Numerical operator calculus in higher dimensions

Gregory Beylkin* and Martin J. Mohlenkamp

Applied Mathematics, University of Colorado, Boulder, CO 80309

When an algorithm in dimension one is extended to dimension *d*, in nearly every case its computational cost is taken to the power *d*. This fundamental difficulty is the single greatest impediment to solving many important problems and has been dubbed the *curse of dimensionality*. For numerical analysis in dimension *d*, we propose to use a representation for vectors and matrices that generalizes separation of variables while allowing controlled accuracy. Basic linear algebra operations can be performed in this representation using one-dimensional operations, thus bypassing the exponential scaling with respect to the dimension. Although not all operators and algorithms may be compatible with this representation, we believe that many of the most important ones are. We prove that the multiparticle Schrödinger operator, as well as the inverse Laplacian, can be represented very efficiently in this form. We give numerical evidence to support the conjecture that eigenfunctions inherit this property by computing the ground-state eigenfunction for a simplified Schrödinger operator with 30 particles. We conjecture and provide numerical evidence that functions of operators inherit this property, in which case numerical operator calculus in higher dimensions becomes feasible.

In almost all problems that arise from physics there is an underlying physical dimension, and in almost every case the algorithm to solve the problem will have computational complexity that grows exponentially in the physical dimension. In other words, when an algorithm in dimension one is extended to dimension d, its computational cost is taken to the power d. In this paper we present an approach that, in several important cases, allows one-dimensional algorithms to be extended to d dimensions without their computational complexity growing exponentially in d. In moderate dimensions ($d = 2, 3, 4$) our approach greatly accelerates a number of algorithms. In higher dimensions, such as those arising from the multiparticle Schrödinger equation, where the wave function for p particles has $d = 3p$ variables, our approach makes algorithms feasible that would be unthinkable in a traditional approach.

As an example of the exponential growth in d, consider ordinary matrix–matrix multiplication. In dimension d a matrix has $(N^2)^d$ entries, and matrix–matrix multiplication takes $(N^3)^d$ operations. Using a "fast" one-dimensional algorithm does not help: a banded matrix has $(bN)^d$ entries, and matrix–matrix multiplication takes $(b^2N)^d$ operations. This fundamental difficulty is the single greatest impediment to solving many real-world problems and has been dubbed the *curse of dimensionality* (1).

In problems in physics where the underlying assumptions permit, separation of variables has been the most successful approach for avoiding the high cost of working in d dimensions. Instead of trying to find a d-dimensional function that solves the given equation (e.g., the multiparticle Schrödinger equation), one only considers functions that can be represented as a product:

$$f(x_1, \ldots, x_d) = \phi_1(x_1) \cdots \phi_d(x_d). \qquad [1]$$

By substituting Eq. **1** into the original equation, one often can produce a system of d weakly coupled one-dimensional equations for the functions $\{\phi_i(x_i)\}$ [e.g., via Hartree or Kohn–Sham formulations (2)]. By iteratively solving the equation in x_i for the function ϕ_i, one obtains an approximate solution to the original

equation using only one-dimensional operations and thus avoiding the exponential dependence on d. However, if the best approximate solution of the form (Eq. **1**) is not good enough, there is no way to improve the accuracy.

The natural extension of Eq. **1** is the form

$$f(x_1, \ldots, x_d) = \sum_{l=1}^{r} s_l \phi_1^l(x_1) \cdots \phi_d^l(x_d). \qquad [2]$$

The key quantity in Eq. **2** is r, which we call the *separation rank*. By increasing r, the approximate solution can be made as accurate as desired. One way to use Eq. **2** is to fix the functions $\{\phi_i^l(x_i)\}$ from some (basis) set and try to solve for the coefficients s_l. This option includes the use of a tensor product basis as well as configuration interaction methods. Although a wise choice of functions $\{\phi_i^l(x_i)\}$ may reduce the number of degrees of freedom necessary to discretize the problem in each direction, N, it does not affect the exponential scaling with the dimension. A second option is to substitute Eq. **2** into the original equation. Unfortunately, the resulting equations for the functions $\{\phi_i^l(x_i)\}$ are generally intractable, since the equations are strongly coupled. There are many variations on these two approaches within computational quantum mechanics (e.g., see ref. 2). Our approach is distinct from both of these traditional approaches.

Analytic consideration of Eq. **2** raises two questions. The first question is, what class of functions can be represented efficiently in this form? One can, for instance, characterize a class based on the mixed derivatives of order k, for which the approximation error is $O(r^{-kd/(d-1)})$ (3). Along the same lines, the "sparse-grids" (e.g., refs. 4 and 5) methods identify classes of functions compatible with certain representations. For these functions, reduction in the computational complexity from $O(N^d)$ to $O(N^{d-1}\ln N)$ can be achieved, but the complexity is still exponential. Clearly, classes of functions in multiple dimensions are extremely rich, and such approaches face great difficulties. In contrast, our approach relies on properties of physically significant operators.

The second question is, how does one find the representation in Eq. **2** for a given function? Optimized separated representations such as in Eq. **2** have been studied for more than 30 years for statistical applications, and we refer to refs. 6–8 and the references therein. Since the problems considered for statistics have as input a dense d-dimensional data cube, their applications have been limited to small dimensions ($d \ll 10$, mostly $d = 3$).

Summary of the Paper

The contribution of this paper is twofold. First, we present a computational paradigm. With hindsight it is very natural, but this perspective was the most difficult part to achieve, and it has far-reaching consequences. In our approach, we use the natural extension of separation of variables in Eq. **2** but neither fix the set $\{\phi_i^l(x_i)\}$ nor try to find and solve equations for $\{\phi_i^l(x_i)\}$. We use Eq. **2** simply as an approximation technique for functions and operators and try to minimize the number of terms at each step

of the algorithm. Second, we start the development of a theory that demonstrates that separation ranks are low for many problems of interest. We show here that certain physically significant operators have small separation rank.

Operators can be represented in a form similar to Eq. **2**,

$$\mathcal{A} = \sum_{l=1}^{r} s_l \mathcal{B}_1^l \otimes \cdots \otimes \mathcal{B}_d^l. \qquad [3]$$

We observe that many linear algebra operations can be performed while keeping all objects in the form of Eqs. **3** or **2**. We can perform operations in d dimensions using combinations of one-dimensional operations and so achieve computational complexity that scales *linearly* in d. We then solve the original equation directly by some appropriate algorithm while maintaining the intermediate functions and operators in the forms of Eqs. **2** and **3**, with adaptively chosen r. The same approach applies to computing functions of operators, in particular polynomials, exponentials, inverses, and sign functions, which form the foundation for a numerical operator calculus.

Although each linear algebra operation leads to an object in the form of Eqs. **3** or **2**, the result will have larger separation rank. If we allow r to grow uncontrollably, then the representation will quickly become untenable. Therefore, at each step of the algorithm, we seek to minimize the separation rank r by adaptively changing the coefficients $\{s_l\}$ and the functions $\{\phi_j^l\}$ while maintaining the required accuracy. We present a numerical algorithm to perform this reduction. We note that our algorithm is similar to an algorithm used in statistics (e.g., ref. 8). In our approach, however, we never handle a d-dimensional cube.

We have observed that the reduced separation rank produced by our algorithm is typically optimal or nearly optimal. It is an interesting question if such algorithms can, in general, guarantee optimality. For our purposes optimal representations are desired but not required.

In this paper, we prove that both the multiparticle Schrödinger operator and the inverse Laplacian can be represented with separation rank $r = O(\log d)$. We feel strongly that the class of operators with low separation rank is much wider than we can demonstrate at present.

We conjecture that functions associated with such operators inherit low separation rank. We know, for example, that if the Green's function and the initial/boundary condition have low separation rank, then it follows immediately that the same holds for the solution of the equation. We conjecture that eigenfunctions inherit a low separation rank from the operator. We present numerical results for the computation of the ground-state eigenfunction of a simplified model of a 30-electron Schrödinger operator using the power method, with accuracies ranging from 10^{-2} to 10^{-7}. Using a direct method this computation would be impossible, since it would require on the order of 10^{80} operations, but with our approach it took a few hours on a desktop computer. We conjecture that functions of operators inherit a low separation rank and present a numerical example of the computation of a sign function.

Finally, we note that some problems considered in statistical analysis and other applications can be recast as scattered data interpolation in d dimensions, which is feasible with our approach. Although the implications of this observation are very interesting, we do not address them here.

The Separated Representation

In order to emphasize the underlying physical dimension, we define operators and functions in d dimensions. To avoid confusion between, e.g., a "vector in two dimensions" and a "matrix," we clarify our notation and nomenclature. A function

f in dimension d is a map $f : \mathbf{R}^d \to \mathbf{R}$ from d-dimensional Euclidean space to the real numbers. We write f as $f(x_1, \ldots, x_d)$, where $x_i \in \mathbf{R}$. A vector $\mathbf{F}$ in dimension d is a discrete representation of a function in dimension d on a rectangular domain. We write it as $\mathbf{F} = F(j_1, \ldots, j_d)$, where $j_i = 1, \ldots, N_i$. A linear operator $\mathcal{A}$ in dimension d is a linear map $\mathcal{A} : S \to S$ where S is a space of functions in dimension d. A matrix $\mathbb{A}$ in dimension d is a discrete representation of a linear operator in dimension d. We write $\mathbb{A} = A(j_1, j_1'; \ldots; j_d, j_d')$, where $j_i = 1, \ldots, N_i$ and $j_i' = 1, \ldots, N_i'$. For simplicity we assume $N_i' = N_i = N$ for all i.

Definition 1 (separated matrix representation): For a given ε, we represent a matrix $\mathbb{A} = A(j_1, j_1'; j_2, j_2'; \ldots; j_d, j_d')$ in dimension d as

$$\sum_{l=1}^{r} s_l V_1^l(j_1, j_1') V_2^l(j_2, j_2') \cdots V_d^l(j_d, j_d'), \qquad [4]$$

where s_l is a scalar, $s_1 \geq \cdots \geq s_r > 0$, and $\mathbb{V}_i^l$ are matrices in dimension one with norm 1. We require the error to be less than ε:

$$\left\| \mathbb{A} - \sum_{l=1}^{r} s_l \mathbb{V}_1^l \otimes \mathbb{V}_2^l \otimes \cdots \otimes \mathbb{V}_d^l \right\| < \varepsilon. \qquad [5]$$

We call the scalars s_l *separation values* and the integer r the *separation rank*. The smallest r that yields such a representation for a given ε is the *optimal separation rank*.

When possible, it is preferable to use the operator norm; otherwise the Frobenius norm is used. The definition for a vector is similar, with the matrices $\mathbb{V}_i^l$ replaced by the vectors $\mathbf{V}_i^l$.

In dimension $d = 2$, the separated representation in Eq. **4** reduces to a form similar to the singular value decomposition (SVD), and in fact we can construct an optimal representation using an ordinary SVD algorithm but with an unusual pairing of indices. Instead of separating vectors in dimension 2 in the input coordinate (j_1', j_2') from vectors in dimension 2 in the output coordinate (j_1, j_2), we separate matrices in dimension 1 in the j_1 direction from matrices in dimension 1 in the j_2 direction. Thus, common matrix operators that have full rank as operators may still have low separation rank. For example, the identity is trivially represented as $\mathcal{I}_1 \otimes \mathcal{I}_2 \otimes \cdots \otimes \mathcal{I}_d$, with separation rank one.

When $d > 2$ this representation is not unique even when r is the optimal separation rank. The optimal representations for different values of ε are not nested, so we cannot simply add or delete terms from Eq. **4** when ε changes and retain a representation with optimal separation rank. The numbers s_l, matrices $\mathbb{V}_i^l$, and separation rank r all will change as ε changes. Issues related to such generalizations of the SVD have been studied extensively (e.g., see refs. 6 and 7 and the references therein).

To illustrate many of these issues for $d > 2$, we consider a sine wave in the diagonal direction, $\sin(x_1 + \cdots + x_d)$, and attempt to represent it in the separated form using only real functions. We can use the usual trigonometric formulas for sums of angles to obtain a separated representation, but then we will have $r = 2^{d-1}$ terms. The numerical algorithm described below, however, uncovered a trigonometric identity in d dimensions, using exactly d terms.

LEMMA 1.

$$\sin\left(\sum_{j=1}^{d} x_j\right) = \sum_{j=1}^{d} \sin(x_j) \prod_{k=1, k\neq j}^{d} \frac{\sin(x_k + \alpha_k - \alpha_j)}{\sin(\alpha_k - \alpha_j)}, \qquad [6]$$

for all choices of $\{\alpha_j\}$ such that $\sin(\alpha_k - \alpha_j) \neq 0$ for all $j \neq k$.

This identity illustrates several key points. First, the "obvious" (analytic) separated representation may be woefully inefficient. Second, even when $\varepsilon = 0$ and r is optimal, there may be entire families of separated representations. Third, among these separated representations, some may have large separation values, leading potentially to cancellations and, hence, poor conditioning.

The Multiparticle Schrödinger Operator

We consider the Schrödinger operator $\mathcal{H}$ for a d-electron system and show that the separation rank of an appropriate approximation of $\mathcal{H}$ grows only logarithmically in d. Without changing the basic in formalism in Eq. **4**, we choose to use three-dimensional operators as our elementary building blocks. The operator $\mathcal{H}$ is equal to $-\Delta + \mathcal{N} + \mathcal{E}$, where the Laplacian Δ is defined by

$$\Delta = (\Delta_1 \otimes \mathcal{I}_2 \otimes \cdots \mathcal{I}_d) + \cdots + (\mathcal{I}_1 \otimes \cdots \otimes \Delta_d), \qquad [7]$$

the nuclear potential portion $\mathcal{N}$ is defined by

$$\mathcal{N} = (\mathcal{V}_1 \otimes \mathcal{I}_2 \otimes \cdots \mathcal{I}_d) + \cdots + (\mathcal{I}_1 \otimes \cdots \otimes \mathcal{V}_d), \qquad [8]$$

and the electron–electron interaction portion $\mathcal{E}$ is defined by

$$\mathcal{E} = \sum_{i=1}^{d-1} \sum_{m=i+1}^{d} \mathcal{W}_{im}, \qquad [9]$$

where $\mathcal{V}_i$ is the operator that multiplies by the nuclear potential function $v(x)$ in the (three-dimensional) variable x_i, and $\mathcal{W}_{im}$ is multiplication by the electron–electron interaction (Coulomb) potential $w(x_i - x_m)$. Since any numerical treatment must be on a finite domain and use a finite step size, we state our results for an approximation to $\mathcal{H}$ that has been suitably limited in space and in wave number. We choose a fixed but arbitrary precision ε to which to approximate $\mathcal{H}$ and assume that we are working in finite precision arithmetic with roundoff error μ.

The following theorem provides bounds on the separation rank for Δ and $\mathcal{N}$, and its proof provides a construction for their separated representations.

THEOREM 2. *Let* $\mathcal{A} = \Sigma_{i=1}^{d} \mathcal{B}_i$, *where* $\mathcal{B}_i$ *is the bounded operator* $\mathcal{B}$ *acting in direction* i. *We can represent* $\mathcal{A}$ *to within* ε *in the operator norm with separation rank*

$$r = O\left[\frac{\log(d\|\mathcal{B}\|/\varepsilon)}{\log(1/\mu) - \log(d\|\mathcal{B}\|/\varepsilon)}\right]. \qquad [10]$$

Proof: Consider the auxiliary operator-valued function of the real variable t

$$\mathcal{G}(t) = \|\mathcal{B}\| \bigotimes_{i=1}^{d} (\mathcal{I}_i + t\mathcal{B}_i/\|\mathcal{B}\|), \qquad [11]$$

and note that $\mathcal{G}'(0) = \mathcal{A}$. Using an appropriate finite difference formula of order r, we approximate

$$\mathcal{G}'(0) \approx \sum_{j=1}^{r} \alpha_j \mathcal{G}(t_j) \equiv \|\mathcal{B}\| \sum_{j=1}^{r} \alpha_j \bigotimes_{i=1}^{d} (\mathcal{I}_i + t_j \mathcal{B}_i/\|\mathcal{B}\|), \qquad [12]$$

thus providing a separation rank r approximation for $\mathcal{A}$. As with all finite difference approximations, the error bound for an order r formula has the form $\mu/h + h^r\|\mathcal{G}^{(r+1)}\|$, where h is the step size. The bound on r follows by optimizing over h.

The estimate in Eq. **10** implies that, as long as $\|\mathcal{B}\|d/\varepsilon \ll 1/\mu$, the separation rank is $O[\log(\|\mathcal{B}\|d/\varepsilon)]$. Thus, for fixed ε, the separation rank grows only as $\log(d)$. Another implication of these estimates is the restriction $\varepsilon > \|\mathcal{B}\|d\mu$ on the accuracy attainable using the form of Eq. **12**. Note that if we choose to approximate with relative precision and assume $\|\mathcal{A}\| \approx d\|\mathcal{B}\|$, then the separation rank is independent of the dimension.

For $\mathcal{E}$, we use a similar theorem and construction.

THEOREM 3. *Let* $\mathcal{A} = \Sigma_{i=1}^{d-1} \Sigma_{m=i+1}^{d} \mathcal{B}_i \mathcal{B}_m$, *where* $\mathcal{B}_i$ *is the bounded operator* $\mathcal{B}$ *acting in direction* i. *We can represent* $\mathcal{A}$ *to within* ε *in the operator norm with separation rank*

$$r = O\left(\frac{\log(d^2\|\mathcal{B}\|^2/\varepsilon)}{\log(1/\mu) - \log(d^2\|\mathcal{B}\|^2/\varepsilon)}\right). \qquad [13]$$

Proof: With $\mathcal{G}$ defined in Eq. **11**, note that $\mathcal{G}''(0) = 2\mathcal{A}/\|\mathcal{B}\|$. We can thus use the same approach as in Eq. **12**, simply substituting a second derivative finite difference.

To use this result for $\mathcal{E}$, we first symmetrically separate a single term in Eq. **9** to obtain the representation

$$\left\| \mathcal{W}_{im} - \sum_{k=1}^{K_\varepsilon} \mathcal{W}_i^k \mathcal{W}_m^k \right\| < \varepsilon \qquad [14]$$

for some separation rank K_ε and collection of (multiplication) operators $\mathcal{W}_i^k$. This can be accomplished using, for example, the SVD. We then substitute Eq. **14** into **9** and exchange the order of summation to obtain

$$\sum_{k=1}^{K_\varepsilon} \sum_{i=1}^{d-1} \sum_{m=i+1}^{d} \mathcal{W}_i^k \mathcal{W}_m^k, \qquad [15]$$

which gives a separated representation with separation rank $K_\varepsilon d(d-1)/2$. For each value of k, however, we apply Theorem 3. Since K_ε is independent of d, we conclude that the separation rank of $\mathcal{E}$ grows only logarithmically in d rather than quadratically.

The Inverse Laplacian

It can be shown (9) that for $0 < \delta < y < D$ and any $\varepsilon > 0$, there exist $M = O\{\log[D/(\delta\varepsilon)]\}$ numbers $\alpha_l, \tau_l > 0$ such that

$$\left\| \frac{1}{y} - \sum_{l=1}^{M} \alpha_l \exp(-\tau_l y) \right\| < \varepsilon. \qquad [16]$$

We can use this approximation to construct a separated representation for the inverse of $-\Delta$ on part of its spectrum. Similar arguments can be made for other classical Green's functions.

THEOREM 4. *For the interval* $[\delta, D]$ *of the spectrum of* $-\Delta$, *we have*

$$\left\| -\Delta^{-1} - \sum_{l=1}^{r} \alpha_l \exp(\tau_l \Delta) \right\| < \varepsilon. \qquad [17]$$

with separation rank $r = O\{\log[D/(\delta\varepsilon)]\}$.

For functions with Fourier transform supported in the annulus $(\Sigma_{i=1}^{d} \xi_i^2)^{1/2} \in [\sqrt{\delta}, \sqrt{D}]$, we thus have a separated representation for the Green's function:

$$\sum_{l=1}^{r} \alpha_l \bigotimes_{i=1}^{d} \frac{1}{\sqrt{4\pi\tau_l}} \exp(-x_i^2/4\tau_l). \qquad [18]$$

For periodic problems it may be more natural to restrict the wave number to the "cubic annulus" $\max_{1 \leq i \leq d} |\xi_i| \in [\sqrt{\delta}, \sqrt{D}]$, in which case the approximation (Eq. 17) needs to be valid on the interval $[\delta/d, dD]$, and thus the separation rank grows as $r = O\{\log[d^2D/(\delta\varepsilon)]\}$ with the dimension d.

Basic Linear Algebra

The main point of the separated representation is that the elementary objects on which we operate are one-dimensional, so that linear algebra in dimension d is performed using only one-dimensional operations. The computational complexities are linear in d rather than exponential.

We assume that all objects have been discretized using N points in each direction, so a vector in dimension d has N^d entries. In d dimensions, a dense matrix has $(N^2)^d$ entries, whereas a banded matrix has $(bN)^d$ entries. In the separated representation (Eq. 4), we will need to store $d \cdot r \cdot N^2$ entries if the matrices $\mathbb{V}_i^l$ are dense or $d \cdot r \cdot bN$ entries if they are banded. The banded case demonstrates the effect of combining the separated representation with a fast one-dimensional algorithm.

Addition of two matrices in d dimensions takes $(N^2)^d$ operations if they are dense and $(bN)^d$ if they are banded. In the separated representation the addition $\tilde{\mathbb{A}} + \hat{\mathbb{A}}$ is merely the merging of two sums and then resorting. Thus addition is essentially free but yields a matrix with separation rank $r = \hat{r} + \tilde{r}$.

Multiplication of matrices is the most important operation for the applications we have in mind. In d dimensions, multiplication takes $(N^3)^d$ operations for dense matrices and $(b^2N)^d$ operations for banded matrices. In the separated representation, multiplication can be done using only one-dimensional operations:

$$\hat{\mathbb{A}}\tilde{\mathbb{A}} = \sum_{\hat{l}=1}^{\hat{r}} \sum_{\tilde{l}=1}^{\tilde{r}} \hat{s}_{\hat{l}} \tilde{s}_{\tilde{l}} (\hat{\mathbb{V}}_1^{\hat{l}} \tilde{\mathbb{V}}_1^{\tilde{l}}) \otimes \cdots \otimes (\hat{\mathbb{V}}_d^{\hat{l}} \tilde{\mathbb{V}}_d^{\tilde{l}}). \qquad \textbf{[19]}$$

We have $r = \hat{r}\tilde{r}$ pairings, each of which costs d one-dimensional matrix–matrix multiplications. Thus, we need $d \cdot \hat{r} \cdot \tilde{r} \cdot N^3$ operations if the one-dimensional matrices are dense and $d \cdot \hat{r} \cdot \tilde{r} \cdot b^2 \cdot N$ operations if they are banded. The resulting matrix has separation rank $r = \hat{r}\tilde{r}$.

Several other operations are also efficient in the separated representation, in particular the computation of inner products, Frobenius norm, trace, matrix–vector multiplication, etc. We will not discuss these, since they follow the same pattern as matrix–matrix multiplication. It is also possible to solve a linear system in the separated representation by posing the problem as a least-squares problem and then doing a variant of the separation rank reduction algorithm described below.

The following are examples of algorithms that use only matrix and vector operations that can be done in the separated representation with computational cost linear in d.

1. Power method ($\mathbf{F}_{k+1} = \mathbb{A}\mathbf{F}_k$) to determine the largest eigenvalue and its eigenvector for a matrix $\mathbb{A}$.
2. Schulz iteration ($\mathbb{B}_{k+1} = 2\mathbb{B}_k - \mathbb{B}_k\mathbb{A}\mathbb{B}_k$) to construct $\mathbb{A}^{-1}$.
3. Sign iteration [$\mathbb{A}_{k+1} = (3\mathbb{A}_k - \mathbb{A}_k^3)/2$] to construct sign($\mathbb{A}$) (10).
4. Scaling and squaring ($[\exp(\mathbb{A}/2^n)]^{2^n}$) to construct the matrix exponential $\exp(\mathbb{A})$.
5. Inverse power method ($\mathbb{A}\mathbf{F}_{k+1} = \mathbf{F}_k$) for computing the smallest eigenvalue and its eigenvector for a matrix $\mathbb{A}$.

Since the basic linear algebra operations increase the separation rank, after each step in these algorithms we reduce it using the algorithm in the following section. In all our experiments we observe that the final and intermediate matrices have low separation rank, and we conjecture that this is true for a wide class of problems.

Finding "Optimal" Representations

The key to the success of our approach is finding a separated representation with low separation rank for matrices of interest. We assume that $\mathbb{A}$ is given to us in the separated representation but with larger separation rank r than necessary, as occurs, for example, when we multiply two matrices to form $\mathbb{A}$. We have found the algorithm described here to be effective in practice, although it is not guaranteed to find the optimal representation. This algorithm uncovered the trigonometric identity (Eq. 6) and the derivative formulation (Eq. 12) for the multiparticle Schrödinger operator and produced the numerical results described below. Although it is a problem of ongoing interest to find the optimal separation rank (e.g., see refs. 6–8, 11, and 12), it is not required for our approach. In fact, a suboptimal representation is preferred when it has better conditioning or can be obtained faster. For example, when $d = 2$ we find it much more efficient to use a faster algorithm that produces a suboptimal solution, even though the SVD is available in this dimension.

Our overall strategy is to find the best approximation with separation rank $\tilde{r}$, beginning with $\tilde{r} = 1$, and then increase $\tilde{r}$ and try again until the residual is less than ε. This strategy allows us to take our solution at one value of $\tilde{r}$, add another term (obtained e.g. by a power method) and use it as our initial approximation for separation rank $\tilde{r} + 1$. Although the best representations of different separation ranks are not actually nested, this strategy provides us with a good initial approximation at each $\tilde{r}$. Other initialization strategies exist (see also ref. 11).

Before starting, we reduce our search space by rotating each direction into an efficient basis. For each fixed direction i, the matrices $\{\mathbb{V}_i^l\}_{l=1}^r$ each have N^2 (or bN) entries and span a vector space of dimension $M_i \leq \min(r, N^2)$. By computing a basis for the span, we can express the matrix $\mathbb{V}_i^l$ in this basis as a vector $\mathbf{V}_i^l$ of length M_i. The matrix $\mathbb{A}$ is thereby expressed as the vector

$$\mathbf{A} = \sum_{l=1}^r s_l \mathbf{V}_1^l \otimes \mathbf{V}_2^l \otimes \cdots \otimes \mathbf{V}_d^l. \qquad \textbf{[20]}$$

We will reduce the separation rank of $\mathbf{A}$ and then undo the change of basis to recover the matrix $\mathbb{A}$. For simplicity we will assume $M_i = M$ for all i.

Alternating Least Squares

For a given separation rank $\tilde{r}$, the best separated representation is that which minimizes the error (Eq. 5), i.e. solves the nonlinear least-squares problem. To make this problem tractable, we exploit the multilinearity of the problem and use an alternating least-squares approach. This approach is also used in statistics (e.g., see refs. 8 and 11). Besides a few technical details, the key conceptual difference in our approach is that our input is a vector in the separated representation rather than a dense data vector in dimension d, and thus we can consider much larger values of N and d.

In the alternating least-squares approach one starts with an initial approximation

$$\tilde{\mathbf{A}} = \sum_{\tilde{l}=1}^{\tilde{r}} \tilde{s}_{\tilde{l}} \tilde{\mathbf{V}}_1^{\tilde{l}} \otimes \tilde{\mathbf{V}}_2^{\tilde{l}} \otimes \cdots \otimes \tilde{\mathbf{V}}_d^{\tilde{l}} \qquad \textbf{[21]}$$

to $\mathbf{A}$ in Eq. 20, and then iteratively refines it. We loop through the directions $k = 1, \ldots, d$. For each direction k, fix the vectors in the other directions $\{\tilde{\mathbf{V}}_i^{\tilde{l}}\}_{i \neq k}$ and then solve a linear least-squares problem for new $\tilde{\mathbf{V}}_k^l$ (and $\tilde{s}_{\tilde{l}}$). Repeat the loop in k iteratively until convergence is detected or $\|\mathbf{A} - \tilde{\mathbf{A}}\| < \varepsilon$. Although it is easy to prove that the norm of the residual ($\|\mathbf{A} - \tilde{\mathbf{A}}\|$) decreases at each step, in general this process is not guaranteed to converge to the best representation with separation rank $\tilde{r}$.

This linear least-squares problem naturally divides into separate problems for each coordinate. For fixed direction k, form the matrix $\mathbb{B}$ with entries

$$B(\hat{l}, \tilde{l}) = \prod_{i \neq k} \langle \tilde{\mathbf{V}}_i^{\tilde{l}}, \tilde{\mathbf{V}}_i^{\hat{l}} \rangle. \qquad [22]$$

Then, for a fixed coordinate j_k, form the vector $\mathbf{b}_{j_k}$ with entries

$$b_{j_k}(\hat{l}) = \sum_{l=1}^{r} s_l V_k^l(j_k) \prod_{i \neq k} \langle \mathbf{V}_i^l, \tilde{\mathbf{V}}_i^{\hat{l}} \rangle. \qquad [23]$$

The normal equations for the direction k and coordinate j_k become

$$\mathbb{B} c_{j_k}(\tilde{l}) = \mathbf{b}_{j_k}, \qquad [24]$$

which we solve for $c_{j_k}(\tilde{l})$ as a vector in $\tilde{l}$. After computing $c_{j_k}(\tilde{l})$ for all coordinates j_k, we let $\tilde{s}_{\tilde{l}} = \|c_{j_k}(\tilde{l})\|$ and $\tilde{V}_k^l(j_k) = c_{j_k}(\tilde{l})/\tilde{s}_{\tilde{l}}$, where the norm is taken with respect to the coordinate j_k.

For fixed direction k and coordinate j_k, it requires $\tilde{r}^2 \cdot d \cdot M$ operations to compute $\mathbb{B}$, $\tilde{r} r \cdot d \cdot M$ to compute $\mathbf{b}_{j_k}$, and $\tilde{r}^3$ to solve the system. Since $\mathbb{B}$ and the inner products in $\mathbf{b}_{j_k}$ are independent of the coordinate j_k, computing for another value of j_k has incremental cost $r\tilde{r} + \tilde{r}^2$. Similarly, many of the computations involved in $\mathbb{B}$ and $\mathbf{b}_{j_k}$ are the same for different k. Thus, one full alternating least-squares iteration costs $O[d \cdot \tilde{r}(\tilde{r}^2 + r \cdot M)]$. Because this algorithm uses inner products that can only be computed to within roundoff error μ, the best accuracy obtainable is $\varepsilon = \sqrt{\mu}$.

We have found it prudent to monitor the conditioning of the matrix $\mathbb{B}$ in Eq. **22** by detecting small pivots during the linear solve of Eq. **24**. If a small pivot is detected, it means the set of vectors $\{\tilde{\mathbf{V}}_1^{\tilde{l}} \otimes \cdots \otimes \tilde{\mathbf{V}}_d^{\tilde{l}}\}$ that make up $\tilde{\mathbb{A}}$ have become nearly linearly dependent. If the separation values are large, it signals a legitimate but poorly conditioned representation, in which case we should raise $\tilde{r}$ to alleviate the conditioning. If the separation values are not large, we can instead discard the vector corresponding to the small pivot and reduce $\tilde{r}$ by one.

Numerical Examples

In this section we provide numerical evidence to support our conjectures and to demonstrate that our approach can be used successfully.

Alternating Least-Squares Tests. In the first series of tests we generate a random separated vector of norm one (Eq. **20**) and then look for a representation (Eq. **21**) with the same separation rank using our algorithm. On a typical run with $d = 30$, $M = 100$, and $r = 100$, the residual decreased steadily but slowly as $\tilde{r}$ was increased and was still $2 \cdot 10^{-2}$ when $\tilde{r} = 99$. At $\tilde{r} = 100$ the error dropped to $2 \cdot 10^{-8}$, the best obtainable in double precision. The entire process took 2,900 seconds on a 360-MHz Sun Ultra10.

The second test is to find a representation for $\sin(\Sigma_{j=1}^d x_j)$, starting with the separated representation with $r = 2^{d-1}$ terms, obtained via ordinary trigonometric formulas. When started with $\tilde{r} = d$, the alternating least-squares algorithm quickly finds separated representations with $\tilde{r} = d$ terms, thereby generating an instance of the trigonometric identity (Eq. **6**). When started with $\tilde{r} < d$, however, it pursued poorly conditioned representations.

The third test is the proton-type potential (Eq. **8**), which nominally has $r = d$. In dimensions $d \leq 100$, our algorithm found representations of the correct form (Eq. **12**), which was unknown to us at the time of the experiment. It did not, however, find truly optimal representations and occasionally encountered badly conditioned matrices (Eq. **22**).

Table 1. Run times in seconds for a sign iteration in $d = 2$

	Ordinary		Separated	
N	Dense	Sparse	Dense	Sparse
8	$1.4 \cdot 10^0$	$1.4 \cdot 10^0$	$2.2 \cdot 10^0$	$4.4 \cdot 10^0$
16	$9.8 \cdot 10^1$	$7.9 \cdot 10^1$	$1.3 \cdot 10^1$	$2.3 \cdot 10^1$
32	$2.9 \cdot 10^4$	$3.1 \cdot 10^2$	$9.2 \cdot 10^1$	$7.7 \cdot 10^1$
64	(est.)$\cdot 10^6$	$2.2 \cdot 10^3$	$7.3 \cdot 10^2$	$2.3 \cdot 10^2$
128	(est.)$\cdot 10^8$	(est.)$\cdot 10^4$	$4.9 \cdot 10^3$	$7.8 \cdot 10^2$
256	(est.)$\cdot 10^{10}$	(est.)$\cdot 10^5$	(est.)$\cdot 10^4$	$3.0 \cdot 10^3$
Observed order	N^6	N^3	N^3	$N^{1.5}$

Computation of a Spectral Projector. For a diagonalizable matrix $\mathbb{A}$ with real eigenvalues, we can compute $\text{sign}(\mathbb{A})$ using the following recursion:

$$\mathbb{A}_0 = \mathbb{A}/\|\mathbb{A}\|_2$$
$$\mathbb{A}_{k+1} = (3\mathbb{A}_k - \mathbb{A}_k^3)/2, \quad k = 0, 1, \ldots \qquad [25]$$

Using the sign function, we can compute spectral projectors (10), which are useful for computing electron densities, in some wave-propagation problems and in model reduction.

We now give an example of the computation of a sign function. Here we test the principle of using a separated representation to compute functions of matrices. We compare the run times using the ordinary representation with dense matrices and sparse matrices in wavelet coordinates and the separated representation with dense and sparse one-dimensional matrices $\mathbb{V}_i^l$.

We compute $\text{sign}(\mathcal{H} - 50\mathcal{I})$ for the Hamiltonian

$$\mathcal{H} = -\Delta + 20\cos(x_1^2 + x_2^2), \qquad [26]$$

on a periodic domain in dimension $d = 2$. This Hamiltonian is a simplified single-atom independent-electron model for a crystal. The shift 50 separates out the five eigenfunctions with lowest eigenvalues. Even in this dimension the computation is very expensive. The sign function does not have an explicit solution and cannot be simplified by separation of variables. We perform 30 iterations (Eq. **25**) and truncate the sparse matrices at 10^{-5} relative precision. Table 1 gives run times on a Sun Ultra2 with a 300-MHz processor, thus demonstrating that the separated representation removes the dimension $d = 2$ from the exponent.

Computation of the Smallest Eigenvalue via the Power Method in Dimension 30

For a diagonalizable matrix $\mathbb{A}$, we can compute the spectral radius $\rho(\mathbb{A})$ with the power method. Beginning with a random vector $\mathbf{F}_0$, we compute

$$\mathbf{G}_k = \mathbb{A}\mathbf{F}_k$$
$$\mathbf{F}_{k+1} = \mathbf{G}_k/\|\mathbf{G}_k\|, \quad k = 0, 1, \ldots. \qquad [27]$$

The norm $\|\mathbf{G}_k\|$ will converge to $\rho(\mathbb{A})$, and if the corresponding eigenvalue is distinct, then $\mathbf{F}_k$ is the eigenvector.

For the operator, we choose the Hamiltonian

$$\mathcal{H} = -\Delta + 2d \sum_{i=1}^{d} \cos(x_i) + \sum_{i=1}^{d} \sum_{k>i}^{d} \cos(x_i - x_k), \qquad [28]$$

which is a simplified single-atom interacting d-electron model. We discretize $\mathcal{H}$ to form the matrix $\mathbb{H}$. Since the smallest eigenvalue of $\mathbb{H}$ corresponds to the lowest energy state, we apply the power method to $\mathbb{A} = C_d\mathbb{I} - \mathbb{H}$ for a suitable shift $C_d \approx \|\mathbb{H}\|/2$ to determine that eigenvalue. Usually the power method

Table 2. Computation of the smallest eigenvalue for a model Schrödinger operator in dimension 30

ε	Iterations	$r(\mathbf{F}_k)$	Time, sec
10^{-2}	521	1	$2.7 \cdot 10^1$
10^{-3}	2,557	2	$1.4 \cdot 10^2$
10^{-4}	4,130	5	$1.5 \cdot 10^3$
10^{-5}	5,230	6	$6.4 \cdot 10^3$
10^{-6}	6,160	11	$2.0 \cdot 10^4$
10^{-7}	6,368	12	$2.2 \cdot 10^4$

is not used to compute the smallest eigenvalue, in particular due to the large number of iterations required. However, it provides a simple demonstration of the ability to compute in higher dimensions.

The initial vector F_0 is chosen with separation rank one. After each iteration, we reduce the separation rank of $\mathbf{F}_k$ using $\mathbf{F}_{k-1}$ as the initial approximation in the alternating least-squares algorithm. For small values of k, $\mathbf{F}_k$ does not have the properties of the target eigenvector, and so it may have large separation rank. To prevent the power method from slowing down, we use an adaptively changing accuracy ε_k.

For this example we choose dimension $d = 30$ and a one-directional discretization with $N = 20$. These choices make $\|\mathbb{H}\| \approx 8 \cdot 10^4$, so we choose $C_d = 5 \cdot 10^4$. The matrix $\mathbb{H}$ has apparent separation rank $2d + d(d - 1) = 930$ using trigonometric identities, but we represent it with separation rank $r = 22$ using the derivative formulation in Eq. **12**, accurate to relative precision $\varepsilon = 10^{-7}$ in the operator norm. We iterate until the norm $\|\mathbf{G}_k\|$ has converged within 10^{-7} relative accuracy and use this as the correct value for $\rho(\mathbb{A})$. We then examine earlier iterates that are accurate to within ε, for various values of ε. In Table 2 we vary ε and present the number of iterations needed to obtain that precision, the separation rank of $\mathbf{F}_k$, and the run time in seconds on a Sun Ultra10 with a 360-MHz processor.

Although we did not enforce antisymmetry in this example, it is possible to guide the iteration to a fermionic rather than bosonic eigenspace. We plan to perform computations with the full Schrödinger operator, including the antisymmetry condition, and will present results as they become available.

1. Bellman, R. (1961) *Adaptive Control Processes: A Guided Tour* (Princeton Univ. Press, Princeton).
2. Yarkony, D. R., ed. (1995) *Modern Electronic Structure Theory, Advanced Series in Physical Chemistry* (World Scientific, River Edge, NJ).
3. Temlyakov, V. N. (1989) *Proc. Steklov Inst. Math.* **181,** 275–293.
4. Bittner, K. (1999) *SIAM J. Sci. Comput.* **20,** 1192–1213.
5. Strömberg, J.-O. (1998) *Doc. Math. Extra* **III,** 523–532.
6. De Lathauwer, L., De Moor, B. & Vandewalle, J. (2000) *SIAM J. Matrix Anal. Appl.* **21,** 1253–1278.
7. Kolda, T. G. (2001) *SIAM J. Matrix Anal. Appl.* **23,** 243–255.
8. Kroonenberg, P. M. & de Leeuw, J. (1980) *Psychometrika* **45,** 69–97.
9. Yarvin, N. & Rokhlin, V. (1999) *SIAM J. Sci. Comput.* **20,** 699–718.
10. Beylkin, G., Coult, N. & Mohlenkamp, M. J. (1999) *J. Comput. Phys.* **152,** 32–54.
11. Leurgans, S. E., Moyeed, R. A. & Silverman, B. W. (1993) *J. R. Stat. Soc. Ser. B* **55,** 725–740.
12. Cardoso, J.-F. (1999) *Neural Comput.* **11,** 157–192.

APPLIED MATHEMATICS