Toward Solving the Multiparticle Schrödinger Equation via an Unconstrained Sum of Slater Determinants

Gregory Beylkin, Martin Mohlenkamp, and Fernando Pérez

University of Colorado at Boulder and Ohio University

Pacific Northwest National Laboratory
Richland, WA
January 25-26, 2007

Numerical algorithms based on separated representations

We use separated multiresolution representations of operators and functions to solve problems in high dimensions

Last summer we started testing multiresolution multiparticle algorithms and report here some initial results

---

Our goals include the development of

- a new approach to multiparticle quantum computations that is fully adaptive and algorithmically size-extensive
- new tools for adaptive computations within two-particle theories in quantum chemistry

There is no "low hanging fruit" when computing in spaces of high dimension

Multiparticle Computations in Quantum Mechanics

CI (configuration interaction), CC (coupled cluster), MCSCF (multi-configuration self-consistent field) etc.: problems with computational complexity and accuracy control.

Monte Carlo algorithms: expensive for high accuracy and resolution.

We solve the multiparticle Schrödinger equation using:

- Nonlinear Approximations: unconstrained representation of the wavefunction
- Adaptive multiresolution application of operators in 3D
- Adaptive accuracy control

Multiparticle Schrödinger equation

Consider the non-relativistic Hamiltonian of a system with K nuclei and N_e electrons,

$$\mathcal{H} = \sum_{j=1}^{N_e}(-\frac{1}{2}\nabla_j^2 - \sum_{k=1}^{K}\frac{Z_k}{||\mathbf{x}_j - \mathbf{R}_k||}) + \sum_{i>j}\frac{1}{||\mathbf{x}_i - \mathbf{x}_j||}.$$

The ultimate goal is to solve the eigenvalue problem $\mathcal{H}\psi = E\psi$ without resorting to a reduced model.

We need numerical methods to transition from one particle theories (e.g. Kohn-Sham, LDA, HF) to a multiparticle approach.

The "curse of dimensionality", the requirement of antisymmetry and size extensivity are just several of many issues that need to be resolved.

Algorithmic road map

We have developed algorithms to work with nonlinear approximations and integral equations:

- Separated representations for Green's functions via Gaussians
- Reduction of the separation rank, alternating least-squares
- Combining antisymmetry and separated representations
- Incorporation of the electron-electron interaction within our formalism
- Adaptive application of operators in 3D

The Separated Representation

The standard separation of variables: $f(x_1, x_2, \ldots, x_d) = \phi_1(x_1) \cdot \phi_2(x_2) \cdot \cdots \cdot \phi_d(x_d)$

<u>*Definition*</u>: 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

$$\sum_{l=1}^{r} s_l \, A_1^l(j_1, j_1') \, A_2^l(j_2, j_2') \, \cdots A_d^l(j_d, j_d'),$$

where s_l is a scalar, $s_1 \geq \cdots \geq s_r > 0$, and $\mathbb{A}_i^l$ are matrices with entries $A_i^l(j_i, j_i')$ and norm one. We require:

$$||\mathbb{A} - \sum_{l=1}^{r} s_l \, \mathbb{A}_1^l \otimes \mathbb{A}_2^l \otimes \cdots \otimes \mathbb{A}_d^l|| \leq \epsilon.$$

We call the scalars s_l separation values and the rank r the separation rank.

Separated representations are not unique. If $d \geq 3$ then the analogy with SVD breaks down: by changing ϵ we change all terms rather than add/delete terms

Example: a trigonometric identity

Consider the function $\sin(x_1 + x_2 + \cdots + x_d)$ and represent it in the separated form. The usual trigonometric formulas produce 2^{d-1} terms. We have

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

for all choices of $\alpha_1, \alpha_2, \ldots, \alpha_d$, such that $\sin(\alpha_k - \alpha_j) \neq 0$ for all $j \neq k$.

Observations: effective, not unique, may be ill-conditioned

We have developed an alternating least-squares algorithm for reducing the separation rank while controlling the condition number

Nonlinear approximations

Bases provide an approximation of a function f on a subspace spanned, for example, by n elements of a Hilbert space $\mathcal{H}$. The sum of two n-term approximations (on the same subspace) is a new n-term approximation.

By nonlinear approximation we refer to using *any* n elements of $\mathcal{H}$ to find the best (say in L^∞ norm) n-term approximation.

In this case, the addition of two n-term approximations does not (typically) provide an n-term approximation and, thus, requires constructing a new n-term approximation.

In general nonlinear approximations are far superior to linear approximations but require robust algorithms that are not so easy to construct

Note that early quantum chemistry used nonlinear approximations (see Boys, Singer, Longstaff, 1960), which later were replaced by linear approximations via orbitals

Antisymmetry

Since electrons are fermions, the wave function must be antisymmetric, e.g., $\psi(\gamma_2, \gamma_1, \gamma_3, \ldots) = -\psi(\gamma_1, \gamma_2, \gamma_3, \ldots)$, where $\gamma = ((x, y, z), \sigma)$ and σ is the spin.

For functions of N variables, the "antisymmetrizer" is defined by

$$\mathcal{A} = \frac{1}{N!} \sum_{p \in S_N} (-1)^p \mathcal{P},$$

where S_N is the permutation group on N elements. If $\mathcal{A}$ is applied to a separable function, then the result can be expressed as a Slater determinant,

$$\mathcal{A} \prod_{j=1}^{N} \phi_j(\gamma_j) = \frac{1}{N!} \begin{vmatrix} \phi_1(\gamma_1) & \phi_1(\gamma_2) & \cdots & \phi_1(\gamma_N) \\ \phi_2(\gamma_1) & \phi_2(\gamma_2) & \cdots & \phi_2(\gamma_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\gamma_1) & \phi_N(\gamma_2) & \cdots & \phi_N(\gamma_N) \end{vmatrix}.$$

Weak formulation

The number of terms in $\mathcal{A}\prod_{j=1}^{N}\phi_j(\gamma_j)$ grows exponentially fast (although this number can algebraically be reduced somewhat).

However, if we care only about computing inner products with $\mathcal{A}\prod_{j=1}^{N}\phi_j(\gamma_j)$, then the so-called Löwdin rules provide a solution,

$$\langle \mathcal{A}\prod_{j=1}^{N}\phi_j(\gamma_j), \mathcal{A}\prod_{j=1}^{N}\tilde{\phi}_j(\gamma_j)\rangle = \frac{1}{N!}\begin{vmatrix} \langle\phi_1,\tilde{\phi}_1\rangle & \langle\phi_1,\tilde{\phi}_2\rangle & \cdots & \langle\phi_1,\tilde{\phi}_N\rangle \\ \langle\phi_2,\tilde{\phi}_1\rangle & \langle\phi_2,\tilde{\phi}_2\rangle & \cdots & \langle\phi_2,\tilde{\phi}_N\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\phi_N,\tilde{\phi}_1\rangle & \langle\phi_N,\tilde{\phi}_2\rangle & \cdots & \langle\phi_N,\tilde{\phi}_N\rangle \end{vmatrix}.$$

Computing determinant of a matrix with scalar entries costs at most $\mathcal{O}(N^3)$.

For large N the matrix is banded and the cost is $\mathcal{O}(N)$.

The Wavefunction as an Unconstrained Sum of Slater Determinants

We consider approximations to the wavefunction of the form

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

We impose no constraints (such as orthogonality). Without constraints, we can have

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

where $\{\phi_j\}_{j=1}^{2N}$ form an orthonormal set. Methods that constrain the factors to come from an orthogonal set are forced to multiply out the second term and, thus, produce 2^N terms.

Basic Green's Function Iteration

The Hamiltonian $\mathcal{H} = \mathcal{T} + \mathcal{V} + \mathcal{W}$ =kinetic + nuclear + electronic parts, so

$$\mathcal{H}\psi = (\mathcal{T} + \mathcal{V} + \mathcal{W})\psi = E\psi \Leftrightarrow (\mathcal{T} - E\mathcal{I})\psi = -(\mathcal{V} + \mathcal{W})\psi$$
$$\Leftrightarrow \psi = -(\mathcal{T} - E\mathcal{I})^{-1}(\mathcal{V} + \mathcal{W})\psi$$

Define the Green's function

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

for $\mu < 0$. The Green's function iteration (*a la* Kalos) is

$$\begin{aligned}
\tilde{\psi}_n &= -\mathcal{G}_{\mu_n}[(\mathcal{V} + \mathcal{W})\psi_n] \\
\mu_{n+1} &= \mu_n - \langle(\mathcal{V} + \mathcal{W})\psi_n, \psi_n - \tilde{\psi}_n\rangle / \|\tilde{\psi}_n\|^2 \\
\psi_{n+1} &= \tilde{\psi}_n / \|\tilde{\psi}_n\|
\end{aligned}$$

and has $\mu_n \longrightarrow E$

A Modified Green's Function Iteration

Since computing

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

does not preserve the separation rank, we define $\tilde{\psi}_n$ to be of the same separation rank as ψ_n and minimizing

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

In order to assure convergence to an antisymmetric solution, we use the pseudo-norm induced by the pseudo inner product

$$\langle \cdot, \cdot \rangle_{\mathcal{A}} = \langle \mathcal{A}(\cdot), \mathcal{A}(\cdot) \rangle.$$

This makes non-antisymmetric subspaces "invisible".

Alternating Least Squares Fitting

To find $\tilde{\psi}_n$ we loop through the directions, optimizing the functions in that coordinate while fixing the functions in other coordinates.

For each direction the minimization is a linear problem, solving the normal equations

$$\mathbb{A}\mathbf{x} = \mathbf{b}\,.$$

Rather than optimize the coefficients in some basis, we optimize the point values, resulting in a linear system of integral equations. The entries in $\mathbb{A}$ are *integral operators* and the entries in $\mathbf{x}$ and $\mathbf{b}$ are functions.

Construction of the System of Integral Equations

For direction $i = 1$, the kernels in $\mathbb{A}$ are defined by

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

where $\delta(\cdot)$ is the delta function.

The functions in $\mathbf{b}$ are defined by

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

Details in the upcoming paper.

Antisymmetric Inner Products

Let $\mathbf{\Phi}$ denote the column vector containing the functions in the product $\prod_{i=1}^{N} \phi_i(\gamma_i)$.
Define the matrix $\mathbb{L} = \mathbb{L}(\tilde{\mathbf{\Phi}}, \mathbf{\Phi})$ with entries

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

A simple derivation gives us Löwdin's rule, that

$$\langle \tilde{\mathbf{\Phi}}, \mathbf{\Phi} \rangle_{\mathcal{A}} = \frac{|\mathbb{L}|}{N!}.$$

Entries in $\mathbb{A}$ are computed via Löwdin's rule (and some trickery with the delta functions).

Expansion of the Green's Function

Using a separated representation $\mathcal{G}_\mu \approx \sum_{p=1}^{L} \bigotimes_{i=1}^{N} \mathcal{F}^p_{\mathbf{r}_i}$ we can apply $\mathcal{G}_\mu$ via

$$\left\langle \tilde{\mathbf{\Phi}}, \mathcal{G}_\mu \left[\mathcal{V} + \mathcal{W}\right] \mathbf{\Phi} \right\rangle_{\mathcal{A}} \approx \sum_{p=1}^{L} \left\langle \mathcal{F}^p \tilde{\mathbf{\Phi}}, [\mathcal{V} + \mathcal{W}] \mathbf{\Phi} \right\rangle_{\mathcal{A}} .$$

To construct this approximation, we discretize an integral representation of its kernel

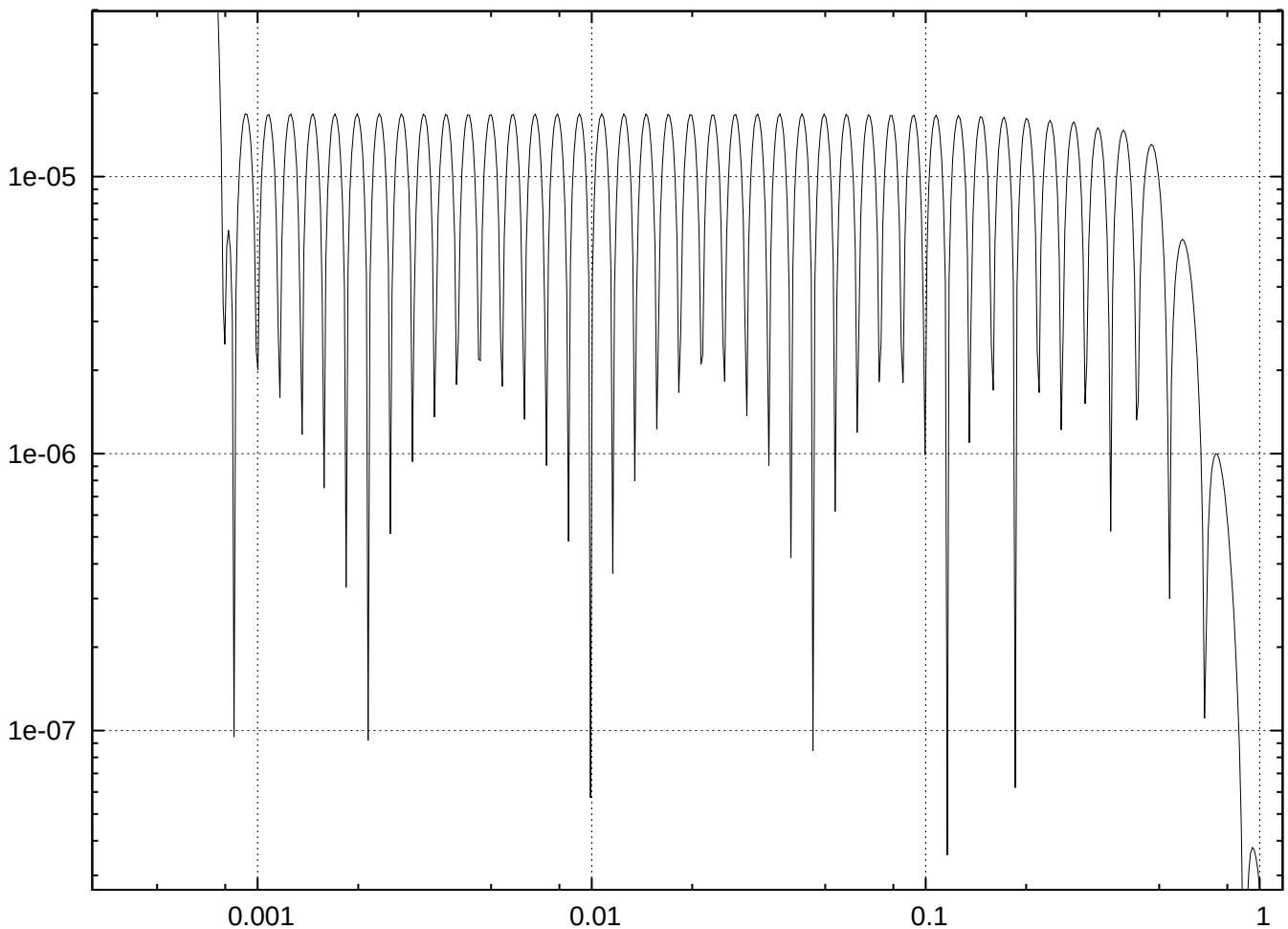
$$G_N(\rho, \mu) = \pi^{-\frac{3N}{2}} (\frac{\mu}{2\rho})^{\frac{3N}{2}-1} \mathrm{K}_{\frac{3N}{2}-1}(\mu\rho) = \pi^{-\frac{3N}{2}} \int_{\mathbb{R}} e^{-\rho^2 e^{2s} - \frac{2\mu}{4} e^{-2s} + (3N-2)s} ds,$$

and obtain τ_p and w_p (depending on μ) such that $\mathcal{F}^p_{\mathbf{r}_i}$ is the convolution defined by

$$\mathcal{F}^p_{\mathbf{r}_i} f(\gamma_1, \ldots, \gamma_N) = w_p \int \exp\left(-\tau_p \|\mathbf{r}_i - \mathbf{r}'\|^2\right) f(\gamma_1, \ldots, \gamma_{i-1}, (\mathbf{r}', \sigma_i), \ldots, \gamma_N) d\mathbf{r}' .$$

Two-particle Green's function for accuracy $\approx 10^{-4}$

The relative error of approximating $G_2(\rho, \mu)$ by a Gaussian sum with 24 terms,



Antisymmetric Inner Products with Operators

Defining

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

and denoting convolution with the Poisson kernel by

$$\mathcal{W}[f](\mathbf{r}) = \int \frac{1}{\|\mathbf{r}-\mathbf{r}'\|} f(\gamma')d\gamma' \,,$$

we can generalize Löwdin's rules to obtain

$$\left\langle \tilde{\mathbf{\Phi}}, \mathcal{W}\mathbf{\Phi} \right\rangle_{\mathcal{A}} = \frac{1}{2}\frac{|\mathbb{L}|}{N!}\int \mathbf{\Phi}^*\mathbf{\Theta}\mathcal{W}[\mathbf{\Phi}^*\mathbf{\Theta}] - \mathbf{\Phi}^*\mathcal{W}[\mathbf{\Theta}\mathbf{\Phi}^*]\mathbf{\Theta}d\gamma$$

$$\left\langle \tilde{\mathbf{\Phi}}, \mathcal{V}\mathbf{\Phi} \right\rangle_{\mathcal{A}} = \frac{|\mathbb{L}|}{N!}\int V(\mathbf{r})\mathbf{\Phi}^*\mathbf{\Theta}d\gamma \,.$$

Entries in $\mathbf{b}$ are computed using these rules, the expansion of $\mathcal{G}_\mu$, and some trickery with the delta functions.

Adaptive numerical algorithms in 3D

Adaptive numerical algorithms in 3D

- We have developed a fast (multiresolution, adaptive) algorithm for applying in 3D:

$$G_0^\alpha(\mu) = (-\frac{1}{2}\nabla^2 + \frac{1}{2}\mu^2\mathcal{I})^{-\alpha},$$

where μ is real and $0 < \alpha < 3/2$. The kernel of G_0^α is the radial function

$$\mathcal{G}_0^\alpha(r) = \frac{2^{-\frac{1}{2}}}{\Gamma(\alpha)\pi^{\frac{3}{2}}}(\frac{\mu}{r})^{\frac{3}{2}-\alpha}\mathrm{K}_{\frac{3}{2}-\alpha}(\mu r),$$

where K is the modified Bessel function.

- Adaptive multiresolution pointwise multiplication of functions

- Adaptive multiresolution inner product

Example: an adaptive multiresolution Poisson solver in 3D

Example: $\nabla^2\phi(\mathbf{r}) = -f(\mathbf{r}),\ f(\mathbf{r};\alpha) = \sum_{i=1}^{3}(6\alpha 4\alpha^2 r_i^2 - 6\alpha)e^{-\alpha r_i^2}, \quad \alpha = 300.$

We compute

$$\phi(\mathbf{r}) = \frac{1}{4\pi}\int_{\mathbb{R}^3}\frac{f(\mathbf{r}')}{||\mathbf{r}-\mathbf{r}'||}d\mathbf{r}' = \frac{1}{4\pi}\mathcal{W}[f].$$

Platform: Pentium 4 2.8 GHz

Timings:

Requested error	Resulting ϵ_{L^2}	Basis order	Application time (s)	MFLOPS
10^{-4}	2.3×10^{-5}	12	2.1	1770
10^{-6}	8.4×10^{-7}	12	12	1670
10^{-8}	2.0×10^{-9}	14	33	1880

Performance of the Poisson solver in 3D

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	$8.0 \cdot 10^4$

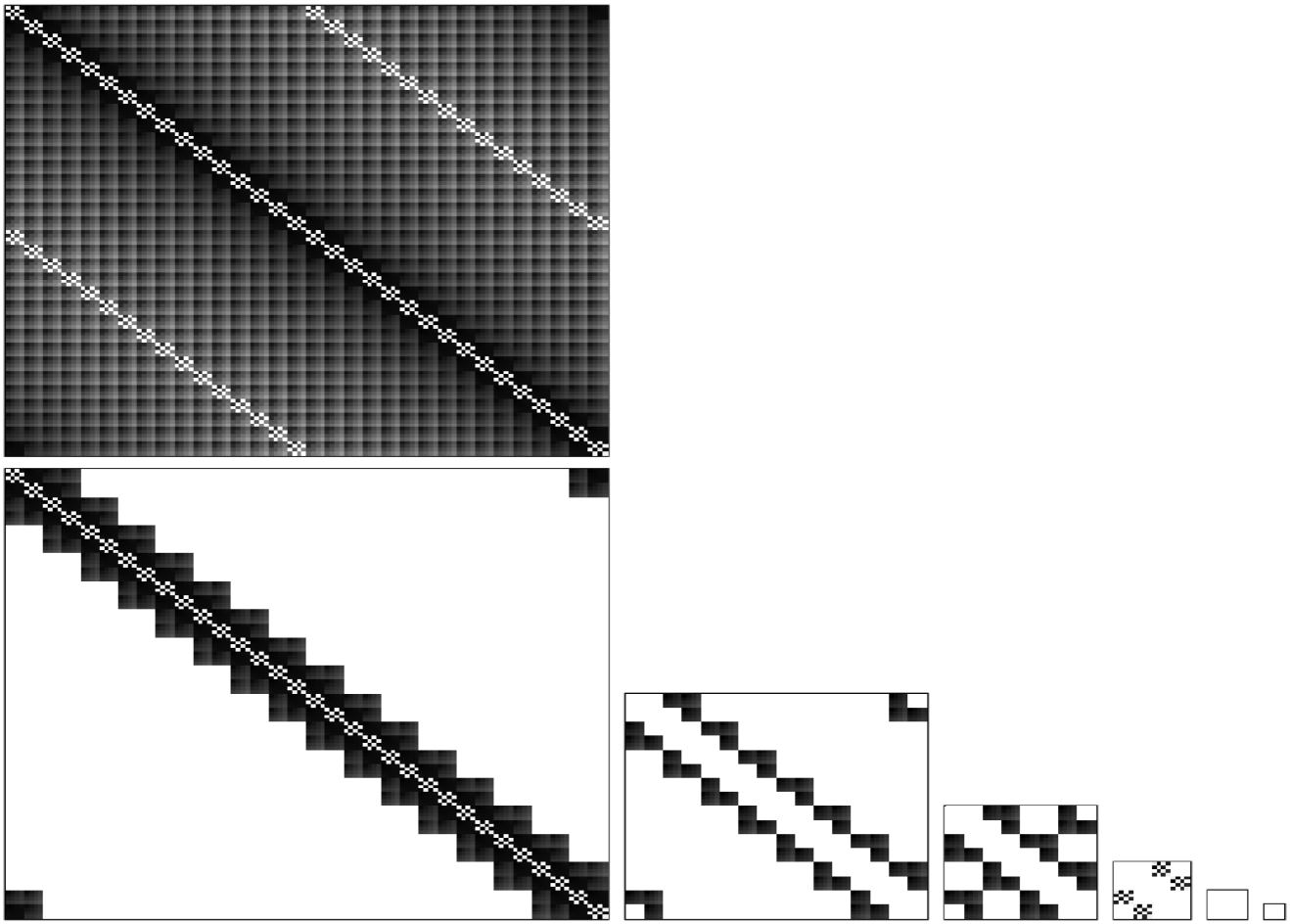
Requested $\epsilon = 10^{-9}$

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 1: Accuracy and timings for the adaptive solution of the Poisson equation

Multiresolution: the core of the matter

Matrix of $Cot[\pi(x-y)]$ and its multiresolution version



Features of the Algorithm for Wavefunction Approximation

The integral equations formulation means we do not have to work in a fixed basis, but can adapt as necessary. We use adaptive polynomial multiwavelets, which also allow us to compute $\mathcal{W}[\cdot]$ efficiently.

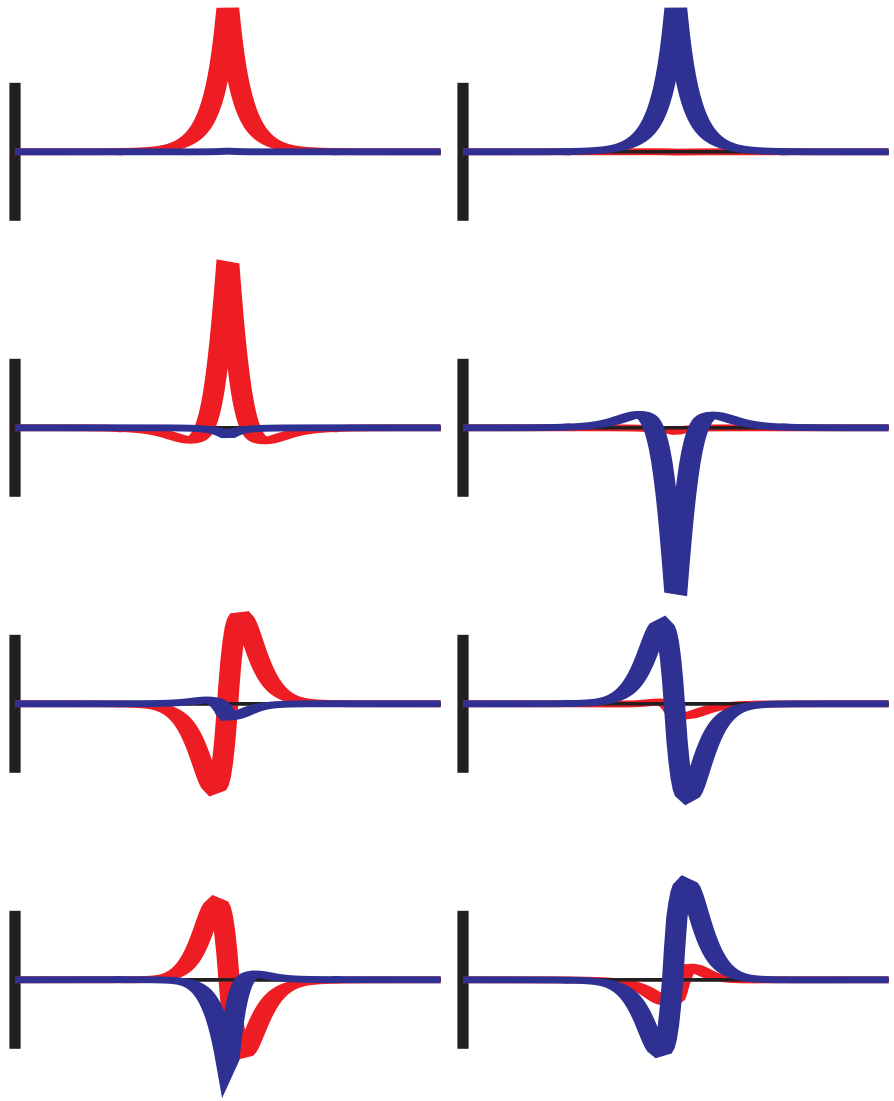
With respect to the number of terms r, the number of electrons N, and the cost M to represent a function of γ, the computational cost of algorithms is

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

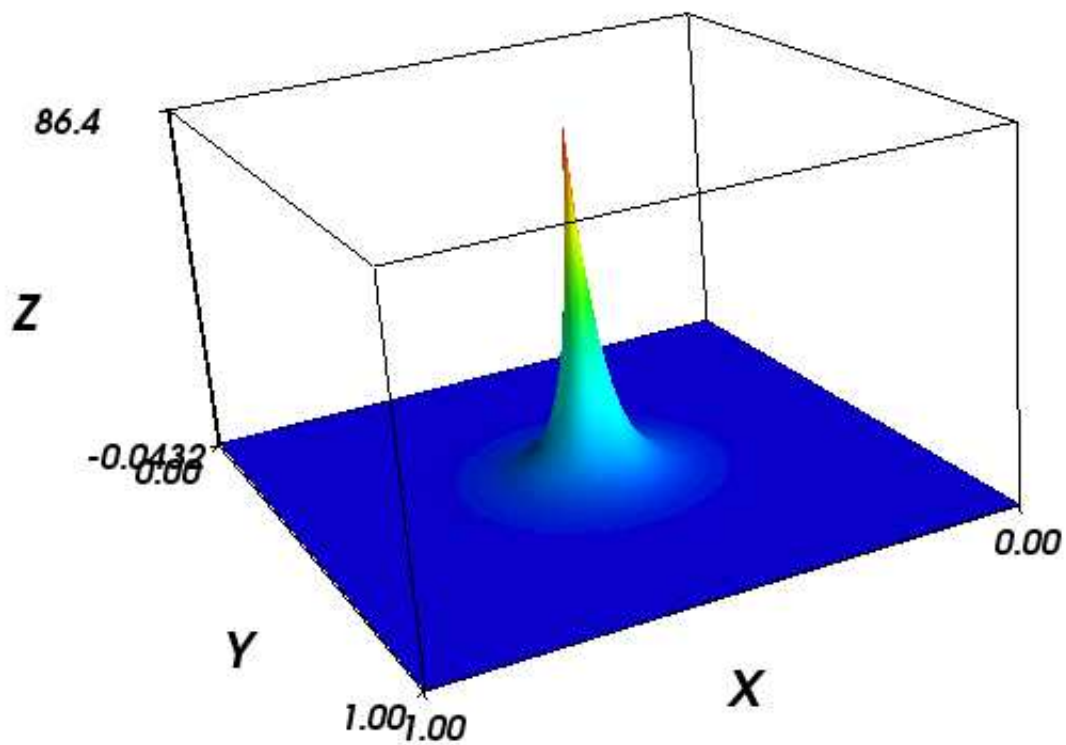
For comparison, the cost to evaluate a single instance of Löwdin's rules is

$$\mathcal{O}(N^2 (N + M)).$$

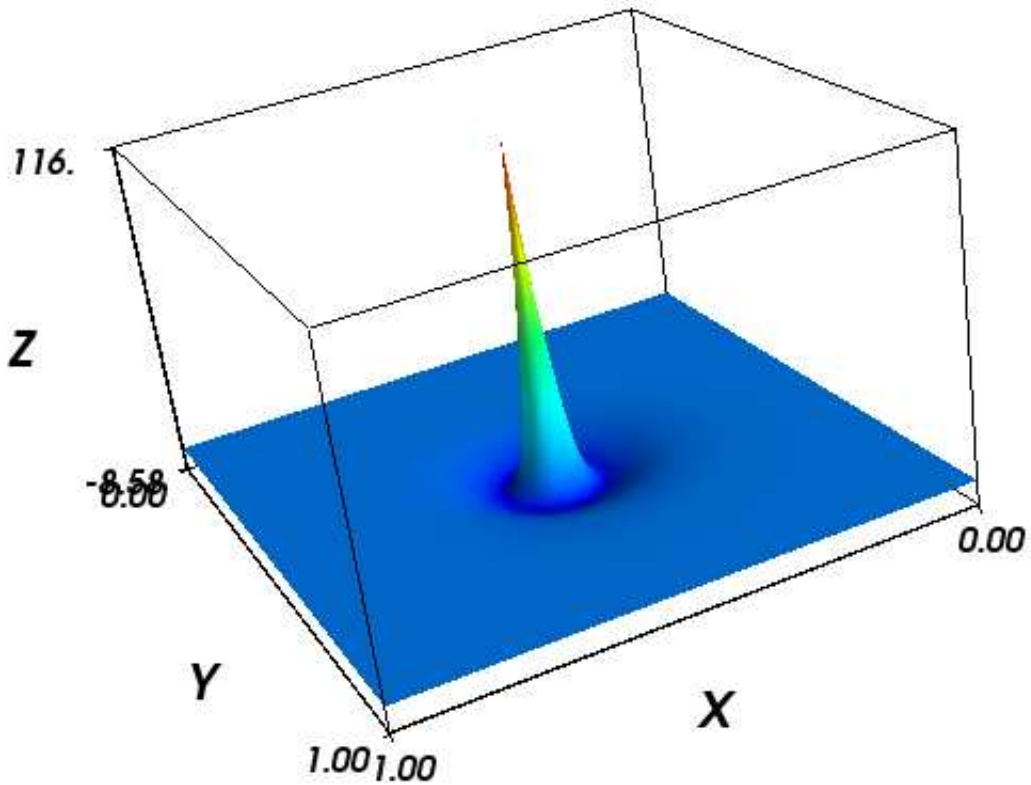
Computed slices of a He wavefunction with $r = 4$



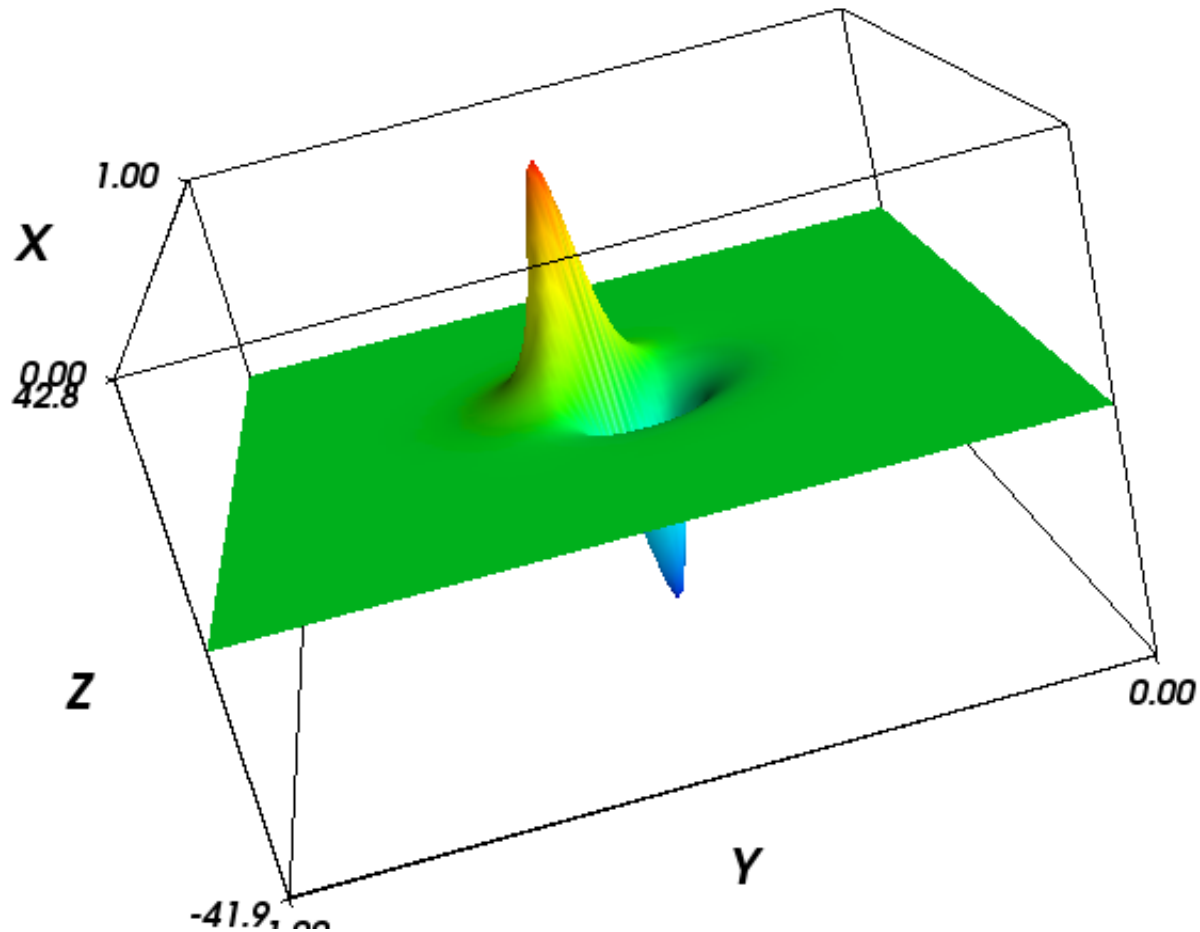
Computed $\phi_1^1(\cdot,\cdot,0;-1/2)$, $s_1 = 0.99725$



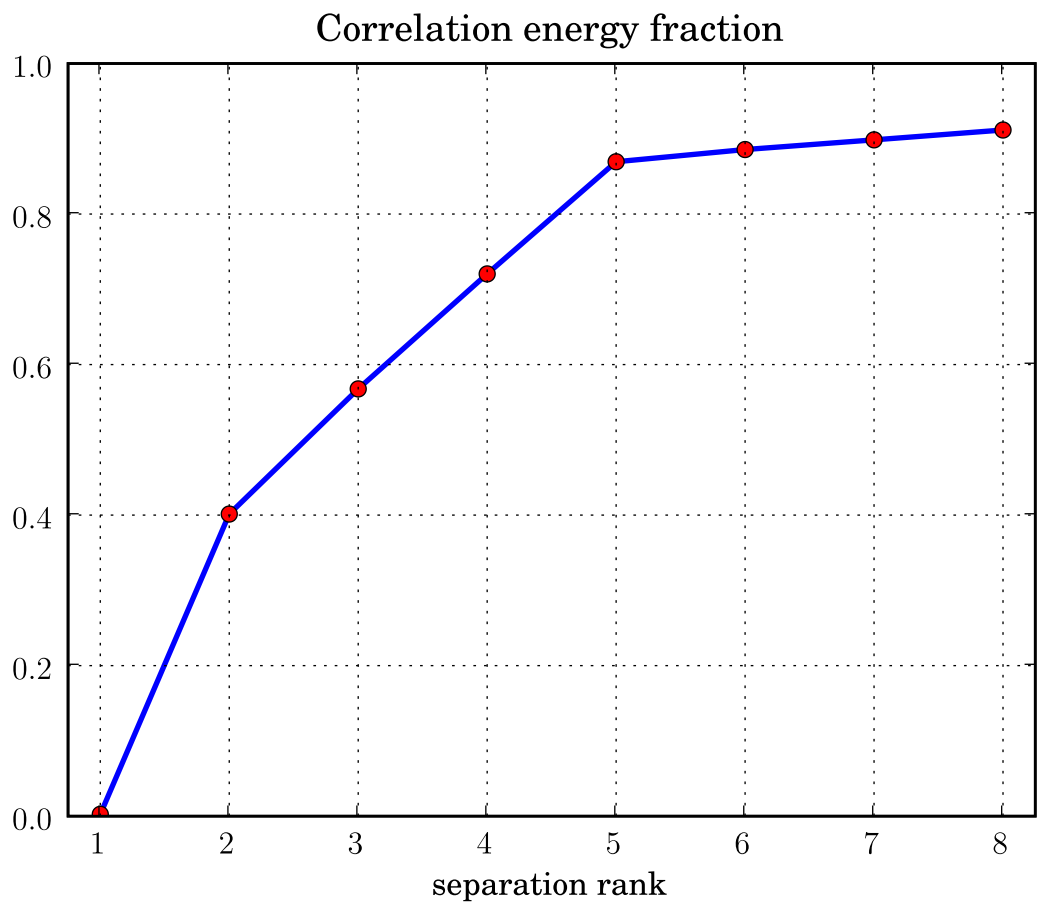
Computed $\phi_1^2(\cdot,\cdot,0;-1/2)$, $s_2 = 0.06716$



Computed $\phi_1^3(\cdot,\cdot,0;-1/2)$, $s_3 = 0.03827$



Computation of the ground state of the Helium atom



$$\frac{E_{comp} - E_{HF}}{E_{exact} - E_{HF}}$$

Summary

We implemented a prototype of a fully adaptive, input free, multiparticle solver.

Our formalism is shown to be correct and, so far, accurate wavefunctions appear to require a relatively small separation rank

Currently we are testing and debugging the code using He and LiH

As might be expected, the prototype version is too slow and we work toward:

- Improving convergence: old ideas in a new environment
- Alternative adaptive representations
- Parallelization of alternating least squares
- Algorithmic size-extensivity

We made a number of interesting analytic observations that we will try to use in our formulation

Paper detailing our approach is in preparation