdot 10^{-10}$
14	[3,4,5]	16	10^{-12}	$4.4 \cdot 10^{-13}$

Table 1
Results from evaluating (13) with our algorithm. The order of the basis p is adjusted as a function of the requested precision ϵ. The second column indicates scales present in the adaptive tree for the input. The third column shows the total number of blocks of coefficients in this tree. The last column (E_2) shows the actual error of the computed solution in the ℓ^2 norm.

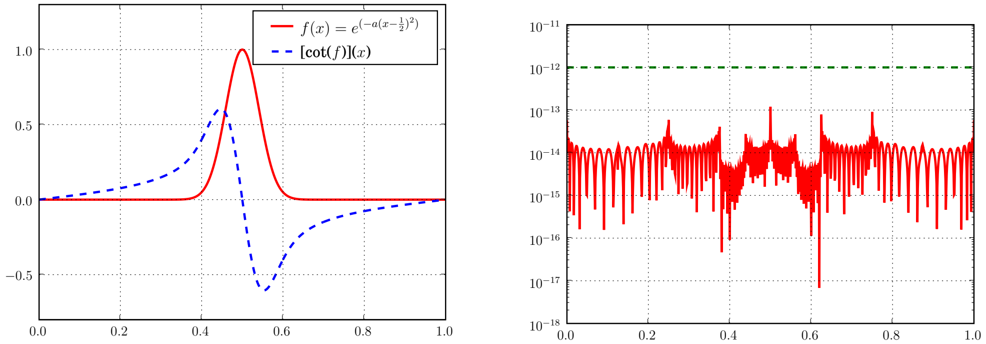


Figure 3. Results of applying the cotangent kernel to a periodized Gaussian using basis of order $p = 14$ (the last row in Table 1). The pointwise error is shown on the right for a requested accuracy of $\epsilon = 10^{-12}$.

physically important kernels belong to this category (e.g. the Poisson kernels in $d = 2$ and d=3), additional tools are needed. We now describe the key idea that allows us to perform this generalization to $d > 1$.

We use the separated representation of operators introduced in [6,24] to reduce the computational cost of the straightforward generalization of the multiresolution approach in [4]. Such representations are particularly simple for convolution operators and are based on approximating kernels by a sum of Gaussians [24,8,9,11,10]. This approximation has a multiresolution character by itself and requires a remarkably small number of terms. In fact, our algorithm uses the coefficients and the exponents of the Gaussian terms as the only input from which it selects the necessary terms, scale by scale, according the desired accuracy threshold ϵ. Therefore, our algorithm works for all operators with kernels that admit approximation by a sum of Gaussians. Examples of such operators include the Poisson and the bound state Helmholtz Green's functions, the projector on divergence free functions, as well as regular and fractional derivative operators. Let us consider a particular family of operators $(-\Delta + \mu^2 I)^{-\alpha}$, where $\mu \geq 0$ and $0 < \alpha < 3/2$. The kernel of this operator

$$K_{\mu,\alpha}(r) = 2^{-\frac{3}{2}+\alpha} \cdot C_\alpha \cdot (\frac{\mu}{r})^{\frac{3}{2}-\alpha} \mathrm{K}_{\frac{3}{2}-\alpha}(\mu r),$$

where $\mathrm{K}_{\frac{3}{2}-\alpha}$ is the modified Bessel function, $r = ||\mathbf{x} - \mathbf{y}||$ and $C_\alpha = 2 \cdot 2^{-2\alpha}\pi^{-\frac{3}{2}}/\Gamma(\alpha)$, has an integral representation

$$K_{\mu,\alpha}(r) = C_\alpha \int_{-\infty}^{\infty} e^{-r^2 e^{2s}} e^{-\frac{1}{4}\mu^2 e^{-2s} + (3-2\alpha)s} ds. \tag{14}$$

Using the trapezoidal rule, we construct an approximation valid over a range of values $\delta \leq r \leq R$ with accuracy ϵ, of the form

$$\left| K_{\mu,\alpha}(r) - \sum_{m=1}^{M} w_m e^{-\tau_m r^2} \right| \leq \epsilon K_{0,\alpha}(r) = \epsilon \frac{\Gamma(3/2 - \alpha) \cdot C_\alpha}{2r^{3-2\alpha}}, \tag{15}$$

where $\tau_m = e^{2s_m}$, $w_m = h\, C_\alpha\, e^{-\mu^2 e^{-2s_m}/4 + (3-2\alpha)s_m}$, $h = (B - A)/M$ and $s_m = A + mh$. The limits of integration, A, B and the step size h are selected as indicated in [24], where it is shown that for a fixed accuracy ϵ the number of terms M in (15) is proportional to $\log(R\delta^{-1})$. Although it is possible to select δ and R following the estimates in [25] and optimize the number of terms for a desired accuracy ϵ, in this paper we start with an approximation that has an obviously excessive range of validity and thus, an excessive number of terms.

An example of such approximation is shown in Figure 4. For a requested tolerance of $\epsilon = 10^{-10}$, roughly 300 terms are enough to provide a valid approximation over a range of 15 decades. We then let the algorithm choose the necessary terms, scale by scale, to satisfy the user-supplied accuracy requirement ϵ. This approach may end up with a few extra terms on some scales in comparison with that using a nearly optimal number of terms [8,10,11,9]. Whereas the cost of applying a few extra terms is negligible, we gain significantly in having a much more flexible and general algorithm.

We note that approximation in (15) clearly reduces the problem of applying the operator to that of applying a sequence of Gauss transforms [26,27], one by one. From this point of view, the algorithm that we present may be considered as a procedure for applying a linear combination of Gauss transforms simultaneously.

In order to represent the kernel K of the operator in multiwavelet bases, we need to compute the integrals,

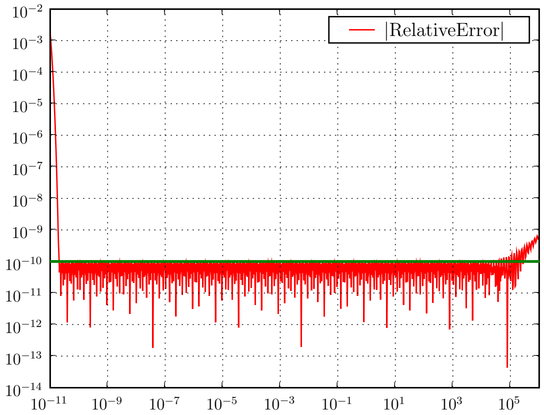


Figure 4. Relative error of the Gaussian approximation for the Poisson kernel in 3 dimensions. This unoptimized expansion uses 299 terms to cover a dynamic range of roughly 15 decades with $\epsilon = 10^{-10}$ relative accuracy.

$$r^{j;\ell}_{i_1 i'_1, i_2 i'_2, i_3 i'_3} = \sum_{m=1}^{M} 2^{-3j} \int_B w_m \, e^{-\tau_m \|2^{-j}(\mathbf{x}+\ell)\|^2} \, \Phi_{i_1 i'_1}(x_1) \, \Phi_{i_2 i'_2}(x_2) \, \Phi_{i_3 i'_3}(x_3) \, d\mathbf{x} \,, \tag{16}$$

where $\Phi_{ii'}(x)$ are the cross-correlations of the scaling functions (see Appendix). We obtain

$$r^{j;\ell}_{i_1 i'_1, i_2 i'_2, i_3 i'_3} = \sum_{m=1}^{M} w_m \, F^{j;\,m,l_1}_{i_1 i'_1} \, F^{j;\,m,l_2}_{i_2 i'_2} \, F^{j;\,m,l_3}_{i_3 i'_3} \,, \tag{17}$$

where

$$F^{j;\,m,l}_{ii'} = \frac{1}{2^j} \int_{-1}^{1} e^{-\tau_m (x+l)^2/4^j} \, \Phi_{ii'}(x) dx. \tag{18}$$

Since the Gaussian kernel is not homogeneous, we have to compute integrals (18) for each scale. Although in principle $l \in \mathbb{Z}$, in the next section we explain how to restrict it to a limited range on each scale, for a given accuracy ϵ.

4 Modified ns-form of a multidimensional operator

In this section, we describe how the separated representation approximations of Section 3 can be used to construct a multidimensional extension of the ns-form representation from Section 2.3, using only one-dimensional quantities and norm estimates. This makes our approach viable for $d > 1$. We use the modified ns-form as in the one-dimensional case described in Section 2.3. We find the ns-form essential for adaptive algorithms in more than one dimension, since:

(1) Scales do not interact as the operator is applied. All interactions between scales are accounted for by the (inexpensive) projection at the final step of the algorithm.
(2) For the same reason, the subdivision of space at different scales naturally maps into the supporting data structures. We note that one of the main difficulties in developing adaptive algorithms is in organizing computations with blocks of an adaptive decomposition of a function from different scales but with a common boundary. The methods for such computations are known as mortar elements methods. In our approach this issue does not present any obstacle, as all relevant interactions are naturally accounted for by the data structures.

The key feature that makes our approach efficient in dimensions $d \geq 2$ is the separated structure of the modified ns-form. Namely, the blocks of $\mathbf{T}_j - \uparrow (\mathbf{T}_{j-1})$ are of the form (for $d = 3$)

$$\begin{aligned}
\mathbf{T}_j^\ell - \uparrow (\mathbf{T}_{j-1}^\ell) &= \sum_{m=1}^{M} w_m F^{j;ml_1} F^{j;ml_2} F^{j;ml_3} \\
&\quad - \uparrow \left(\sum_{m=1}^{M} w_m F^{j-1;ml_1} F^{j-1;ml_2} F^{j-1;ml_3} \right) \\
&= \sum_{m=1}^{M} w_m F^{j;ml_1} F^{j;ml_2} F^{j;ml_3} \qquad (19) \\
&\quad - \sum_{m=1}^{M} w_m \uparrow (F^{j-1;ml_1}) \cdot \uparrow (F^{j-1;ml_2}) \cdot \uparrow (F^{j-1;ml_3}).
\end{aligned}$$

As in the case $d = 1$, the norm of the operator blocks of $\mathbf{T}_j^\ell - \uparrow (\mathbf{T}_{j-1}^\ell)$ decays rapidly with $\|\ell\|$, $\ell = (l_1, l_2, l_3)$, and the rate of decay depends on the number of vanishing moments of the basis [4]. Moreover, we limit the range of shift indices $\|\ell\|$ using only one-dimensional estimates of the differences

$$F_{ii'}^{j;\,m,2l} - \uparrow F_{ii'}^{j;\,m,l} \quad \text{and} \quad F_{ii'}^{j;\,m,2l+1} - \uparrow F_{ii'}^{j;\,m,l} \qquad (20)$$

of operator blocks computed via (18). The norms of individual blocks $F_{ii'}^{j;\,m,l}$ are illustrated in Figure 5 (a), for scale $j = 1$.

By selecting the number of vanishing moments for a given accuracy, it is sufficient to use $\|\ell\|_\infty \leq 2$ in practical applications that we have encountered. Also, not all terms in the Gaussian expansion of an operator need to be included since, depending on the scale j, their contribution may be negligible for a given accuracy, as shown in Figure 5 (b). Below we detail how we select terms of the Gaussian expansion on a given scale as well as the significant blocks of $\mathbf{T}_j^\ell - \uparrow(\mathbf{T}_{j-1}^\ell)$. This procedure establishes the band structure of the operator. We then project the *banded* operator $\uparrow(\mathbf{T}_{j-1}^{b_j})$ back to the scale $j-1$ and then combine blocks on the natural scale for each projection in order to apply the operator efficiently as was explained in Section 2.3 for the one-dimensional case.

We note that in deciding which terms to keep in (19), we do not compute the difference between the full three dimensional blocks as it would carry a high computational cost; instead we use estimates based on the one dimensional blocks of the separated representation. We note that since the resulting band structure depends only on the operator and the desired accuracy of its approximation, one of the options is to store such band information as it is likely to be reused.

In order to efficiently identify the significant blocks in $\mathbf{T}_j^\ell - \uparrow(\mathbf{T}_{j-1}^\ell)$ as a function of ℓ, we develop norm estimates based only on the one-dimensional blocks. The difference between two terms of the separated representation, say $F_1F_2F_3 - G_1G_2G_3$, may be written as

$$F_1F_2F_3 - G_1G_2G_3 = (F_1 - G_1)F_2F_3 + G_1(F_2 - G_2)F_3 + G_1G_2(F_3 - G_3).$$

We average six different combinations of the three terms to include all directions in a symmetric manner, which results in the norm estimate

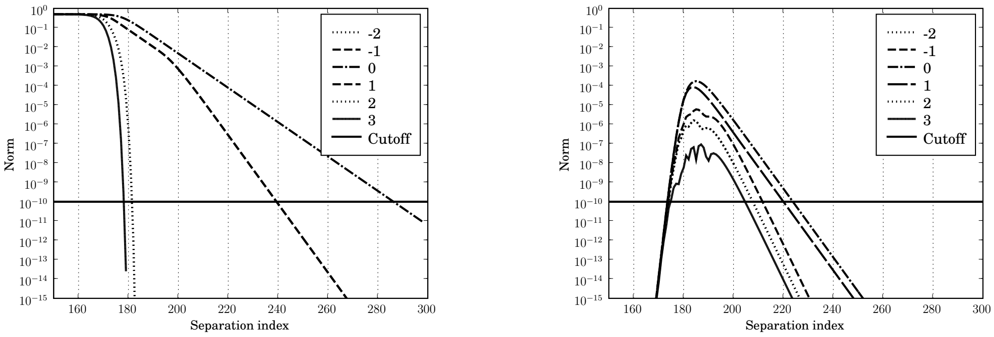
$$\begin{aligned}\|F_1F_2F_3 - G_1G_2G_3\| \le \frac{1}{6}\mathrm{sym}\,[&\|F_1 - G_1\|\|F_2\|\|F_3\| + \\ &\|G_1\|\|F_2 - G_2\|\|F_3\| + \|G_1\|\|G_2\|\|F_3 - G_3\|]\,,\end{aligned} \tag{21}$$

where the symmetrization is over the three directions and generates 18 terms. For the rotationally symmetric operators with Gaussian expansion as in (15) computing the right hand side in this estimate involves just three types of one dimensional blocks and their norms,

$$\begin{aligned} N_{\mathrm{dif}}^{j;m;l} &= \left\| F^{j;m;l} - \uparrow(F^{j-1;m;l}) \right\|, \\ N_F^{j;m;l} &= \left\| F^{j;m;l} \right\|, \\ N_{\uparrow F}^{j;m;l} &= \left\| \uparrow(F^{j-1;m;l}) \right\|, \end{aligned} \tag{22}$$

where index j indicates the scale, m the term in the Gaussian expansion (15), and l the position of the block in a given direction.

These estimates allow us to discard blocks whose norm falls below a given threshold of accuracy, namely, for each multi-index ℓ, we estimate



(a) Norms of each one-dimensional block computed via (18) for scale $j = 1$, as a function of the index m in the separated representation.

(b) Norm estimates (23) for scale $j = 1$ as a function of the index m in the separated representation. Based on these estimates, only terms above the cutoff are actually applied.

Figure 5. Comparison of norms of matrix blocks generated by the Gaussian approximation for the Poisson kernel in dimension $d = 3$. In each picture, the curves correspond to the different offsets l for which blocks are generated. Figure (b) illustrates the estimate in (23) (see main text for details).

$$\begin{aligned}\left\|\mathbf{T}_j^{\ell}- \uparrow (\mathbf{T}_{j-1}^{\ell})\right\| \leq \frac{1}{6}\sum_{m=1}^{M} w_m \ \mathrm{sym}\Big[& N_{\mathrm{dif}}^{j;m;l_1} N_F^{j;m;l_2} N_F^{j;m;l_3} + \\ & N_{\uparrow F}^{j;m;l_1} N_{\mathrm{dif}}^{j;m;l_2} N_F^{j;m;l_3} + N_{\uparrow F}^{j;m;l_1} N_{\uparrow F}^{j;m;l_2} N_{\mathrm{dif}}^{j;m;l_3}\Big].\end{aligned} \tag{23}$$

For each scale j and each block $\mathbf{T}_j^{\ell}- \uparrow (\mathbf{T}_{j-1}^{\ell})$ labeled by the multi-index $\ell = (l_1, l_2, l_3)$, we compute all terms of the sum in (23) and identify the range $[m_1, m_2]$ which we need to keep for that block, by discarding from the sum terms whose cumulative contribution is below ϵ. If the entire sum falls below ϵ, this range may be empty and the entire $\mathbf{T}_j^{\ell}- \uparrow (\mathbf{T}_{j-1}^{\ell})$ is discarded. The range $[m_1, m_2]$ differs significantly depending whether or not the block is affected by the singularity of the kernel as is illustrated in Figure 5. In Figure 5 (a) the rate of decay for the blocks with shift $|l| = 2, 3$ is significantly faster than for the blocks with $|l| \leq 1$ affected by the singularity. We note that all blocks of the first 150 terms in the separated representation (17) have norm 1 (and rank 1 as matrices) and are not shown in Figure 5 (a).

Since the difference in (23) involves blocks upsampled from a coarser scale, all shifts $|l| \leq 3$ are affected by the singularity. Figure 5 (b) shows the r.h.s. of the estimate in (23) for different shifts along one of the directions, where the blocks along the other two directions are estimated by the maximum norm over all possible shifts.

After discarding blocks with norms less that ϵ using the estimate in (23), we downsample the remaining blocks of $\uparrow (\mathbf{T}_{j-1}^{\ell})$ back to the original scale. This leaves only blocks of $\mathbf{T}_j^\ell$ on the scale j and we remove additional blocks of $\mathbf{T}_j^\ell$ for the shifts $|l| = 2, 3$ where the decay is sufficiently fast to make their contribution less than ϵ.

This leads us to arrange the blocks on each scale into several subsets by the effect the singularity of the kernel has on them and find the appropriate range $[m_1, m_2]$ for each subset. There are three such sets in dimension $d = 2$ and four sets in dimension $d = 3$. For each index l, we will say that the index belongs to the *core* if $l = -1, 0, 1$ and to the *boundary* otherwise. The core indices correspond to one-dimensional blocks whose defining integrals include the singularity of the kernel. We then divide all possible values of the multi-index $\ell = (l_1, l_2, l_3)$, according to the number of core indices it has. In $d = 3$ this gives us four sets:

- Core: all indices (l_1, l_2, l_3) belong to the core.
- Boundary-1: two of the indices belong to the core and one to the boundary
- Boundary-2: one of the indices belongs to the core, the other two to the boundary
- Boundary-3: all indices (l_1, l_2, l_3) belong to the boundary.

We then find the range $[m_1, m_2]$ for each subset and apply blocks of each subset separately, thus avoiding unnecessary computations with blocks whose contribution is negligible. This range analysis only needs to be done once per operator and the desired accuracy, and the results may be saved for repeated use.

5 Multidimensional adaptive application of ns-form

In this section we present an algorithm for applying the modified non-standard form which is an extension of (2) (based on (8) and (10)) to higher dimensions. We are now seeking to compute

$$g(\mathbf{x}) = [Tf](\mathbf{x}) = \int K(\mathbf{y} - \mathbf{x}) f(\mathbf{y}) d\mathbf{y},$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ for $d = 2, 3$. The separated approximation (19) reduces the complexity of applying the operator by allowing partial factorization of the nested loops in each scale indicated by the order of summation and illustrated for $d = 3$,

$$
\begin{aligned}
g^{j;l_1l_2l_3} = \sum_m w_m \Bigg\{ & \sum_{l'_1} F^{j;m;l_1-l'_1} \sum_{l'_2} F^{j;m,l_2-l'_2} \sum_{l'_3} F^{j;m,l_3-l'_3} - \\
& \uparrow \Bigg[\sum_{l'_1} F^{j-1;m;l_1-l'_1} \sum_{l'_2} F^{j-1;m,l_2-l'_2} \sum_{l'_3} F^{j-1;m,l_3-l'_3} \Bigg] \Bigg\} f^{j;l'_1l'_2l'_3}.
\end{aligned}
$$

As described in the previous section, this evaluation is done by regions of indices. These regions of indices are organized so that (for a given accuracy) the number of retained terms of the separated representation is roughly the same for all blocks within each region. Thus, we perform the summation over the terms of the separated representation last, applying only the terms that actually contribute to the result above the requested accuracy threshold, according to estimate (23). Therefore, we avoid introducing checks per individual block and the resulting loss of performance.

Just as in the one-dimensional case, we use (19) in a 'natural scale' manner. That is, blocks belonging to scale j are only applied on that scale. As in one-dimensional case, the interaction between scales is achieved by the projection (10) that redistributes blocks accumulated in this manner properly between the scales to obtain the adaptive tree for the resulting function. The overall approach is the same as described in (2). We note that, as expected, the separated representation requires more detailed bookkeeping when constructing the data structures for the operator.

Remark 1 Our multiresolution decomposition corresponds to the geometrically varying refinement in finite element methods. In this case the adjoining boxes do not necessarily share common vertices, forming what corresponds to the so-called non-conforming grid. In finite element methods such situation requires additional construction provided by the mortar element methods. Mortar element methods were introduced by Patera and his associates, see e.g. [19,20,21,22]. These methods permit coupling discretizations of different types in non-overlapping domains. Such methods are fairly complicated and involve, for example, the introduction of interface conditions through an L^2 minimization. In our approach we do not face these issues at all and do not have to introduce any additional interface conditions. The proper construction for adjoining boxes is taken care by the redundant tree data structure and Step 2 of Algorithm 3 for applying the kernel, which generates the necessary missing boxes on appropriate scales.

Remark 2 Although Algorithm 3 applies convolution operators, only minor changes are needed to use it for non-convolutions. Of course in such case, the separated representation for the modified ns-form should be constructed by a different approach.

Algorithm 3 Adaptive non-standard form operator application in multiple dimensions (illustrated for $d = 2$), $g = \mathbf{T}f$

```
**Initialization:** Construct the redundant tree for $f$ and copy it as skeleton tree for $g$ (see Section 2.2.1).
**for all** scales $j = 0, \ldots, n-1$ **do**
  **for all** function blocks $g^j_{l_1 l_2}$ in the tree for $g$ at scale $j$ **do**
    # *Step 1. Determine the list of all Core, Boundary-1 and Boundary-2 contributing operator blocks of the modified ns-form $F^{j;m;l_1-l'_1}, F^{j;m;l_2-l'_2}$ (see Section 4):*
    **if** $g^j_{l_1 l_2}$ belongs to a *leaf* **then**
      Read operator blocks $F^{j;m;l_1-l'_1}, F^{j;m;l_2-l'_2}$ for Core, Boundary-1 and Boundary-2, and their weights $w_m$ and corresponding ranges from the separated representation.
    **else**
      Read operator blocks $F^{j;m;l_1-l'_1}, F^{j;m;l_2-l'_2}$ for Boundary-1 and Boundary-2, and their weights $w_m$ and corresponding ranges from the separated representation.
    **end if**
    # *Step 2. Find the required blocks $f^j_{l'_1 l'_2}$ of the input function $f$:*
    **if** function block $f^j_{l'_1 l'_2}$ exists in the redundant tree for $f$ **then**
      Retrieve it.
    **else**
      Create it by interpolating from a coarser scale and cache for reuse.
    **end if**
    # *Step 3. Output function block computation:*
    For each set $S$ of indices determined in *Step 1 (Core, Boundary-1, Boundary-2) and the corresponding ranges of terms in the separated representation, compute the sum*

    $$\hat{g}^{j;S}_{l_1 l_2} = \sum_m w_m \sum_{l'_1} F^{j;m;l_1-l'_1} \sum_{l'_2} F^{j;m;l_2-l'_2} f^j_{l'_1 l'_2},$$

    *Add all computed sums to obtain $\hat{g}^j_{l_1 l_2}$.*
  **end for**
**end for**
# *Step 4. Adaptive projection:*
Project resulting output function blocks $\hat{g}^j_{l_1 l_2}$ on all scales into a proper adaptive tree by using Eq. (5).
Discard from the resulting tree unnecessary function blocks at the requested accuracy.
**Return:** the function $g$ represented by its adaptive tree.
```

5.1 Operation count estimates

Adaptive decomposition of functions

The cost of adaptively decomposing a function in d dimensions is essentially that of an adaptive wavelet transform. Specifically, it takes $\mathcal{O}(N_{\text{blocks}} \cdot p^{d+1})$ operations to compute such representation, where N_{blocks} is the final number of significant blocks in the representation and p is the order of multiwavelet basis chosen. In comparison with the usual wavelet transform, it appears to be significantly more expensive. However, these $\mathcal{O}(p^{d+1})$ operations process $\mathcal{O}(p^d)$ points, thus in counting significant coefficients as it is done in the usual adaptive wavelet transform, we end up with $\mathcal{O}(p)$ operations per point.

Operator application

The cost of applying an operator in the modified ns-form is $\mathcal{O}(N_{\text{blocks}} M p^{d+1})$, where N_{blocks} is the number of blocks in the adaptive representation of the input function, M is the separation rank of the kernel in (15) and p is the order of the multiwavelet basis. For a given desired accuracy ϵ, we typically select $p \propto \log \epsilon^{-1}$; M has been shown to be proportional to $(\log \epsilon^{-1})^{\nu}$, where ν depends on the operator [24]. In our numerical experiments, M is essentially proportional to $\log \epsilon^{-1}$, since we never use the full separated representation, as discussed in Section 4 and illustrated in Figure 5.

This operation count can be potentially reduced to $\mathcal{O}(N_{\text{blocks}} M p^{d})$ by using the structure of the matrices in 18, and we plan to address this in the future.

Final projection

After the operator has been applied to a function in a scale-independent fashion, a final projection step is required as discussed in Section 2.3. This step requires $\mathcal{O}(N_{\text{blocks}} \cdot p^{d})$ operations, the same as in the original function decomposition. In practice, this time is negligible compared to the actual operator application.

6 Numerical examples

6.1 The Poisson equation

We illustrate the performance of the algorithm by solving the Poisson equation in $d = 3$

$$\nabla^2 \phi(\mathbf{r}) = -\rho(\mathbf{r}) \tag{24}$$

with free space boundary conditions, $\phi(\mathbf{r}) \rightarrow 0$ and $\partial\phi/\partial r \rightarrow 0$ as $r \rightarrow \infty$. We write the solution as

$$\phi(\mathbf{r}) = \frac{1}{4\pi} \int_{\mathbb{R}^3} \frac{1}{|\mathbf{r} - \mathbf{r}'|} \rho(\mathbf{r}') d\mathbf{r}'$$

and adaptively evaluate this integral. We note that our method can equally be used for $d = 2$, since the corresponding Green's function can also be represented as a sum of Gaussians, and the operator application algorithm has been implemented in for both $d = 2$ and $d = 3$.

For our test we select

$$\phi(\mathbf{r}) = \sum_{i=1}^{3} e^{-\alpha|\mathbf{r}-\mathbf{r}'|^2},$$

so that we solve the Poisson equation with

$$\rho(\mathbf{r}) = -\nabla^2 \phi(\mathbf{r}) = -\sum_{i=1}^{3} (4\alpha^2 \left|\mathbf{r} - \mathbf{r}_i\right|^2 - 6\alpha) e^{-\alpha|\mathbf{r}-\mathbf{r}_i|^2}.$$

Our parameters are chosen as follows: $\alpha = 300$, $r_1 = (0.5, 0.5, 0.5)$, $r_2 = (0.6, 0.6, 0.5)$ and $r_3 = (0.35, 0.6, 0.5)$. These ensure that $\rho(r)$ is well below our requested thresholds on the boundary of the computational domain. All numerical experiments were performed on a Pentium-4 running at 2.8 GHz, with 2 GB of RAM. The results are summarized in Table 2.

In order to gauge the speed of algorithm in reasonably computer-independent terms, we use a similar approach to that of [17] and also provide timings of the Fast Fourier Transform (FFT). Specifically, we display timings for two FFTs as an estimate of the time needed to solve the Poisson equation with a smooth right hand side and periodic boundary conditions in a cube. As in [17], we compute the rate that estimates the number of processed points per second. We observe that for our adaptive algorithm such rate varies between $3.4 \cdot 10^4$ and $1.1 \cdot 10^5$ (see Table 2), whereas for the FFTs it is around 10^6 (see Table 3). We note that our algorithm is not fully optimized, namely, we do not use the structure of the matrices in (18) and the symmetries afforded by the radial kernels. We expect a substantial impact on the speed by introducing these improvements and will report them separately.

Requested $\epsilon = 10^{-3}$

p	E_2	E_∞	Time (s)	Rate (pts/s)
6	$5.0 \cdot 10^{-3}$	$7.9 \cdot 10^{-1}$	1.2	$7.2 \cdot 10^4$
8	$1.7 \cdot 10^{-3}$	$1.2 \cdot 10^{-1}$	0.51	$7.3 \cdot 10^4$
10	$4.4 \cdot 10^{-4}$	$3.7 \cdot 10^{-2}$	0.68	$1.1 \cdot 10^5$

Requested $\epsilon = 10^{-6}$

p	E_2	E_∞	Time (s)	Rate (pts/s)
10	$4.7 \cdot 10^{-6}$	$3.6 \cdot 10^{-4}$	10.3	$5.7 \cdot 10^4$
12	$8.5 \cdot 10^{-6}$	$4.3 \cdot 10^{-5}$	13.5	$7.5 \cdot 10^4$
14	$6.9 \cdot 10^{-8}$	$5.2 \cdot 10^{-6}$	20.0	$8.0 \cdot 10^4$

Requested $\epsilon = 10^{-9}$

p	E_2	E_∞	Time (s)	Rate (pts/s)
16	$2.5 \cdot 10^{-10}$	$2.2 \cdot 10^{-8}$	68.1	$3.5 \cdot 10^4$
18	$7.7 \cdot 10^{-11}$	$3.5 \cdot 10^{-9}$	100.3	$3.4 \cdot 10^4$
20	$1.2 \cdot 10^{-10}$	$1.8 \cdot 10^{-8}$	133.4	$3.5 \cdot 10^4$

Table 2
Accuracy and timings for the adaptive solution of the Poisson equation in (24).

size	Time (s)	Rate (pts/s)
32^3	0.02	$1.7 \cdot 10^6$
64^3	0.22	$1.2 \cdot 10^6$
128^3	2.21	$9.5 \cdot 10^5$
256^3	20.7	$8.1 \cdot 10^5$

Table 3
Timings of two 3D FFTs to estimate the speed of a non-adaptive, periodic Poisson solver on a cube for smooth functions.

We note that the multigrid method (see e.g. [28,29]) is frequently used as a tool for solving the Poisson equation (and similar problems) in differential form. The FFT-based gauge suggested in [17] is useful for comparisons with these algorithms as well.

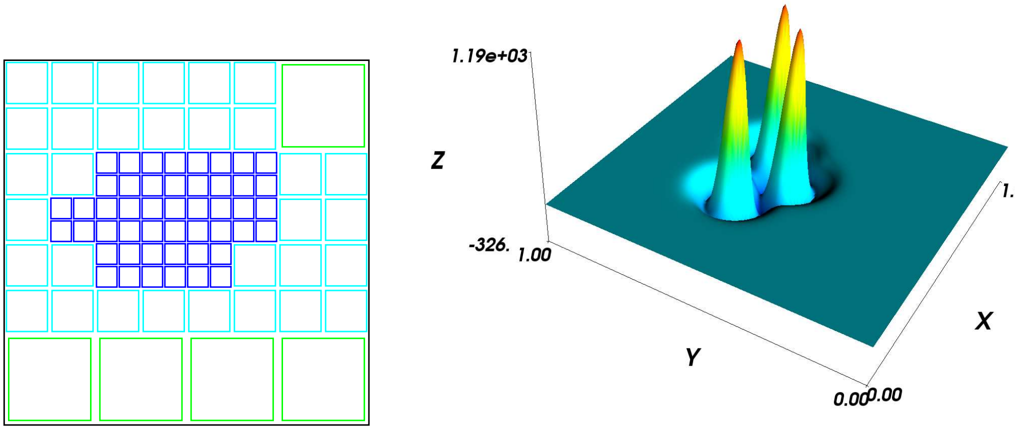


Figure 6. A two-dimensional slice of the three-dimensional subdivision of space by the scaling functions and an illustration of the source term for the Poisson equation (24).

6.2 The ground state of the hydrogen atom

A simple example of computing the ground state of the hydrogen atom illustrates the numerical performance of the algorithms developed in this paper, and their utility for constructing more complex codes. The eigenfunctions ψ for the hydrogen atom satisfy the time-independent Schrödinger equation (written in atomic units and spherical coordinates),

$$-\frac{1}{2}\Delta\psi - \frac{1}{r}\psi = E\psi, \tag{25}$$

where E is the energy eigenvalue. For the ground state, $E = -1/2$ and the (unnormalized) wave function is $\psi = e^{-r}$. Following [30], we write

$$\phi = -2G_\mu V\phi, \tag{26}$$

where $G_\mu = (-\Delta + \mu^2\mathcal{I})^{-1}$ is the Green's function for some μ and $V = -1/r$ is the nuclear potential. For $\mu = \sqrt{-2E}$ the solution ϕ of (26) has $\|\phi\|_2 = 1$ and coincides with that of (25). We solve (26) by a simple iteration starting from some value μ_0 and changing μ to obtain the solution with $\|\phi\|_2 = 1$. The algorithm proceeds as follows:

(1) Initialize with some value μ_0 and function ϕ. The number of iterations of the algorithm is only weakly sensitive to these choices.
(2) Compute the product of the potential V and the function ϕ.
(3) Apply the Green's function G_μ to the product $V\phi$ via the algorithm of

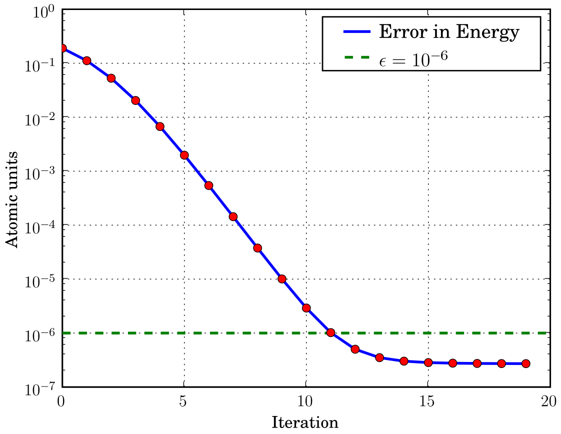


Figure 7. Convergence of the iteration to obtain the ground state of the hydrogen atom computed via formulation in (26) for the non-relativistic Schrödinger equation. The requested accuracy in applying the Green's function is set to 10^{-6}.

this paper to compute

$$\phi_{new} = -2G_{\mu}V\phi.$$

(4) Compute the energy for ϕ_{new},

$$E_{new} = \frac{\frac{1}{2}\langle\nabla\phi_{new}, \nabla\phi_{new}\rangle + \langle V\phi_{new}, \phi_{new}\rangle}{\langle\phi_{new}, \phi_{new}\rangle}.$$

(5) Set $\mu = \sqrt{-2E_{new}}$, $\phi = \phi_{new}/\|\phi_{new}\|$ and return to Step 2.

The iteration is terminated as the change in μ and $\|\phi\| - 1$ falls below the desired accuracy. The progress of the iteration is illustrated in Figure 7. The computations in Steps 2 and 4 use the three dimensional extension of the approach described in [13], to compute point-wise multiplications of adaptively represented functions and weak differential operators of the same.

This example illustrates an application of our algorithm to problems in quantum chemistry. Multiresolution quantum chemistry developed in [8,9,10,11] also uses separated representations. The main technical difference with [8,9,10,11] is that we use the modified non-standard form and apply operator blocks on their "natural" scale, thus producing a fully adaptive algorithm. We are currently using this algorithm as part of a new method for solving the multiparticle Schrödinger equation and will report the results elsewhere.

7 Discussion and conclusions

We have shown that a combination of separated and multiresolution representations of operators yields a new multidimensional algorithm for applying a class of integral operators with radial kernels. We note that the same algorithm is used for all such operators as they are approximated by a weighted sum of Gaussians. This fact makes our approach applicable across multiple fields, where a single implementation of the core algorithm can be reused for different specific problems. The algorithm is fully adaptive and avoids issues usually addressed by mortar methods.

The method of approximation underlying our approach is distinct from that of the FMM, has similar efficiency and has the advantage of being more readily extendable to higher dimensions. We also note that semi-analytic approximations via weighted sums of Gaussians provide additional advantages in some applications. Although we described the application of kernels in free space, there is a simple extension to problems with radial kernels subject to periodic, Dirichlet or Neumann boundary conditions on a cube that we will describe separately.

The algorithm may be extended to classes of non-convolution operators, e.g., the Calderon-Zygmund operators. For such extensions the separated representation may not be available in analytic form, as it is for the operators of this paper, and may require a numerical construction. The separated representation of the kernel permits further generalization of our approach to dimensions $d \gg 3$ for applying operators to functions in separated representation.

A notable remaining challenge is an efficient, high order extension of this technique to the application of operators on domains with complicated geometries and surfaces.

8 Appendix

8.1 Scaling functions

We use either the Legendre polynomials $P_0, \dots, P_{p-1}$ or the interpolating polynomials on the Gauss-Legendre nodes in $[-1, 1]$ to construct an orthonormal basis for each subspace $\mathbf{V}_j$ [12,13].

Let us briefly describe some properties of the Legendre scaling functions ϕ_k, $k = 0, \dots, p-1$, defined as

$$\phi_k(x) = \begin{cases} \sqrt{2k+1} P_k(2x-1), & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}, \tag{27}$$

and forming a basis for $\mathbf{V}_0$. The subspace $\mathbf{V}_j$ is spanned by $2^j p$ functions obtained from $\phi_0, \ldots, \phi_{p-1}$ by dilation and translation,

$$\phi^j_{kl}(x) = 2^{j/2}\phi_k(2^j x - l), \quad k = 0, \ldots, p-1, \quad l = 0, \ldots, 2^j - 1. \tag{28}$$

These functions have support on $[2^{-j}l, 2^{-j}(l+1)]$ and satisfy the orthonormality condition

$$\int_{-\infty}^{\infty} \phi^j_{kl}(x)\phi^j_{k'l'}(x)dx = \delta_{kk'}\delta_{ll'}\,. \tag{29}$$

A function f, defined on $[0, 1]$, is represented in the subspace $\mathbf{V}_j$ by its normalized Legendre expansion

$$f(x) = \sum_{l=0}^{2^j-1}\sum_{k=0}^{p-1} s^j_{kl}\phi^j_{kl}(x), \tag{30}$$

where the coefficients s^j_{kl} are computed via

$$s^j_{kl} = \int_{2^{-j}l}^{2^{-j}(l+1)} f(x)\phi^j_{kl}(x)\,dx. \tag{31}$$

As long as $f(x)$ is smooth enough and is well approximated on $[2^{-j}l, 2^{-j}(l+1)]$ by a polynomial of order up to $2p-1$, we may use Gauss-Legendre quadratures to calculate the s^j_{kl} via

$$s^j_{kl} = 2^{-j/2}\sum_{i=0}^{p-1} f(2^{-j}(x_i + l))\phi_k(x_i)w_i, \tag{32}$$

where $x_0, \ldots, x_{p-1}$ are the roots of $P_p(2x-1)$ and $w_{0,} \ldots, w_{p-1}$ are the corresponding quadrature weights.

In more than one dimension, the above formulas are extended by using a tensor product basis in each subspace. For example, in two dimensions equation (30) becomes

$$f(x, x') = \sum_{l=0}^{2^j-1}\sum_{k=0}^{p-1}\sum_{l'=0}^{2^j-1}\sum_{k'=0}^{p-1} s^j_{kk'll'}\phi^j_{kl}(x)\phi^j_{k'l'}(x'). \tag{33}$$

8.2 Multiwavelets

We use piecewise polynomial functions $\psi_{0,} \ldots, \psi_{p-1}$ as an orthonormal basis for $\mathbf{W}_0$ [12,13],

$$\int_0^1 \psi_i(x)\psi_j(x)dx = \delta_{ij}. \tag{34}$$

Since $\mathbf{W}_0 \perp \mathbf{V}_0$, the first p moments of all $\psi_{0,} \ldots, \psi_{p-1}$ vanish:

$$\int_0^1 \psi_i(x)x^i dx = 0, \quad i, j = 0, 1, \ldots, p-1. \tag{35}$$

The space $\mathbf{W}_j$ is spanned by $2^j p$ functions obtained from $\psi_0, \ldots, \psi_{p-1}$ by dilation and translation,

$$\psi^j_{kl}(x) = 2^{j/2}\psi_k(2^j x - l), \quad k = 0, \ldots, p-1, \quad l = 0, \ldots, 2^j - 1, \tag{36}$$

and supported in the interval $I_{jl} = [2^{-j}l, 2^{-j}(l+1)]$. A function $f(x)$ defined on $[0, 1]$ is represented in the multiwavelet basis on n scales by

$$f(x) = \sum_{k=0}^{p-1} s^0_{k,0}\phi_k(x) + \sum_{j=0}^{n-1}\sum_{l=0}^{2^j-1}\sum_{k=0}^{p-1} d^j_{kl}\psi^j_{kl}(x) \tag{37}$$

with the coefficients d^j_{kl} computed via

$$d^j_{kl} = \int_{2^{-j}l}^{2^{-j}(l+1)} f(x)\psi^j_{kl}(x)\, dx. \tag{38}$$

8.3 Two-scale relations

The relation between subspaces, $\mathbf{V}_0 \oplus \mathbf{W}_0 = \mathbf{V}_1$, is expressed via the two-scale difference equations,

$$\phi_k(x) = \sqrt{2}\sum_{k'=0}^{p-1}\left(h^{(0)}_{kk'}\phi_{k'}(2x) + h^{(1)}_{kk'}\phi_{k'}(2x-1)\right), \quad k = 0, \ldots, p-1, \tag{39}$$

$$\psi_k(x) = \sqrt{2}\sum_{k'=0}^{p-1}\left(g^{(0)}_{kk'}\phi_{k'}(2x) + g^{(1)}_{kk'}\phi_{k'}(2x-1)\right), \quad k = 0, \ldots, p-1, \tag{40}$$

where the coefficients $h^{(0)}_{ij}$, $h^{(1)}_{ij}$ and $g^{(0)}_{ij}$, $g^{(1)}_{ij}$ depend on the type of polynomial basis used (Legendre or interpolating) and its order p. The matrices of coefficients

$$H^{(0)} = \{h^{(0)}_{kk'}\}, \quad H^{(1)} = \{h^{(1)}_{kk'}\}, \quad G^{(0)} = \{g^{(0)}_{kk'}\}, \quad G^{(1)} = \{g^{(1)}_{kk'}\}$$

are the multiwavelet analogues of the quadrature mirror filters in the usual wavelet construction, e.g., [15]. These matrices satisfy a number of important orthogonality relations and we refer to [13] for complete details, including the construction of the H, G matrices themselves. Let us only state how these matrices are used to connect the scaling s^j_{kl} and wavelet d^j_{kl} coefficients on neighboring scales j and $j+1$. The *decomposition procedure* $(j+1 \to j)$ is based on

$$s^j_{kl} = \sum_{k'=0}^{p-1} \left(h^{(0)}_{kk'} s^{j+1}_{k,2l} + h^{(1)}_{kk'} s^{j+1}_{k,2l+1} \right), \tag{41}$$

$$d^j_{kl} = \sum_{k'=0}^{p-1} \left(g^{(0)}_{kk'} s^{j+1}_{k,2l} + g^{(1)}_{kk'} s^{j+1}_{k,2l+1} \right); \tag{42}$$

the *reconstruction* ($j \to j+1$) is based on

$$s^{j+1}_{k,2l} = \sum_{k'=0}^{p-1} \left(h^{(0)}_{kk'} s^j_{kl} + g^{(0)}_{kk'} d^j_{kl} \right), \tag{43}$$

$$s^{j+1}_{k,2l+1} = \sum_{k'=0}^{p-1} \left(h^{(1)}_{kk'} s^j_{kl} + g^{(1)}_{kk'} d^j_{kl} \right). \tag{44}$$

8.4 Cross-correlation of the scaling functions

For convolution operators we only need to compute integrals with the cross-correlation functions of the scaling functions,

$$\Phi_{ii'}(x) = \int_{-\infty}^{\infty} \phi_i(x+y)\phi_{i'}(y)dy. \tag{45}$$

Since the support of the scaling functions is restricted to $[0,1]$, the functions $\Phi_{ii'}$ are zero outside the interval $[-1,1]$ and are polynomials on $[-1,0]$ and $[0,1]$ of degree $i+i'+1$,

$$\Phi_{ii'}(x) = \begin{cases} \Phi^+_{ii'}(x), & 0 \le x \le 1, \\ \Phi^-_{ii'}(x), & -1 \le x < 0, \\ 0, & 1 < |x|, \end{cases} \tag{46}$$

where $i, i' = 0, \ldots, p-1$ and

$$\Phi^+_{ii'}(x) = \int_0^{1-x} \phi_i(x+y)\,\phi_{i'}(y)dy\,, \quad \Phi^-_{ii'}(x) = \int_{-x}^{1} \phi_i(x+y)\,\phi_{i'}(y)dy\,. \tag{47}$$

We summarize relevant properties of the cross-correlation functions $\Phi_{ii'}$ in

Proposition 3

(1) Transposition of indices: $\Phi_{ii'}(x) = (-1)^{i+i'}\Phi_{i'i}(x)$,

(2) Relations between Φ^+ *and* Φ^-*:* $\Phi^-_{i,i'}(-x) = (-1)^{i+i'}\Phi^+_{i,i'}(x)$ *for* $0 \le x \le 1$,

(3) Values at zero: $\Phi_{ii'}(0) = 0$ *if* $i \neq i'$*, and* $\Phi_{ii}(0) = 1$ *for* $i = 0, \dots, p-1$*,*

(4) Upper bound: $\max_{x\in[-1,1]} |\Phi_{ii'}(x)| \leq 1$ *for* $i, i' = 0, \dots, p-1$*,*

(5) Connection with the Gegenbauer polynomials: $\Phi^{+}_{00}(x) = \frac{1}{2}\, C^{(-1/2)}_{1}(2x-1) + \frac{1}{2}$ *and* $\Phi^{+}_{l0}(x) = \frac{1}{2}\sqrt{2l+1}\, C^{(-1/2)}_{l+1}(2x-1)$*, for* $l = 1, 2, \dots,$ *where* $C^{(-1/2)}_{l+1}$ *is the Gegenbauer polynomial,*

(6) Linear expansion: if $i' \geq i$ *then we have*

$$\Phi^{+}_{ii'}(x) = \sum_{l=i'-i}^{i'+i} c^{l}_{ii'}\Phi^{+}_{l0}(x), \tag{48}$$

where

$$c^{l}_{ii'} = \begin{cases} 4l(l+1)\int_0^1 \Phi^{+}_{ii'}(x)\Phi^{+}_{l0}(x)(1-(2x-1)^2)^{-1}dx, & i' > i, \\ \\ 4l(l+1)\int_0^1 (\Phi^{+}_{ii}(x) - \Phi^{+}_{00}(x))\, \Phi^{+}_{l0}(x)(1-(2x-1)^2)^{-1}dx, & i' = i, \end{cases} \tag{49}$$

for $l \geq 1$ *and* $c^{0}_{ii'} = \delta_{ii'}$*.*

(7) Vanishing moments: we have $\int_{-1}^{1} \Phi_{00}(x)\, dx = 1$ *and* $\int_{-1}^{1} x^k \Phi_{ii'}(x)\, dx = 0$ *for* $i + i' \geq 1$ *and* $0 \leq k \leq i + i' - 1$*.*

Proof of these properties can be found in [25].

8.5 Separated representations of radial functions

As an example, consider approximating the function $1/r^{\alpha}$ by a collection of Gaussians. The number of terms needed for this purpose is mercifully small. We have [24]

Proposition 4 *For any* $\alpha > 0$*,* $0 < \delta \leq 1$*, and* $0 < \epsilon \leq \min\left\{\frac{1}{2}, \frac{8}{\alpha}\right\}$*, there exist positive numbers* τ_m *and* w_m *such that*

$$\left| r^{-\alpha} - \sum_{m=1}^{M} w_m e^{-\tau_m r^2} \right| \leq r^{-\alpha}\epsilon, \text{ for all } \delta \leq r \leq 1 \tag{50}$$

with

$$M = \log \epsilon^{-1}[c_0 + c_1 \log \epsilon^{-1} + c_2 \log \delta^{-1}], \tag{51}$$

where c_k *are constants that only depend on* α*. For fixed power* α *and accuracy* ϵ*, we have* $M = \mathcal{O}(\log \delta^{-1})$*.*

The proof of Proposition 4 in [24] is based on using the trapezoidal rule to discretize an integral representation of $1/r^{\alpha}$. Similar estimates may be obtained for more general radial kernels using their integral representations as in (14).

We note that approximations of the function $1/r$ via sums of Gaussians have been also considered in [31,32,33].

8.6 Estimates

By selecting appropriate ϵ and δ in the separated representations of a radial kernel K (as in Proposition 4), we obtain a separated approximation for the coefficients

$$t^{j;\ell}_{ii',jj',kk'} = 2^{-3j} \int_{[-1,1]^3} K(2^{-j}(\mathbf{x}+\ell))\, \Phi_{ii'}(x_1)\, \Phi_{jj'}(x_2)\, \Phi_{kk'}(x_3)\, d\mathbf{x}\,. \qquad (52)$$

Since the number of terms, M, depends logarithmically on ϵ and δ, we achieve any finite accuracy with a very reasonable number of terms. For example, for the Poisson kernel $K(r) = 1/4\pi r$, we have the following estimate [25],

Proposition 5 *For any $\epsilon > 0$ and $0 < \delta \leq 1$ the coefficients $t^{j;\ell}_{ii',jj',kk'}$ in (52) have an approximation with a low separation rank,*

$$r^{j;\ell}_{ii',jj',kk'} = \sum_{m=1}^{M} w_m\, F^{m,l_1}_{ii'}\, F^{m,l_2}_{jj'}\, F^{m,l_3}_{kk'}\,, \qquad (53)$$

such that if $\max_i |l_i| \geq 2$, *then*

$$|t^{j;\ell}_{ii',jj',kk'} - r^{j;\ell}_{ii',jj',kk'}| \leq c_0 2^{-2j}\epsilon, \qquad (54)$$

and if $\max_i |l_i| \leq 1$, *then*

$$|t^{j;\ell}_{ii',jj',kk'} - r^{j;\ell}_{ii',jj',kk'}| \leq 2^{-2j}(c_1\delta^2 + c_0\epsilon) \qquad (55)$$

where ϵ, δ, M, τ_m, w_m, $m = 1,\ldots,M$ are described in Proposition 4 for $\alpha = 1$ and c_0 and c_1 are (small) constants.

As described in Section 4, our adaptive algorithm selects only some of the terms, as needed on a given scale for the desired accuracy ϵ.

8.7 Evaluation of integrals with the cross-correlation functions

We need to compute integrals in (18), where the cross-correlation functions $\Phi_{ii'}$ are given in (45). We note that using (48), it is sufficient to compute

$$F_{i0}^{j;m,l} = \frac{1}{2^j} \int_{-1}^{1} e^{-\tau_m (x+l)^2/4^j} \Phi_{i0}(x) dx$$

for $0 \leq i \leq 2p-1$ rather than p^2 integrals in (18). Using the relation between Φ^- and Φ^+ in Proposition 3, we have

$$\int_{-1}^{0} \Phi_{i0}^{-}(x) e^{-\tau_m (x+l)^2} dx = \int_{0}^{1} \Phi_{i0}^{-}(-x) e^{-\tau_m (-x+l)^2} dx = (-1)^i \int_{0}^{1} \Phi_{i0}^{+}(x) e^{-\tau_m (x-l)^2} dx,$$

so that

$$F_{i0}^{j;m,l} = \int_{0}^{1} [e^{-\tau_m (x+l)^2} + (-1)^i e^{-\tau_m (x-l)^2}] \Phi_{i0}^{+}(x) dx.$$

References

[1] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comp. Phys. 73 (1) (1987) 325–348.

[2] J. Carrier, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm for particle simulations, SIAM Journal of Scientific and Statistical Computing 9 (4), yale University Technical Report, YALEU/DCS/RR-496 (1986).

[3] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (2) (1999) 468–498.

[4] G. Beylkin, R. Coifman, V. Rokhlin, Fast wavelet transforms and numerical algorithms. I, Comm. Pure Appl. Math. 44 (2) (1991) 141–183, Yale Univ. Technical Report YALEU/DCS/RR-696, August 1989.

[5] G. Beylkin, R. Cramer, A multiresolution approach to regularization of singular operators and fast summation, SIAM J. Sci. Comput. 24 (1) (2002) 81–117.

[6] G. Beylkin, M. J. Mohlenkamp, Numerical operator calculus in higher dimensions, Proc. Natl. Acad. Sci. USA 99 (16) (2002) 10246–10251, `http://www.pnas.org/cgi/content/abstract/112329799v1`.

[7] G. Beylkin, M. J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, SIAM J. Sci. Comput. 26 (6) (2005) 2133–2159, `http://amath.colorado.edu/pub/wavelets/papers/BEY-MOH2005.pdf`.

[8] R. Harrison, G. Fann, T. Yanai, G. Beylkin, Multiresolution quantum chemistry in multiwavelet bases, in: P.M.A. Sloot et. al. (Ed.), Lecture Notes in Computer Science. Computational Science-ICCS 2003, Vol. 2660, Springer, 2003, pp. 103–110.

[9] R. Harrison, G. Fann, T. Yanai, Z. Gan, G. Beylkin, Multiresolution quantum chemistry: basic theory and initial applications, J. Chem. Phys. 121 (23) (2004) 11587–11598, `http://amath.colorado.edu/pub/wavelets/papers/mrqc.pdf`.

[10] T. Yanai, G. Fann, Z. Gan, R. Harrison, G. Beylkin, Multiresolution quantum chemistry: Hartree-Fock exchange, J. Chem. Phys. 121 (14) (2004) 6680–6688.

[11] T. Yanai, G. Fann, Z. Gan, R. Harrison, G. Beylkin, Multiresolution quantum chemistry: Analytic derivatives for Hartree-Fock and density functional theory, J. Chem. Phys. 121 (7) (2004) 2866–2876.

[12] B. Alpert, A class of bases in L^2 for the sparse representation of integral operators, SIAM J. Math. Anal 24 (1) (1993) 246–262.

[13] B. Alpert, G. Beylkin, D. Gines, L. Vozovoi, Adaptive solution of partial differential equations in multiwavelet bases, J. Comput. Phys. 182 (1) (2002) 149–190.

[14] G. Beylkin, Approximations and fast algorithms, in: A.Laine, M.Unser, A.Aldroubi (Eds.), Proc. SPIE: Wavelets: Applications in Signal and Image Processing IX, Vol. 4478, 2001, pp. 1–9.

[15] I. Daubechies, Ten Lectures on Wavelets, CBMS-NSF Series in Applied Mathematics, SIAM, 1992.

[16] C. Chui, An Introduction to Wavelets, Academic Press, 1992.

[17] F. Ethridge, L. Greengard, A new fast-multipole accelerated Poisson solver in two dimensions, SIAM J. Sci. Comput. 23 (3) (2001) 741–760 (electronic).

[18] G. Beylkin, M. J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, APPM preprint #519, Univ. of Colorado, `http://amath.colorado.edu/pub/wavelets/papers/BEY-MOH2004P.pdf`, accepted for publication in *SIAM J. Sci. Comput.* (February 2004).

[19] Y. Maday, C. Mavriplis, A. T. Patera, Nonconforming mortar element methods: application to spectral discretizations, in: Domain decomposition methods (Los Angeles, CA, 1988), SIAM, Philadelphia, PA, 1989, pp. 392–418.

[20] G. Anagnostou, Y. Maday, C. Mavriplis, A. T. Patera, On the mortar element method: generalizations and implementation, in: Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989), SIAM, Philadelphia, PA, 1990, pp. 157–173.

[21] C. Bernardi, Y. Maday, A. T. Patera, Domain decomposition by the mortar element method, in: Asymptotic and numerical methods for partial differential equations with critical parameters (Beaune, 1992), Vol. 384 of NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Kluwer Acad. Publ., Dordrecht, 1993, pp. 269–286.

[22] C. Bernardi, Y. Maday, A. T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, in: Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991), Vol. 299 of Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, 1994, pp. 13–51.

[23] M. Abramowitz, I. A. Stegun, Handbook of Mathematical Functions, ninth Edition, Dover Publications, 1970.

[24] G. Beylkin, L. Monzón, On approximation of functions by exponential sums, Appl. Comput. Harmon. Anal. 19 (1) (2005) 17–48, `http://amath.colorado.edu/pub/wavelets/papers/afes.pdf`.

[25] G. Beylkin, R. Cramer, G. Fann, R. Harrison, Multiresolution separated representations of singular and weakly singular operators, Appl. Comput. Harmon. Anal.To appear.

[26] L. Greengard, J. Strain, The fast Gauss transform, SIAM J. Sci. Stat. Comput. 12 (1) (1991) 79–94.

[27] L. Greengard, X. Sun, A new version of the fast Gauss transform, in: Proceedings of the International Congress of Mathematicians, Vol. III (Berlin, 1998), no. Extra Vol. III, 1998, pp. 575–584 (electronic).

[28] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comp. 31 (1977) 333–390.

[29] A. Brandt, Multilevel computations of integral transforms and particle interactions with oscillatory kernels, Computer Physics Communications 65 (1991) 24–38.

[30] M. H. Kalos, Monte Carlo calculations of the ground state of three- and four-body nuclei, Phys. Rev. (2) 128 (1962) 1791–1795.

[31] W. Kutzelnigg, Theory of the expansion of wave functions in a Gaussian basis., Internat. J. Quantum Chem. 51 (1994) 447–463.

[32] D. Braess, Asymptotics for the approximation of wave functions by exponential sums, J. Approx. Theory 83 (1) (1995) 93–103.

[33] D. Braess, W. Hackbusch, Approximation of $1/x$ by exponential sums in $[1, \infty)$, IMA J. Numer. Anal. 25 (4) (2005) 685–697.

Approximating a Wavefunction as an Unconstrained Sum of Slater Determinants

Gregory Beylkin[*] Martin J. Mohlenkamp[†] Fernando Pérez[‡]

Submitted for publication July 2007.
University of Colorado APPM preprint number 554.

Abstract

The wavefunction for the multiparticle Schrödinger equation is a function of many variables and satisfies an antisymmetry condition, so it is natural to approximate it as a sum of Slater determinants. Many current methods do so, but they impose additional structural constraints on the determinants, such as orthogonality between orbitals or an excitation pattern. We present a method without any such constraints, by which we hope to obtain much more efficient expansions, and insight into the inherent structure of the wavefunction. We use an integral formulation of the problem, a Green's function iteration, and a fitting procedure based on the computational paradigm of separated representations. The core procedure is the construction and solution of a matrix-integral system derived from antisymmetric inner products involving the potential operators. We show how to construct and solve this system with computational complexity competitive with current methods.

AMS Subject Classification: 65Z05 65D15 81-08
Keywords: Multiparticle Schrödinger equation; Slater determinant; curse of dimensionality.

[*] Department of Applied Mathematics, University of Colorado at Boulder, 526 UCB, Boulder CO 80309-0526; beylkin@colorado.edu.

[†] Department of Mathematics, Ohio University, 321 Morton Hall, Athens OH 45701; mjm@math.ohiou.edu.

[‡] Department of Applied Mathematics, University of Colorado at Boulder, 526 UCB, Boulder CO 80309-0526; Fernando.Perez@colorado.edu.

I Introduction

Given the difficulties of solving the multiparticle Schrödinger equation, current numerical methods in quantum chemistry/physics are remarkably successful. Part of their success comes from efficiencies gained by imposing structural constraints on the wavefunction to match physical intuition. However, such methods scale poorly to high accuracy, and are biased to only reveal structures that were part of their own construction. Our goal is to develop a method that scales well to high accuracy and allows an unbiased exploration of the structure of the wavefunction. In this paper we take a step toward this goal by developing a method to approximate the wavefunction as an unconstrained sum of Slater determinants.

Since the multiparticle fermionic wavefunction is an antisymmetric function of many variables, it is natural to approximate it as a sum of Slater determinants, at least as a first step. Motivated by the physical intuition that electrons may be excited into higher energy states, the Configuration Interaction (CI) family of methods choose a set of determinants with predetermined orbitals, and then optimize the coefficients used to combine them. When it is found insufficient, methods to optimize the orbitals, work with multiple reference states, etc., are introduced (along with an alphabet of acronyms). A common feature of all these methods is that they impose some structural constraints on the Slater determinants, such as orthogonality of orbitals or an excitation pattern. As the requested accuracy increases, these structural constraints trigger an explosion in the number of determinants used, making the computation intractable for high accuracy.

The a priori structural constraints present in CI-like methods also force the wavefunction to comply with such structure, whether or not it really is the case. For example, if you use a method that approximates the wavefunction as a linear combination of a reference state and excited states, you could not learn that the wavefunction is better approximated as a linear combination of several non-orthogonal, near-reference states. Thus the choice of numerical method is not just a computational issue; it can help or hinder our understanding of the wavefunction.

For these reasons, our goal is to construct an adaptive numerical method without imposing a priori structural constraints besides that of antisymmetry. In this paper we derive and present an algorithm for approximating a wavefunction with an unconstrained sum of Slater determinants, with fully-adaptive single-electron functions. In particular we discard the notions of reference state and excitation of orbitals. The functions comprising the Slater determinants need not come from a particular basis set, be orthogonal, or follow some excitation pattern. They are computed so as to optimize the overall representation. In this respect we follow the philosophy of separated representations [4, 5], which allow surprisingly accurate expansions with remarkably few terms.

Our construction generates a solution using an iterative procedure based on nonlinear approximations via separated representations. To accomplish this nonlinear approximation, we derive a system of integral equations that describe the fully-correlated many-particle problem. The computational core of the method is the repeated construction and solution of a matrix-integral system of equations.

Specifically, our approach has the following distinctive features:

- We use an adaptive representation for single-electron functions, but our method does not depend on its details.

- We use an integral formulation of the multiparticle Schrödinger equation and a Green's function iteration to converge to the ground-state wavefunction. The Green's function is decomposed and applied using separated approximations obtained by expanding the kernel into Gaussians.

- We use a variant of the so-called alternating least squares algorithm to reduce the error of our approximation using a sum of a given number of Slater determinants.
- We compute antisymmetric inner products involving portions of the Hamiltonian operator by reducing them to formulas involving only combinations of standard integrals. In particular, we avoid the direct application of the electron-electron potential and instead compute convolutions with the Poisson kernel.

By doing this, we hope to represent the effects of correlations in the most natural and concise way possible, thus providing both computational efficiency and physical insight. We believe that this algorithm and the system of integral equations underlying it provide the foundation for a new approach to solving the multiparticle Schrödinger equation. We defer to the sequels several important issues, such as algorithmic size-consistency/extensivity and the treatment of the inter-electron cusp.

In Section II we formulate the problem more carefully, make precise some of the statements that we made in this introduction, and give a high-level description of the method. We then present the derivations and proofs in the following sections.

II Problem Formulation and Description of the Method

II.1 Formulation of the Problem

We consider the time-independent, nonrelativistic, multiparticle Schrödinger equation, and fix the nuclei according to the Born-Oppenheimer approximation, so the equation describes the steady state of an interacting system of electrons. For each of the N electrons in the system there are three spatial variables $\mathbf{r} = (x, y, z)$ and a discrete spin variable σ taking the values $\{-\frac{1}{2}, \frac{1}{2}\}$, which we combine and denote $(\mathbf{r}, \sigma)$ by γ. The Hamiltonian operator $\mathcal{H}$ is a sum of a kinetic energy operator $\mathcal{T}$, a nuclear potential operator $\mathcal{V}$, and an electron-electron interaction operator $\mathcal{W}$, defined in atomic units by

$$\mathcal{H} = \mathcal{T} + \mathcal{V} + \mathcal{W} = -\frac{1}{2}\sum_{i=1}^{N} \Delta_i + \sum_{i=1}^{N} v(\mathbf{r}_i) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j \neq i}^{N} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|}\,, \tag{1}$$

where Δ_i is the three-dimensional Laplacian acting in the variable $\mathbf{r}_i$ and $v(\mathbf{r})$ is a sum of terms of the form $-Z_a/\|\mathbf{r} - \mathbf{R}_a\|$ from a nucleus at position $\mathbf{R}_a$ with charge Z_a. The antisymmetric eigenfunctions of $\mathcal{H}$ represent electronic states of the system and are called wavefunctions. Antisymmetric means that under the exchange of any two coordinates, the wavefunction is odd, e.g. $\psi(\gamma_2, \gamma_1, \ldots) = -\psi(\gamma_1, \gamma_2, \ldots)$. The bound-state wavefunctions have negative eigenvalues, and are of greatest interest. We will focus on the ground-state wavefunction, which has the most negative eigenvalue, although the techniques can be used for other states. In summary, our goal is to find E and ψ, with E the most negative eigenvalue in

$$\mathcal{H}\psi = E\psi\,, \tag{2}$$

subject to the antisymmetry condition on ψ. Analytic methods can give qualitative results about the solutions, and determine limiting cases, but most quantitative results must be obtained numerically. Although the equation is 'just' an eigenvalue problem, its numerical solution presents several serious difficulties, among them the large number of variables and the antisymmetry condition on the solution. The simplest method that addresses these two difficulties is Hartree-Fock (HF) (see e.g. [28]), which uses the antisymmetrization of a single product, called a *Slater determinant*, to approximate the N-particle wavefunction, i.e.

$$\psi_{\mathrm{HF}} = \mathcal{A}\prod_{i=1}^{N}\phi_i(\gamma_i) = \frac{1}{N!}\begin{vmatrix} \phi_1(\gamma_1) & \phi_1(\gamma_2) & \cdots & \phi_1(\gamma_N) \\ \phi_2(\gamma_1) & \phi_2(\gamma_2) & \cdots & \phi_2(\gamma_N) \\ \vdots & \vdots & & \vdots \\ \phi_N(\gamma_1) & \phi_N(\gamma_2) & \cdots & \phi_N(\gamma_N) \end{vmatrix} . \tag{3}$$

Any antisymmetric approximation $\tilde{\psi}$ to the wavefunction ψ can be substituted into

$$\frac{\langle \mathcal{H}\tilde{\psi}, \tilde{\psi}\rangle}{\langle \tilde{\psi}, \tilde{\psi}\rangle} , \tag{4}$$

where $\langle\cdot,\cdot\rangle$ is the usual inner product, to obtain an estimate for E. This estimate gives an upper bound on the lowest value of E that solves (2). Substituting (3) into (4), one can iteratively solve for ϕ_i to minimize (4). The resulting ψ_{HF} will best approximate ψ, in the sense of providing the best estimate (4).

To improve upon HF, it is natural to consider the antisymmetrization of a sum of products

$$\psi_{(r)} = \mathcal{A}\sum_{l=1}^{r} s_l \prod_{i=1}^{N}\phi_i^l(\gamma_i) , \tag{5}$$

which could also be written as a sum of Slater determinants. The coefficients s_l are introduced in order to have $\|\phi_i^l\| = 1$. Many methods are based on this form, but they use it in different ways. The Configuration Interaction (CI) method (see e.g. [57]) chooses the functions ϕ_i^l from a preselected master set of orthogonal functions and decides on a large number r of combinations to consider, based on excitation level. Substituting (5) into (4) leads to a matrix eigenvalue problem that can be solved for the scalar coefficients s_l. The Multi-Configuration Self-Consistent Field (MCSCF) method (e.g. [20, 11]) solves for the master set of orthogonal functions as well as the scalar coefficients. There are numerous variations and combinations of these methods, too many to describe here.

II.1.1 What is New Here

In this work we construct and demonstrate a method that also uses a wavefunction of the form (5) but without constraints on the ϕ_i^l. We remove both structural constraints, such as an excitation pattern or orthogonality between single-electron functions, and representation constraints, such as those imposed by using a predetermined basis set.

Many methods (e.g. [55, 47, 39, 1, 19, 15, 18, 2, 16, 60, 41]) have loosened the constraints on the Slater determinants in one way or another, often with encouraging results. These works, however, only partially removed the constraints, and so, we claim, did not achieve the full potential of an unconstrained approximation. By removing these constraints we hope to produce much better approximations at much smaller separation rank r than existing methods allow. We also hope to provide new perspective from which to analyze and understand the wavefunction, free from the biases that physical intuition imposes.

Our hopes are based on our work in [4, 5, 43], where we developed general methods to represent and compute with functions and operators in many dimensions. We used sums of separable functions, dubbed *separated representations*, similar to (5). We found rather natural examples where removing constraints produces expansions that are *exponentially* more efficient, i.e. $r = N$ instead of 2^N or $r = \log N$ instead of N. For example, in our approach we can have a two-term representation

$$\psi = \mathcal{A}\prod_{i=1}^{N}\phi_i(\gamma_i) + \mathcal{A}\prod_{i=1}^{N}(\phi_i(\gamma_i) + \phi_{i+N}(\gamma_i)) \tag{6}$$

where $\{\phi_j\}_{j=1}^{2N}$ form an orthogonal set. To represent the same function as (6) while imposing the constraint that factors come from a master orthogonal set would force one to multiply out the second term, and thus use a representation with 2^N terms.

At present we have no proof that the wavefunction is well-approximated by a structure that would benefit from the removal of constraints. The size r needed in practice, and how it depends on the various parameters in the problem, is thus still an open question. In [4, 5, 43], the most interesting examples came from “reverse-engineering” the numerical results to obtain formulas and proofs. We therefore expect that the tools we provide here will allow an exploration of the wavefunction, perhaps revealing unexpected structure, and a strategy for a proof.

II.2 Description of the Algorithm

The removal of constraints in (5), and, thus, the basis sets, coefficients, and other structure that went along with them, also eliminates the conventional strategies for constructing (5) to minimize (4). It leads one to consider how one would compute the ground-state wavefunction if its numerical representation were not an issue. We choose to use an integral iteration, which we sketch in Section II.2.1. In Appendix A we sketch an alternative iteration based on gradient descent.

To use the form (5) we must choose some value of r, which determines the quality of the approximation. In Section II.2.2 we show how to incorporate a nonlinear fitting step within the integral iteration in order to maintain fixed r. Accomplishing this fitting requires a significant amount of machinery, which makes up the body of the paper. Eventually one would want to adaptively determine r, but we do not address that issue here.

II.2.1 A Green’s Function Iteration

The eigenvalue equation (2) contains the differential operator $\mathcal{H}$, which has both the discrete negative eigenvalue(s) that we are interested in and unbounded, continuous, positive spectrum. In [31, 32] this differential equation was reformulated as an integral equation, producing an operator with only discrete, bounded spectrum. Such integral formulations are in general far superior to differential formulations, since, e.g. numerical noise is suppressed rather than amplified. An iteration based on the integral formulation with Green’s functions was introduced in [31, 32] and used in e.g. [12, 26]. A rigorous analysis of this iteration is given in [44] based on classical theorems from [30, 33, 52, 53, 54]. In this section we review this iteration, and then modify it in Section II.2.2 to preserve our wavefunction representation (5).

Define the Green’s function

$$\mathcal{G}_\mu = (\mathcal{T} - \mu\mathcal{I})^{-1}\,, \tag{7}$$

for $\mu < 0$, and consider the Lippmann-Schwinger integral equation

$$\lambda_\mu \psi_\mu = -\mathcal{G}_\mu[(\mathcal{V}+\mathcal{W})\psi_\mu]\,. \tag{8}$$

The subscript μ on λ_μ and ψ_μ are to emphasize the dependence of the eigenvalues and eigenfunctions on μ. The operator $\mathcal{G}_\mu[(\mathcal{V}+\mathcal{W})]$ is compact, so (8) has only discrete spectrum. If $\mu = E$, then there is an eigenvalue $\lambda_\mu = 1$ and the corresponding eigenfunction ψ_μ of (8) is the desired ground-state eigenfunction of (2), as one can see by rearranging (8) into (2). We note that other eigenfunctions may be obtained by deflation.

When $\mu = E$, $\lambda_\mu = 1$ is the largest eigenvalue, so a simple iteration like the power method yields the desired ground-state eigenfunction. The eigenvalues λ_μ depend analytically on μ, so when μ is sufficiently close to E the power method will still yield an eigenfunction of (8) with energy near the minimum of (4). From ψ_μ and λ_μ one can construct an update rule for μ, based for example on applying Newton's method to solve $\lambda_\mu = 1$.

The convergence rate of the power method to produce ψ_μ and λ_μ is linear, and depends, as usual, on the gap between the eigenvalues in (8). The convergence rate of Newton's method to solve $\lambda_\mu = 1$ is quadratic, so μ will converge to E quadratically, provided that λ_μ and ψ_μ have been found at each step. In the practical use of this approach, one does not wait for the power method to converge at each step, but instead intertwines it with the update of μ. Beginning with an approximation to the energy $\mu_0 \approx E$ and an approximate wavefunction ψ_0, one converts (8) to an iteration

$$\tilde{\psi}_n = -\mathcal{G}_{\mu_n}[(\mathcal{V} + \mathcal{W})\psi_n] \,. \tag{9}$$

After each iteration one normalizes by setting

$$\psi_{n+1} = \tilde{\psi}_n / \|\tilde{\psi}_n\| \,. \tag{10}$$

Following the approach of [26], we can use the update rule

$$\mu_{n+1} = \mu_n - \langle(\mathcal{V} + \mathcal{W})\psi_n, \psi_n - \tilde{\psi}_n\rangle / \|\tilde{\psi}_n\|^2 \,, \tag{11}$$

which is equivalent to using Newton's method.

II.2.2 Approximating with Fixed Separation Rank r

We restrict the method to approximate wavefunctions of the form (5), with r fixed, by replacing the definition of $\tilde{\psi}_n$ in (9). We define $\tilde{\psi}_n$ to be the function of the form (5) that minimizes the (least-squares) error

$$\|\tilde{\psi}_n - (-\mathcal{G}_{\mu_n}[(\mathcal{V} + \mathcal{W})\psi_n])\|. \tag{12}$$

Since using (12) instead of (9) introduces an error, the update rule (11) may no longer give quadratic convergence, and in any case is not expected to converge to the true energy. One may choose to replace the update rule (11) with the more robust but slower converging rule

$$\mu_{n+1} = \frac{\langle \mathcal{H}\psi_{n+1}, \psi_{n+1}\rangle}{\|\psi_{n+1}\|^2} \,, \tag{13}$$

which is based on (4). Other rules may be possible as well. At present we do not have enough numerical experience to decide which rule to prefer.

The Green's function iteration itself does not enforce the antisymmetry condition. In order to assure convergence to an antisymmetric solution, we use the pseudo-norm induced by the pseudo inner product $\langle \cdot, \cdot \rangle_{\mathcal{A}} = \langle \mathcal{A}(\cdot), \mathcal{A}(\cdot)\rangle$, as we did in [5].

The least-squares problem (12) is non-linear, and so very difficult in general. To simplify notation in the description of our method, we now suppress the index n in (12) and consider a single problem of that form. We begin by setting $\tilde{\psi} = \psi$, and then iteratively improve $\tilde{\psi}$ to reduce (12). Although we can see several strategies for improving $\tilde{\psi}$, for concreteness we will restrict our description to the strategy most similar to [5]. To improve the approximation $\tilde{\psi}$ we loop through the variables (electrons). The functions in variables other than the current variable are fixed, and the functions in the current variable are modified to minimize the overall error (12). The error (12) depends linearly on the functions in a single variable, so the minimization becomes much easier. This general Alternating Least-Squares (ALS) approach is well-known (see e.g. [27, 36, 38, 10, 14, 58]). Although to minimize (12) one may need to loop through the variables multiple times, it appears to be more cost effective to loop only once and then do the next Green's function iteration. We alternate through the directions, but for ease of exposition we describe the $k = 1$ case. So, $\tilde{\phi}^l_k$ is fixed for $k > 1$, and we will solve for the values of $\tilde{\phi}^l_1$ for all l.

To refine in the current variable, we set up and solve a linear least-squares problem. The normal equations for a least-squares problem are derived by taking a gradient with respect to the free parameters and setting the result equal to zero. As long as the approximating function is linear and not degenerate in these parameters, the resulting equations are linear and have a unique solution, which minimizes the error with respect to these parameters. Usually these free parameters are coefficients of the representation in some fixed basis. For example, to find the coefficients $\{c_i\}$ to minimize

$$\left\| f - \sum_i c_i g_i \right\|^2 = \left\langle f - \sum_i c_i g_i, f - \sum_i c_i g_i \right\rangle , \tag{14}$$

construct the normal equations

$$\mathbb{A}\mathbf{x} = \mathbf{b} \,, \tag{15}$$

with

$$A(k, i) = \langle g_k, g_i \rangle \quad \text{and} \quad b(k) = \langle g_k, f \rangle \,, \tag{16}$$

solve them, and set $c_i = x(i)$. Instead of using coefficients in some basis as our parameters, we take the parameters to be the point values of our functions $\tilde{\phi}^l_1$, so that the gradient becomes a variational derivative. Formally, we consider a basis of delta functions $\delta(\gamma - \cdot)$ and let their coefficients be our parameters. We still obtain linear normal equations (15), but now $\mathbf{b}$ and $\mathbf{x}$ are vectors of functions, and $\mathbb{A}$ is a matrix of integral operators. Specifically, $b(l)$ is a function of γ, $x(l')$ is a function of γ', and $A(l, l')$ is an integral operator mapping functions of γ' to functions of γ. The kernels in $\mathbb{A}$ are formally defined by

$$A(l, l')(\gamma, \gamma') = \tilde{s}_l \tilde{s}_{l'} \left\langle \delta(\gamma - \gamma_1) \prod_{i=2}^{N} \tilde{\phi}^l_i(\gamma_i), \delta(\gamma' - \gamma_1) \prod_{i=2}^{N} \tilde{\phi}^{l'}_i(\gamma_i) \right\rangle_{\mathcal{A}} , \tag{17}$$

and the functions in $\mathbf{b}$ are defined by

$$b(l)(\gamma) = \tilde{s}_l \sum_m^r s_m \left\langle \delta(\gamma - \gamma_1) \prod_{i=2}^{N} \tilde{\phi}^l_i(\gamma_i), -\mathcal{G}_\mu[\mathcal{V} + \mathcal{W}] \prod_{i=1}^{N} \phi^m_i(\gamma_i) \right\rangle_{\mathcal{A}} . \tag{18}$$

Once we solve (15), we set $\tilde{\phi}^l_1 = x(l)$. To enforce the normalization convention $\|\tilde{\phi}^l_1\| = 1$ we can divide $\tilde{\phi}^l_1$ by its norm and incorporate the norm into $\tilde{s}_l$.

To solve the matrix-integral system (15), we need an iterative method for solving linear systems that uses only operations compatible with integral operators, such as matrix-vector products, vector scales and additions, and vector inner products. Typically the matrix $\mathbb{A}$ in normal equations is positive-definite. Our operator $\mathbb{A}$ is only semidefinite due to the nullspace in the antisymmetric pseudonorm. Fortunately, $\mathbf{b}$ was computed with the same pseudonorm and has no component in the nullspace of $\mathbb{A}$, so we can still use methods for positive-definite matrices. Based on these considerations, we choose to use the Conjugate Gradient iterative method (see e.g. [21]) to solve (15). One initializes with $\mathbf{r} = \mathbf{b} - \mathbb{A}\mathbf{x}$, $\mathbf{v} = \mathbf{r}$, and $c = \langle \mathbf{r}, \mathbf{r} \rangle$, and then the core of the method is the sequence of assignments $\mathbf{z} \leftarrow \mathbb{A}\mathbf{v}$, $t \leftarrow c/\langle \mathbf{v}, \mathbf{z} \rangle$, $\mathbf{x} \leftarrow \mathbf{x} + t\mathbf{v}$, $\mathbf{r} \leftarrow \mathbf{r} - t\mathbf{z}$, $d \leftarrow \langle \mathbf{r}, \mathbf{r} \rangle$, $\mathbf{v} \leftarrow \mathbf{r} + (d/c)\mathbf{v}$, and $c \leftarrow d$, applied iteratively.

One advantage of using this iterative method with integral operators is that our algorithm does not rely on any particular basis. The representation of $\mathbf{x}$ can naturally be adaptive in γ, for example refining near the nuclei as indicated by the refinement in $\mathbf{b}$. We assume the availability of some adaptive, high-accuracy representation for single-electron functions, such as the polynomial multiwavelet representation demonstrated in [25, 26], which effectively eliminates the basis-set error. For the estimates of computational cost, we use M to denote the cost to represent a function of γ, or integrate such a function. The antisymmetry constraint requires $N \leq M$, and in general we expect M to be much larger than N.

II.2.3 Summary of the Remainder of the Paper

The core of the paper is the development of the methods needed to construct $\mathbb{A}$ in (17) and $\mathbf{b}$ in (18). First, in Section III, we develop the machinery and algorithms for computing antisymmetric inner products involving the operators $\mathcal{T}$, $\mathcal{V}$, and $\mathcal{W}$. Our formulation uses low-rank perturbations of matrices, thus avoiding cofactor expansions. We also avoid explicit construction of $\mathcal{W}$ by incorporating its effect via spatial convolutions with the Poisson kernel in three dimensions. Second, in Section IV, we show how to compute antisymmetric inner products involving these operators and the delta function $\delta(\gamma - \gamma_1)$. Again the key is to use low-rank perturbations of matrices.

In Section V we assemble all our tools to demonstrate how to perform our main algorithm, and in particular how to construct $\mathbb{A}$ in (17) and $\mathbf{b}$ in (18). We also gather the computational cost for the whole method. The cost depends on the number of electrons N, the separation rank r, the one-particle representation cost M, the number of Green's function iterations I (see Section II.2.1), and the number of conjugate gradient iterations S (see Section II.2.2). Although S in theory could be as many as the number of degrees of freedom rM, we have a very good starting point, and so expect only a very small constant number to be needed. We use $M \log M$ to denote the generic cost to convolve a function of γ with the Poisson kernel $1/\|\mathbf{r}\|$. A Fourier-based Poisson solver on a uniform grid would achieve this complexity; for adaptive methods such as we use it is very difficult to state the cost (see [7, 17]). We use L to denote the number of terms used to approximate the Green's function to relative error ϵ with Gaussians, and prove in Section V.1 that $L = \mathcal{O}((\ln \epsilon)^2)$ independent of μ and N. The final computational cost is then

$$\mathcal{O}(Ir^2N^2[L(N + M \log M) + S(N + M)]). \tag{19}$$

For comparison, the cost to evaluate a single antisymmetric inner product via Löwdin's rules is $\mathcal{O}(N^2(N + M))$.

II.3 Further Considerations

We have implemented the method developed here and tested it sufficiently to verify the correctness of the algorithm as presented. The numerical results are too preliminary to allow us to make any particular claims at this point, however, so we will present them separately. The linear algebra accelerations based on Appendix B have not yet been implemented.

We develop the method in terms of the total variable γ without specifying the spin states. If a specific spin state is imposed on our initial trial wavefunction ψ_0, the iteration will preserve this state.

The representation (5) does not account for the inter-electron cusp (see e.g. [56, 46, 35, 49, 50, 34, 37]), and thus we cannot hope to achieve small error ϵ in the wavefunction with small r. As with Configuration Interaction methods, we may still be able to achieve small error in the energy difference of two systems, which is often the quantity of interest in Chemistry. For the current work, we fix r and adapt $\phi_i^l(\gamma_i)$ and s_l to minimize the error ϵ, rather than fixing ϵ and adaptively determining r. We are developing an extension to (5) that incorporates the cusp, and hope to achieve small error ϵ through it.

Similarly, (5) is not size-consistent/extensive, and thus is not suitable for large systems. We are also developing an extension to (5) suitable for large systems, and hope to achieve linear scaling through it.

Although we have focused on the multiparticle Schrödinger equation, the tools that we have developed are another step towards general-purpose, automatically adaptive methods for solving high-dimensional problems.

III Antisymmetric Inner Products

In this section we develop methods for computing antisymmetric inner products involving $\mathcal{W}$, $\mathcal{V}$, and $\mathcal{T}$. For this purpose, after setting notation, we develop methods for computing with low rank perturbations of matrices, review the antisymmetry constraint and define a notion of maximum coincidence. With these tools we then derive the main formulas.

III.1 Notation

We denote a column *vector* with suppressed indices by $\mathbf{F}$ and with explicit indices by $F(i)$. We denote its conjugate transpose by $\mathbf{F}^*$. We use $\mathbf{e}_i$ to denote the column vector that is one in coordinate i and zero otherwise. A linear *operator* is written $\mathcal{L}$. We denote a *matrix* with suppressed indices by $\mathbb{L}$ and with explicit indices by $L(i,j)$. Recalling that $\mathbf{r} = (x,y,z) \in \mathbf{R}^3$, we combine spatial integration with summation over spins and define the integral

$$\int f(\gamma) d\gamma = \sum_{\sigma \in \{-1/2, 1/2\}} \int f(\mathbf{r}, \sigma) d\mathbf{r}. \tag{20}$$

We define the action of the single-electron kinetic and nuclear potential operators by

$$(\mathcal{T}_* + \mathcal{V}_*)\,[f](\gamma) = \left(-\frac{1}{2}\Delta + v(\mathbf{r})\right) f(\gamma) = \left(-\frac{1}{2}\Delta + v(\mathbf{r})\right) f(\mathbf{r},\sigma). \tag{21}$$

In what follows we will reduce the action of the inter-electron potential operator $\mathcal{W}$ to convolutions with the Poisson kernel, so we define

$$\mathcal{W}_{\mathcal{P}}\,[f](\mathbf{r}) = \int \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} f(\gamma') d\gamma' = \sum_{\sigma' \in \{-1/2, 1/2\}} \int \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} f(\mathbf{r}', \sigma') d\mathbf{r}'\,. \tag{22}$$

We allow these operators to be applied componentwise to vectors and matrices of functions.

Next, we define $\Phi = \prod_{i=1}^N \phi_i(\gamma_i)$, so for example we can write $\langle \tilde{\Phi}, \Phi \rangle_{\mathcal{A}}$ instead of $\left\langle \prod_{i=1}^N \tilde{\phi}_i(\gamma_i), \prod_{i=1}^N \phi_i(\gamma_i) \right\rangle_{\mathcal{A}}$. We also associate with the product Φ a vector of N functions of a single variable,

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_N \end{bmatrix}. \tag{23}$$

We can then, for example, construct a new vector of functions $\mathbf{\Theta}$ by applying a matrix to an old one, as in $\mathbf{\Theta} = \mathbb{L}^{-1}\tilde{\mathbf{\Phi}}$. Although we do linear algebra operations on these vectors, we note that $\mathbf{\Phi} + \tilde{\mathbf{\Phi}}$ does not correspond to $\Phi + \tilde{\Phi}$, so there is not a true vector-space structure. Our formulas contain fairly complicated expressions with such vectors, such as $\int \mathbf{\Phi}^* \mathcal{W}_{\mathcal{P}}\,[\mathbf{\Theta}\mathbf{\Phi}^*]\mathbf{\Theta} d\gamma$. To parse this expression, we note that $\mathbf{\Theta}$ is a column vector of functions and $\mathbf{\Phi}^*$ is a row vector of functions, so $\mathbf{\Theta}\mathbf{\Phi}^*$ is a matrix of functions. Then $\mathcal{W}_{\mathcal{P}}\,[\mathbf{\Theta}\mathbf{\Phi}^*]$ is still a matrix of functions, but applying $\mathbf{\Phi}^*$ on the left and $\mathbf{\Theta}$ on the right yields a single function, which is integrated in the implied variable γ to yield a number. When explicit specification of the variable involved is needed, the notation $\mathbf{\Phi}(\gamma)$ indicates that the single variable γ is used in all the functions.

III.2 Determinants of Low-Rank Perturbations of Matrices

Since the antisymmetric inner product involves determinants, we will use some linear algebra relations for them. Proposition 2 in this section is used heavily, and is the key to avoiding rather unpleasant cofactor expansions.

Proposition 1 (Determinant via Schur Complement) *Let $\mathbb{A}$ be a nonsingular square matrix, $\mathbb{D}$ a square matrix, and $\mathbb{B}$ and $\mathbb{C}$ matrices of appropriate size. Then*

$$\begin{vmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{C} & \mathbb{D} \end{vmatrix} = |\mathbb{A}| \left|\mathbb{D} - \mathbb{C}\mathbb{A}^{-1}\mathbb{B}\right| . \tag{24}$$

Proof: (see e.g. [51]) It is easy to verify directly that

$$\begin{bmatrix} \mathbb{A} & \mathbb{B} \\ \mathbb{C} & \mathbb{D} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & 0 \\ \mathbb{C}\mathbb{A}^{-1} & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbb{A} & 0 \\ 0 & \mathbb{D} - \mathbb{C}\mathbb{A}^{-1}\mathbb{B} \end{bmatrix} \begin{bmatrix} \mathbb{I} & \mathbb{A}^{-1}\mathbb{B} \\ 0 & \mathbb{I} \end{bmatrix} . \tag{25}$$

Since the determinants of the first and third matrices are equal to one, the determinant of the middle matrix gives the desired result. □

Proposition 2 (Determinant of a Perturbation of the Identity) *Let $\{\mathbf{u}_q\}_{q=1}^{Q}$ and $\{\mathbf{v}_q\}_{q=1}^{Q}$ be two sets of vectors of the same length, and $\mathbf{u}_q\mathbf{v}_q^*$ denote the outer product of $\mathbf{u}_q$, and $\mathbf{v}_q$. Then*

$$\left| \mathbb{I} + \sum_{q=1}^{Q} \mathbf{u}_q \mathbf{v}_q^* \right| = \begin{vmatrix} 1 + \mathbf{v}_1^*\mathbf{u}_1 & \mathbf{v}_1^*\mathbf{u}_2 & \cdots & \mathbf{v}_1^*\mathbf{u}_Q \\ \mathbf{v}_2^*\mathbf{u}_1 & 1 + \mathbf{v}_2^*\mathbf{u}_2 & \cdots & \mathbf{v}_2^*\mathbf{u}_Q \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_Q^*\mathbf{u}_1 & \mathbf{v}_Q^*\mathbf{u}_2 & \cdots & 1 + \mathbf{v}_Q^*\mathbf{u}_Q \end{vmatrix} . \tag{26}$$

Proof: Let $\mathbb{U}$ be the matrix with the vectors $\{\mathbf{u}_q\}$ as its columns, and $\mathbb{V}$ the matrix with the vectors $\{\mathbf{v}_q\}$ as its columns. Note that $\mathbb{U}$ and $\mathbb{V}$ are of the same size. By Proposition 1 we have

$$\begin{vmatrix} \mathbb{I} & \mathbb{U} \\ -\mathbb{V}^* & \mathbb{I} \end{vmatrix} = |\mathbb{I} + \mathbb{V}^*\mathbb{U}| , \tag{27}$$

which evaluates to the right side of (26). Exchanging the roles of $\mathbb{A}$ and $\mathbb{D}$ in Proposition 1 we have

$$\begin{vmatrix} \mathbb{I} & \mathbb{U} \\ -\mathbb{V}^* & \mathbb{I} \end{vmatrix} = |\mathbb{I} + \mathbb{U}\mathbb{V}^*| , \tag{28}$$

which evaluates to the left side of (26). □

The $Q = 1$ case is well-known (see e.g. [51]) but we have not found the general case in the literature.

III.3 The Modified Pseudo-inverse

The singular value decomposition (SVD) (e.g. [21]) of a $N \times N$ matrix is

$$\mathbb{A} = \sum_{i=1}^{N} s_i \mathbf{u}_i \mathbf{v}_i^* = \mathbb{U}\mathbb{S}\mathbb{V}^* \,, \tag{29}$$

where the matrices $\mathbb{U}$ and $\mathbb{V}$ are unitary and the singular values $\{s_i\}$ are non-neganive and in descending order. The left singular vectors $\{\mathbf{u}_i\}$ form an orthonormal set, as do the right singular vectors $\{\mathbf{v}_i\}$. The pseudo-inverse is defined as

$$\mathbb{A}^{\dagger} = \sum_{i=1}^{N-Q} s_i^{-1} \mathbf{v}_i \mathbf{u}_i^* \,, \tag{30}$$

where Q is the dimension of the (numerical) nullspace. We also define a projection matrix onto the nullspace

Definition 3

$$\mathbb{A}^{\perp} = \sum_{i=N-Q+1}^{N} \mathbf{v}_i \mathbf{u}_i^* \tag{31}$$

and a modified pseudo-inverse

Definition 4 (Modified Pseudo-Inverse)

$$\mathbb{A}^{\ddagger} = \mathbb{A}^{\dagger} + \mathbb{A}^{\perp} \,. \tag{32}$$

Note that $\mathbb{A}^{\perp}$ and thus $\mathbb{A}^{\ddagger}$ are not uniquely defined since the choice of basis for the nullspace is not unique. For our purposes any consistent choice works. The modified pseudo-inverse behaves much like the pseudo-inverse, but always has a non-zero determinant,

$$|\mathbb{A}^{\ddagger}| = \left(|\mathbb{U}||\mathbb{V}^*| \prod_{s_i \neq 0} s_i \right)^{-1} \neq 0 \,. \tag{33}$$

III.4 The Antisymmetrizer and Löwdin's Rule

Given a separable function, its antisymmetric projection can be found by applying the *antisymmetrizer* $\mathcal{A}$ (see e.g. [48]), also called the *skew-symmetrization* or *alternation* (see e.g. [45, 51]), resulting in a Slater determinant. In the vector notation (23), we have

$$\mathcal{A}\Phi = \frac{1}{N!} \left| \begin{bmatrix} \boldsymbol{\Phi}(\gamma_1) & \cdots & \boldsymbol{\Phi}(\gamma_N) \end{bmatrix} \right| = \frac{1}{N!} \begin{vmatrix} \phi_1(\gamma_1) & \phi_1(\gamma_2) & \cdots & \phi_1(\gamma_N) \\ \phi_2(\gamma_1) & \phi_2(\gamma_2) & \cdots & \phi_2(\gamma_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\gamma_1) & \phi_N(\gamma_2) & \cdots & \phi_N(\gamma_N) \end{vmatrix} . \tag{34}$$

One cannot explicitly form a Slater determinant $\mathcal{A}\Phi$ for large N since it would have $N!$ terms. However, one can compute the antisymmetric pseudo inner product

$$\langle \tilde{\Phi}, \Phi \rangle_{\mathcal{A}} \overset{\mathrm{def}}{=} \langle \mathcal{A}\tilde{\Phi}, \mathcal{A}\Phi \rangle = \langle \tilde{\Phi}, \mathcal{A}\Phi \rangle = \langle \mathcal{A}\tilde{\Phi}, \Phi \rangle, \tag{35}$$

where the first equality is a definition and the others follow since $\mathcal{A}$ is an orthogonal projector. It is not a true inner product because it has a nullspace. To compute (35), first construct the matrix $\mathbb{L}$ with entries

$$L(i,j) = \langle \tilde{\phi}_i, \phi_j \rangle \tag{36}$$

at cost $\mathcal{O}(N^2M)$. Then use $\langle \tilde{\Phi}, \Phi \rangle_{\mathcal{A}} = \langle \mathcal{A}\tilde{\Phi}, \Phi \rangle$ and move the integrals inside the determinant to obtain

$$\langle \tilde{\Phi}, \Phi \rangle_{\mathcal{A}} = \frac{1}{N!}|\mathbb{L}|\,, \tag{37}$$

which is the so-called Löwdin's rule (e.g. [40, 48]). Since $\mathbb{L}$ is an ordinary matrix, its determinant can be computed with cost $\mathcal{O}(N^3)$ (or less). The denominator $N!$ need never be computed, since it will occur in every term in our equations, and so cancels.

Our method for enforcing the antisymmetry constraint, as described in [5], is to use the pseudo-norm based on the antisymmetric inner product $\langle \cdot, \cdot \rangle_{\mathcal{A}}$ for the least-squares fitting (12).

III.5 Maximum Coincidence

Consider two products, $\Phi = \prod_{i=1}^N \phi_i(\gamma_i)$ and $\tilde{\Phi} = \prod_{i=1}^N \tilde{\phi}_i(\gamma_i)$, stored in the vector notation of (23) as $\mathbf{\Phi}$ and $\tilde{\mathbf{\Phi}}$. To specify which functions were used to compute $\mathbb{L}$ in (36), we use the notation $\mathbb{L}(\tilde{\Phi}, \Phi)$. The matrix of inner products $\mathbb{L} = \mathbb{L}(\tilde{\Phi}, \Phi)$ is in general full. Defining

$$\mathbf{\Theta} = \mathbb{L}^{-1}\tilde{\mathbf{\Phi}}\,, \tag{38}$$

we have

$$\begin{aligned}\mathcal{A}\Theta = \frac{1}{N!}\left|\left[\begin{array}{ccc} (\mathbb{L}^{-1}\tilde{\mathbf{\Phi}})(\gamma_1) & \cdots & (\mathbb{L}^{-1}\tilde{\mathbf{\Phi}})(\gamma_N) \end{array}\right]\right| \\ = |\mathbb{L}^{-1}|\frac{1}{N!}\left|\left[\begin{array}{ccc} \tilde{\mathbf{\Phi}}(\gamma_1) & \cdots & \tilde{\mathbf{\Phi}}(\gamma_N) \end{array}\right]\right| = |\mathbb{L}^{-1}|\mathcal{A}\tilde{\Phi}\,.\end{aligned} \tag{39}$$

Thus the antisymmetrizations of $\tilde{\Phi}$ and Θ are the same up to a constant, and we can use Θ instead of $\tilde{\Phi}$ in calculations. The advantage of using Θ is that the resulting matrix of inner products $\hat{\mathbb{L}} = \mathbb{L}(\Theta, \Phi) = \mathbb{I}$; in other words, we have the biorthogonality property $\langle \theta_i, \phi_j \rangle = \delta_{ij}$. To show this, write the matrix $\hat{\mathbb{L}}$ as $\int \mathbf{\Theta}\mathbf{\Phi}^* d\gamma$, where the integration is elementwise. Substituting for $\mathbf{\Theta}$, we have $\int (\mathbb{L}^{-1}\tilde{\mathbf{\Phi}})\mathbf{\Phi}^* d\gamma$. Since the integration is elementwise it commutes with $\mathbb{L}^{-1}$ and we have $\mathbb{L}^{-1}\int \tilde{\mathbf{\Phi}}\mathbf{\Phi}^* d\gamma = \mathbb{L}^{-1}\mathbb{L} = \mathbb{I}$. The computational cost to construct $\mathbf{\Theta}$ is $\mathcal{O}(N^2(N+M))$.

When the matrix $\mathbb{L}$ in (36) is singular, we define $\mathbf{\Theta} = \mathbb{L}^{\ddagger}\tilde{\mathbf{\Phi}}$ using the modified pseudo-inverse of Definition 4. By the same argument as before, we have $|\mathbb{L}^{\ddagger}|^{-1}\mathcal{A}\Theta = \mathcal{A}\tilde{\Phi}$. The matrix $\int \mathbf{\Theta}\mathbf{\Phi}^* d\gamma$ evaluates to $\mathbb{L}^{\ddagger}\mathbb{L} = \mathbb{I} - \sum_{i=N-Q+1}^N \mathbf{v}_i\mathbf{v}_i^*$. For notational convenience in later sections, we will re-index our singular values and vectors so that the first Q generate the nullspace, rather than the last Q.

Remark 5 *Within Configuration Interaction methods, the functions in* $\mathbf{\Phi}$ *and* $\tilde{\mathbf{\Phi}}$ *are taken from a master set of orthonormal functions, and* $\mathbf{\Theta}$ *is simply a signed permutation of* $\tilde{\mathbf{\Phi}}$ *so that* $\phi_j = \theta_j$ *for as many* j *as possible. This is known as the 'maximum coincidence' ordering. The construction we use generalizes this notion.*

III.6 Antisymmetric Inner Product with the Electron-Electron Potential $\mathcal{W}$ Present

In this section we derive formulas for computing antisymmetric inner products that include the electron-electron interaction potential. Although the derivation is somewhat messy, the resulting formulas are rather clean, and we use them verbatim in the computations. The main ideas are given in this section, and then reused in later sections for other cases.

Proposition 6 *When $\mathbb{L}$ from (36) is nonsingular,*

$$\left\langle \tilde{\Phi}, \mathcal{W}\Phi \right\rangle_{\mathcal{A}} \overset{\text{def}}{=} \left\langle \mathcal{A}\prod_{j=1}^{N} \tilde{\phi}_j(\gamma_j), \left(\frac{1}{2}\sum_{i\neq j} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|} \right) \prod_{j=1}^{N} \phi_j(\gamma_j) \right\rangle \tag{40}$$

is equal to

$$\frac{1}{2}\frac{|\mathbb{L}|}{N!} \int \mathbf{\Phi}^*\mathbf{\Theta}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^*\mathbf{\Theta}\right] - \mathbf{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Theta}\mathbf{\Phi}^*\right]\mathbf{\Theta} d\gamma\,, \tag{41}$$

where $\mathbf{\Theta} = \mathbb{L}^{-1}\tilde{\mathbf{\Phi}}$.

Proof: Using the maximum-coincidence procedure in Section III.5, (40) is equal to $|\mathbb{L}|\langle\Theta, \mathcal{W}\Phi\rangle_{\mathcal{A}}$. We reorganize and find that we must compute

$$\frac{1}{2}\frac{|\mathbb{L}|}{N!} \int \left(\sum_{i\neq j} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|} \right) \prod_{j=1}^{N} \overline{\phi}_j(\gamma_j) \begin{vmatrix} \theta_1(\gamma_1) & \theta_1(\gamma_2) & \cdots & \theta_1(\gamma_N) \\ \theta_2(\gamma_1) & \theta_2(\gamma_2) & \cdots & \theta_2(\gamma_N) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_N(\gamma_1) & \theta_N(\gamma_2) & \cdots & \theta_N(\gamma_N) \end{vmatrix} d\gamma_1 \cdots d\gamma_N\,. \tag{42}$$

By moving the sum outside of the integral, we can integrate in all directions except γ_i and γ_j. Using $\langle\theta_m, \phi_n\rangle = \delta_{mn}$, we obtain

$$\begin{aligned} &\frac{1}{2}\frac{|\mathbb{L}|}{N!} \sum_{i\neq j} \int \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \begin{vmatrix} 1 & \cdots & \overline{\phi}_i(\gamma)\theta_1(\gamma) & \cdots & \overline{\phi}_j(\gamma')\theta_1(\gamma') & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \overline{\phi}_i(\gamma)\theta_i(\gamma) & \cdots & \overline{\phi}_j(\gamma')\theta_i(\gamma') & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \overline{\phi}_i(\gamma)\theta_j(\gamma) & \cdots & \overline{\phi}_j(\gamma')\theta_j(\gamma') & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & \overline{\phi}_i(\gamma)\theta_N(\gamma) & \cdots & \overline{\phi}_j(\gamma')\theta_N(\gamma') & \cdots & 1 \end{vmatrix} d\gamma d\gamma' \\ &= \frac{1}{2}\frac{|\mathbb{L}|}{N!} \sum_{i\neq j} \int \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \left| \mathbb{I} + \left(\overline{\phi}_i(\gamma)\mathbf{\Theta}(\gamma) - \mathbf{e}_i\right)\mathbf{e}_i^* + \left(\overline{\phi}_j(\gamma')\mathbf{\Theta}(\gamma') - \mathbf{e}_j\right)\mathbf{e}_j^* \right| d\gamma d\gamma'\,. \end{aligned} \tag{43}$$

Since the inner matrix is a low-rank perturbation of the identity, we reduce its determinant using Proposition 2 and obtain

$$\frac{1}{2}\frac{|\mathbb{L}|}{N!} \sum_{i\neq j} \int \frac{1}{\|\mathbf{r} - \mathbf{r}'\|} \overline{\phi}_i(\gamma)\overline{\phi}_j(\gamma') \begin{vmatrix} \theta_i(\gamma) & \theta_i(\gamma') \\ \theta_j(\gamma) & \theta_j(\gamma') \end{vmatrix} d\gamma d\gamma'\,. \tag{44}$$

The determinant is zero if $j = i$, so we do not need to explicitly prohibit it as we needed to in (43) and above. The antisymmetrization has caused a convenient cancellation of a fictitious self-interaction, and, thus, allowed us to decouple the two sums. Expanding out the determinant and rearranging the terms, we obtain

$$\frac{1}{2}\frac{|\mathbb{L}|}{N!}\int\left(\sum_i \overline{\phi}_i(\gamma)\theta_i(\gamma)\right)\left[\int \frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\left(\sum_j \overline{\phi}_j(\gamma')\theta_j(\gamma')\right)d\gamma'\right]d\gamma$$
$$-\frac{1}{2}\frac{|\mathbb{L}|}{N!}\int\sum_i\sum_j \overline{\phi}_i(\gamma)\theta_j(\gamma)\left[\int \frac{1}{\|\mathbf{r}-\mathbf{r}'\|}\overline{\phi}_j(\gamma')\theta_i(\gamma')d\gamma'\right]d\gamma\,. \quad (45)$$

In our compact notation, this yields (41). □

We now consider the computational cost of (41). In the first term in (41), computing $\mathbf{\Phi}^*\mathbf{\Theta}$ costs $\mathcal{O}(NM)$, applying $\mathcal{W}_{\mathcal{P}}[\cdot]$ to it costs $\mathcal{O}(M\log M)$, and the integral in γ costs $\mathcal{O}(M)$. In the second term, $\mathbf{\Phi}\mathbf{\Theta}^*$ costs $\mathcal{O}(N^2M)$, applying $\mathcal{W}_{\mathcal{P}}[\cdot]$ to it costs $\mathcal{O}(N^2M\log M)$, applying $\mathbf{\Theta}^*$ and then $\mathbf{\Phi}$ costs $\mathcal{O}(N^2M)$, and then the integral in γ costs $\mathcal{O}(M)$. Including the cost to construct $\mathbf{\Theta}$, our total cost is $\mathcal{O}(N^2(N+M\log M))$.

III.6.1 The Singular Case

In this section we investigate the case when the matrix $\mathbb{L}$ from (36) is singular. Inserting the definition $\mathbf{\Theta}=\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}$ into our main formula (41), we have

$$\frac{1}{2}\frac{|\mathbb{L}|}{N!}\int \mathbf{\Phi}^*\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^*\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}\right]-\mathbf{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}\mathbf{\Phi}^*\right]\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}d\gamma\,. \quad (46)$$

In terms of the SVD (29), we can express

$$\mathbb{L}^{-1}=\sum_{j=1}^{N}s_j^{-1}\mathbf{v}_j\mathbf{u}_j^* \quad \text{and} \quad |\mathbb{L}|=|\mathbb{U}||\mathbb{V}^*|\prod_i s_i\,. \quad (47)$$

Inserting these expressions into (46), we have

$$\frac{1}{2}\frac{|\mathbb{U}||\mathbb{V}^*|\prod_i s_i}{N!}\int \mathbf{\Phi}^*\sum_{j=1}^{N}s_j^{-1}\mathbf{v}_j\mathbf{u}_j^*\tilde{\mathbf{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^*\sum_{k=1}^{N}s_k^{-1}\mathbf{v}_k\mathbf{u}_k^*\tilde{\mathbf{\Phi}}\right]$$
$$-\mathbf{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\sum_{j=1}^{N}s_j^{-1}\mathbf{v}_j\mathbf{u}_j^*\tilde{\mathbf{\Phi}}\mathbf{\Phi}^*\right]\sum_{k=1}^{N}s_k^{-1}\mathbf{v}_k\mathbf{u}_k^*\tilde{\mathbf{\Phi}}d\gamma$$
$$=\frac{1}{2}\frac{|\mathbb{U}||\mathbb{V}^*|}{N!}\sum_{j=1}^{N}\sum_{k=1}^{N}\prod_{i\neq j,k}s_i\int \mathbf{\Phi}^*\mathbf{v}_j\mathbf{u}_j^*\tilde{\mathbf{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^*\mathbf{v}_k\mathbf{u}_k^*\tilde{\mathbf{\Phi}}\right]$$
$$-\mathbf{\Phi}^*\mathbf{v}_j\mathcal{W}_{\mathcal{P}}\left[\mathbf{u}_j^*\tilde{\mathbf{\Phi}}\mathbf{\Phi}^*\mathbf{v}_k\right]\mathbf{u}_k^*\tilde{\mathbf{\Phi}}d\gamma\,. \quad (48)$$

If $\mathbb{L}$ is singular then at least one s_i is zero, and only terms that exclude those from the product in (48) are nonzero. Since we exclude two indices in the product, if more than two s_i are zero then the entire inner product is zero. If exactly two are zero then only one term in the sum survives. If exactly one is zero then we can simplify from a double to a single sum, using symmetry. Recalling the modified pseudo inverse from Definition 4 and sorting the zero s_i to the beginning for notational convenience, we obtain the following propositions.

Proposition 7 *When the rank-deficiency of* $\mathbb{L}$ *is more than two, the antisymmetric inner product (40) evaluates to zero.*

Proposition 8 *When the rank-deficiency of $\mathbb{L}$ is equal to two, the antisymmetric inner product (40) is equal to*

$$\frac{1}{|\mathbb{L}^{\ddagger}|N!}\int \mathbf{\Phi}^{*}\mathbf{v}_{1}\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^{*}\mathbf{v}_{2}\mathbf{u}_{2}^{*}\tilde{\mathbf{\Phi}}\right]-\mathbf{\Phi}^{*}\mathbf{v}_{1}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^{*}\mathbf{v}_{2}\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}\right]\mathbf{u}_{2}^{*}\tilde{\mathbf{\Phi}}d\gamma\,. \tag{49}$$

Proposition 9 *When the rank-deficiency of $\mathbb{L}$ is equal to one, defining $\mathbf{\Theta}=\mathbb{L}^{\dagger}\tilde{\mathbf{\Phi}}$ or $\mathbf{\Theta}=\mathbb{L}^{\ddagger}\tilde{\mathbf{\Phi}}$, the antisymmetric inner product (40) is equal to*

$$\frac{1}{|\mathbb{L}^{\ddagger}|N!}\int \mathbf{\Phi}^{*}\mathbf{v}_{1}\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^{*}\mathbf{\Theta}\right]-\mathbf{\Phi}^{*}\mathbf{v}_{1}\mathcal{W}_{\mathcal{P}}\left[\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}\mathbf{\Phi}^{*}\right]\mathbf{\Theta}d\gamma\,. \tag{50}$$

In computing (49), constructing $\mathbf{\Phi}^{*}\mathbf{v}_{1}$, $\mathbf{\Phi}^{*}\mathbf{v}_{2}$, $\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}$, and $\mathbf{u}_{2}^{*}\tilde{\mathbf{\Phi}}$ costs $\mathcal{O}(NM)$, applying $\mathcal{W}_{\mathcal{P}}\left[\cdot\right]$ costs $\mathcal{O}(M\log M)$ and, finally, the integral in γ costs $\mathcal{O}(M)$. In computing (50), the first term costs $\mathcal{O}(NM)$ to form $\mathbf{\Phi}^{*}\mathbf{\Theta}$, $\mathcal{O}(M\log M)$ to apply $\mathcal{W}_{\mathcal{P}}\left[\cdot\right]$, and $\mathcal{O}(M)$ to integrate in γ. The second term costs $\mathcal{O}(NM)$ to form $\mathbf{u}_{1}^{*}\tilde{\mathbf{\Phi}}\mathbf{\Phi}$, $\mathcal{O}(NM\log M)$ to apply $\mathcal{W}_{\mathcal{P}}\left[\cdot\right]$, $\mathcal{O}(NM)$ to apply $\mathbf{\Theta}$, and $\mathcal{O}(M)$ to integrate in γ. In total, the computational cost for the singular cases are less than the cost of the nonsingular case.

Remark 10 *In the Configuration Interaction context, rank-deficiency two corresponds to a double excitation. The vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ would be zero except for a single entry, and so select the locations of the excited electrons out of $\mathbf{\Phi}$ and $\tilde{\mathbf{\Phi}}$. Proposition 8 then reduces to the Slater-Condon rules [13].*

III.7 Antisymmetric Inner Product with $\mathcal{T}$ and/or $\mathcal{V}$ Present

Since $\mathcal{T}$ and $\mathcal{V}$ both have the structure of a sum of one-directional operators, we state the formulas for their sum, although of course they can be treated individually.

Proposition 11 *If $\mathbb{L}$ from (36) is nonsingular,*

$$\left\langle\tilde{\Phi},(\mathcal{T}+\mathcal{V})\Phi\right\rangle_{\mathcal{A}}\stackrel{\mathrm{def}}{=}\left\langle\mathcal{A}\prod_{j=1}^{N}\tilde{\phi}_{j}(\gamma_{j}),\left(\sum_{i}-\frac{1}{2}\Delta_{i}+v(\mathbf{r}_{i})\right)\prod_{j=1}^{N}\phi_{j}(\gamma_{j})\right\rangle \tag{51}$$

is equal to

$$\frac{|\mathbb{L}|}{N!}\int\left(\mathcal{T}_{*}+\mathcal{V}_{*}\right)\left[\mathbf{\Phi}\right]^{*}\mathbf{\Theta}d\gamma\,. \tag{52}$$

Proof: We follow the same procedure as we used for the electron-electron operator $\mathcal{W}$ in Section III.6. Instead of (43) we have the simpler expression

$$\frac{|\mathbb{L}|}{N!}\sum_{i}\int\left|\mathbb{I}+\left(\left(\mathcal{T}_{*}+\mathcal{V}_{*}\right)\left[\overline{\phi}_{i}\right](\gamma)\mathbf{\Theta}(\gamma)-\mathbf{e}_{i}\right)\mathbf{e}_{i}^{*}\right|d\gamma\,. \tag{53}$$

Applying Proposition 2 we obtain (52). □

To analyze the computational cost to compute (52), we note that it costs $\mathcal{O}(NM)$ to apply $\left(\mathcal{T}_{*}+\mathcal{V}_{*}\right)\left[\cdot\right]$. Including the cost for the maximum coincidence transformation, our total cost is thus $\mathcal{O}(N^{2}(N+M))$.

III.7.1 The Singular Case

We now state the formula when $\mathbb{L}$ is singular. The analysis is similar to that for $\mathcal{W}$ in Section III.6.1.

Proposition 12 *If the rank-deficiency of* $\mathbb{L}$ *is greater than one, (51) evaluates to zero. If it is equal to one we have*

$$\frac{1}{|\mathbb{L}^{\ddagger}|N!}\int (\mathcal{T}_* + \mathcal{V}_*)\,[\mathbf{\Phi}^*\mathbf{v}_1]\mathbf{u}_1^*\tilde{\mathbf{\Phi}}d\gamma\,. \tag{54}$$

To compute (54), it costs $\mathcal{O}(NM)$ to form $\mathbf{\Phi}^*\mathbf{v}_1$ and $\mathbf{u}_1^*\tilde{\mathbf{\Phi}}$, and $\mathcal{O}(M)$ to apply $(\mathcal{T}_* + \mathcal{V}_*)\,[\cdot]$.

IV Incorporating Delta Functions into the Antisymmetric Inner Products

In this section we show how to compute antisymmetric inner products when one of the component functions is replaced by a delta function. For concreteness, we will replace $\tilde{\phi}_1(\gamma_1)$ by $\delta(\gamma - \gamma_1)$.

IV.1 Löwdin's Rule with $\delta(\gamma - \gamma_1)$ Present

The matrix $\mathbb{L}$ from (36) is defined by $L(i,j) = \langle\tilde{\phi}_i, \phi_j\rangle$. If we replace $\tilde{\phi}_1(\gamma_1)$ by $\delta(\gamma - \gamma_1)$, then the first row depends on γ and is given by $L(1,j) = \langle\delta(\gamma - \cdot), \phi_j\rangle = \phi_j(\gamma)$. We thus have a matrix that depends on γ,

$$\mathbb{L}(\gamma) = \begin{bmatrix} \phi_1(\gamma) & \phi_2(\gamma) & \cdots & \phi_N(\gamma) \\ \langle\tilde{\phi}_2, \phi_1\rangle & \langle\tilde{\phi}_2, \phi_2\rangle & \cdots & \langle\tilde{\phi}_2, \phi_N\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\tilde{\phi}_N, \phi_1\rangle & \langle\tilde{\phi}_N, \phi_2\rangle & \cdots & \langle\tilde{\phi}_N, \phi_N\rangle \end{bmatrix}. \tag{55}$$

To compute with $\mathbb{L}(\gamma)$ without resorting to cofactor expansions, we express $\mathbb{L}(\gamma)$ as a rank-one perturbation of a matrix of numbers. Define

$$\mathbb{E} = \begin{bmatrix} \overline{d}(1) & \overline{d}(2) & \cdots & \overline{d}(N) \\ \langle\tilde{\phi}_2, \phi_1\rangle & \langle\tilde{\phi}_2, \phi_2\rangle & \cdots & \langle\tilde{\phi}_2, \phi_N\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\tilde{\phi}_N, \phi_1\rangle & \langle\tilde{\phi}_N, \phi_2\rangle & \cdots & \langle\tilde{\phi}_N, \phi_N\rangle \end{bmatrix}, \tag{56}$$

where the vector $\mathbf{d}^*$ is chosen to be a unit vector orthogonal to the remaining rows of $\mathbb{E}$. This choice assures that the rank deficiency of $\mathbb{E}$ will be smaller than or equal to the rank deficiency of the matrix with any other first row. It also gives us some convenient properties, namely $\mathbb{E}\mathbf{d} = \mathbf{e}_1$, $\mathbf{d}^*\mathbb{E}^{\ddagger} = \mathbf{e}_1^*$, $\mathbb{E}^{\ddagger}\mathbf{e}_1 = \mathbf{d}$, and $\mathbf{e}_1^*\mathbb{E} = \mathbf{d}^*$, where $\mathbb{E}^{\ddagger}$ is the modified pseudo-inverse of Definition 4. It costs $\mathcal{O}(N^2M)$ to construct $\mathbb{E}$ and $\mathcal{O}(N^3)$ to compute $\mathbb{E}^{\ddagger}$ and $|\mathbb{E}|$.

We then have

$$\mathbb{L}(\gamma) = \mathbb{E} + \mathbf{e}_1(\mathbf{\Phi}(\gamma) - \mathbf{d})^* \tag{57}$$

and, with the help of Proposition 2, compute

$$|\mathbb{L}(\gamma)| = |\mathbb{E}||\mathbb{I} + \mathbf{d}(\mathbf{\Phi}(\gamma) - \mathbf{d})^*| = |\mathbb{E}|\,(1 + (\mathbf{\Phi}(\gamma) - \mathbf{d})^*\mathbf{d}) = |\mathbb{E}|\,\mathbf{\Phi}(\gamma)^*\mathbf{d}\,, \tag{58}$$

which yields

Proposition 13

$$\left\langle \delta(\gamma-\gamma_1)\prod_{i=2}^{N}\tilde{\phi}_i(\gamma_i), \prod_{i=1}^{N}\phi_i(\gamma_i)\right\rangle_{\mathcal{A}} = |\mathbb{E}|\,\mathbf{\Phi}(\gamma)^*\mathbf{d}\,, \tag{59}$$

where $\mathbb{E}$ *and* $\mathbf{d}$ *are defined as above.*

Remark 14 *If* $i>1$ *then*

$$\langle|\mathbb{E}|\,\mathbf{\Phi}^*\mathbf{d},\tilde{\phi}_i\rangle = |\mathbb{E}|\langle\mathbf{\Phi},\tilde{\phi}_i\rangle^*\mathbf{d} = |\mathbb{E}|E(i,\cdot)^*\mathbf{d} = 0\,, \tag{60}$$

since $\mathbf{d}$ *is orthogonal to* $E(i,\cdot)$*, which is row number* i *of* $\mathbb{E}$*. Thus the function (59) is orthogonal to* $\tilde{\phi}_i$ *for* $i>1$*. The same property will hold when the operators* $\mathcal{T}$*,* $\mathcal{V}$*, and* $\mathcal{W}$ *are present in the antisymmetric inner product, as described in the following sections.*

IV.2 Antisymmetric Inner Product with $\delta(\gamma-\gamma_1)$ and ($\mathcal{T}$ and/or $\mathcal{V}$) Present

To compute antisymmetric inner products involving operators, we will modify formulas from Section III. The first (trivial) modification is to denote the variable of integration in those formulas by γ', so as not to confuse it with the variable γ in $\delta(\gamma-\gamma_1)$. Next we replace $|\mathbb{L}|$ with $|\mathbb{L}(\gamma)|$ given by (58). Using (57), we can express

$$\begin{aligned}\mathbb{L}(\gamma)^{-1} = \left(\mathbb{E}+\mathbf{e}_1(\mathbf{\Phi}(\gamma)-\mathbf{d})^*\right)^{-1} &= \left(\mathbb{E}\left(\mathbb{I}+\mathbf{d}(\mathbf{\Phi}(\gamma)-\mathbf{d})^*\right)\right)^{-1}\\ &= \left(\mathbb{I}+\mathbf{d}(\mathbf{\Phi}(\gamma)-\mathbf{d})^*\right)^{-1}\mathbb{E}^{-1}\,. \end{aligned}\tag{61}$$

Using the Sherman-Morrisson Formula (see e.g. [21] and (B5) in Appendix B) we then have

$$\mathbb{L}(\gamma)^{-1} = \left(\mathbb{I}-\frac{\mathbf{d}(\mathbf{\Phi}(\gamma)-\mathbf{d})^*}{1+(\mathbf{\Phi}(\gamma)-\mathbf{d})^*\mathbf{d}}\right)\mathbb{E}^{-1} = \left(\mathbb{I}+\mathbf{d}\frac{(\mathbf{d}-\mathbf{\Phi}(\gamma))^*}{\mathbf{\Phi}(\gamma)^*\mathbf{d}}\right)\mathbb{E}^{-1}\,. \tag{62}$$

The vector of functions $\mathbf{\Theta}$, which was defined by $\mathbb{L}^{-1}\tilde{\mathbf{\Phi}}$, now depends on the variable γ in $\delta(\gamma-\gamma_1)$ as well as its own internal variable γ'. Replacing $\mathbb{L}^{-1}$ with (62) and $\tilde{\mathbf{\Phi}}$ with $\tilde{\mathbf{\Phi}}(\gamma')+\mathbf{e}_1(\delta(\gamma-\gamma')-\tilde{\phi}_1(\gamma'))$, we obtain

$$\mathbf{\Theta}(\gamma,\gamma') = \left(\mathbb{I}+\mathbf{d}\frac{(\mathbf{d}-\mathbf{\Phi}(\gamma))^*}{\mathbf{\Phi}(\gamma)^*\mathbf{d}}\right)\mathbb{E}^{-1}\left(\tilde{\mathbf{\Phi}}(\gamma')+\mathbf{e}_1(\delta(\gamma-\gamma')-\tilde{\phi}_1(\gamma'))\right)\,. \tag{63}$$

To compute it, we first compute the base case $\tilde{\mathbf{\Theta}}(\gamma')=\mathbb{E}^{-1}\tilde{\mathbf{\Phi}}(\gamma')$. Multiplying out (63) and noting $\mathbf{d}^*\tilde{\mathbf{\Theta}}=\mathbf{d}^*\mathbb{E}^{\ddagger}\tilde{\mathbf{\Phi}}=\tilde{\phi}_1$, we obtain

$$\begin{aligned}\mathbf{\Theta}(\gamma,\gamma') = \tilde{\mathbf{\Theta}}(\gamma')+\mathbf{d}&\frac{\mathbf{d}^*\tilde{\mathbf{\Theta}}(\gamma')-\mathbf{\Phi}(\gamma)^*\tilde{\mathbf{\Theta}}(\gamma')+\delta(\gamma-\gamma')-\tilde{\phi}_1(\gamma')}{\mathbf{\Phi}(\gamma)^*\mathbf{d}}\\ &= \tilde{\mathbf{\Theta}}(\gamma')-\mathbf{d}\frac{\mathbf{\Phi}(\gamma)^*\tilde{\mathbf{\Theta}}(\gamma')-\delta(\gamma-\gamma')}{\mathbf{\Phi}(\gamma)^*\mathbf{d}}\,.\end{aligned}\tag{64}$$

We are now ready to state our main formulas.

Proposition 15 *When* $\mathbb{E}$ *is nonsingular,*

$$\left\langle \delta(\gamma-\gamma_1)\prod_{i=2}^{N}\tilde{\phi}_i(\gamma_i), (\mathcal{T}+\mathcal{V})\prod_{i=1}^{N}\phi_i(\gamma_i)\right\rangle_{\mathcal{A}} \tag{65}$$

is equal to

$$\frac{|\mathbb{E}|}{N!}\left[\mathbf{\Phi}(\gamma)^*\left(\mathbf{d}\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}]^*\tilde{\mathbf{\Theta}}d\gamma'-\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}^*\mathbf{d}]\tilde{\mathbf{\Theta}}d\gamma'\right)\right.$$
$$\left.+\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}^*\mathbf{d}](\gamma)\right], \quad (66)$$

which can be computed with total cost $\mathcal{O}(N^3+N^2M)$.

Proof: To compute (65), we start with $\frac{|\mathbb{L}|}{N!}\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}]^*\mathbf{\Theta}d\gamma'$ from (52) and substitute in (58) and (64) to obtain

$$\frac{|\mathbb{E}|\;\mathbf{\Phi}(\gamma)^*\mathbf{d}}{N!}\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}](\gamma')^*\left(\tilde{\mathbf{\Theta}}(\gamma')-\mathbf{d}\frac{\mathbf{\Phi}(\gamma)^*\tilde{\mathbf{\Theta}}(\gamma')-\delta(\gamma-\gamma')}{\mathbf{\Phi}(\gamma)^*\mathbf{d}}\right)d\gamma'\,. \quad (67)$$

Distributing out and rearranging, we have

$$\frac{|\mathbb{E}|}{N!}\int\mathbf{\Phi}(\gamma)^*\mathbf{d}(\mathcal{T}_*+\mathcal{V}_*)\left[\mathbf{\Phi}\right]^*(\gamma')\tilde{\mathbf{\Theta}}(\gamma')-\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}](\gamma')^*\mathbf{d}\mathbf{\Phi}(\gamma)^*\tilde{\mathbf{\Theta}}(\gamma')$$
$$+\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}](\gamma')^*\mathbf{d}\delta(\gamma-\gamma')d\gamma'\,, \quad (68)$$

which yields (66). Although in (62) and (64) we divide by $\mathbf{\Phi}^*\mathbf{d}$, which could be zero, this denominator cancels in the final expression, so we can argue by continuity that the final expression is still valid. One can also prove this directly by determining the nullspace of $\mathbb{L}$ and then using (54). □

Remark 16 *It is the term with pointwise multiplication, $\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}^*\mathbf{d}]$ in (66), that allows adaptive refinement around the nuclei in the numerical algorithm.*

To obtain the formulas when $\mathbb{E}$ is singular, we follow the same logic as in Section III.6.1. Denote the singular vectors in the nullspace of $\mathbb{E}$ by $\{(\tilde{\mathbf{u}}_i,\tilde{\mathbf{v}}_i)\}$.

Proposition 17 *When $\mathbb{E}$ has rank deficiency greater than one, (65) is zero. When $\mathbb{E}$ has rank deficiency one, (65) is equal to*

$$\frac{1}{|\mathbb{E}^\ddagger|N!}\mathbf{\Phi}(\gamma)^*\left(\mathbf{d}\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}^*\tilde{\mathbf{v}}_1]\tilde{\mathbf{u}}_1^*\tilde{\mathbf{\Phi}}d\gamma'-\tilde{\mathbf{v}}_1\int\left(\mathcal{T}_*+\mathcal{V}_*\right)[\mathbf{\Phi}^*\mathbf{d}]\tilde{\mathbf{u}}_1^*\tilde{\mathbf{\Phi}}d\gamma'\right)\,, \quad (69)$$

which can be computed with total cost $\mathcal{O}(N^3+N^2M)$.

IV.3 Antisymmetric Inner Product with $\delta(\gamma-\gamma_1)$ and $\mathcal{W}$ Present

Conceptually the derivation if $\mathcal{W}$ is present in the inner product is the same and we obtain the following propositions.

Proposition 18 *When $\mathbb{E}$ is nonsingular,*

$$\left\langle\delta(\gamma-\gamma_1)\prod_{i=2}^{N}\tilde{\phi}_i(\gamma_i),\mathcal{W}\prod_{i=1}^{N}\phi_i(\gamma_i)\right\rangle_{\mathcal{A}} \quad (70)$$

is equal to

$$\begin{aligned}\frac{1}{2}\frac{|\mathbb{E}|}{N!}\Big[2\left(\boldsymbol{\Phi}(\gamma)^*\mathbf{d}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\right](\gamma) - \boldsymbol{\Phi}(\gamma)^*\mathcal{W}_{\mathcal{P}}\left[\tilde{\boldsymbol{\Theta}}\boldsymbol{\Phi}^*\mathbf{d}\right](\gamma)\right)& \\ + \boldsymbol{\Phi}(\gamma)^*\Bigg(\mathbf{d}\int \boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\right] - \boldsymbol{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\tilde{\boldsymbol{\Theta}}\boldsymbol{\Phi}^*\right]\tilde{\boldsymbol{\Theta}}d\gamma'& \\ -2\int \tilde{\boldsymbol{\Theta}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\right]\boldsymbol{\Phi}^*\mathbf{d} - \tilde{\boldsymbol{\Theta}}\boldsymbol{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\tilde{\boldsymbol{\Theta}}\boldsymbol{\Phi}^*\mathbf{d}\right]d\gamma'\Bigg)\Big]&\,, \qquad (71)\end{aligned}$$

which can be computed with total cost $\mathcal{O}(N^3 + N^2M\log M)$.

Proposition 19 *When* $\mathbb{E}$ *has rank deficiency one, (70) is equal to*

$$\begin{aligned}\frac{1}{|\mathbb{E}^{\ddagger}|N!}\Big[\left(\boldsymbol{\Phi}(\gamma)^*\mathbf{d}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\right](\gamma) - \boldsymbol{\Phi}(\gamma)^*\tilde{\mathbf{v}}_1\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\mathbf{d}\right](\gamma)\right)& \\ +\boldsymbol{\Phi}(\gamma)^*\Bigg(\mathbf{d}\int \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\left(\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\right] - \mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\right]\tilde{\boldsymbol{\Theta}}\right)d\gamma'& \\ + \int \tilde{\boldsymbol{\Theta}}\left(\boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\mathbf{d}\right] - \mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\right]\boldsymbol{\Phi}^*\mathbf{d}\right)d\gamma'& \\ - \tilde{\mathbf{v}}_1\int \boldsymbol{\Phi}^*\mathbf{d}\left(\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\boldsymbol{\Theta}}\right] - \mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\right]\tilde{\boldsymbol{\Theta}}\right)d\gamma'\Bigg)\Big]&\,, \qquad (72)\end{aligned}$$

which can be computed with total cost $\mathcal{O}(N^3 + N^2M + NM\log M)$.

Proposition 20 *When* $\mathbb{E}$ *has rank deficiency two, (70) is equal to*

$$\begin{aligned}\frac{1}{|\mathbb{E}^{\ddagger}|N!}\boldsymbol{\Phi}(\gamma)^*\Big[\mathbf{d}\int \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\tilde{\mathbf{v}}_2\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}\right] - \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_2\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\right]\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}d\gamma& \\ - \tilde{\mathbf{v}}_1\int \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_2\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\mathbf{d}\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\right] - \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_2\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\mathbf{d}\right]\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}d\gamma& \\ - \tilde{\mathbf{v}}_2\int \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\mathcal{W}_{\mathcal{P}}\left[\boldsymbol{\Phi}^*\mathbf{d}\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}\right] - \boldsymbol{\Phi}^*\tilde{\mathbf{v}}_1\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{u}}_1^*\tilde{\boldsymbol{\Phi}}\boldsymbol{\Phi}^*\mathbf{d}\right]\tilde{\mathbf{u}}_2^*\tilde{\boldsymbol{\Phi}}d\gamma\Big]&, \qquad (73)\end{aligned}$$

which can be computed with total cost $\mathcal{O}(N^3 + NM + M\log M)$.

V Details of the Green's Function Iteration

In this section we fill in the missing pieces in the Green's function iteration algorithm outlined in Section II.2. First we give a representation for the Green's function itself. Then we use the methods in the previous sections to construct the vector $\mathbf{b}$ in (18) and the matrix $\mathbb{A}$ in (17) to form the normal equations (15). Next we give the algorithm from Section II.2 in outline form as pseudocode. Finally we gather the computational cost of the whole method, and present some linear algebra techniques to reduce it.

V.1 Representing the Green's Function

In this section we construct a separated representation for the Green's function $\mathcal{G}_\mu$ in (7), following the ideas in [4, 5] (see also [22, 23]). We will use this representation in Section V.2 when constructing the right-hand-side of the normal equations.

We begin by constructing an approximation of $1/t$ with exponentials such that

$$\left| \frac{1}{t} - \sum_{p=1}^{L} w_p \exp(-\tau_p t) \right| < \epsilon \,, \tag{74}$$

on the interval $t \in [1, \infty)$, with w_p and τ_p positive. Expansions of $1/t$ into exponentials have been used in several applications and constructed by diverse techniques; see [8, 29, 59, 6, 9, 24] and the references therein. The interval $[1, \infty)$ is addressed specifically in [9], where it is shown that the error rate $\epsilon = \mathcal{O}(\exp(-c\sqrt{L}))$ can be achieved, which means we can achieve $L = \mathcal{O}((\ln \epsilon)^2)$.

Substituting $t = s/(-\mu)$ for $\mu < 0$ into (74) and dividing by $-\mu$, one has

$$\left| \frac{1}{s} - \sum_{p=1}^{L} \frac{w_p}{-\mu} \exp(-\frac{\tau_p}{-\mu} s) \right| < \frac{\epsilon}{-\mu} \,, \tag{75}$$

valid on the interval $s \in [-\mu, \infty)$. In Fourier coordinates, we can express

$$\mathcal{G}_\mu = \frac{1}{2\pi^2 \sum \xi_i^2 - \mu} \,, \tag{76}$$

from which we see that $\|\mathcal{G}_\mu\| = 1/(-\mu)$. Since the denominator is at least $-\mu > 0$, we can substitute into (75) and obtain

$$\left| \mathcal{G}_\mu - \sum_{p=1}^{L} \frac{w_p}{-\mu} e^{-\tau_p} \bigotimes_{i=1}^{N} \exp(-\frac{2\pi^2 \tau_p}{-\mu} \xi_i^2) \right| < \frac{\epsilon}{-\mu} = \epsilon \|\mathcal{G}_\mu\| \,. \tag{77}$$

Thus we obtain an approximation of $\mathcal{G}_\mu$ with relative error ϵ in norm using L terms, with L independent of N and μ. To construct $\mathcal{G}_\mu$ as an integral operator in spatial coordinates, we apply the inverse Fourier transform to obtain

$$\mathcal{G}_\mu \approx \sum_{p=1}^{L} \bigotimes_{i=1}^{N} \mathcal{F}^p_{\mathbf{r}_i} \,, \tag{78}$$

where the convolution operator $\mathcal{F}^p_{\mathbf{r}_i}$, which depends implicitly on μ, is defined by

$$\mathcal{F}^p_{\mathbf{r}_i} f(\gamma_1, \ldots, \gamma_N) = \left(\frac{w_p}{-\mu e^{\tau_p}} \right)^{1/N} \left(\frac{-\mu}{2\pi\tau_p} \right)^{3/2} \times$$
$$\int \exp\left(-\frac{-\mu}{2\tau_p} \|\mathbf{r}_i - \mathbf{r}'\|^2 \right) f(\gamma_1, \ldots, \gamma_{i-1}, (\mathbf{r}', \sigma_i), \gamma_{i+1}, \ldots, \gamma_N) d\mathbf{r}' \,. \tag{79}$$

This construction has theoretical value, since it has proved the following theorem.

Theorem 21 *For any $\epsilon > 0$, $\mu < 0$, and N, the N-particle Green's function $\mathcal{G}_\mu$ has a separated representation with relative error in operator norm bounded by ϵ using $L = \mathcal{O}((\ln \epsilon)^2)$ terms, with L independent of μ and N.*

V.2 Constructing the Right-Hand-Side Vector b in (18)

In order to do a step in the iteration, we need to construct the right-hand-side $\mathbf{b}$ in the normal equations (15) in Section II.2.2. Since $\mathcal{A}$ is an orthogonal projection, $\mathcal{A}$ and $\mathcal{G}_\mu$ commute, and $\mathcal{G}_\mu$ is self-adjoint, the entry (18) is equal to

$$b(l)(\gamma) = -\tilde{s}_l \sum_{m}^{r} s_m \left\langle \mathcal{A}\, \mathcal{G}_\mu \delta(\gamma - \gamma_1) \prod_{i=2}^{N} \tilde{\phi}_i^l(\gamma_i), [\mathcal{V} + \mathcal{W}] \prod_{i=1}^{N} \phi_i^m(\gamma_i) \right\rangle \,. \tag{80}$$

Substituting (78) in for $\mathcal{G}_\mu$ and rearranging, we have

$$b(l)(\gamma) = -\tilde{s}_l \sum_{m}^{r} s_m \sum_{p=1}^{L} \left\langle \mathcal{A}\mathcal{F}^p_{\mathbf{r}_1}\delta(\gamma-\gamma_1)\prod_{i=2}^{N}\mathcal{F}^p_{\mathbf{r}_i}\tilde{\phi}^l_i(\gamma_i), [\mathcal{V}+\mathcal{W}]\prod_{i=1}^{N}\phi^m_i(\gamma_i)\right\rangle . \tag{81}$$

The computation is of the same form for each value of the indices l, m, and p, so we can consider a single term and suppress the indices.

To evaluate a single term $\langle \mathcal{A}\mathcal{F}_{\mathbf{r}_1}\delta(\gamma-\gamma_1)\prod_{i=2}\mathcal{F}_{\mathbf{r}_i}\tilde{\phi}_i(\gamma_i), [\mathcal{V}+\mathcal{W}]\prod_{i=1}\phi_i(\gamma_i)\rangle$ we use the formulas in Propositions 15–20 in Sections IV.2 and IV.3, with two modifications. The first modification is that $\tilde{\mathbf{\Phi}}$ is replaced with $\mathcal{F}\tilde{\mathbf{\Phi}}$ throughout. This replacement causes no structural change to the formulas; it just changes the inputs. The second modification is caused by the replacement of $\delta(\gamma-\gamma_1)$ by $\mathcal{F}_{\mathbf{r}_1}\delta(\gamma-\gamma_1)$. The first row of $\mathbb{L}(\gamma)$ in (55) becomes $\mathcal{F}\mathbf{\Phi}(\gamma)^*$, which makes $|\mathbb{L}(\gamma)| = |\mathbb{E}|\,\mathcal{F}\mathbf{\Phi}(\gamma)^*\mathbf{d}$. Similarly, (64) becomes

$$\mathbf{\Theta}(\gamma,\gamma') = \tilde{\mathbf{\Theta}}(\gamma') - \mathbf{d}\frac{\mathcal{F}\mathbf{\Phi}(\gamma)^*\tilde{\mathbf{\Theta}}(\gamma') - \mathcal{F}\delta(\gamma-\gamma')}{\mathcal{F}\mathbf{\Phi}(\gamma)^*\mathbf{d}} . \tag{82}$$

Tracking $\mathcal{F}$ through the formulas, we find that all we need to do is to modify the formulas in Sections IV.2 and IV.3 by applying $\mathcal{F}$ to the final result.

V.3 Constructing the Matrix $\mathbb{A}$ in (17)

In this section we construct the kernels in (17) for the normal equations (15), using the same ideas as in Section IV. We fix l and l' and define

$$K(\gamma,\gamma') = \frac{A(l,l')(\gamma,\gamma')}{\tilde{s}_l\tilde{s}_{l'}} \tag{83}$$

$$\mathbf{w}(\gamma') = \left[\begin{array}{ccc} \tilde{\phi}^l_2(\gamma') & \dots & \tilde{\phi}^l_N(\gamma') \end{array}\right]^* \tag{84}$$

$$\mathbf{y}(\gamma) = \left[\begin{array}{ccc} \tilde{\phi}^{l'}_2(\gamma) & \dots & \tilde{\phi}^{l'}_N(\gamma) \end{array}\right]^* \tag{85}$$

$$\mathbb{D} = \left[\begin{array}{ccc} \langle\tilde{\phi}^l_2,\tilde{\phi}^{l'}_2\rangle & \cdots & \langle\tilde{\phi}^l_2,\tilde{\phi}^{l'}_N\rangle \\ \vdots & \ddots & \vdots \\ \langle\tilde{\phi}^l_N,\tilde{\phi}^{l'}_2\rangle & \cdots & \langle\tilde{\phi}^l_N,\tilde{\phi}^{l'}_N\rangle \end{array}\right] . \tag{86}$$

Using Löwdin's rules (37) we have

$$K(\gamma,\gamma') = \frac{|\mathbb{L}|}{N!} = \frac{1}{N!}\left|\begin{array}{cc} \delta(\gamma-\gamma') & \mathbf{y}^*(\gamma) \\ \mathbf{w}(\gamma') & \mathbb{D} \end{array}\right| . \tag{87}$$

Expressing $\mathbb{L}$ as a low-rank perturbation of $\left[\begin{array}{cc} 1 & 0 \\ 0 & \mathbb{D}\end{array}\right]$, we have

$$\begin{aligned} K(\gamma,\gamma') &= \frac{1}{N!}\left|\left[\begin{array}{cc} 1 & 0 \\ 0 & \mathbb{D}\end{array}\right] + \left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{cc} 0 & \mathbf{y}^*(\gamma)\end{array}\right] + \left[\begin{array}{c} \delta(\gamma-\gamma')-1 \\ \mathbf{w}(\gamma')\end{array}\right]\left[\begin{array}{cc} 1 & 0 \end{array}\right]\right| \\ &= \frac{1}{N!}\left|\begin{array}{cc} 1 & 0 \\ 0 & \mathbb{D}\end{array}\right|\left|\mathbb{I} + \left[\begin{array}{c} 1 \\ 0 \end{array}\right]\left[\begin{array}{cc} 0 & \mathbf{y}^*(\gamma)\end{array}\right] + \left[\begin{array}{c} \delta(\gamma-\gamma')-1 \\ \mathbb{D}^{-1}\mathbf{w}(\gamma')\end{array}\right]\left[\begin{array}{cc} 1 & 0 \end{array}\right]\right| \\ &= \frac{|\mathbb{D}|}{N!}\left|\begin{array}{cc} 1 & \mathbf{y}^*(\gamma)\mathbb{D}^{-1}\mathbf{w}(\gamma') \\ 1 & \delta(\gamma-\gamma')\end{array}\right| = \frac{|\mathbb{D}|}{N!}\left(\delta(\gamma-\gamma') - \mathbf{y}^*(\gamma)\mathbb{D}^{-1}\mathbf{w}(\gamma')\right) . \end{aligned} \tag{88}$$

If $\mathbb{D}$ is singular then we apply the same logic as in Section III.6.1. If $\mathbb{D}$ has rank-deficiency greater than one then $K(\gamma, \gamma') = 0$. If it has rank-deficiency one then we have $K(\gamma, \gamma') =$

$$\begin{aligned}
&\frac{1}{|\mathbb{D}^{\ddagger}|N!}\left|\mathbb{I} + \begin{bmatrix} 0 \\ -\mathbf{v} \end{bmatrix}\begin{bmatrix} 0 & \mathbf{v}^* \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 0 & \mathbf{y}^*(\gamma) \end{bmatrix} + \begin{bmatrix} \delta(\gamma-\gamma')-1 \\ \mathbb{D}^{\ddagger}\mathbf{w}(\gamma') \end{bmatrix}\begin{bmatrix} 1 & 0 \end{bmatrix}\right| \\
&\qquad = \frac{1}{|\mathbb{D}^{\ddagger}|N!}\begin{vmatrix} 0 & 0 & \mathbf{v}^*\mathbb{D}^{\ddagger}\mathbf{w}(\gamma') \\ -\mathbf{y}^*(\gamma)\mathbf{v} & 1 & \mathbf{y}^*(\gamma)\mathbb{D}^{\ddagger}\mathbf{w}(\gamma') \\ 0 & 1 & \delta(\gamma-\gamma') \end{vmatrix} = \frac{-(\mathbf{y}^*(\gamma)\mathbf{v})(\mathbf{v}^*\mathbb{D}^{\ddagger}\mathbf{w}(\gamma'))}{|\mathbb{D}^{\ddagger}|N!} \\
&\qquad = \frac{-(\mathbf{y}^*(\gamma)\mathbf{v})(\mathbf{u}^*\mathbf{w}(\gamma'))}{|\mathbb{D}^{\ddagger}|N!}, \qquad (89)
\end{aligned}$$

where $\mathbb{D}^{\ddagger}$ is the modified pseudo-inverse of Definition 4.

In the nonsingular case, we can construct $\mathbb{D}$ at cost $\mathcal{O}(N^2M)$ and compute $\mathbb{D}^{-1}$ at cost $\mathcal{O}(N^3)$. Applying this kernel costs $\mathcal{O}(NM)$ to integrate against a function in γ', $\mathcal{O}(N^2)$ to apply $\mathbb{D}^{-1}$, and then $\mathcal{O}(NM)$ to apply $\mathbf{y}^*$ to the result. In the singular case, we can compute $\mathbb{D}^{\ddagger}$ at cost $\mathcal{O}(N^3)$ and construct $\mathbf{y}^*\mathbf{v}$ and $\mathbf{u}^*\mathbf{w}$ at cost $\mathcal{O}(NM)$. Since the variables separate, applying this kernel costs $\mathcal{O}(M)$.

Remark 22 *In the case $r = 1$, which corresponds to the Hartree-Fock formulation, $\mathbb{D} = \mathbb{I}$ and $K(\gamma, \gamma')$ is just the projector orthogonal to $\{\tilde{\phi}_i\}_{i=2}^N$.*

V.4 Pseudocode

In this section we give the algorithm in outline form as pseudocode. We do not indicate when objects can be recalled or updated from previous computations.

```
Loop through $I$ Green's function iterations (9,10,13). For each of these:
  Construct $\mathcal{G}_\mu$ as in Section V.1, obtaining the operators $\mathcal{F}^p$ in (79).
  Loop through the $N$ directions (electrons). For each of these:
    Compute $A(l, l')$ via (88) for all $(l, l')$.
    Compute $b(l)(\gamma)$ in (81) by:
      Loop in the $r$ values of $l$ and for each:
        Sum over the $L$ values of $p$ and for each:
          Compute $\mathcal{F}^p\phi_i^l$ for all $i$.
          Sum over the $r$ values of $m$ and for each:
            Using $\mathcal{F}^p\tilde{\mathbf{\Phi}}$ in place of $\tilde{\mathbf{\Phi}}$, construct $\mathbb{E}$ in (56).
            Compute $|\mathbb{E}|$ and $\mathbb{E}^{-1}$.
            Construct $\tilde{\mathbf{\Theta}} = \mathbb{E}^{-1}\mathcal{F}\tilde{\mathbf{\Phi}}$.
            Construct $\mathbf{\Phi}^*\tilde{\mathbf{\Theta}}$, $\mathbf{\Phi}^*\mathbf{d}$, and $\tilde{\mathbf{\Theta}}\mathbf{\Phi}^*$.
            Compute $\mathcal{W}_{\mathcal{P}}\left[\mathbf{\Phi}^*\tilde{\mathbf{\Theta}}\right]$ and $\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{\Theta}}\mathbf{\Phi}^*\right]$.
            Compute (66) and (71) using these ingredients.
            Apply $\mathcal{F}^p$ to ((66) + (71)).
    Apply conjugate gradient to solve the normal equations (15).
  Renormalize as in (10).
  Update $\mu$ via (13).
```

Remark 23 *We have presented the algorithm in serial form for clarity. The loop in l, sum in p, and sum in m can be trivially parallelized. Parallelizing the loop through the N electrons would represent a change in the algorithm, which we will develop elsewhere.*

V.5 Overall Computational Cost

The computational cost is dominated by the repeated construction and solution of the normal equations (15). For a fixed direction, the construction cost is dominated by (81), which has r^2L inner products. The most costly portion of the inner products is (71), which requires $\mathcal{O}(N^3 + N^2M\log M)$ operations, giving us the net construction cost

$$\mathcal{O}(r^2LN^2(N + M\log M))\,. \tag{90}$$

The operation count to solve the normal equations (15) by applying the matrix of integral operators $\mathbb{A}$ S times is

$$\mathcal{O}(r^2SN(N + M))\,. \tag{91}$$

As we loop through the directions, we may reuse several quantities, so the total cost of the construction is less than N times the cost for one direction. In fact, the construction cost for the entire loop through N directions is of the same order as the cost for one direction. The application cost is simply multiplied by N. In the sections below we show how to update the construction for direction $k = 2$ using what we already have for direction $k = 1$, and then determine the cost for one loop through the directions. We defer the development of the technical linear algebra rules on low-rank updates to Appendix B, and here only show how to apply them to our problem. Our final conclusion is the computational cost

$$\mathcal{O}(Ir^2N^2[L(N + M\log M) + S(N + M)]), \tag{92}$$

where I the number of Green's function iterations.

V.5.1 Reuse in Computing $\mathbb{A}$

Let $\mathbb{D}_1$ denote $\mathbb{D}$ in (86) for directions one, and $\mathbb{D}_2$ the version for direction two. We let $\hat{\phi}_1^l$ denote the updated version of $\tilde{\phi}_1^l$. To construct $\mathbb{D}_2$ requires only the first column and row of $\mathbb{D}_1$ to be updated, specifically

$$\mathbb{D}_2 = \mathbb{D}_1 + \mathbf{e}_1 \left[\; 0 \quad (\langle \hat{\phi}_1^l, \tilde{\phi}_3^{l'} \rangle - \langle \tilde{\phi}_2^l, \tilde{\phi}_3^{l'} \rangle) \quad \ldots \;\right] + \left[\begin{array}{c} \langle \hat{\phi}_1^l, \hat{\phi}_1^{l'} \rangle - \langle \tilde{\phi}_2^l, \tilde{\phi}_2^{l'} \rangle \\ \vdots \end{array} \right] \mathbf{e}_1^*\,. \tag{93}$$

Computing those inner products involving $\hat{\phi}_1^l$ and $\hat{\phi}_1^{l'}$ costs $\mathcal{O}(NM)$. Using Proposition 24 twice, we compute $\mathbb{D}_2^{\ddagger}$, $|\mathbb{D}_2^{\ddagger}|$, and if appropriate $\mathbf{v}$, all at cost $\mathcal{O}(N^2)$. The formulas (87) and following are modified by inserting the extra column and row in the second place instead of the first, but otherwise the procedure is unchanged. The cost for one loop through the N directions is thus $\mathcal{O}(N^3 + N^2M)$.

V.5.2 Reuse in Computing Antisymmetric Inner Products with $\delta(\gamma - \gamma_1)$ and Operators

We again let $\hat{\phi}_1^l$ denote the updated version of $\tilde{\phi}_1^l$ computed during the $k = 1$ solve. The inner products needed to construct $\mathbb{E}_2$ require only the one row involving $\hat{\phi}_1$ to be updated, at cost $\mathcal{O}(NM)$. The vector $\mathbf{d}_1$ can be constructed by doing the SVD of $\mathbb{E}_1$ with the first row set to zero and then selecting one of the right singular vectors $\mathbf{v}_i$ with zero singular value. Using Proposition 24 we obtain the SVD of $\mathbb{E}_2$ with first row set to zero and second row containing the new inner products, and thus can find $\mathbf{d}_2$. Putting the first and second rows back in proper position, we then have

$$\begin{aligned} \mathbb{E}_2 = \mathbb{E}_1 + \mathbf{e}_1 \big(\big[\; \langle \mathcal{F}\hat{\phi}_1, \phi_1 \rangle \quad \cdots \quad \langle \mathcal{F}\hat{\phi}_1, \phi_N \rangle \;\big] - \mathbf{d}_1^*\big) \\ + \mathbf{e}_2 \big(\mathbf{d}_2^* - \big[\; \langle \mathcal{F}\tilde{\phi}_2, \phi_1 \rangle \quad \cdots \quad \langle \mathcal{F}\tilde{\phi}_2, \phi_N \rangle \;\big]\big)\,, \end{aligned} \tag{94}$$

and we can compute $|\mathbb{E}_2^{\ddagger}|$ and $\mathbb{E}_2^{\ddagger}$ using Proposition 24 twice, at cost $\mathcal{O}(N^2)$.

Proposition 24 produces a rank two update and we must apply it twice. For notational ease we will show how to use a rank one update applied once; the method easily extends. Assuming $\mathbb{E}_2^{\ddagger} = \mathbb{E}_1^{\ddagger} + \mathbf{f}\mathbf{g}^*$, we next update

$$\begin{aligned} \tilde{\mathbf{\Theta}}_2 = \mathbb{E}_2^{\ddagger}\mathcal{F}\tilde{\mathbf{\Phi}}_2 = (\mathbb{E}_1^{\ddagger} + \mathbf{f}\mathbf{g}^*)(\mathcal{F}\tilde{\mathbf{\Phi}}_1 + \mathbf{e}_1(\hat{\phi}_1 - \tilde{\phi}_1)) \\ = \tilde{\mathbf{\Theta}}_1 + \mathbf{d}_1(\hat{\phi}_1 - \tilde{\phi}_1) + \mathbf{f}\mathbf{g}^*\mathcal{F}\tilde{\mathbf{\Phi}}_1 + \mathbf{f}\mathbf{g}^*\mathbf{e}_1(\hat{\phi}_1 - \tilde{\phi}_1) \end{aligned} \quad (95)$$

at cost $\mathcal{O}(NM)$. It is insufficient to just update $\tilde{\mathbf{\Theta}}_2$ in this way, since it would still cost $\mathcal{O}(N^2 M \log M)$ to compute $\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{\Theta}}_2\mathbf{\Phi}^*\right]$ in (71). Instead we update the combined quantity

$$\begin{aligned} \mathbf{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{\Theta}}_2\mathbf{\Phi}^*\right] = \mathbf{\Phi}^*\mathcal{W}_{\mathcal{P}}\left[\tilde{\mathbf{\Theta}}_1\mathbf{\Phi}^*\right] + \mathbf{\Phi}^*\mathbf{d}_1\mathcal{W}_{\mathcal{P}}\left[(\hat{\phi}_1 - \tilde{\phi}_1)\mathbf{\Phi}^*\right] \\ + \mathbf{\Phi}^*\mathbf{f}\mathcal{W}_{\mathcal{P}}\left[\mathbf{g}^*\mathcal{F}\tilde{\mathbf{\Phi}}_1\mathbf{\Phi}^*\right] + \mathbf{\Phi}^*\mathbf{f}\mathbf{g}^*\mathbf{e}_1\mathcal{W}_{\mathcal{P}}\left[(\hat{\phi}_1 - \tilde{\phi}_1)\mathbf{\Phi}\right] \end{aligned} \quad (96)$$

at cost $\mathcal{O}(NM \log M)$. With this quantity and $\tilde{\mathbf{\Theta}}_2$ we can compute (71) at cost $\mathcal{O}(NM \log M)$. The singular cases work similarly. The cost for one loop through the N directions is thus $\mathcal{O}(N^2 M \log M)$.

Acknowledgments

We would like to thank Dr. Robert Harrison (U. of Tennessee and Oak Ridge National Lab) and Dr. Lucas Monzón (U. of Colorado) for many useful conversations. Portions of this research were conducted while G.B. and M.J.M. were in residence at the Institute for Pure and Applied Mathematics during the fall of 2004.

This material is based upon work supported by the National Science Foundation under Grants DMS-0219326 (G.B and M.J.M.), DMS-0612358 (G.B. and F.P.), and DMS-0545895 (M.J.M.), the DARPA/ARO under Grants W911NF-04-1-0281 and W911NF-06-1-0254 (G.B and M.J.M.), and the Department of Energy under Grants DE-FG02-03ER25583 and DOE/ORNL Grant 4000038129 (G.B. and F.P.).

A Appendix: Algorithms Based on Gradient Descent

We prefer the integral iteration in Section II.2.1 due to the generally superior numerical properties of integral formulations. One could, however, try to minimize (4) directly with a method based on gradients. Since the machinery that we have constructed applies to these methods as well, we sketch how it can be used.

To minimize (4) we could use a gradient descent, starting at some initial guess for ψ. Inserting our current approximation ψ and formally taking the gradient, we have

$$2\frac{\langle\mathcal{H}\psi, \nabla\psi\rangle_{\mathcal{A}}\langle\psi,\psi\rangle_{\mathcal{A}} - \langle\mathcal{H}\psi,\psi\rangle_{\mathcal{A}}\langle\psi,\nabla\psi\rangle_{\mathcal{A}}}{\langle\psi,\psi\rangle_{\mathcal{A}}^2}\,. \quad \text{(A1)}$$

Defining μ to be our current value of (4), the gradient reduces to

$$\frac{2}{\langle\psi,\psi\rangle_{\mathcal{A}}}\left(\langle\mathcal{H}\psi,\nabla\psi\rangle_{\mathcal{A}} - \mu\langle\psi,\nabla\psi\rangle_{\mathcal{A}}\right)\,. \quad \text{(A2)}$$

The gradient is with respect to the parameters that are used to minimize (4). In our case that is the values of the functions ϕ_j^l. Taking the gradient with respect to the point values of ϕ_j^l results in a vector $\mathbf{g}$ of functions, defined by

$$g_j^l(\gamma) = \frac{2}{\langle \psi, \psi \rangle_{\mathcal{A}}} s_l \sum_{m=1}^{r} s_m \left\langle \delta(\gamma - \gamma_j) \prod_{i \neq j}^{N} \phi_i^l(\gamma_i), (\mathcal{H} - \mu\mathcal{I}) \prod_{i=1}^{N} \phi_i^m(\gamma_i) \right\rangle_{\mathcal{A}}, \tag{A3}$$

where $\delta(\gamma - \gamma_j)$ is the delta function. The methods in Section IV can be used to construct $\mathbf{g}$.

Moving t in the direction opposite the gradient replaces ψ with

$$\sum_{l=1}^{r} s_l \prod_{i=1}^{N} (\phi_i^l - t g_i^l). \tag{A4}$$

Some search procedure can then be used to find an appropriate t. Then ψ is updated and the procedure repeated.

Alternatively, we can use an alternating direction approach and take the gradient with respect to the functions ϕ_i^l for one direction i, while fixing the functions in the other directions, and then loop through the directions. This loop through the directions is then repeated I times until we obtain the desired accuracy. We describe the $i = 1$ case. Moving t in the direction opposite the gradient replaces ψ with

$$\sum_{l=1}^{r} s_l (\phi_1^l - t g_1^l) \prod_{i=2}^{N} \phi_i^l = \psi - t \sum_{l=1}^{r} s_l g_1^l \prod_{i=2}^{N} \phi_i^l = \psi - t\tilde{\psi}. \tag{A5}$$

Inserting (A5) into (4) results in

$$\frac{\left\langle \mathcal{H}(\psi - t\tilde{\psi}), \psi - t\tilde{\psi} \right\rangle_{\mathcal{A}}}{\left\langle \psi - t\tilde{\psi}, \psi - t\tilde{\psi} \right\rangle_{\mathcal{A}}} = \frac{\langle \mathcal{H}\psi, \psi \rangle_{\mathcal{A}} - 2t \left\langle \mathcal{H}\psi, \tilde{\psi} \right\rangle_{\mathcal{A}} + t^2 \left\langle \mathcal{H}\tilde{\psi}, \tilde{\psi} \right\rangle_{\mathcal{A}}}{\langle \psi, \psi \rangle_{\mathcal{A}} - 2t \left\langle \psi, \tilde{\psi} \right\rangle_{\mathcal{A}} + t^2 \left\langle \tilde{\psi}, \tilde{\psi} \right\rangle_{\mathcal{A}}}. \tag{A6}$$

Once the inner products have been computed, we can find the minimizer for (A6) by solving a quadratic equation, and then update ψ via (A5). The cost to construct $\mathbf{g}$ for one direction is r^2 times the cost for one inner product. The dominant cost for the inner product comes from (71), which costs $\mathcal{O}(N^3 + N^2 M \log M)$, giving us the net construction cost

$$\mathcal{O}(r^2 N^2 (N + M \log M)). \tag{A7}$$

As described in Section V.5.2, many of the computations can be reused, so the cost for a single loop through the N directions is of the same order. Thus, for I loops through the directions the overall computational cost is

$$\mathcal{O}(I r^2 N^2 (N + M \log M)). \tag{A8}$$

B Appendix: Low-rank Updates

In this section we develop formulas for low-rank updates to $\mathbb{A}^{\dagger}$, $\mathbb{A}^{\perp}$ and $|\mathbb{A}^{\ddagger}|$, based on [42, 3].

Proposition 24 *Given* $\mathbb{A}$, $\mathbb{A}^{\dagger}$, $\mathbb{A}^{\perp}$, $|\mathbb{A}^{\ddagger}|$, $\mathbf{b}$, *and* $\mathbf{c}$, *let* $\mathbb{A}_1 = \mathbb{A} + \mathbf{b}\mathbf{c}^*$ *and compute*

$$\begin{array}{cccc} \mathbf{d} = \mathbb{A}^{\dagger}\mathbf{b}, & \mathbf{e} = (\mathbb{A}^{\dagger})^*\mathbf{c}, & \mathbf{f} = (\mathbb{I} - \mathbb{A}\mathbb{A}^{\dagger})\mathbf{b}, & \mathbf{g} = (\mathbb{I} - \mathbb{A}^{\dagger}\mathbb{A})\mathbf{c}, \\ d = \mathbf{d}^*\mathbf{d}, & e = \mathbf{e}^*\mathbf{e}, & f = \mathbf{f}^*\mathbf{f}, & g = \mathbf{g}^*\mathbf{g}, \\ \lambda = 1 + \mathbf{c}^*\mathbb{A}^{\dagger}\mathbf{b}, & \mu = |\lambda|^2 + dg, & \nu = |\lambda|^2 + ef, & \\ \mathbf{p} = \bar{\lambda}\mathbf{d} + d\mathbf{g}, & \mathbf{q} = \lambda\mathbf{e} + e\mathbf{f}. & & \end{array} \tag{B1}$$

1. *If $\lambda = 0$, $f = 0$, and $g = 0$, then $rank(\mathbb{A}_1) = rank(\mathbb{A}) - 1$ and*

$$\mathbb{A}_1^\dagger = \mathbb{A}^\dagger - d^{-1}\mathbf{d}\mathbf{d}^*\mathbb{A}^\dagger + e^{-1}(-\mathbb{A}^\dagger\mathbf{e} + d^{-1}(\mathbf{d}^*\mathbb{A}^\dagger\mathbf{e})\mathbf{d})\mathbf{e}^* \tag{B2}$$

$$\mathbb{A}_1^\perp = \mathbb{A}^\perp + (1/\sqrt{de})\mathbf{d}\mathbf{e}^* \tag{B3}$$

$$|\mathbb{A}_1^\ddagger| = -(1/\sqrt{de})|\mathbb{A}^\ddagger| \,. \tag{B4}$$

2. *If $\lambda \neq 0$, $f = 0$, and $g = 0$, then $rank(\mathbb{A}_1) = rank(\mathbb{A})$ and*

$$\mathbb{A}_1^\dagger = \mathbb{A}^\dagger - \lambda^{-1}\mathbf{d}\mathbf{e}^* \tag{B5}$$

$$\mathbb{A}_1^\perp = \mathbb{A}^\perp \tag{B6}$$

$$|\mathbb{A}_1^\ddagger| = |\mathbb{A}^\ddagger|\lambda^{-1} \,. \tag{B7}$$

3. *If $f = 0$ and $g \neq 0$, then $rank(\mathbb{A}_1) = rank(\mathbb{A})$ and*

$$\mathbb{A}_1^\dagger = \mathbb{A}^\dagger - \mu^{-1}\mathbf{d}(g\mathbf{d}^*\mathbb{A}^\dagger + \bar{\lambda}\mathbf{e}^*) + \mu^{-1}\mathbf{g}(-d\mathbf{e}^* + \lambda\mathbf{d}^*\mathbb{A}^\dagger) \tag{B8}$$

$$\mathbb{A}_1^\perp = \mathbb{A}^\perp - \frac{|\lambda|(\sqrt{\mu} - |\lambda|)\mathbf{g} + \lambda g\mathbf{d}}{g|\lambda|\sqrt{\mu}}\mathbf{g}^*\mathbb{A}^\perp \tag{B9}$$

$$|\mathbb{A}_1^\ddagger| = |\mathbb{A}^\ddagger|\frac{(\bar{\lambda} - \lambda)|\lambda|^2 + \lambda\mu}{\mu|\lambda|\sqrt{\mu}} \,. \tag{B10}$$

4. *If $f \neq 0$ and $g = 0$, then $rank(\mathbb{A}_1) = rank(\mathbb{A})$ and*

$$\mathbb{A}_1^\dagger = \mathbb{A}^\dagger - \nu^{-1}(f\mathbb{A}^\dagger\mathbf{e} + \bar{\lambda}\mathbf{d})\mathbf{e}^* + \nu^{-1}(-e\mathbf{d} + \lambda\mathbb{A}^\dagger\mathbf{e})\mathbf{f}^* \tag{B11}$$

$$\mathbb{A}_1^\perp = \mathbb{A}^\perp - \mathbb{A}^\perp\mathbf{f}\frac{(|\lambda|(\sqrt{\nu} - |\lambda|)\mathbf{f} + \bar{\lambda}f\mathbf{e})^*}{f|\lambda|\sqrt{\nu}} \tag{B12}$$

$$|\mathbb{A}_1^\ddagger| = |\mathbb{A}^\ddagger|\frac{(\lambda - \bar{\lambda})|\lambda|^2 + \bar{\lambda}\nu}{\nu|\lambda|\sqrt{\nu}} \,. \tag{B13}$$

5. *If $f \neq 0$ and $g \neq 0$, then $rank(\mathbb{A}_1) = rank(\mathbb{A}) + 1$ and*

$$\mathbb{A}_1^\dagger = \mathbb{A}^\dagger - f^{-1}\mathbf{d}\mathbf{f}^* + g^{-1}\mathbf{g}(-\mathbf{e}^* + \lambda f^{-1}\mathbf{f}^*) \tag{B14}$$

$$\mathbb{A}_1^\perp = \mathbb{A}^\perp - (1/\sqrt{gf})\mathbf{g}\mathbf{f}^* \tag{B15}$$

$$|\mathbb{A}_1^\ddagger| = |\mathbb{A}^\ddagger|\left[1 + (g^{-1}f^{-1} - (1/\sqrt{gf}))\mathbf{g}^*\mathbb{A}^\perp\mathbf{f}\right] \,. \tag{B16}$$

The cost to compute $\mathbb{A}_1^\dagger$, $\mathbb{A}_1^\perp$, and $|\mathbb{A}_1^\ddagger|$ is $\mathcal{O}(N^2)$.

Proof: The overall method, update rules for $rank(\mathbb{A}_1)$, and update rules for $\mathbb{A}_1^\dagger$ are taken from [3], who also list the useful properties

$$\begin{gathered}\mathbf{c}^*\mathbf{d} = \mathbf{e}^*\mathbf{b} = \lambda - 1, \quad \mathbf{b}^*\mathbf{f} = f, \quad \mathbf{c}^*\mathbf{g} = g, \quad \mathbf{d}^*\mathbf{g} = 0, \quad \mathbf{e}^*\mathbf{f} = 0, \\ \mathbb{A}^\dagger\mathbb{A}\mathbf{d} = \mathbf{d}, \quad \mathbb{A}\mathbb{A}^\dagger\mathbf{e} = \mathbf{e}, \quad \mathbb{A}^*\mathbf{f} = \mathbb{A}^\dagger\mathbf{f} = 0, \quad \mathbb{A}\mathbf{g} = (\mathbb{A}^\dagger)^*\mathbf{g} = 0.\end{gathered} \tag{B17}$$

They give update rules for the row and column spans of $\mathbb{A}_1$, which we translate into update rules for $\mathbb{A}^\perp$. The cases (B3), (B6), and (B15) follow directly. Corresponding to (B9), their update rule is that the row span of $\mathbb{A}^\perp$ should be extended (orthogonally) by $\mathbf{d}$ and then reduced by projecting orthogonal to $\mathbf{p}$. We translate this into a (Householder) reflection of the vector $\mathbf{g}$ into a vector in the span of $\mathbf{d}$ and $\mathbf{g}$ perpendicular to $\mathbf{p}$. Adjusting these vectors to have equal norm and real inner product yields the reflection of the vector $\bar{\lambda}\sqrt{\mu}\,\mathbf{g}$ to $-|\lambda|(g\mathbf{d}-\bar{\lambda}\mathbf{g})$, resulting in

$$\left(\mathbb{I}-\frac{2(\bar{\lambda}\sqrt{\mu}\mathbf{g}+|\lambda|(g\mathbf{d}-\bar{\lambda}\mathbf{g}))(\bar{\lambda}\sqrt{\mu}\mathbf{g}+|\lambda|(g\mathbf{d}-\bar{\lambda}\mathbf{g}))^*}{\|(\bar{\lambda}\sqrt{\mu}\mathbf{g}+|\lambda|(g\mathbf{d}-\bar{\lambda}\mathbf{g}))\|^2}\right)\mathbb{A}^{\perp}\,, \tag{B18}$$

which simplifies to (B9). To obtain (B12) we use the same process, extending the column span by $\mathbf{e}$ and then projecting orthogonal to $\mathbf{q}$ by a reflection of $\lambda\sqrt{\nu}\,\mathbf{f}$ to $-|\lambda|(f\mathbf{e}-\lambda\mathbf{f})$.

To derive the update rules for $|\mathbb{A}_1^{\ddagger}|$, first add the update rules for $\mathbb{A}_1^{\dagger}$ and $\mathbb{A}_1^{\perp}$ and then take the determinant. On the right hand side factor out a copy of $\mathbb{A}^{\ddagger}$ leaving a low-rank perturbation of the identity, to which we can apply Proposition 2. To simplify the results, we use (B1), (B17), and the further observations

$$\begin{array}{c}(\mathbb{A}^{\ddagger})^{-1}\mathbf{d}=\mathbf{b}-\mathbf{f}, \qquad (\mathbb{A}^{\ddagger})^{-1}\mathbb{A}^{\dagger}\mathbf{e}=\mathbf{c}-\mathbf{g}, \qquad (\mathbb{A}^{\ddagger})^{-1}\mathbf{g}=(\mathbb{A}^{\perp})^*\mathbf{c},\\ \mathbf{e}^*(\mathbb{A}^{\ddagger})^{-1}=\mathbf{c}^*-\mathbf{g}^*, \quad \mathbf{f}^*(\mathbb{A}^{\ddagger})^{-1}=\mathbf{b}^*(\mathbb{A}^{\perp})^*.\end{array} \tag{B19}$$

To obtain (B4) we compute

$$\begin{aligned}|\mathbb{A}_1^{\ddagger}| &= |\mathbb{A}^{\ddagger}|\left|\mathbb{I}-d^{-1}\mathbf{b}\mathbf{d}^*\mathbb{A}^{\dagger}+((1/\sqrt{de})\mathbf{b}-e^{-1}\mathbf{e}+d^{-1}e^{-1}(\mathbf{d}^*\mathbb{A}^{\dagger}\mathbf{e})\mathbf{b})\mathbf{e}^*\right|\\ &= |\mathbb{A}^{\ddagger}|\begin{vmatrix}1-d^{-1}\mathbf{d}^*\mathbb{A}^{\dagger}\mathbf{b} & \mathbf{d}^*\mathbb{A}^{\dagger}((1/\sqrt{de})\mathbf{b}-e^{-1}\mathbf{e}+d^{-1}e^{-1}(\mathbf{d}^*\mathbb{A}^{\dagger}\mathbf{e})\mathbf{b})\\ -d^{-1}\mathbf{e}^*\mathbf{b} & 1+\mathbf{e}^*((1/\sqrt{de})\mathbf{b}-e^{-1}\mathbf{e}+d^{-1}e^{-1}(\mathbf{d}^*\mathbb{A}^{\dagger}\mathbf{e})\mathbf{b})\end{vmatrix}\\ &= |\mathbb{A}^{\ddagger}|\begin{vmatrix}0 & \mathbf{d}^*\mathbb{A}^{\dagger}(1/\sqrt{de})\mathbf{b}\\ d^{-1} & \mathbf{e}^*((1/\sqrt{de})\mathbf{b}+d^{-1}e^{-1}(\mathbf{d}^*\mathbb{A}^{\dagger}\mathbf{e})\mathbf{b})\end{vmatrix} = |\mathbb{A}^{\ddagger}|(-(1/\sqrt{de}))\,.\end{aligned} \tag{B20}$$

For (B7) we have $|\mathbb{A}_1^{\ddagger}|=|\mathbb{A}^{\ddagger}|\left|\mathbb{I}-\lambda^{-1}\mathbf{b}\mathbf{e}^*\right|=|\mathbb{A}^{\ddagger}|(1-\lambda^{-1}\mathbf{e}^*\mathbf{b})=|\mathbb{A}^{\ddagger}|\lambda^{-1}$. To obtain (B10) we compute

$$\begin{aligned}&|\mathbb{A}^{\ddagger}|\left|\mathbb{I}+(\mathbb{A}^{\ddagger})^{-1}\left(\mathbf{d}(-\mu^{-1}(g\mathbf{d}^*\mathbb{A}^{\dagger}+\bar{\lambda}\mathbf{e}^*)-\frac{\lambda\mathbf{g}^*\mathbb{A}^{\perp}}{|\lambda|\sqrt{\mu}})\right.\right.\\ &\qquad\qquad \left.\left.+\mathbf{g}(\mu^{-1}(-d\mathbf{e}^*+\lambda\mathbf{d}^*\mathbb{A}^{\dagger})-\frac{(\sqrt{\mu}-|\lambda|)\mathbf{g}^*\mathbb{A}^{\perp}}{g\sqrt{\mu}})\right)\right|\\ &= |\mathbb{A}^{\ddagger}|\begin{vmatrix}1+(-\mu^{-1}(g\mathbf{d}^*\mathbb{A}^{\dagger}+\bar{\lambda}\mathbf{e}^*))\mathbf{b} & (\mu^{-1}(-d\mathbf{e}^*+\lambda\mathbf{d}^*\mathbb{A}^{\dagger}))^*\mathbf{b}\\ (-\lambda\mathbf{g}^*\mathbb{A}^{\perp}/|\lambda|\sqrt{\mu})(\mathbb{A}^{\perp})^*\mathbf{c} & 1-((\sqrt{\mu}-|\lambda|)\mathbf{g}^*\mathbb{A}^{\perp}/g\sqrt{\mu})^*(\mathbb{A}^{\perp})^*\mathbf{c}\end{vmatrix}\\ &= |\mathbb{A}^{\ddagger}|\begin{vmatrix}\bar{\lambda}/\mu & d/\mu\\ -\lambda g/|\lambda|\sqrt{\mu} & |\lambda|/\sqrt{\mu}\end{vmatrix} = |\mathbb{A}^{\ddagger}|\frac{(\bar{\lambda}-\lambda)|\lambda|^2+\lambda\mu}{\mu|\lambda|\sqrt{\mu}}\,.\end{aligned} \tag{B21}$$

A similar calculation yields (B13). To obtain (B16) we compute

$$\begin{aligned}&|\mathbb{A}^{\ddagger}|\left|\mathbb{I}+(\mathbb{A}^{\ddagger})^{-1}\left(-f^{-1}\mathbf{d}\mathbf{f}^*+\mathbf{g}(g^{-1}(-\mathbf{e}^*+\lambda f^{-1}\mathbf{f}^*)-(1/\sqrt{gf})\mathbf{f}^*)\right)\right|\\ &= |\mathbb{A}^{\ddagger}|\begin{vmatrix}1 & \mathbf{f}^*(\mathbb{A}^{\perp})^*\mathbf{c}\\ g^{-1}f^{-1}(\lambda-1) & 1+(g^{-1}\lambda f^{-1}-(1/\sqrt{gf}))\mathbf{f}^*(\mathbb{A}^{\perp})^*\mathbf{c}\end{vmatrix}\\ &= |\mathbb{A}^{\ddagger}|(1+((-(1/\sqrt{gf}))+g^{-1}f^{-1})\mathbf{f}^*(\mathbb{A}^{\perp})^*\mathbf{c})\\ &= |\mathbb{A}^{\ddagger}|\left[1+(g^{-1}f^{-1}-(1/\sqrt{gf}))\mathbf{g}^*\mathbb{A}^{\perp}\mathbf{f}\right].\end{aligned} \tag{B22}$$

□

When $\mathbb{A}$ and $\mathbb{A}_1$ are nonsingular, (B5) is the Sherman-Morrisson Formula (see e.g. [21]). For our application we need the singular vectors in $\mathbb{A}^{\perp}$, rather than $\mathbb{A}^{\perp}$ itself, but then only when $rank(\mathbb{A}^{\perp})\leq 3$. These singular vectors can be extracted by a simple modification of the power method with deflation.

References

[1] H. Agren, A. Flores-Riveros, and H.J.Aa. Jensen. Evaluation of first- and second-order nonadiabatic coupling elements from large multiconfigurational self-consistent-field wave functions. *Physical Review A*, 34(6):4606–4614, December 1986.

[2] Philippe Y. Ayala and H. Bernhard Schlegel. A nonorthogonal CI treatment of symmetry breaking in sigma formyloxyl radical. *J. Chem. Phys.*, 108(18):7560–7567, May 1998.

[3] Jerzy K. Baksalary, Oskar Maria Baksalary, and Gtz Trenkler. A revisitation of fomulae for the Moore-Penrose inverse of modified matrices. *Linear Algebra Appl.*, 372:207–224, 2003.

[4] G. Beylkin and M. J. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proc. Natl. Acad. Sci. USA*, 99(16):10246–10251, August 2002. http://www.pnas.org/cgi/content/abstract/112329799v1.

[5] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM J. Sci. Comput.*, 26(6):2133–2159, July 2005. http://amath.colorado.edu/pub/wavelets/papers/BEY-MOH2005.pdf.

[6] G. Beylkin and L. Monzón. On approximation of functions by exponential sums. *Appl. Comput. Harmon. Anal.*, 19(1):17–48, 2005. http://amath.colorado.edu/pub/wavelets/papers/afes.pdf.

[7] Gregory Beylkin, Vani Cheruvu, and Fernando Pérez. Fast adaptive algorithms in the non-standard form for multidimensional problems. *J. Comp. Phys.*, 2006. Submitted. APPM Preprint #550.

[8] D. Braess. Asymptotics for the approximation of wave functions by exponential sums. *J. Approx. Theory*, 83(1):93–103, 1995.

[9] Dietrich Braess and Wolfgang Hackbusch. Approximation of $1/x$ by exponential sums in $[1, \infty)$. *IMA J. Numer. Anal.*, 25(4):685–697, 2005.

[10] Rasmus Bro. Parafac. tutorial & applications. In *Chemom. Intell. Lab. Syst., Special Issue 2nd Internet Conf. in Chemometrics (INCINC'96)*, volume 38, pages 149–171, 1997. http://www.models.kvl.dk/users/rasmus/presentations/parafac_tutorial/paraf.htm.

[11] Eric Cancès, Mireille Defranceschi, Werner Kutzelnigg, Claude Le Bris, and Yvon Maday. Computational quantum chemistry: a primer. In *Handbook of Numerical Analysis, Vol. X*, pages 3–270. North-Holland, Amsterdam, 2003.

[12] Bin Chen and James R. Anderson. A simplified released-node quantum monte carlo calculation of the ground state of LiH. *J. Chem. Phys.*, 102(11):4491–4494, March 1995.

[13] E.U. Condon and G.H. Shortley. *The Theory of Atomic Spectra*. Cambridge Uinversity press, 1967.

[14] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \cdots, R_N)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.

[15] Fokke Dijkstra and Joop H. Van Lenthe. On the rapid evaluation of cofactors in the calculation of nonorthogonal matrix elements. *International Journal of Quantum Chemistry*, 67:77–83, 1998.

[16] Fokke Dijkstra and Joop H. Van Lenthe. Gradients in valence bond theory. *Journal of Chemical Physics*, 113(6):2100–2108, August 2000.

[17] Frank Ethridge and Leslie Greengard. A new fast-multipole accelerated Poisson solver in two dimensions. *SIAM J. Sci. Comput.*, 23(3):741–760 (electronic), 2001.

[18] Reinhold Fink and Volker Staemmler. A multi-configuration reference CEPA method based on pair natural orbitals. *Theoretica Chimica Acta*, 87(1/2):129–145, November 1993.

[19] P. Gilbert. The reconstruction of a three-dimensional structure from projections and its applications to electron microscopy II. Direct methods. *Proc. R. Soc. Lond. B.*, pages 89–102, 1972.

[20] T. L. Gilbert. Multiconfiguration self-consistent-field theory for localized orbitals. I. The orbital equations. *Phys. Rev. A*, 6(2), August 1972.

[21] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.

[22] W. Hackbusch and B. N. Khoromskij. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. I. Separable approximation of multi-variate functions. *Computing*, 76(3-4):177–202, 2006.

[23] W. Hackbusch and B. N. Khoromskij. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. II. HKT representation of certain operators. *Computing*, 76(3-4):203–225, 2006.

[24] Wolfgang Hackbusch. Entwicklungen nach exponentialsummen. Technical Report 4, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, Germany, 2005. see also `http://www.mis.mpg.de/scicomp/EXP_SUM/`.

[25] R.J. Harrison, G.I. Fann, T. Yanai, and G. Beylkin. Multiresolution quantum chemistry in multiwavelet bases. In P.M.A. Sloot et. al., editor, *Lecture Notes in Computer Science. Computational Science-ICCS 2003*, volume 2660, pages 103–110. Springer, 2003.

[26] R.J. Harrison, G.I. Fann, T. Yanai, Z. Gan, and G. Beylkin. Multiresolution quantum chemistry: basic theory and initial applications. *J. Chem. Phys.*, 121(23):11587–11598, 2004. `http://amath.colorado.edu/pub/wavelets/papers/mrqc.pdf`.

[27] Richard A. Harshman. Foundations of the parafac procedure: Model and conditions for an "explanatory" multi-mode factor analysis. Working Papers in Phonetics 16, UCLA, 1970. `http://publish.uwo.ca/~harshman/wpppfac0.pdf`.

[28] T. Helgaker and P.R. Taylor. *Modern Electronic Structure Theory*. World Scientific, Singapore, 1995.

[29] Tomasz Hrycak and Vladimir Rokhlin. An improved fast multipole algorithm for potential fields. *SIAM J. Sci. Comput.*, 19(6):1804–1826 (electronic), 1998.

[30] Walter Hunziker. On the spectra of Schrödinger multiparticle Hamiltonians. *Helv. Phys. Acta*, 39:451–462, 1966.

[31] M. H. Kalos. Monte Carlo calculations of the ground state of three- and four-body nuclei. *Phys. Rev. (2)*, 128:1791–1795, 1962.

[32] M. H. Kalos. Monte Carlo integration of the Schrödinger equation. *Trans. New York Acad. Sci. (2)*, 26:497–504, 1963/1964.

[33] Tosio Kato. Fundamental properties of Hamiltonian operators of Schrödinger type. *Trans. Amer. Math. Soc.*, 70:195–211, 1951.

[34] W. Klopper. R12 methods, Gaussian geminals. In J. Grotendorst et. al., editor, *Modern Methods and Algorithms of Quantum Chemistry*, volume 1 of *NIC Series*, pages 153–201. John von Neumann Institute for Computing, 2000.

[35] Wim Klopper and Claire C. M. Samson. Explicitly correlated second-order Møller Plesset methods with auxiliary basis sets. *Journal of Chemical Physics*, 116(15), April 2002.

[36] Pieter M. Kroonenberg and Jan de Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980.

[37] C. Le Bris, editor. *Handbook of Numerical Analysis. Vol. X*. North-Holland, Amsterdam, 2003. Special Volume: Computational Chemistry.

[38] S. E. Leurgans, R. A. Moyeed, and B. W. Silverman. Canonical correlation analysis when the data are curves. *J. Roy. Statist. Soc. Ser. B*, 55(3):725–740, 1993.

[39] Roland Lindh, Jeppe Olsen, and Björn O. Roos. Low-rank configuration interaction with orbital optimization—the LR SCF approach. *Chemical Physics Letters*, 148(4):276–280, July 1988.

[40] Per-Olov Löwdin. Quantum theory of many-particle systems. I. Physical interpretations by means of density matrices, natural spin-orbitals, and convergence problems in the method of configuration interaction. *Physical Review*, 97(6):1474–1489, March 1955.

[41] Arne Lüchow and Reinhold Fink. On the systematic improvement of fixed-node diffusion quantum Monte Carlo energies using natural orbital CI guide functions. *J. Chem. Phys.*, 113(19):8457–8463, November 2000.

[42] Carl D. Meyer, Jr. Generalized inversion of modified matrices. *SIAM J. Appl. Math.*, 24:315–323, 1973.

[43] Martin J. Mohlenkamp and Lucas Monzón. Trigonometric identities and sums of separable functions. *The Mathematical Intelligencer*, 27(2):65–69, 2005. `http://www.math.ohiou.edu/~mjm/research/sine.pdf`.

[44] Martin J. Mohlenkamp and Todd Young. Convergence of Green iterations for Schrödinger equations. In Xiaoping Shen, editor, *Proceedings of International Workshop on Computational Science and its Education 2005*, (to appear).

[45] Thomas Muir. *A Treatise on the Theory of Determinants*. Privately published, Albany New York, 1930. revised and enlarged by William H. Metzler.

[46] Jozef Noga, Werner Kutzelnigg, and Wim Klopper. CC-R12, a correlation cusp corrected coupled-cluster method with a pilot application to the Be_2 potential curve. *Chemical Physics Letters*, 199(5), November 1992.

[47] Jeppe Olsen, Per-Ake Malmquist, Björn O. Roos, Roland Lindh, and Per-Olaf Widmark. A non-linear approach to configuration interaction. The low-rank CI method (LR CI). *Chemical Physics Letters*, 133(2):91–101, January 1987.

[48] Ruben Pauncz. *The Symmetric Group in Quantum Chemistry*. CRC Press, Boca Raton, FL, 1995.

[49] B. Joakim Persson and Peter R. Taylor. Accurate quantum-chemical calculations: The use of gaussian-type geminal functions in the treatment of electron correlation. *J. Chem. Phys.*, 105(14):5915–5926, October 1996.

[50] B. Joakim Persson and Peter R. Taylor. Molecular integrals over gaussian-type geminal basis functions. *Theor. Chem. Acc.*, 97:240–250, 1997.

[51] V.V. Prasolov. *Problems and Theorems in Linear Algebra*, volume 134 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, R.I., 1994.

[52] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics. II. Fourier analysis, self-adjointness*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1975.

[53] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics. IV. Analysis of operators*. Academic Press [Harcourt Brace Jovanovich Publishers], New York, 1978.

[54] Franz Rellich. Störungstheorie der Spektralzerlegung. V. *Math. Ann.*, 118:462–484, 1942.

[55] Sven Peter Rudin. *Configuration Interaction with Non-orthogonal Slater Determinants Applied to the Hubbard Model, Atoms, and Small Molecules*. PhD thesis, The Ohio State University, 1997.

[56] Jacek Rychlewski, Wojciech Cencek, and Jacek Komasa. The equivalence of explicitly correlated slater and gaussian functions in variational quantum chemistry computations. the ground state of H_2. *Chemical Physics Letters*, 229:657–660, November 1994.

[57] C. David Sherrill and Henry F. Schaefer III. The configuration interaction method: Advances in highly correlated approaches. *Advances in Quantum Chemisty*, 127:143–269, 1999.

[58] Age Smilde, Rasmus Bro, and Paul Geladi. *Multi-way Analysis. Applications in the Chemical Sciences*. John Wiley & Sons, 2004.

[59] N. Yarvin and V. Rokhlin. Generalized Gaussian quadratures and singular value decompositions of integral operators. *SIAM J. Sci. Comput.*, 20(2):699–718 (electronic), 1999.

[60] Jürgen Zanghellini. *Multi-Electron Dynamics in the Ionization of Molecules by Strong Laser Pulses*. PhD thesis, Vienna University of Technology, Vienna, Austria, 2004.