MADNESS

Multiresolution
Adaptive
Numerical
Scientific
Simulation

Multiresolution Adaptive Numerical Scientific Simulation

Robert J. Harrison[1], Joel Anderson[1], Bryan Sundahl, Hideo Sekino[1], George I. Fann[2], Gregory Beylkin[4], Lucas Monzon[3], Edward Valeev[4], Florian Bischoff[5], Jakob Kottmann[5]

[1]Stony Brook University, Brookhaven National Laboratory

[2]Oak Ridge National Laboratory

[3]University of Colorado

[4]Virginia Tech

[4]Humboldt-Universität zu Berlin

robert.harrison@stonybrook.edu

BROOKHAVEN
NATIONAL LABORATORY

National Science Foundation
WHERE DISCOVERIES BEGIN

Stony Brook University

Tools in the tool box: fast and accurate computation

George I. Fann[1], Diego Galindo[1], Robert J. Harrison[2], Scott Thornton[2], Joel Anderson[2], Byran Sundahl[2], Judy Hill[1], and Jun Jia[1]

[1]Oak Ridge National Laboratory

[2]Stony Brook University, Brookhaven National Laboratory

In collaboration with

Gregory Beylkin[4], Lucas Monzon[4], Hideo Sekino[5] and Edward Valeev[6]

[4]University of Colorado

[5]Toyohashi Technical University, Japan

[6]Virginia Tech

robert.harrison@gmail.com

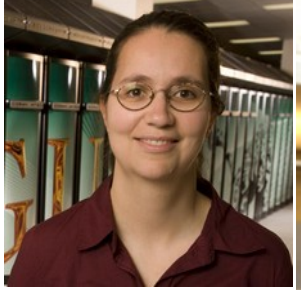
National Science Foundation
WHERE DISCOVERIES BEGIN

Stony Brook University

George Fann

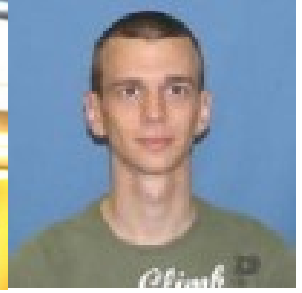
Judy Hill

Gregory Beylkin



Rebecca Hartman-Baker

Jeff Hammond



Joel Anderson

Ariana Beste

Eduard Valeyev

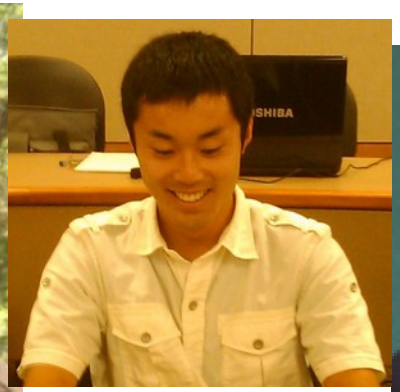
Alvaro Vasquez

Hideo Sekino

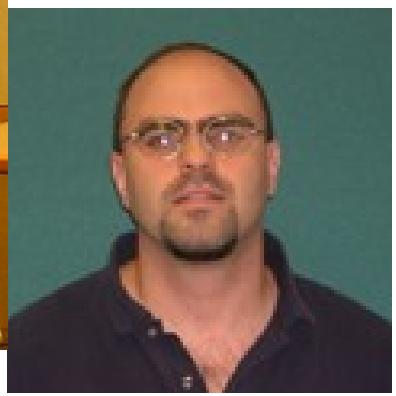
Robert Harrison

Bryan Sundahl

Nicholas Vence



Takahiro Ii



Scott Thornton

Matt Reuter

Nichols Romero

Jia, Kato, Niimura, Calvin, Pei,

Big picture

- Want robust algorithms that scale correctly with system size and are easy to write
- Robust, accurate, fast computation
  - Gaussian basis sets: high accuracy yields dense matrices and linear dependence – $O(N^3)$
  - Plane waves: force pseudo-potentials – $O(N^3)$
  - $O(N \log^m N \log^k \varepsilon)$ is possible with guaranteed ε
- Semantic gap
  - Why are our equations just O(100) lines but programs O(1M) lines?
- Facile path from laptop to exaflop

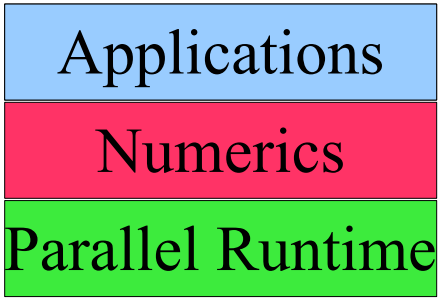
“Fast” algorithms

- Fast in mathematical sense
  - Optimal scaling of cost with accuracy & size
- Multigrid method – Brandt (1977)
  - Iterative solution of differential equations
  - Analyzes solution/error at different length scales
- Fast multipole method – Greengard, Rokhlin (1987)
  - Fast application of dense operators
  - Exploits smoothness of operators
- Multiresolution analysis
  - Exploits smoothness of operators and functions

What is MADNESS?

- A general purpose numerical environment for reliable and fast scientific simulation
  - Chemistry, nuclear physics, atomic physics, material science, nanoscience, climate, fusion, ...
- Want robust and fast algorithms that scale correctly with system size and are easy to write
- Semantic gap
  - Why are equations O(100) lines but codes O(1M) ?
- Facile path from laptop to exaflop

https://github.com/m-a-d-n-e-s-s/madness



Why MADNESS?

- Reduces S/W complexity
  - MATLAB-like level of composition of scientific problems with guaranteed speed and precision
  - Programmer not responsible for managing dependencies, scheduling, or placement
- Reduces numerical complexity
  - Solution of integral not differential equations
  - Framework makes latest techniques in applied math and physics available to wide audience

E.g., with guaranteed precision of 1e-6 form a numerical representation of a Gaussian in the cube $[-20,20]^3$, solve Poisson's equation, and plot the resulting potential (all running in parallel with threads+MPI)

Let

$$\Omega = [-20, 20]^3$$

$$\epsilon = 1e-6$$

$$g = x \rightarrow \exp\left(-\left(x_0^2 + x_1^2 + x_2^2\right)\right) * \pi^{-1.5}$$

In

$$f = \mathcal{F}\, g$$

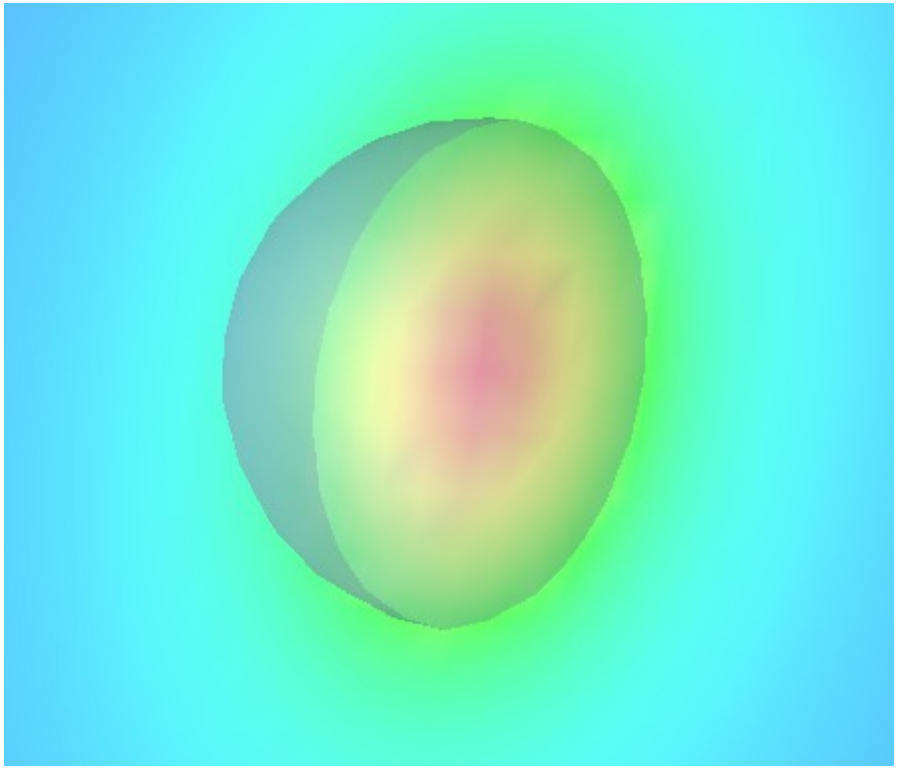
$$u = \nabla^{-2}\left(-4 * \pi * f\right)$$

print "norm of f", $\langle f \rangle$, "energy", $\langle f | u \rangle * 0.5$

plot u

End

output: norm of f 1.00000000e+00 energy 3.98920526e-01



There are only two lines doing real work. First the Gaussian (g) is projected into the adaptive basis to the default precision. Second, the Green's function is applied. The exact results are norm=1.0 and energy=0.3989422804.

Let

$$\Omega = [-20, 20]^3$$

$$r = x \to \sqrt{x_0^2 + x_1^2 + x_2^2}$$

$$g = x \to \exp(-2 * r(x))$$

$$v = x \to -\frac{2}{r(x)}$$

In

$$\nu = \mathcal{F}\, v$$

$$\phi = \mathcal{F}\, g$$

$$\lambda = -1.0$$

for $i \in [0, 10]$

$$\phi = \phi * \|\phi\|^{-1}$$

$$V = \nu - \nabla^{-2}\left(4 * \pi * \phi^2\right)$$

$$\psi = -2 * \left(-2 * \lambda - \nabla^2\right)^{-1}(V * \phi)$$

$$\lambda = \lambda + \frac{\langle V * \phi | \psi - \phi \rangle}{\langle \psi | \psi \rangle}$$

$$\phi = \psi$$

print "iter", i, "norm", $\|\phi\|$, "eval", λ

end

End

He atom Hartree-Fock

Compose directly in terms of functions and operators

This is a Latex rendering of a program to solve the Hartree-Fock equations for the helium atom

The compiler also outputs a C++ code that can be compiled without modification and run in parallel

The math behind the MADNESS

- Multiresolution

$$V_0 \subset V_1 \subset \cdots \subset V_n$$

$$V_n = V_0 + \left(V_1 - V_0\right) + \cdots + \left(V_n - V_{n-1}\right)$$

- Low-separation rank

$$f(x_1, \ldots, x_n) = \sum_{l=1}^{M} \sigma_l \prod_{i=1}^{d} f_i^{(l)}(x_i) + O(\epsilon)$$

$$\left\| f_i^{(l)} \right\|_2 = 1 \qquad \sigma_l > 0$$

- Low-operator rank

$$A = \sum_{\mu=1}^{r} u_\mu \, \sigma_\mu \, v_\mu^T + O(\epsilon)$$

$$\sigma_\mu > 0 \qquad v_\mu^T v_\lambda = u_\mu^T u_\lambda = \delta_{\mu\nu}$$

Why “think” multiresolution?

- It is everywhere in nature/chemistry/physics
  - Core/valence; high/low frequency; short/long range; smooth/non-smooth; atomic/nano/micro/macro scale
- Common to separate just two scales
  - E.g., core orbital heavily contracted, valence flexible
  - More efficient, compact, and numerically stable
- Multiresolution
  - Recursively separates all length/time scales
  - Computationally efficient and numerically stable
  - Coarse-scale models that capture fine-scale detail

How to "think" multiresolution

- Consider a ladder of function spaces

$$V_0 \subset V_1 \subset \cdots \subset V_n$$

  - E.g., increasing quality atomic basis sets, or finer resolution grids, …
- Telescoping series

$$V_n = V_0 + (V_1 - V_0) + (V_2 - V_1) + \cdots (V_n - V_{n-1})$$

  - Instead of using the most accurate representation, use the difference between successive approximations
  - Representation on V_0 small/dense; differences sparse
  - Computationally efficient; possible insights

Another Key Component

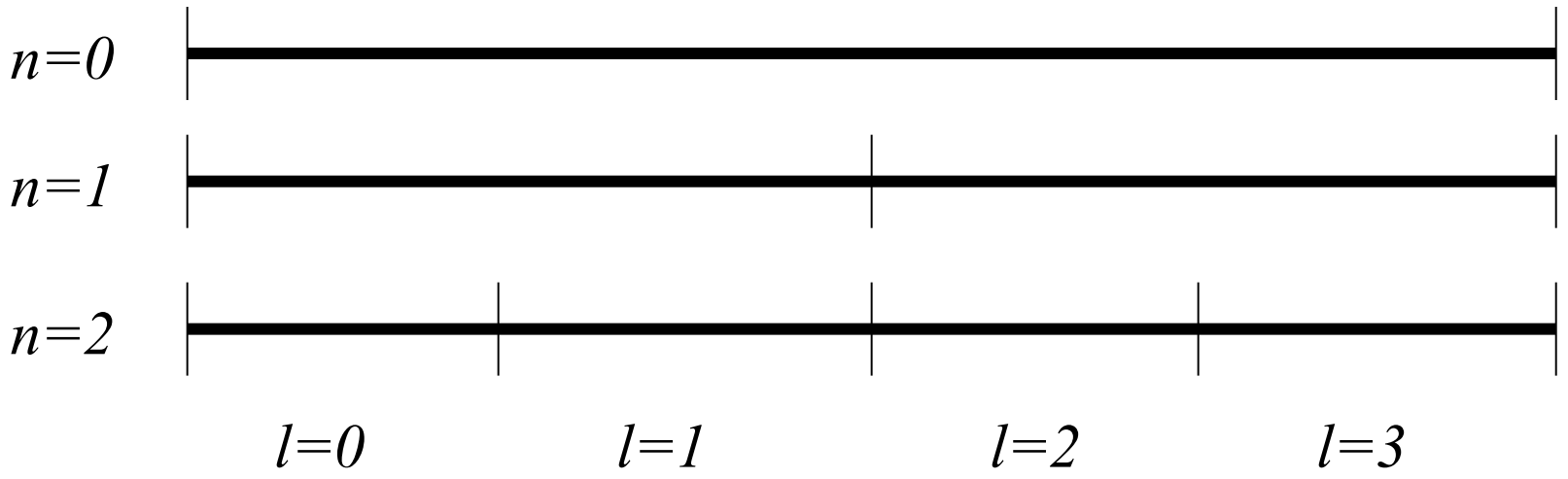
- Trade precision for speed – everywhere
  - Don't do anything exactly
  - Perform everything to $O(\varepsilon)$
  - Require
    - Robustness
    - Speed, and
    - Guaranteed, arbitrary, *finite* precision

Problem Setup

- In 1-D solve on [0,1]
- Tensor product in n-D
- Periodic and infinite domains also feasible

Scaling Function Basis - I

- Divide domain into 2^n pieces
  - Referred to as level n
  - Adaptive sub-division – local refinement
  - Non-uniform division possible (not yet done)
  - l^{th} sub-interval $[l*2^{-n},(l+1)*2^{-n}]$ $l=0,\ldots,n-1$



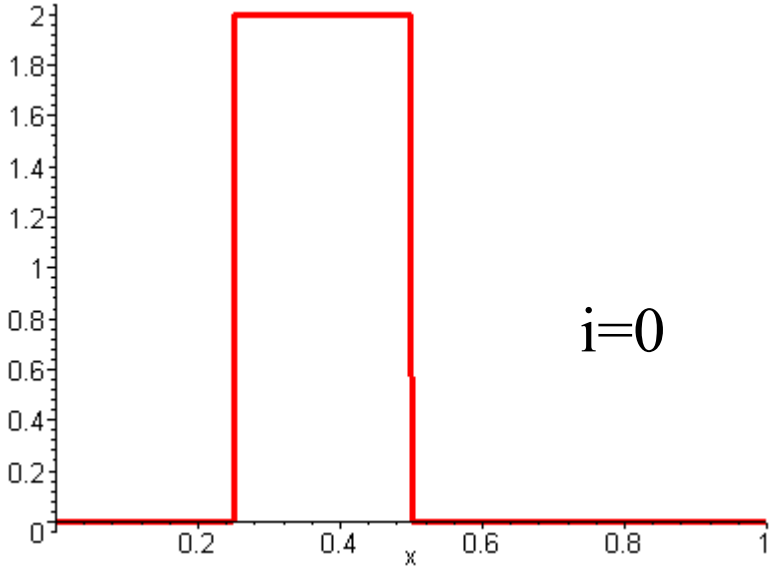
Scaling Function Basis - II

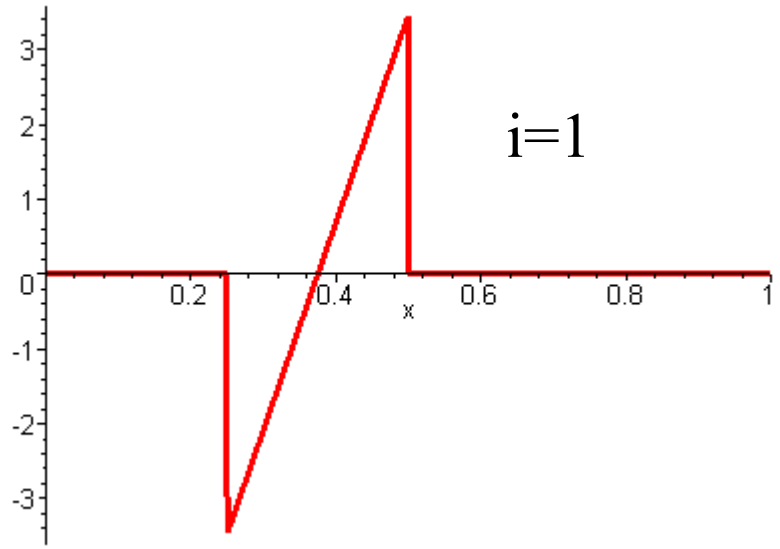
- In each sub-interval define a polynomial basis
  - Currently use the first *k* Legendre polynomials
  - Other bases interesting (e.g, interpolating polynomials, non-polynomial functions)
  - Zero outside of the sub-interval

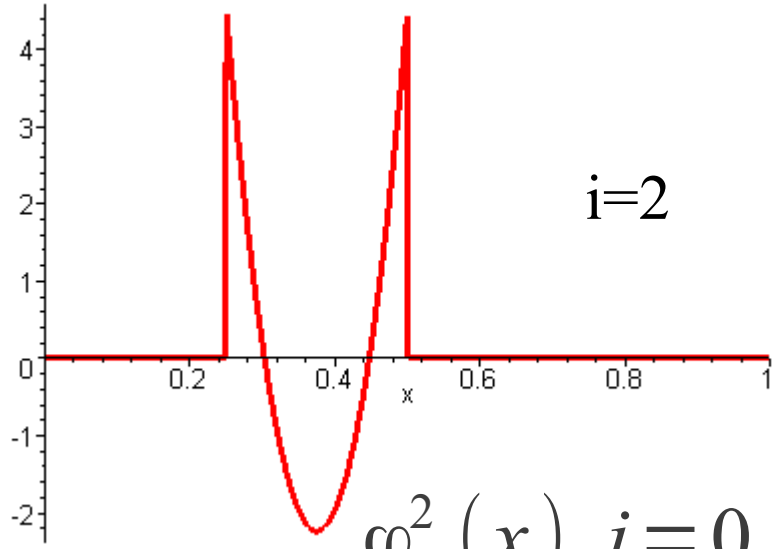
$$\phi_i(x) = \begin{cases} \sqrt{2i+1}\,P_i(2x-1) & 0 \\ 0 \le x \le 1 & \textit{otherwise} \end{cases}$$

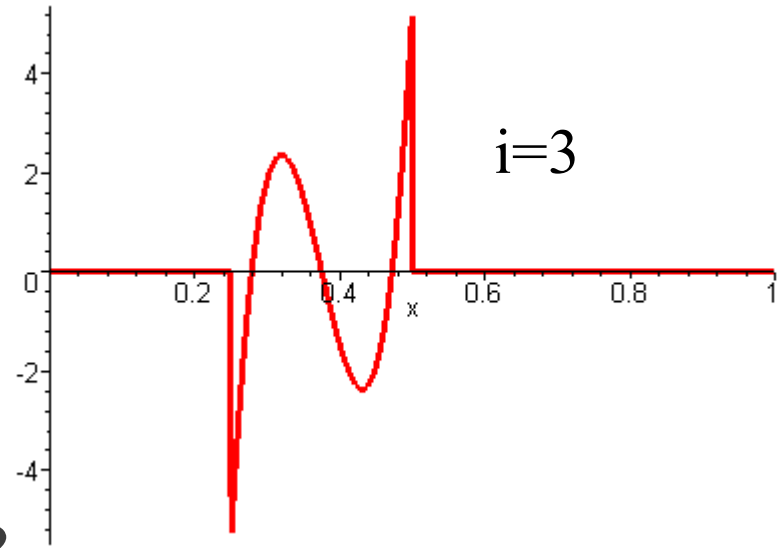
$$\phi_{i\,l}^{n} = 2^{n/2}\phi_i(2^n x - l)$$

Scaling Function Basis - III









$$\varphi_{i1}^{2}(x), i = 0, \ldots, 3$$

Scaling Function Basis - IV

- Translation and dilation $\phi_{il}^{n}(x) = 2^{n/2}\phi_{i}(2^{n}x - l)$
- Orthonormal $\int_{-\infty}^{\infty} \phi_{il}^{n}(x)\phi_{i'l'}^{n}(x)dx = \delta_{ll'}\delta_{ii'}$
- Ladder of spaces $V_0 \subset V_1 \subset \cdots \subset V_n$
- Complete basis for L_2 [0,1] in limit of *either* large *k* or large *n*.

Two-scale relationship

- $V_0 \subset V_1$ – can represent coarse scale basis exactly in the fine scale basis

$$\phi_i(x) = \sqrt{2} \sum_{j=0}^{k-1} \left(h_{ij}^{(0)} \phi_j(2x) + h_{ij}^{(1)} \phi_j(2x-1) \right)$$

$$\phi_{il}^n(x) = \sum_{j=0}^{k-1} \left(h_{ij}^{(0)} \phi_{j2l}^{n+1}(x) + h_{ij}^{(1)} \phi_{j2l+1}^{n+1}(x) \right)$$

- The filter coefficients (H) may be computed analytically or numerically.

Expansion of a function

- Projection of a function into V_n

$$f^n(x) = \sum_{l=0}^{2^n-1} \sum_{i=0}^{k-1} s_{il}^n \phi_{il}^n(x)$$

$$s_{il}^n = \int dx f(x) \phi_{il}^n(x)$$

- Local error is $O(f^{(k)}(\xi)2^{-nk})$ – high-order convergence
- If a function is expanded in V_n, what is the error from truncating to V_{n-1}?
- The error is contained in $W_{n-1} = V_n - V_{n-1}$

Multiwavelet Basis - I

- An orthonormal basis to span $W_{n-1} = V_n - V_{n-1}$
- Also demand
  - Disjoint support (important!)
  - Dilation/translation of wavelets at level 0
- Expand in V_n

$$\psi_i(x) = \sqrt{2} \sum_{j=0}^{k-1} \left(g_{ij}^{(0)} \phi_j(2x) + g_{ij}^{(1)} \phi_j(2x-1) \right)$$

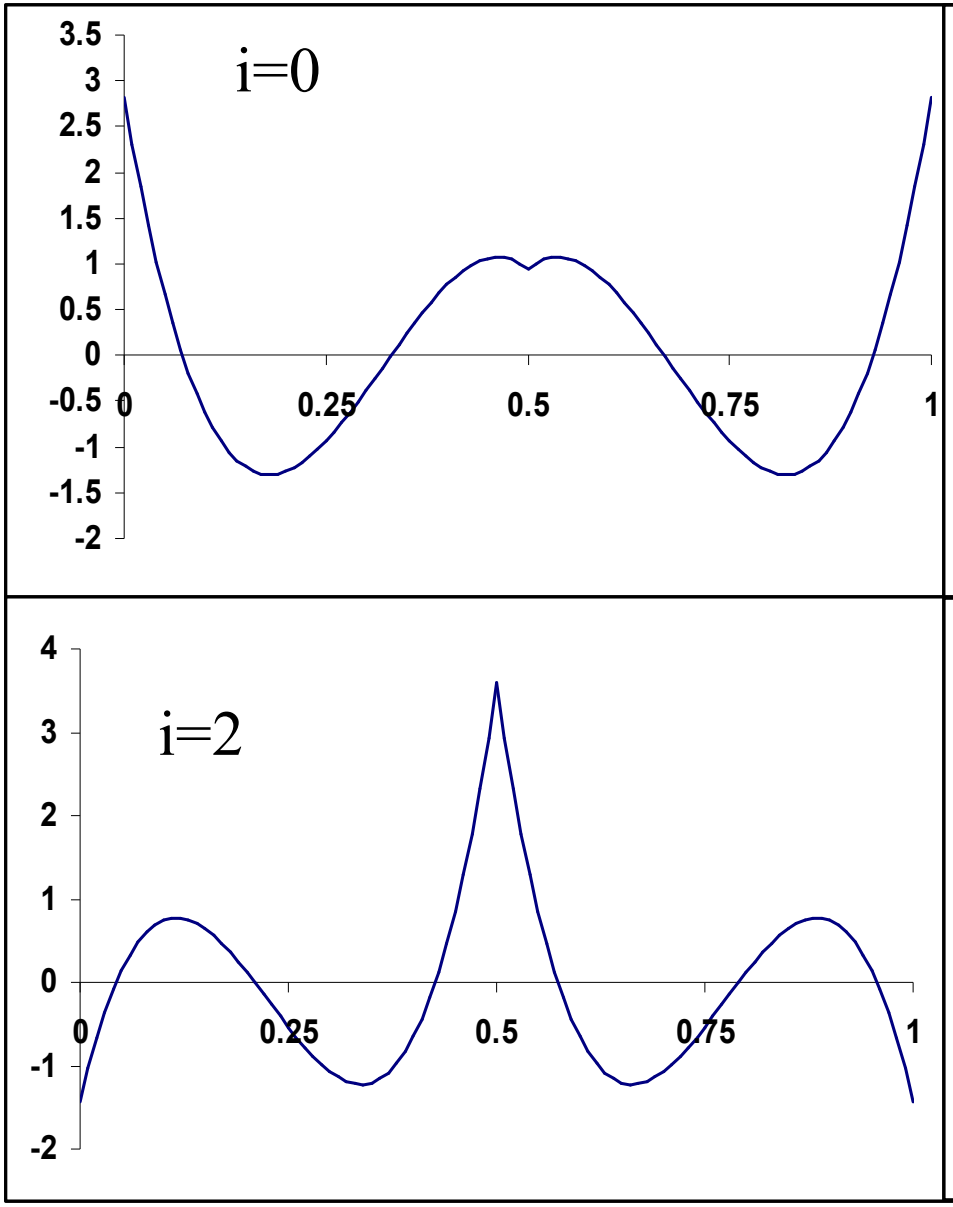
$$\psi_{il}^{n-1}(x) = \sum_{j=0}^{k-1} \left(g_{ij}^{(0)} \phi_{j2l}^{n}(2x) + g_{ij}^{(1)} \phi_{j2l+1}^{n}(2x-1) \right)$$

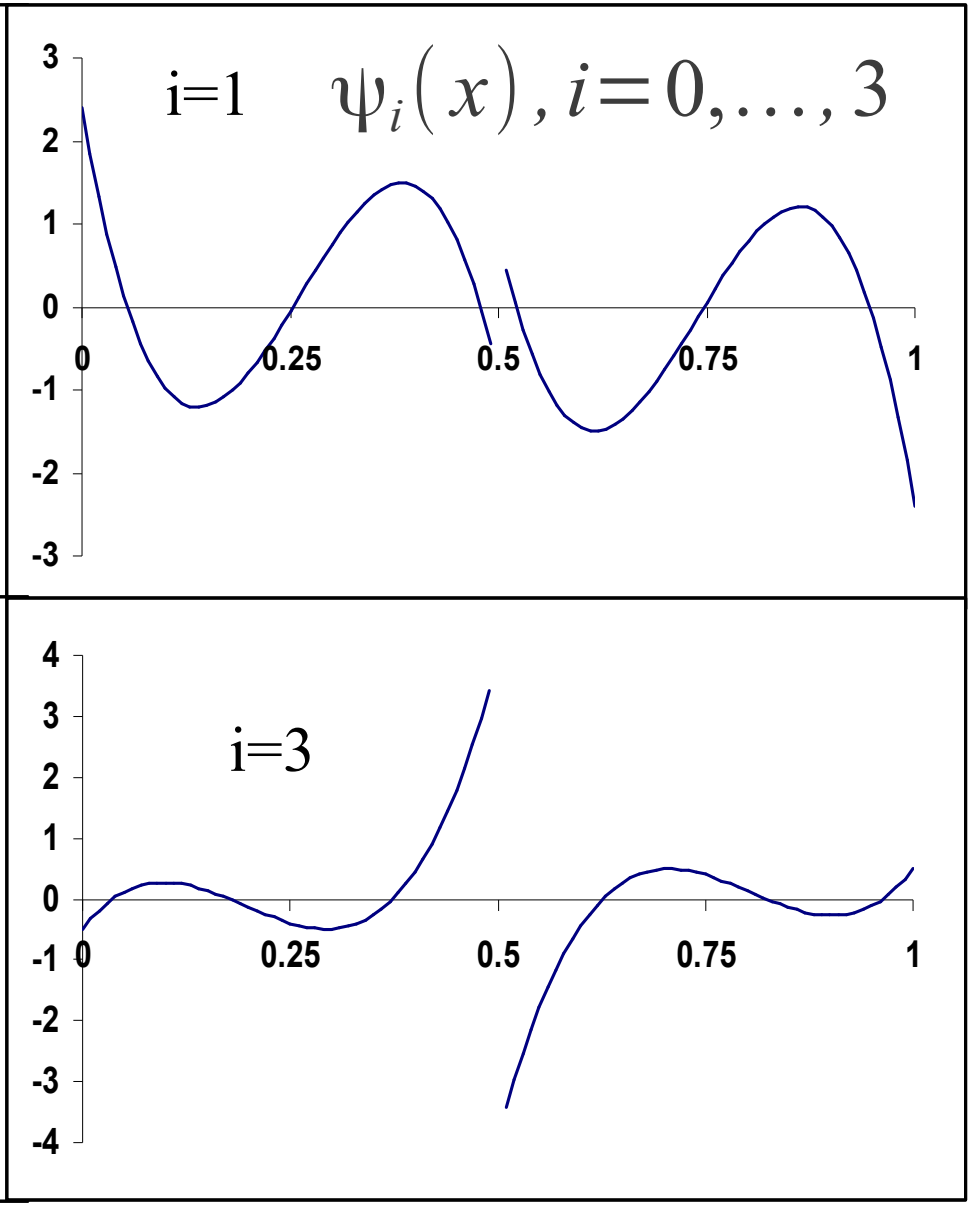
Multiwavelet Basis - II

- Basis not yet completely specified
  - Currently use Alpert's basis (which imposes additional constraints … more about this later)
- The coefficients (G) may be evaluated analytically or numerically
  - Numerical evaluation requires extended precision arithmetic (e.g., 208 bits used for k=12).
- Constructive approach
  - From scaling function make the wavelets
  - Often the reverse (e.g., Daubechies wavelets)

Multiwavelet Basis - III

$k=4$





Multiwavelet Basis - IV

- Translation and dilation $\psi_{li}^{n}(x) = 2^{n/2}\psi_{i}(2^{n}x - l)$
- Orthonormal $\int_{-\infty}^{\infty} \psi_{il}^{n}(x)\psi_{i'l'}^{n'}(x)dx = \delta_{nn'}\delta_{ll'}\delta_{ii'}$
- Direct sum of sub-spaces $V_n = V_0 + W_1 + \cdots + W_{n-1}$
- Complete basis for L_2 [0,1] in limit of *either* large *k* or large *n*.

Multiwavelet Basis - V

- Vanishing moments
  - Critically important property
  - Since W_n is orthogonal to V_n the first k moments of functions in W_n vanish, i.e.,

$$\int dx\, x^i \psi_i(x) = 0 \quad i = 0, \ldots, k-1$$

- In Alpert's basis, additional moments of some of the multiwavelet components also vanish

Some Consequences of Vanishing Moments

$$W_{n-1} = V_n - V_{n-1}$$

- Compact representation of smooth functions
  - Consider Taylor series … the first *k* terms vanish and smooth implies higher order terms are small
- Compact representation of integral operators
  - E.g., 1/|r-s|
  - Consider double Taylor series or multipole expansion
  - Interaction between wavelets decays as r^{-2k-1}
- Derivatives at origin vanish in Fourier space
  - Diminishes effect of singularities at that point

Two-scale relationship - I

$$\begin{pmatrix} \phi(x) \\ \psi(x) \end{pmatrix} = \sqrt{2} \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix} \begin{pmatrix} \phi(2x) \\ \phi(2x-1) \end{pmatrix}$$

Orthonormal transformation – best possible numerical accuracy

Filter coefficients for k=1 (Haar)

$$\begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} s_0^0 \\ d_0^0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} s_0^1 + s_1^1 \\ s_0^1 - s_1^1 \end{pmatrix}$$

Hence, coefficients of scaling functions and wavelets are often referred to as sum and difference coefficients.

Filter coefficients for k=4

```
 7.0711e-01  0.0000e+00  0.0000e+00  0.0000e+00      H0
-6.1237e-01  3.5355e-01  0.0000e+00  0.0000e+00
 0.0000e+00 -6.8465e-01  1.7678e-01  0.0000e+00
 2.3385e-01  4.0505e-01 -5.2291e-01  8.8388e-02

 7.0711e-01  0.0000e+00  0.0000e+00  0.0000e+00      H1
 6.1237e-01  3.5355e-01  0.0000e+00  0.0000e+00
 0.0000e+00  6.8465e-01  1.7678e-01  0.0000e+00
-2.3385e-01  4.0505e-01  5.2291e-01  8.8388e-02

 0.0000e+00  1.5339e-01  5.9409e-01 -3.5147e-01      G0
 1.5430e-01  2.6726e-01  1.7252e-01 -6.1237e-01
 0.0000e+00  8.7867e-02  3.4031e-01  6.1357e-01
 2.1565e-01  3.7351e-01  4.4362e-01  3.4233e-01

 0.0000e+00 -1.5339e-01  5.9409e-01  3.5147e-01      G1
-1.5430e-01  2.6726e-01 -1.7252e-01 -6.1237e-01
 0.0000e+00 -8.7867e-02  3.4031e-01 -6.1357e-01
-2.1565e-01  3.7351e-01 -4.4362e-01  3.4233e-01
```

The point being only that these are not mysterious or weird values.

Two-scale relationship II

$$f^n(x) = \sum_{l=0}^{2^n-1} \sum_{i=0}^{k-1} s_{il}^n \phi_{il}^n(x)$$

- May be rewritten without approximation as

$$f^n(x) = \sum_{l=0}^{2^n-1} \sum_{i=0}^{k-1} \left(s_{il}^{n-1} \phi_{il}^{n-1}(x) + d_{il}^{n-1} \psi_{il}^{n-1}(x) \right)$$

- Where

$$\begin{pmatrix} s_{il}^{n-1} \\ d_{il}^{n-1} \end{pmatrix} = \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix} \begin{pmatrix} s_{i2l}^{n} \\ s_{i2l+1}^{n} \end{pmatrix}$$

Compression of a function

$$V_n = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{n-1}$$

- Recursively apply the two-scale relation
- The basis is the scaling functions at level 0 and the multiwavelets at all levels

$$f^n(x) = \sum_{i=0}^{k-1} s_{i0}^0 \phi_{i0}^0(x) + \sum_{n'=0}^{n-1} \sum_{l=0}^{2^{n'}-1} \sum_{i=0}^{k-1} d_{il}^{n'} \psi_{il}^{n'}(x)$$

- Compression & reconstruction are *O(NlogN)* operations

Two equivalent representations

- Scaling function basis (reconstructed)

$$f^n(x) = \sum_{l=0}^{2^n-1} \sum_{i=0}^{k-1} s_{il}^n \phi_{il}^n(x)$$

- Multi-wavelet basis (compressed)

$$f^n(x) = \sum_{i=0}^{k-1} s_{i0}^0 \phi_{i0}^0(x) + \sum_{n'=0}^{n-1} \sum_{l=0}^{2^{n'}-1} \sum_{i=0}^{k-1} d_{il}^{n'} \psi_{il}^{n'}(x)$$

- Rapid compression/reconstruction
  - Asymptotically faster than the FFT
  - Use most appropriate basis for a given operation

A Third Equivalent Representation

- The function tabulated at the Gauss-Legendre quadrature points in each of the adaptively refined boxes
  - Enables rapid multiplication of functions and application of local functions (e.g., V_{xc})
  - Diagonal transformation from interpolating polynomials

Attributes of the multiwavelet basis

- Good stuff
  - Disjoint support enables true local refinement
  - Discontinuous polynomial basis can compactly represent sharp features
  - High-order convergence for smooth functions
  - Interpolating basis enables many operators
  - Computational kernels highly efficient
- Bad stuff
  - Not strongly band limited
  - High-order beneficial only when locally smooth

Truncation Error

- To satisfy the global error condition

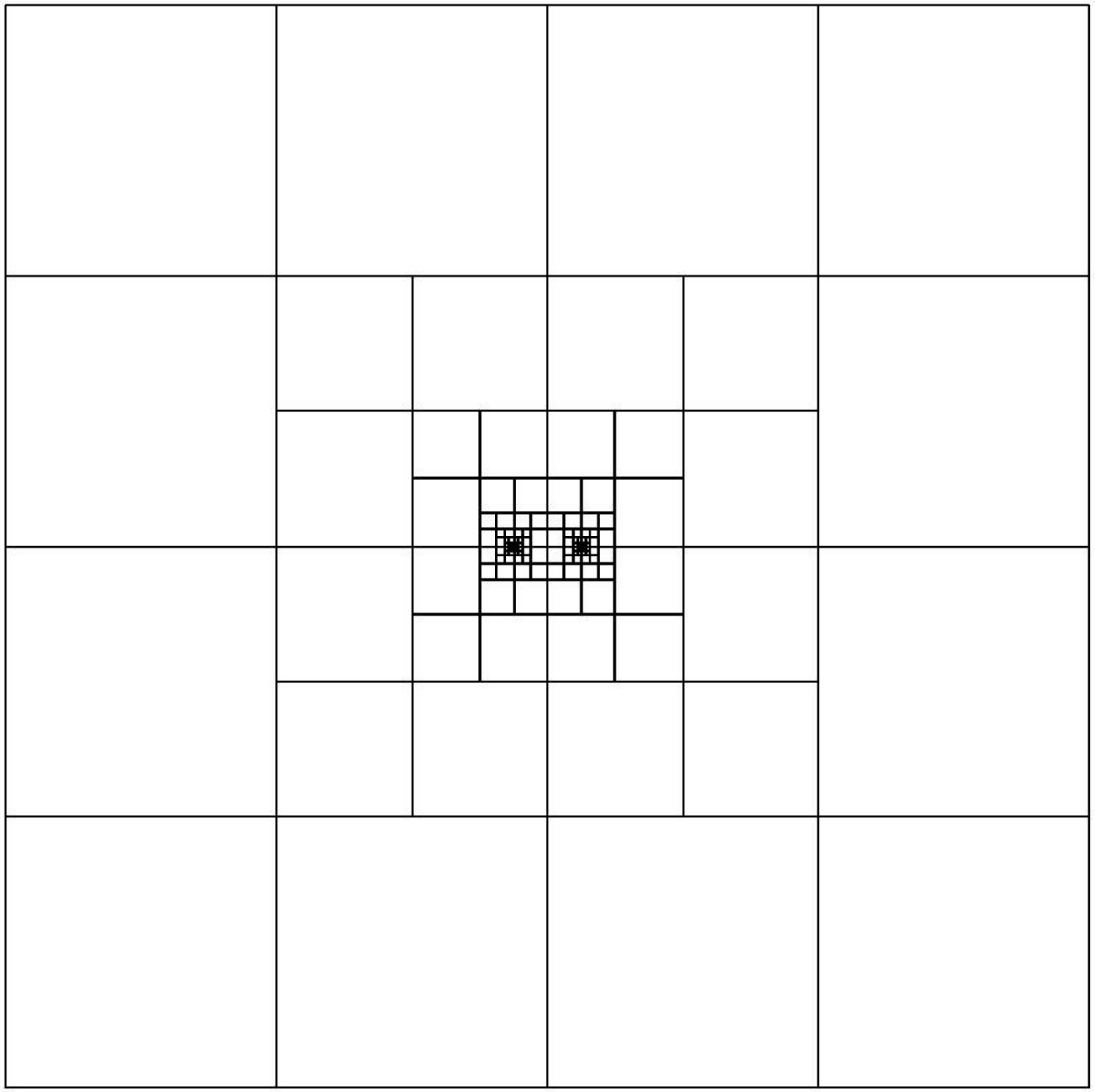
$$\left\| f - f^n \right\|_2 \leq \varepsilon \left\| f \right\|_2$$

- Truncate according to

$$\left\| d_l^n \right\|_2 \leq 2^{-n/2} \varepsilon \left\| f \right\|_2$$

- This is rather conservative – often use

$$\left\| d_l^n \right\|_2 \leq \varepsilon$$



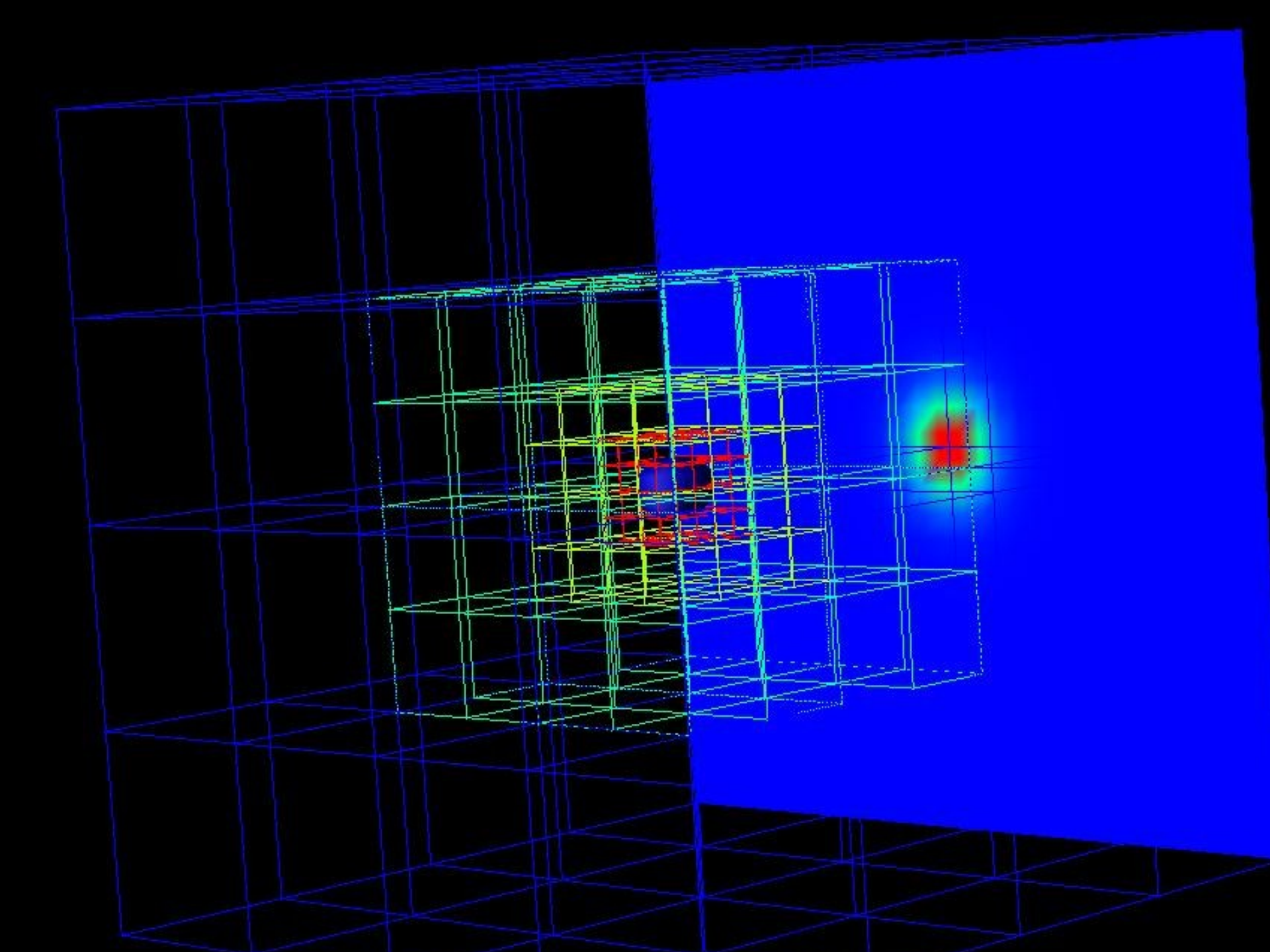
- Slice thru grid used to represent the nuclear potential for H_2 using k=7 to a precision of 10^{-5}.

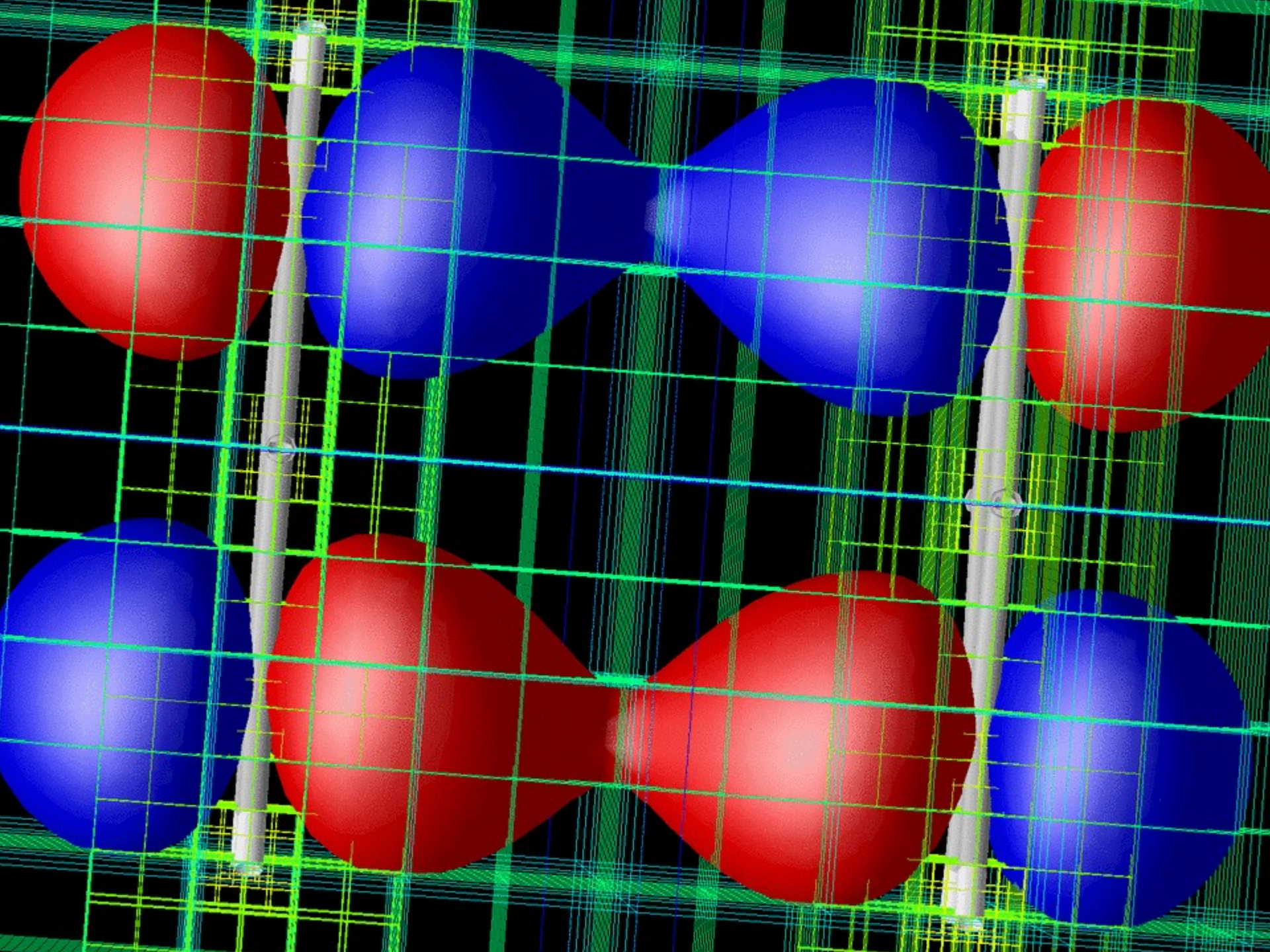
- Automatically adapts – it does not know a priori where the nuclei are.

- Nuclei at dyadic points on level 5 – refinement stops at level 8

- If were at non-dyadic points refinement continues (to level ??) but the precision is still guaranteed.

- In future will unevenly subdivide boxes to force nuclei to dyadic points.





Summary so far

- Scaling functions
  - Familiar orthormal basis, easy to evaluate, integrate, …
- Multi-resolution analysis
  - Separates behavior between length scales
  - Local truncation while preserving global error bound
  - Vanishing moments
- Multi-wavelets
  - High-order convergence with adaptive representation
  - Disjoint support – efficient description of singularities if located at faces/edges/corners (more efficient than smooth wavelets since they do not have disjoint support)
- Fast compression and reconstruction
  - Orthogonal transformations – numerically stable

- Good place to look at the (now ancient) implementation notes
  - Normalization, user/simulation cell, etc.
  - Truncation modes
  - Adaptive projection
  - Addition
  - Differentiation
  - Multiplication with refinement

Compression of a Matrix

- Compression is just a linear transformation
  - Apply separately to each dimension

$$W^T A W$$

  - Result is said to be in “standard form”

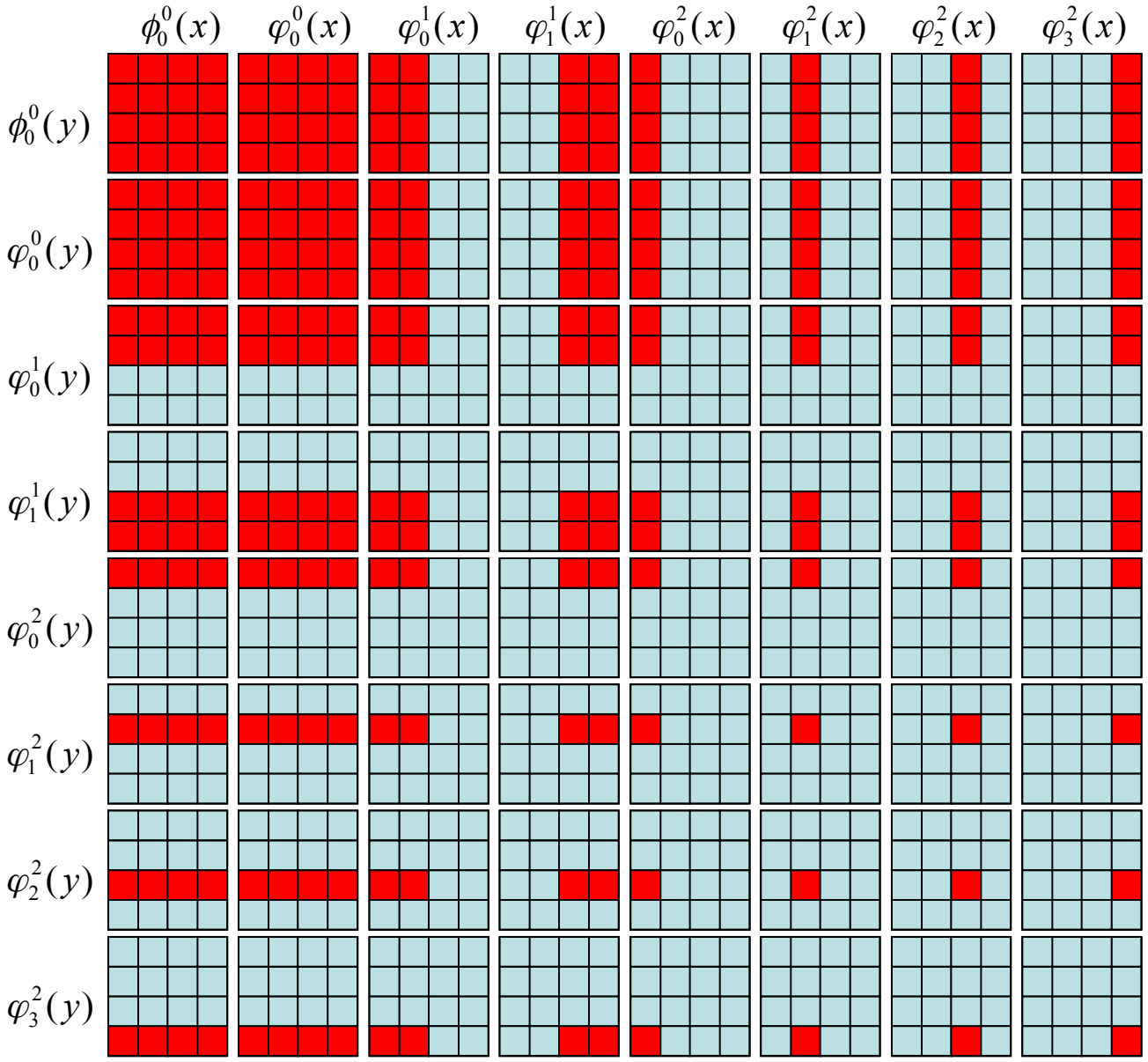
Extension to higher dimensions

- Scaling function basis is tensor product

$$\phi_{il}^{n}(x)\phi_{i'l'}^{n}(y)$$

- Wavelet basis – tensor product is one choice
  - Standard form – compress each dimension just as for a matrix
  - But cannot refine strictly locally since length scales are mixed between dimensions
- To refine locally need to refine all dimensions simultaneously

Standard-Form basis in 2-D



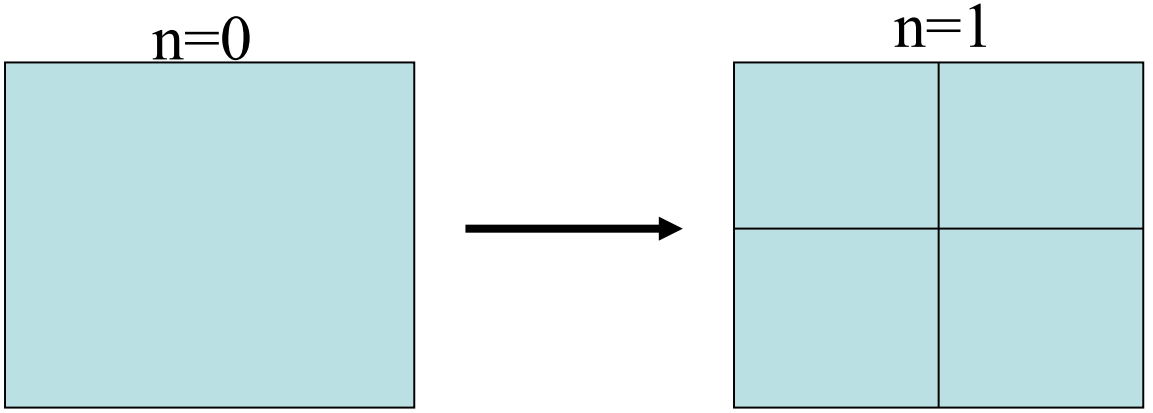
Tensor product basis on level 3 (level 3 scaling functions == level 2 wavelets)

Red indicates the support of the function

Adaptive refinement separately in each dimension … global not local refinement

Locally-refined form of functions

- Construct local basis for $W_{n-1}=V_n - V_{n-1}$



$$V_0=\{\phi(x)\phi(y)\}$$

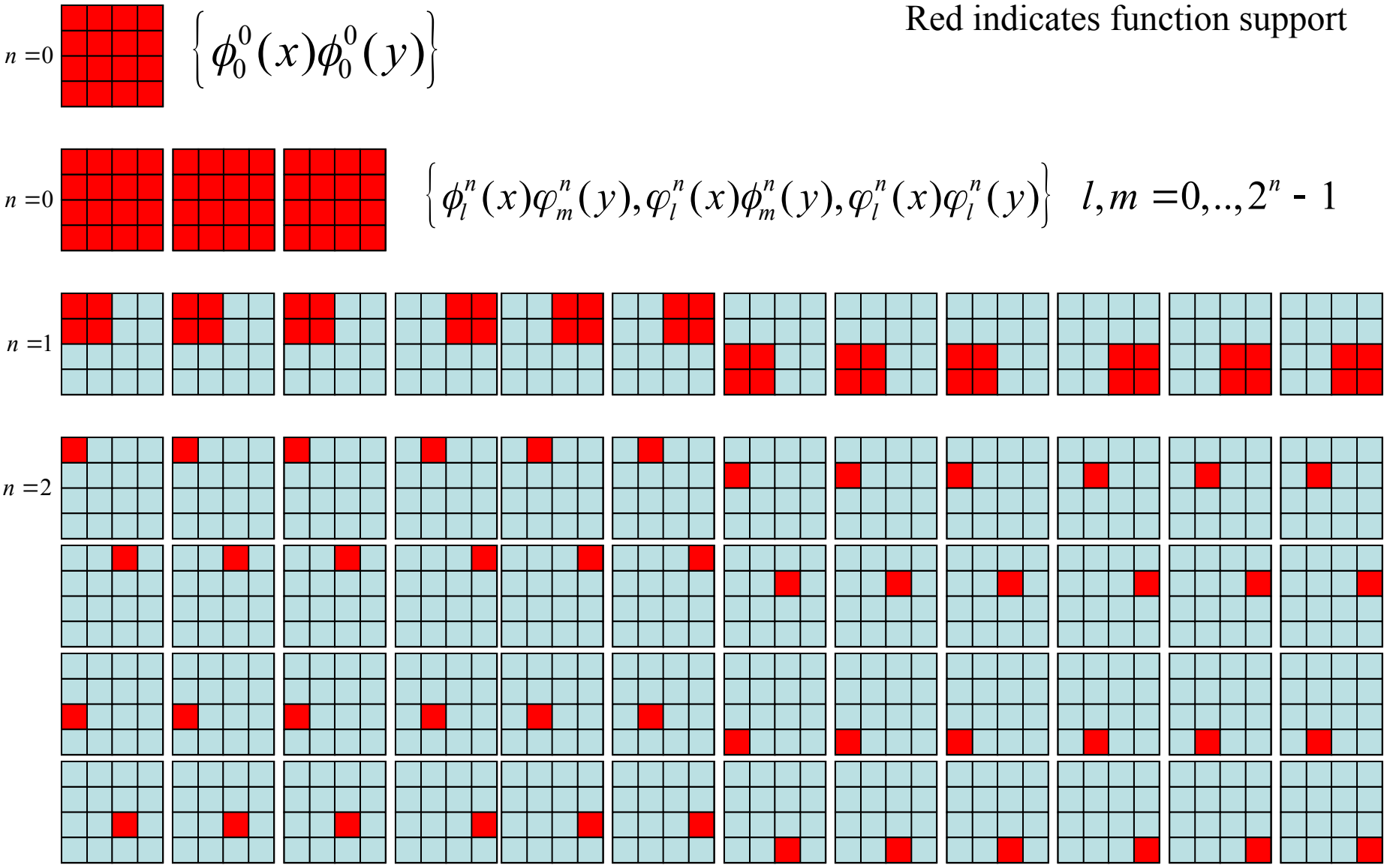
$$V_1=\sqrt{2}\begin{Bmatrix}\phi(2x)\phi(2y), & \phi(2x)\phi(2y-1), \\ \phi(2x-1)\phi(2y), & \phi(2x-1)\phi(2y-1)\end{Bmatrix}$$

$$=\{\phi(x)\phi(y),\ \phi(x)\psi(y),\ \psi(x)\phi(y),\ \psi(x)\psi(y)\}$$

$$W_0=V_1-V_0$$

$$=\{\phi(x)\psi(y),\ \psi(x)\phi(y),\ \psi(x)\psi(y)\}$$

Locally refined basis in 2-D



Non-standard form of operators - I

- Standard form
  - Matrix elements between different length scales
  - Not very efficient on modern computers with deep memory hierarchies
  - Potentially *O(Nlog N)* non-zero terms
  - Hard to calculate matrix elements, high-memory use
- Non-standard form
  - No matrix elements between lengths scales
  - *O(N)* terms
  - Act on (modified) non-standard form of functions
  - Easy to calculate matrix elements
  - Translation invariant operators yield Topelitz matrices
  - Derivation is instructive

Non-standard form of operators - II

- P_n an orthogonal projection into V_n
- Q_n an orthogonal projection into $W_n = V_{n+1} - V_n$

$$P_n + Q_n = P_{n+1}$$

- Consider the projection of an operator T

$$\begin{aligned}
T_n &= P_n T P_n \\
&= \left(P_{n-1} + Q_{n-1}\right) T \left(P_{n-1} + Q_{n-1}\right) \\
&= P_{n-1} T P_{n-1} + Q_{n-1} T Q_{n-1} + Q_{n-1} T P_{n-1} + P_{n-1} T Q_{n-1} \\
&= T_{n-1} + A_{n-1} + B_{n-1} + C_{n-1} \\
&= T_0 + \sum_{n'=0}^{n-1} \left(A_{n'} + B_{n'} + C_{n'}\right)
\end{aligned}$$

NS Form of Operators III

- Matrix elements in the scaling function basis

$$[r^n_{l\,l'}]_{i\,i'} = \int dx\, \phi^n_{i\,l}(x)\, T\, \phi^n_{i'\,l'}$$

- Matrix elements of the NS form

$$\begin{pmatrix} r^{n-1}_{l,l'} & \gamma^{n-1}_{l,l'} \\ \beta^{n-1}_{l,l'} & \alpha^{n-1}_{l,l'} \end{pmatrix} = \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix} \begin{pmatrix} r^n_{2l,2l'} & r^n_{2l,2l'+1} \\ r^n_{2l+1,2l'} & r^n_{2l+1,2l'+1} \end{pmatrix} \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix}^T$$

Vanishing Moments

- Sparse integral operators
  - If the derivatives decay rapidly (i.e., the kernel becomes smoother at long range)
  - See this by Taylor expansion (multipole series)
- Consider NS form of 3D Poisson kernel (*1/r*)

$$[A^n_{ll'}]_{ii'} = \int d^{3x}\, d^{3y}\, \psi^n_{li}(\boldsymbol{x}) \frac{1}{|\boldsymbol{x}-\boldsymbol{y}|} \psi^n_{l'i'}(\boldsymbol{y}) = O\left(|\boldsymbol{l}-\boldsymbol{l}'|^{-2k-1}\right)$$

$$[B^n_{ll'}]_{ii'} = \int d^{3x}\, d^{3y}\, \psi^n_{li}(\boldsymbol{x}) \frac{1}{|\boldsymbol{x}-\boldsymbol{y}|} \phi^n_{l'i'}(\boldsymbol{y}) = O\left(|\boldsymbol{l}-\boldsymbol{l}'|^{-k-1}\right)$$

$$[C^n_{ll'}]_{ii'} = \int d^{3x}\, d^{3y}\, \phi^n_{li}(\boldsymbol{x}) \frac{1}{|\boldsymbol{x}-\boldsymbol{y}|} \psi^n_{l'i'}(\boldsymbol{y}) = O\left(|\boldsymbol{l}-\boldsymbol{l}'|^{-k-1}\right)$$

Integral operators - I

- Consider $T f(x) = \mathrm{pv} \int dy\, K(x-y) f(y)$

$$
\begin{aligned}
[r^n_{l\,l'}]_{i\,i'} &= \int dx \int dy\, K(x-y) \phi^n_{i\,l}(x) \phi^n_{i'\,l}(y) \\
&= 2^{-n} \int dx \int dy\, K\left(2^{-n}(x-y+l-l')\right) \phi_i(x) \phi_{i'}(y) \\
&= 2^{-n} \int dz\, K\left(2^{-n}(z+l-l')\right) \Phi_{i\,i'}(z) \\
\Phi_{i\,i'}(z) &= \int dx\, \phi_i(x) \phi_{i'}(x-z) \\
&= \sum_{p=0}^{2k-1} \left(c^{(-)}_{i\,i'\,p} \phi_p(z+1) + c^{(+)}_{i\,i'\,p} \phi_p(z) \right) \\
[r^n_l]_p &= 2^{-n} \int dz\, K\left(2^{-n}(z+l)\right) \phi_p(z)
\end{aligned}
$$

Integral operators - II

- Matrix elements easy to evaluate from compressed form of kernel $K(x)$
- Application in 1-d is fairly efficient
  - $O(k^2)$ operations
- In 3-d seems to need $O(N_{box}k^6)$ operations
  - Prohibitively expensive
- More intelligent approach
  - $O(N_{box}k^4)$ operations for many "physical" kernels
  - Even better is known to be possible

Relationship to the FMM

- Greengard, Rokhlin
- Separate the behavior on different length scales
- Exploit low-rank form of off-diagonal blocks
- Approaches each kernel as a special case
- Highly-optimized, but complex
  - E.g., latest FMM uses seven different representations
- MRA approach is immediately general
  - Simpler than FMM since don't need to traverse up/down tree
  - Not as fast unless use kernel-specific separated forms
  - Perhaps more straightforward to parallelize

Please forget about wavelets

- They are not central
- Wavelets are a convenient basis to span V_n-V_{n-1} and for understanding its properties
- But you don't actually need to use them
  - MADNESS does still compute wavelet coefficients, but *Beylkin's new code does not*
- Please remember this …
  - Discontinuous spectral element with multi-resolution and separated representations for fast computation with guaranteed precision in many dimensions.

Summary so far

- Extension to many-dimensions
  - Locally-refined basis for functions enables true adaptive refinement
  - Non-standard form for operators enables practical application of operators
- Vanishing moments
  - Turns smoothness into sparsity

Problem with Differential Form

- Consider application of the Laplacian to a function with high-frequency numerical noise

$$\nabla^2\left(f(\boldsymbol{r})+\epsilon e^{i\boldsymbol{k}.r}\right)=\nabla^2 f(\boldsymbol{r})-k^2\epsilon e^{i\boldsymbol{k}.r}$$

- Consider 30 levels of adaptive refinement (and don't forget discontinuities, polynomials)

$$|k|\approx 10^9 \qquad k^2\approx 10^{18}$$

- I.e., we just took numerical noise ($\epsilon=10^{-16}$) and amplified it to *O(100)*

Advantages of Integral Form

- Condition number of inverse Laplacian just as bad (unbounded spectrum & zero eigenvalues)
  - But for the inverse, large eigenvalues correspond to the smooth and usually interesting bits

$$\nabla^{-2}\left(f(\boldsymbol{r})+\epsilon e^{i\boldsymbol{k}.r}\right)=\nabla^{-2} f(\boldsymbol{r})-k^{-2}\epsilon e^{i\boldsymbol{k}.r}$$

  - So the inverse operator damps out the noise
- A key step in applying MADNESS to any problem is rewriting differential equations in integral form

Advantages of Integral Form

- E.g., $\nabla^2 u(\boldsymbol{r}) = -4\pi\rho(\boldsymbol{r})$ v.s. $u(r) = \int G(\boldsymbol{r}, \boldsymbol{r}')\rho(\boldsymbol{r}')d^3r'$
  - Often soluble without any preconditioning and often without any iteration (as in this case)
  - Can obtain higher accuracy
  - In simple domains builds in correct asymptotics
  - Potentially more computationally efficient
- Challenge and solution
  - In most bases integral operator is dense
  - Multiresolution analysis provides fast algorithms with guaranteed precision

Electrostatics $\nabla^2 u(\boldsymbol{r}) = -4\pi\rho(\boldsymbol{r})$

$$u(r) = \int G(\boldsymbol{r},\boldsymbol{r}')\rho(\boldsymbol{r}')d^3r' + \oint_{\partial\Omega}\left[G(\boldsymbol{r},\boldsymbol{r}')\nabla' u(\boldsymbol{r}') - u(\boldsymbol{r}')\nabla' G(\boldsymbol{r},\boldsymbol{r}')\right].\boldsymbol{dS}$$

$$G(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}'|}$$

Quantum mechanics $\left(-\frac{1}{2}\nabla^2 + V(\boldsymbol{r})\right)\psi(\boldsymbol{r}) = E\,\psi(\boldsymbol{r})$

$$\psi(\boldsymbol{r}) = -2\left(\nabla^2 + 2E\right)^{-1} V(\boldsymbol{r})\,\psi(\boldsymbol{r})$$

Time evolution $\hat{L}\,u(\boldsymbol{r},t) + N(u,t) = \frac{du}{dt}$

$$u(\boldsymbol{r},t) = e^{t\hat{L}}u(\boldsymbol{r},0) + \int_0^t e^{(\tau-t)\hat{L}}N(u,\tau)\,d\tau$$

$$e^{t\nabla^2} f(\boldsymbol{r}) = (4\pi t)^{-d/2}\int e^{-\frac{(\boldsymbol{r}-\boldsymbol{r}')^2}{4t}} f(r')\,d^d r'$$

Integral Operator Formulation

- Solving the integral equation
  - Eliminates the derivative operator and related "issues"
  - Converges as fixed point iteration *with no preconditioner*

$$\left(-\tfrac{1}{2}\nabla^2 + V\right)\Psi = E\Psi$$

$$\Psi = -2\left(-\nabla^2 - 2E\right)^{-1} V\Psi$$

$$= -2G*(V\Psi)$$

$$(G*f)(r) = \int ds \frac{e^{-k|r-s|}}{4\pi|r-s|} f(s) \text{ in 3D}; k^2 = -2E$$

Such Green's Functions (bound state Helmholtz, Poisson) can be rapidly and accurately applied with a single, sparse matrix vector product.

Integral operators beyond 1D

$$(T * f)(\boldsymbol{x}) = \int d^3 y \, K(\boldsymbol{x} - \boldsymbol{y}) \, f(\boldsymbol{y})$$

- Consider a block of matrix elements in 3D

$$r^{n, \boldsymbol{l} - \boldsymbol{l}'}_{i i', \, j j', k k'}$$

- In *d* dimensions seems to require $O\left((2k)^{2d}\right)$ memory and operations just for this one block
  - This is not practical
- If we can write in separated form

$$r^{n, \boldsymbol{l} - \boldsymbol{l}'}_{i i', \, j j', k k'} = X^{n, l_x - l'_x}_{i i'} Y^{n, l_y - l'_y}_{j j'} Z^{n, l_z - l'_z}_{k k'}$$

- Then, cost of application reduces to $O\left(d(2k)^{d+1}\right)$
  - Or better by exploiting structure of matrices

Low separation rank approximation of functions

$$f(x_{1,}\ldots,x_n)=\sum_{l=1}^{M}\sigma_l\prod_{i=1}^{d}f_i^{(l)}(x_i)+O(\epsilon)$$

$$\left\|f_i^{(l)}\right\|_2=1 \qquad \sigma_l>0$$

- For efficient approximation we want *M* to depend only weakly upon dimension, domain and accuracy
- For many physically interesting kernels can construct from analytic expressions
  - Efficient expansions may not exist globally, in which case would instead seek local approximations
- Or numerically compute approximations

Separated form for integral operators

$$(T * f)(\boldsymbol{x}) = \int d^3 y\, K(\boldsymbol{x}-\boldsymbol{y})\, f(\boldsymbol{y})$$

- Approach
  - Represent the kernel over a finite range as a sum of products of 1-D operators (often, not always, Gaussian)

$$r^{n,l-l'}_{ii',jj',kk'} = \sum_{\mu=0}^{M} X^{n,l_x-l'_x}_{ii'} Y^{n,l_y-l'_y}_{jj'} Z^{n,l_z-l'_z}_{kk'} + O(\epsilon)$$

  - Only need compute 1D transition matrices (X,Y,Z)
  - SVD the 1-D operators (low rank away from singularity)
  - Apply most efficient choice of low/full rank 1-D operator
  - G. Beylkin, R. Cramer, G. Fann and R. J. Harrison, Multiresolution separated representations of singular and weakly singular operators, Applied and Computational Harmonic Analysis, 23, (2007) 235-253
  - G. Fann, G. Beylkin, R. Harrison and K. Jordan, Singular operators in multiwavelet bases, IBM Journal of Research and Development 48 (2) (2004) 161-171.

Quadratures for separated representations

Seeking representation of form

$$f(r)=\sum_j c_\mu e^{-t_j r^2}$$

If the function is homogeneous of degree k

$$f(\lambda r)=\lambda^k f(r)$$

Then both $c\,e^{-tr^2}$ and $\lambda^k c\,e^{-t\lambda^2 r^2}$ should occur, suggesting the expansion is of the form

$$f(r)=c\sum_j \alpha^{jk} e^{-t\alpha^{2j} r^2}$$

At this point I immediately thought "thanks James" because he taught me a related techique in quadrature.

James N. Lyness, ANL/MCS

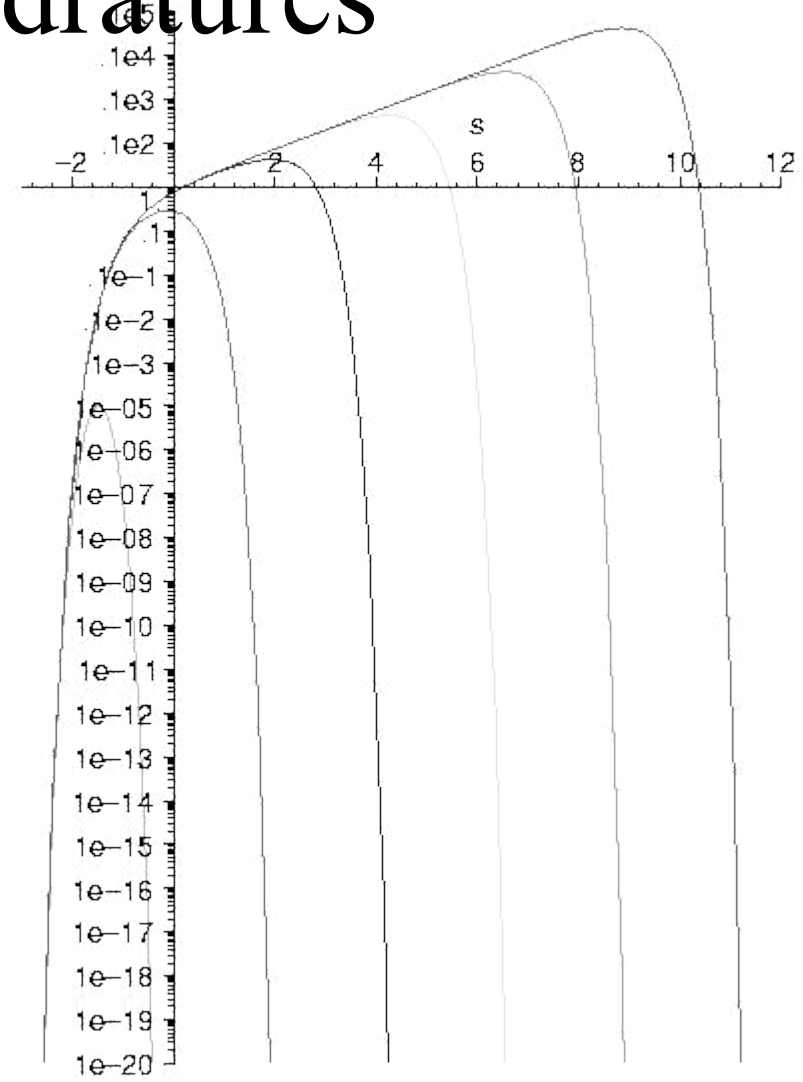
Accurate Quadratures

$$\frac{e^{-\mu r}}{r} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-x^2 t^2 - \mu^2/4t^2}\, dt$$

$$= \frac{2}{\sqrt{\pi}} \int_{-\infty}^\infty e^{-x^2 e^{2s} - \mu^2 e^{-2s}/4 + s}\, ds$$

$$= \sum_\mu c_\mu e^{t_\mu x^2} + O\left(\epsilon(m)\right)$$

- Trapezoidal quadrature
  - Geometric precision for periodic functions with sufficient smoothness
- Beylkin & Monzon
  - Further reductions
- Reuse and caching



The kernel for x=1e-4,1e-3,1e-2,1e-,1e0.

The curve for x=1e-4 is the rightmost

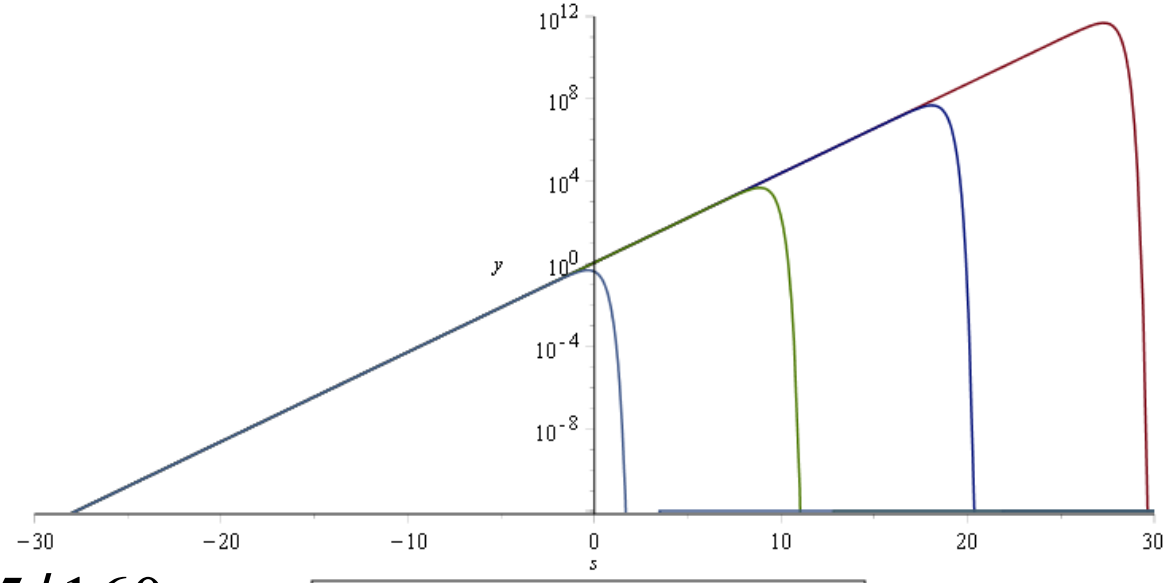
Sample fit for 1/x – relative error

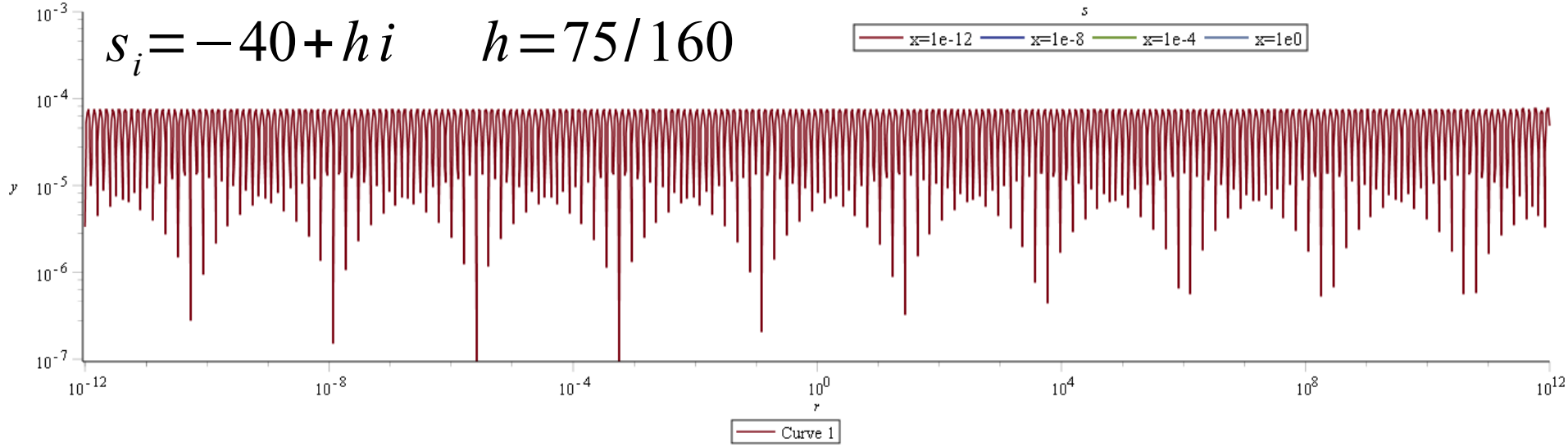
$$x^{-1} = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-r^2 e^{2s} + s}$$

$$f(x) = \sum_{i=0}^{160} c_i e^{-t_i r^2}$$

$$c_i = \frac{2 h e^{s_i}}{\sqrt{\pi}} \qquad t_i = e^{2 s_i}$$

$$s_i = -40 + h i \qquad h = 75/160$$





Construction via Laplace Transform

$$\mathcal{L}(f(t))(s) = F(s) = \int_0^\infty e^{-st} f(t)\, dt \qquad \mathcal{L}^{-1}(F(s))(t) = f(t) = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\gamma - iT}^{\gamma + iT} e^{st} F(s)\, ds$$

- To compute a Gaussian representation of given $F(r^2)$, substitute $s=r^2$ and invert for $f(t)$
- E.g.,
  - $F(r^2) = 1/|r| = 1/\sqrt{r^2}$ need inverse transform of $F(s) = 1/\sqrt{s}$
  - A standard transform is $\frac{\sqrt{\pi}}{\sqrt{s}} = \int_0^\infty \frac{e^{-st}}{\sqrt{t}}\, dt$
  - Substituting $s=r^2$ and $z=t^2$ we obtain a familiar identity $\frac{1}{r} = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-r^2 z^2}\, dz$

Approximation error

- Trefethen and Weideman, "The Exponentially Convergent Trapezoidal Rule," SIAM Review 56, p385 2014

$$I = \int_{-\infty}^{+\infty} \omega(x)\,dx \qquad I_h = h \sum_{k=-\infty}^{k=+\infty} \omega(k\,h)$$

THEOREM 5.1. *Suppose w is analytic in the strip $|\mathrm{Im}(x)| < a$ for some $a > 0$. Suppose further that $w(x) \to 0$ uniformly as $|x| \to \infty$ in the strip, and for some M, it satisfies*

$$\int_{-\infty}^{\infty} |w(x+ib)|dx \le M \tag{5.5}$$

for all $b \in (-a, a)$. Then, for any $h > 0$, I_h as defined by (5.2) exists and satisfies

$$|I_h - I\,| \le \frac{2M}{e^{2\pi a/h} - 1}, \tag{5.6}$$

and the quantity $2M$ in the numerator is as small as possible.

Approximation error 1/r

$$\frac{1}{r} = \int_{-\infty}^{+\infty} \frac{2}{\sqrt{\pi}} \exp\left(-r^2 \exp(2s) + s\right) ds$$

$$\int_{-\infty}^{+\infty} |\omega(x + ib)|\, ds = \frac{1}{r\sqrt{\cos(2b)}}$$

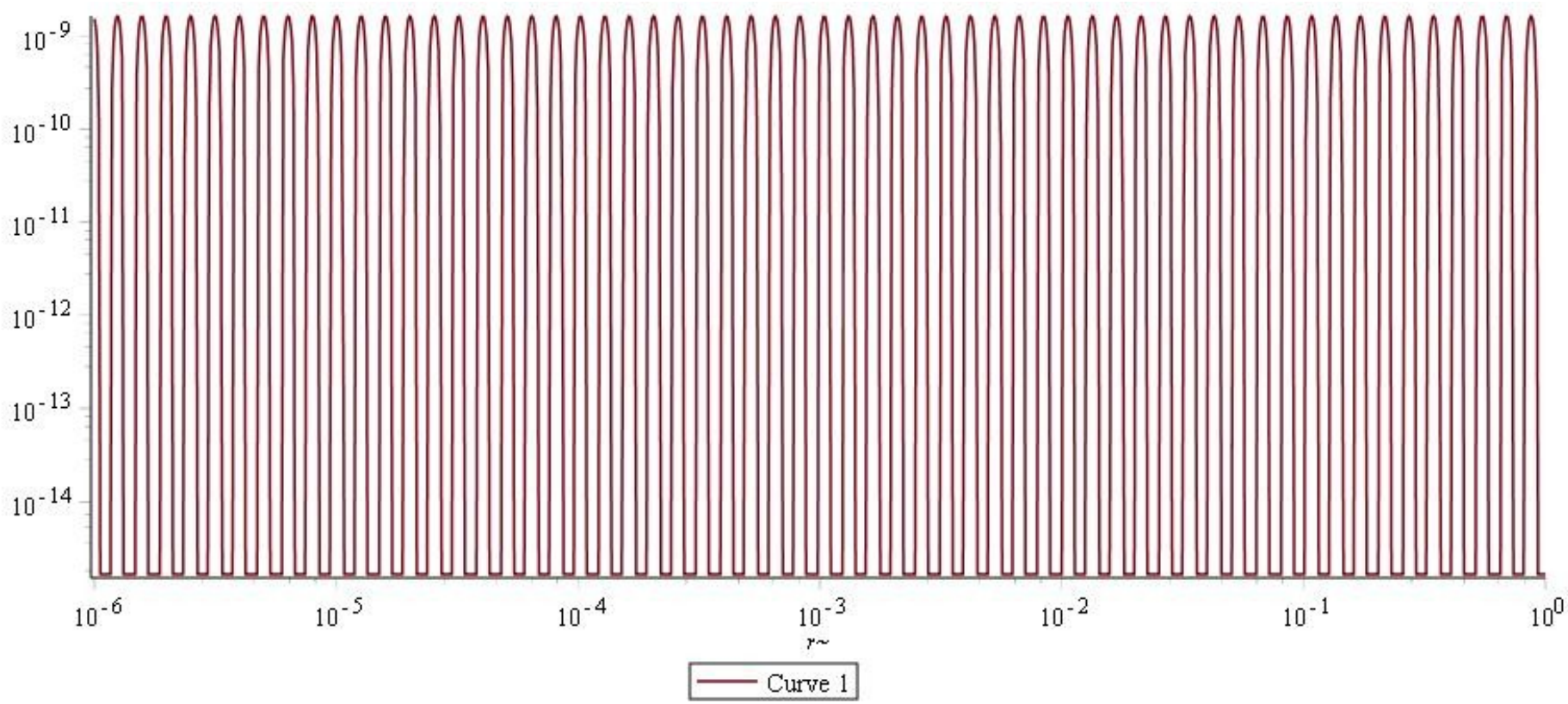
$$|I - I_h| \le \frac{2\exp(-2\pi a/h)}{\sqrt{\cos(2b)}} \quad \text{minimized by } a = \frac{1}{2}\arctan(2\pi/h)$$

$$\epsilon(h) = |I - I_h| \le \frac{1}{r}\sqrt{\frac{8\pi}{h}} \exp\left(-\frac{(\pi^2 - h)}{2h}\right)$$

$$h \approx \frac{\pi^2}{\log(16\pi) - 2\log(\epsilon) + 1}$$

Approximation error 1/r

- Error is relative
- epsilon=1e-8 → h=0.236



Approximation error exp(-t r)/r

$$\frac{\exp(-t\,r)}{r}=\int_{-\infty}^{+\infty}\frac{2}{\sqrt{\pi}}\exp\left(-r^2\exp(2\,s)-\frac{t^2}{4}\exp(-2\,s)+s\right)ds$$

$$\int_{-\infty}^{+\infty}|\omega(x+i\,b)|\,ds=\frac{\exp(-t\,r\cos(2\,b))}{r\sqrt{\cos(2\,b)}}$$

$$|I-I_h|\le\frac{2\exp(-2\,\pi\,a/h)}{\sqrt{\cos(2\,b)}}$$

For small $h\,t\,r$ (i.e., near singularity) also minimized by

$$a=\frac{1}{2}\arctan(2\,\pi/h)$$

$$\epsilon(h)=|I-I_h|\le\frac{1}{r}\sqrt{\frac{8\,\pi}{h}}\exp\left(-\frac{h^2\,t\,r+\pi^3-h\,\pi}{2\,h\,\pi}\right)$$

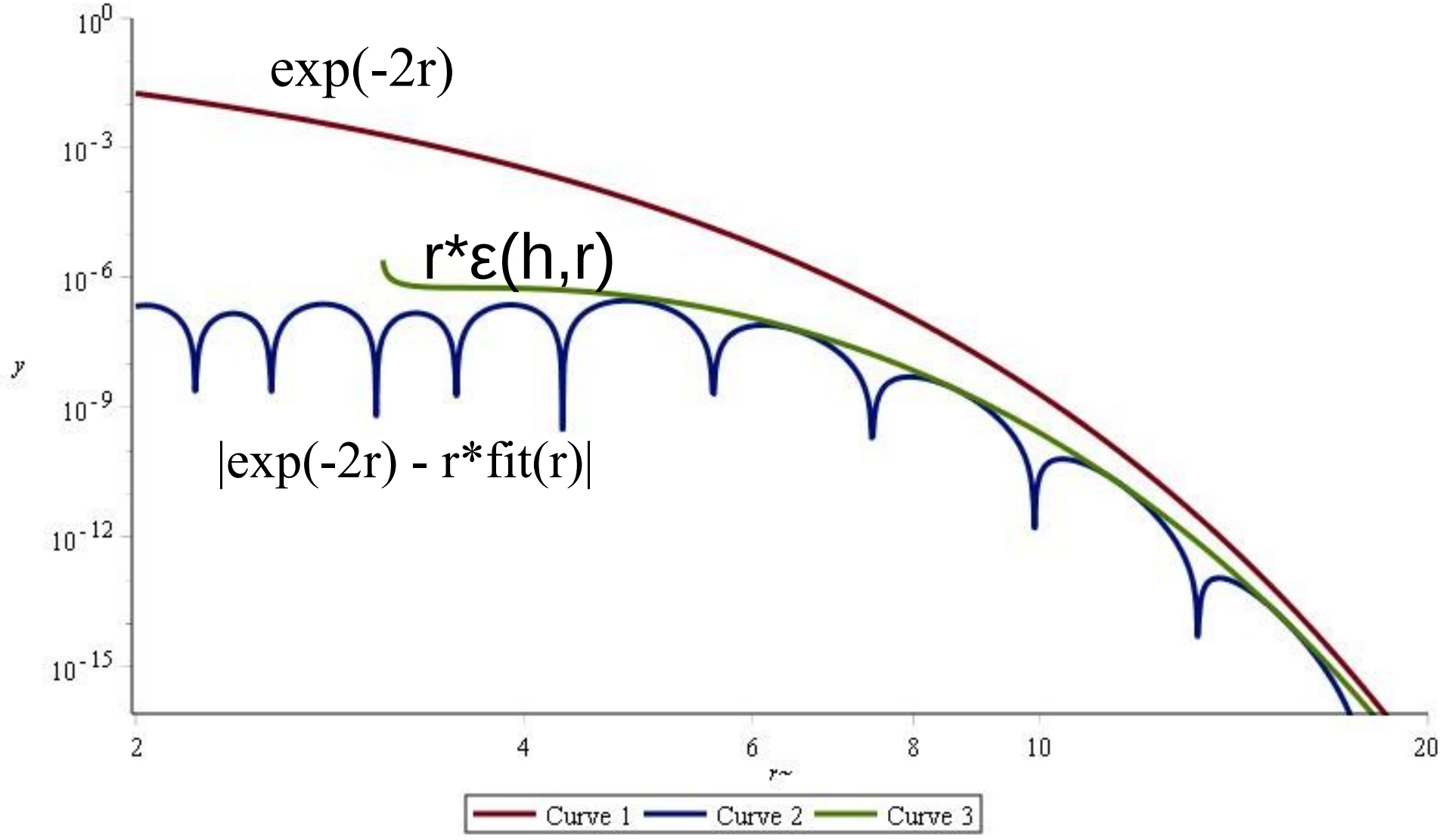
Approximation error exp(-t r)/r

For large $h\,t\,r$ (i.e., in exponential decay) minimized by

$$a = \frac{\pi}{h\left(2\,t\,r+1\right)}$$

$$\epsilon\left(h\right) = \left|I - I_h\right| \leq \frac{2\exp\left(-t\,r\cos\left(2\,\pi/\left(h\left(2\,t\,r+1\right)\right)\right)\right)}{r\left(\sqrt{\left(\cos\left(2\,\pi/\left(h\left(2\,mu\,r+1\right)\right)\right)\right)}\exp\left(2\,\pi^2/\left(h^2\left(2\,mu\,r+1\right)\right)\right)\right)}$$

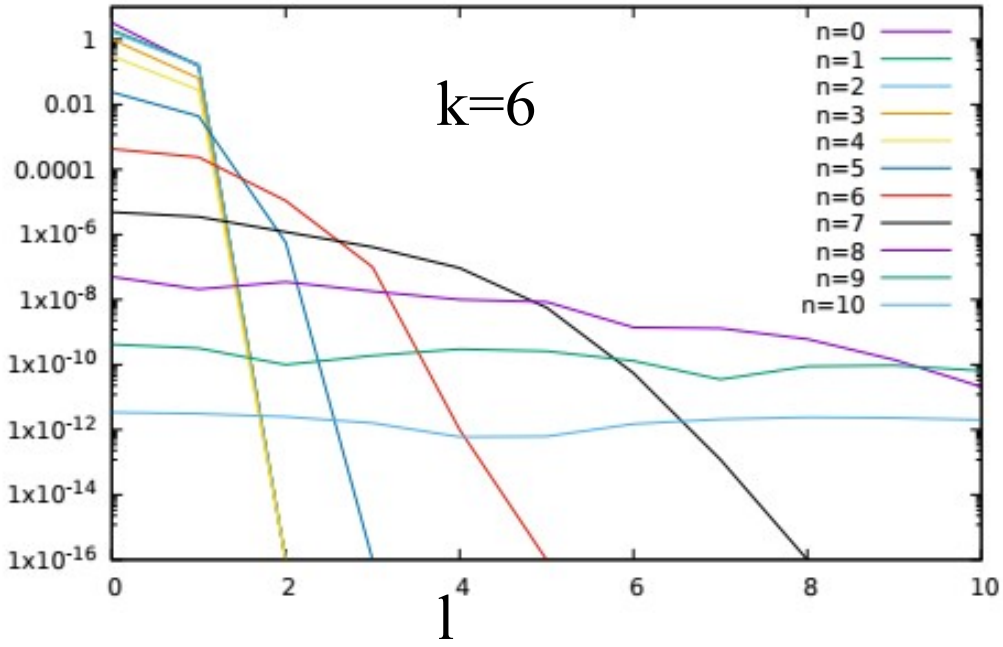
Approximation error – exp(-tr)/r

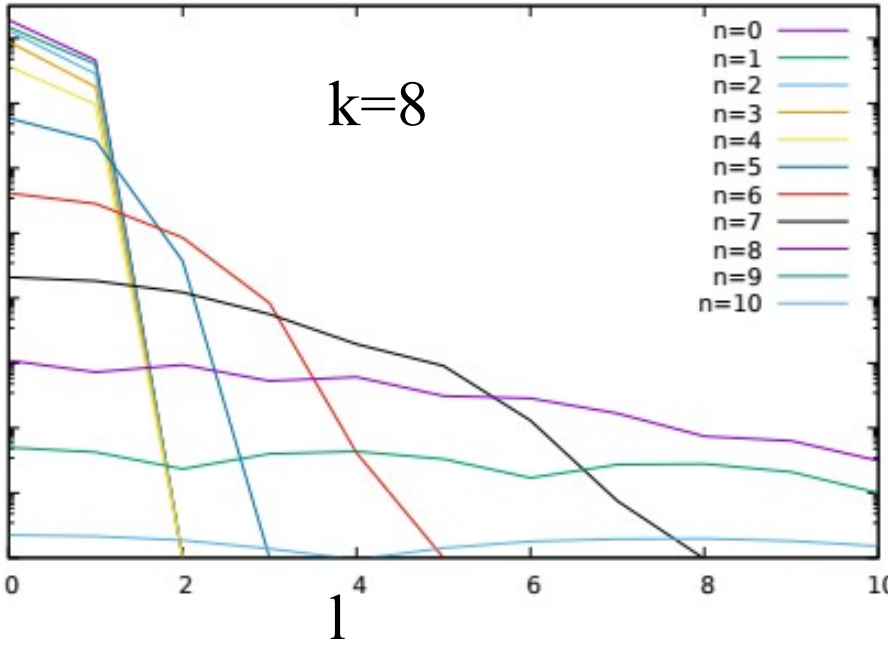


Non-linear approximation – incl. optimal Gaussian expansions

- G. Beylkin and T.S. Haut, Nonlinear approximations for electronic structure calculations, Proceedings of the Royal Society A, 469, (2013) 20130231
- G. Beylkin and L. Monzon, Approximation by exponential sums revisited, Applied and Computational Harmonic Analysis, 28, (2010) 131--149
- G. Beylkin and L. Monzon, On approximation of functions by exponential sums, Applied and Computational Harmonic Analysis, 19 (2005) 17-48

Norm of NS-form of convolution with Gaussian





$\left\| \begin{matrix} 0 & C \\ B & A \end{matrix} \right\|_F$ for kernel $K(x) = \frac{a}{\pi} e^{-10000 r^2}$ for translation $l = 0, \ldots, 10$

Rule of thumb $k = 2 - \log_{10} \epsilon$

Very low rank for $l > 0$

Linear many-body expansions

- G. Beylkin, M. J. Mohlenkamp and F. Perez, Approximating a Wavefunction as an Unconstrained Sum of Slater Determinants, Journal of Mathematical Physics, 49, (2008)
- G. Beylkin, M. J. Mohlenkamp and F. Perez, Preliminary results on approximating a wavefunction as an unconstrained sum of Slater determinants, Proc. Appl. Math. Mech., 7, (2007)

Separated form of reciprocal

- Quadrature: $\frac{1}{x} = \int_0^\infty \exp(-t\,x)\,dt = \sum_{\mu=1}^{M} \omega_i \exp(-x\,t_\mu)$

  Hard to control error over desired range

- Approximation

  $$\frac{1}{x} = \sum_{\mu=1}^{M} \omega_i \exp(-x\,t_\mu) + O(\epsilon) \qquad x \in [a, b]$$

  Better control and more general kernels

- Beylkin, Hackbush (/www.mis.mpg.de/scicomp/EXP_SUM)

Computational kernels

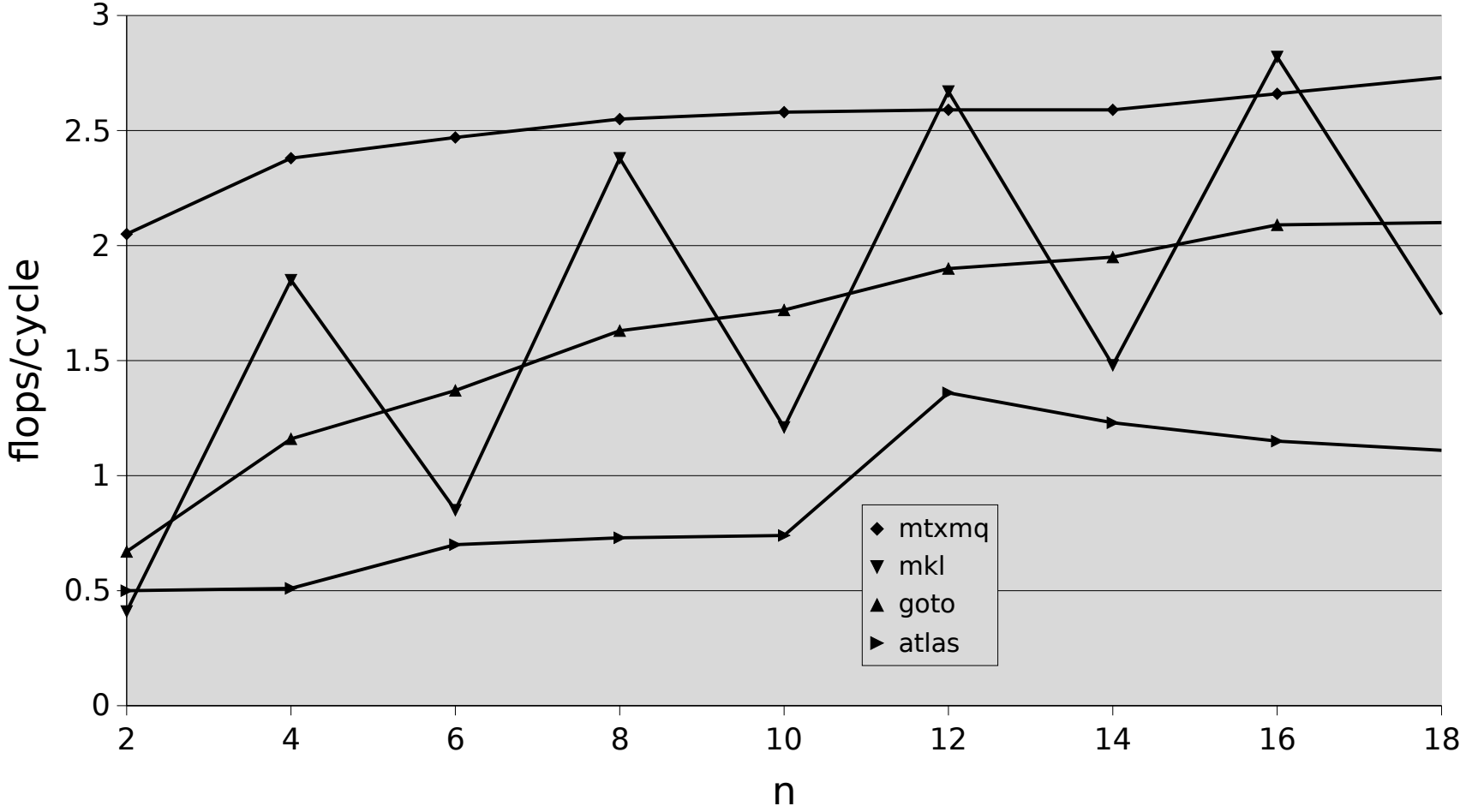
- Discontinuous spectral element
  - In each “box” a tensor product of coefficients
  - Most operations are small matrix-multiplication

$$r_{i'j'k'} = \sum_{ijk} s_{ijk} c_{ii'} c_{jj'} c_{kk'} = \sum_{k} \left(\sum_{j} \left(\sum_{i} s_{ijk} c_{ii'} \right) c_{jj'} \right) c_{kk'}$$

$$\Rightarrow r = \left(\left(s^T c \right)^T c \right)^T c$$

  - Typical matrix dimensions are 2 to 30
  - E.g., $(20,400)^T$ * (20,20)

Comparison with MKL, Goto, ATLAS on Intel Xeon 5355 for $(20,400)^T*(20,n)$.



Cray XT5 single core FLOPs/cycle

(nj, ni)T*(nj,nk)				
ni	nj	nk	MTXMQ	ACML
400	2	20	2.55	0.95
400	4	20	2.62	1.50
400	6	20	2.60	1.79
400	8	20	2.56	2.02
400	10	20	2.58	2.12
400	12	20	2.64	2.27
400	14	20	2.90	2.35
400	16	20	2.80	2.46
400	18	20	2.74	2.49
400	20	20	2.89	2.58

nested transform (nj, ni)T*(nj,nk)				
ni	nj	nk	MTXMQ	ACML
4	2	2	0.10	0.07
16	4	4	1.04	0.51
36	6	6	1.74	0.99
64	8	8	2.33	1.56
100	10	10	2.61	1.80
144	12	12	2.69	2.12
196	14	14	2.94	2.17
256	16	16	2.97	2.41
324	18	18	2.93	2.38
400	20	20	3.03	2.49
484	22	22	3.01	2.52
576	24	24	3.09	2.73
676	26	26	3.02	2.73
784	28	28	2.87	2.87
900	30	30	2.88	2.81

L2 cache is 512Kb = 2*32^3 doubles
- hence good multi-core scaling
- nested transform scales linearly to all cores

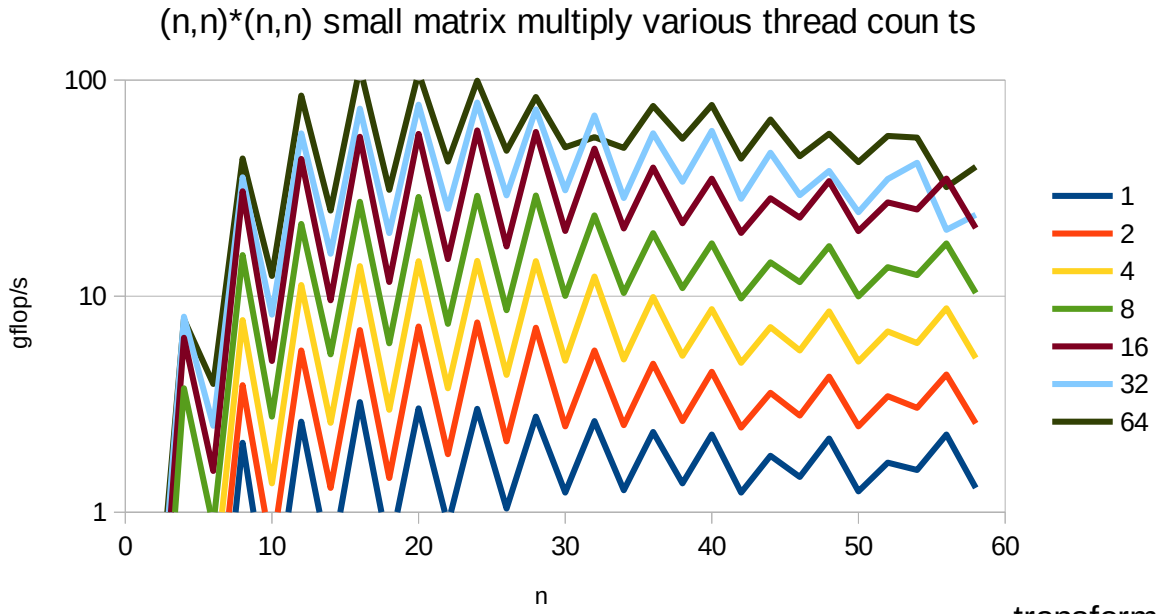
IBM BGQ Team

- ANL
  - Alvaro Vasquez, Jeff Hammond, Nichols Romero
- OSU
  - Kevin Stocks
- SBU
  - Robert Harrison, Scott Thornton
- VT
  - Ed Valeev, Justus Calvin
- Toyohashi
  - Hideo Sekino, Yukina Yokoi
- ORNL
  - George Fann

Early Science Project Activities

- We are testing a new linear-response module to solve TDDFT equations.
- Molecular properties (dipole polarizabilities, NMR chemical shifting, etc.)
- Support for well known hard pseudopotentials (i.e. Krack, Goedecker,etc.)
- Speedup of Hartree-Fock exchange evaluations via screening parameters.
- Implementation of new DFT functionals.
- Improving parallel scalability for current supercomputer architectures.

MtXM performance on BGQ

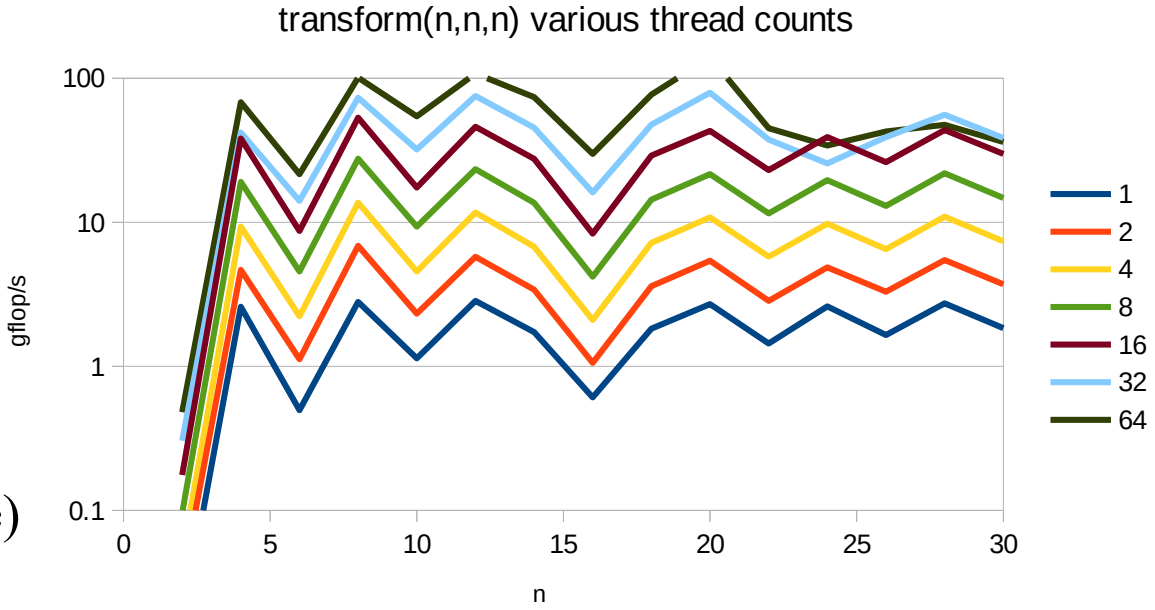


Kevin Stocks OSU

64 threads, best performance is 139.7 GFLOPS (trans(400,20,20))

* Theoretical peak is 204.8
* Linpack is approx. 166.3

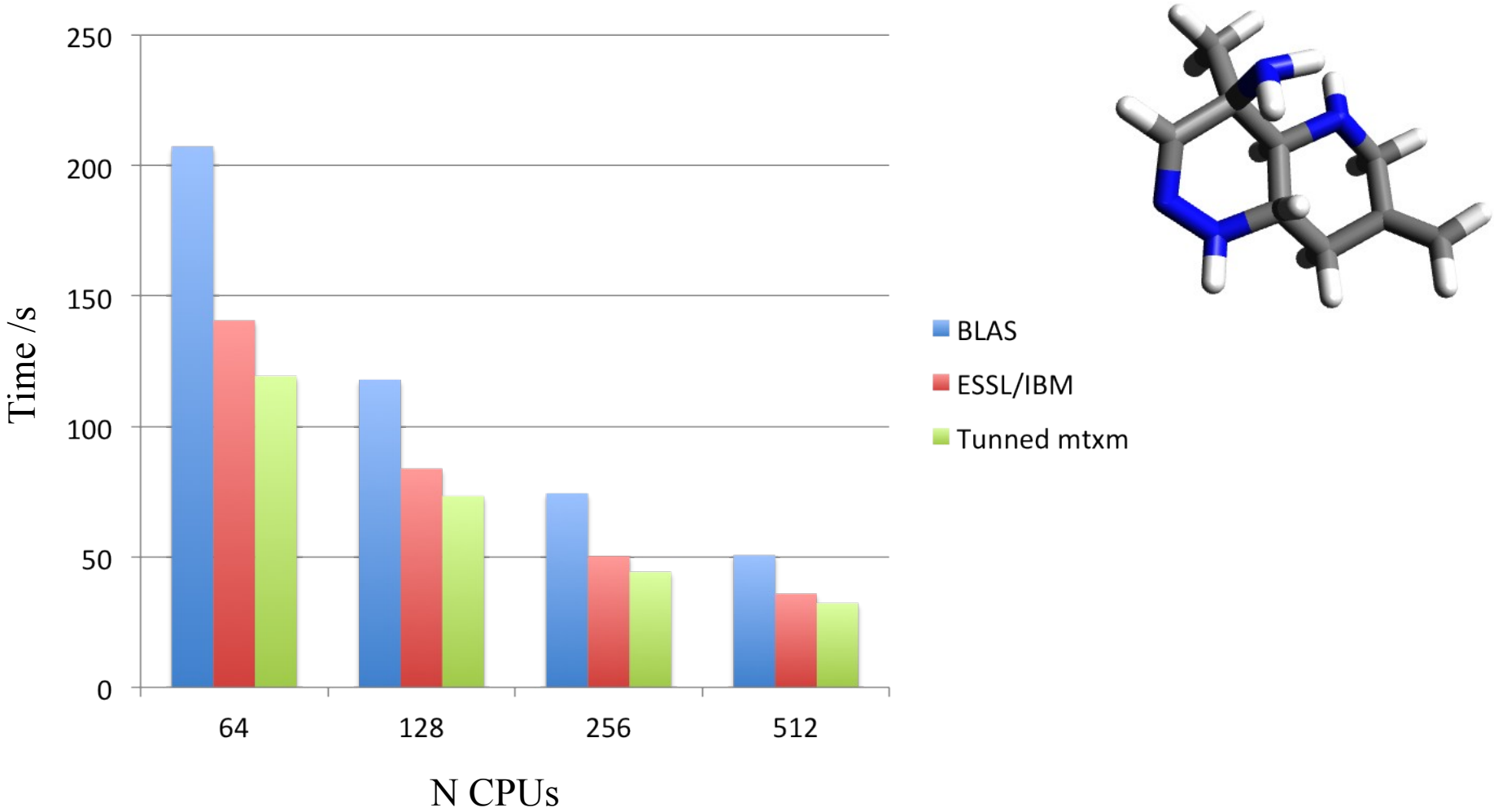
(scaling top500 results to one node)



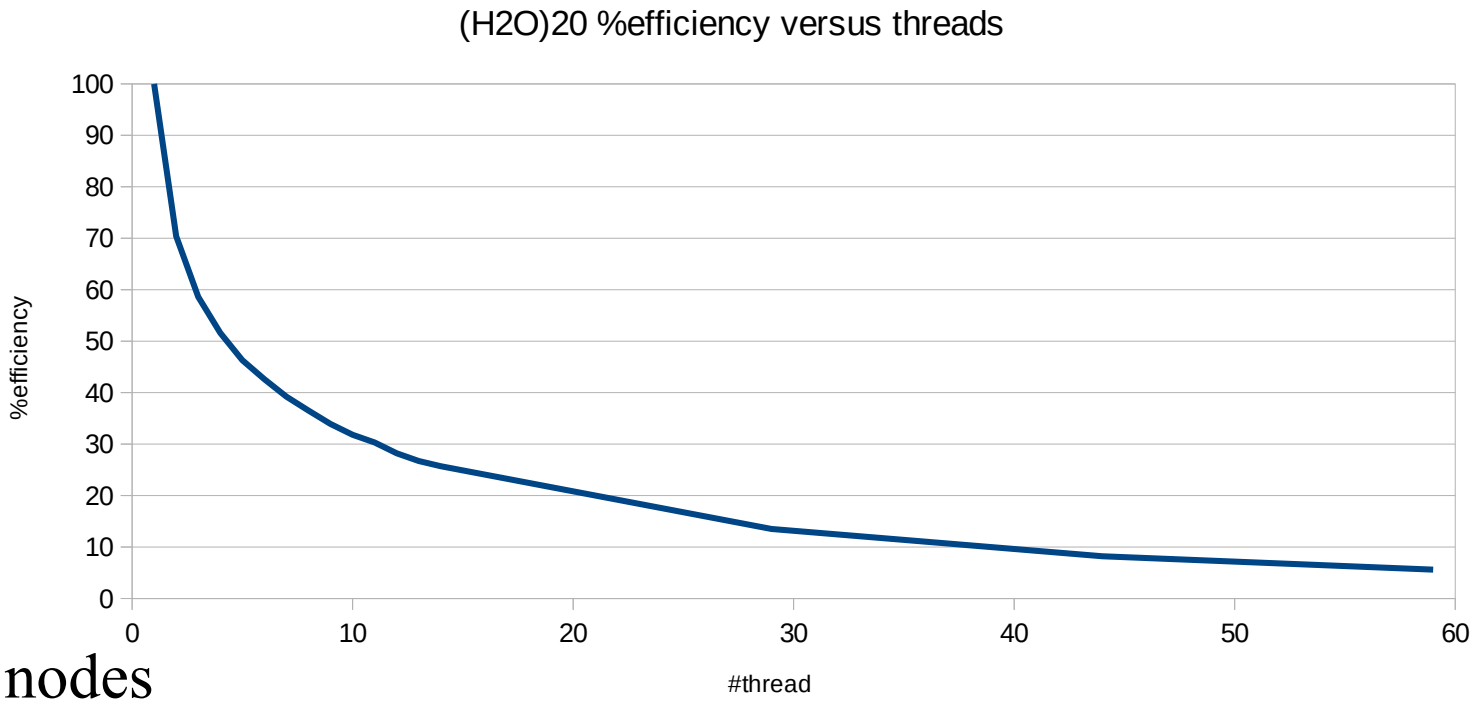
Benefit of tuned mTxm in BG/P Performance

Strong scaling

Molecular system with 13 heavy atoms, DFT, k=8, one iteration



Preliminary scaling w.r.t # threads



128 nodes

Clearly a problem in task queue … it was designed for Cray XT5

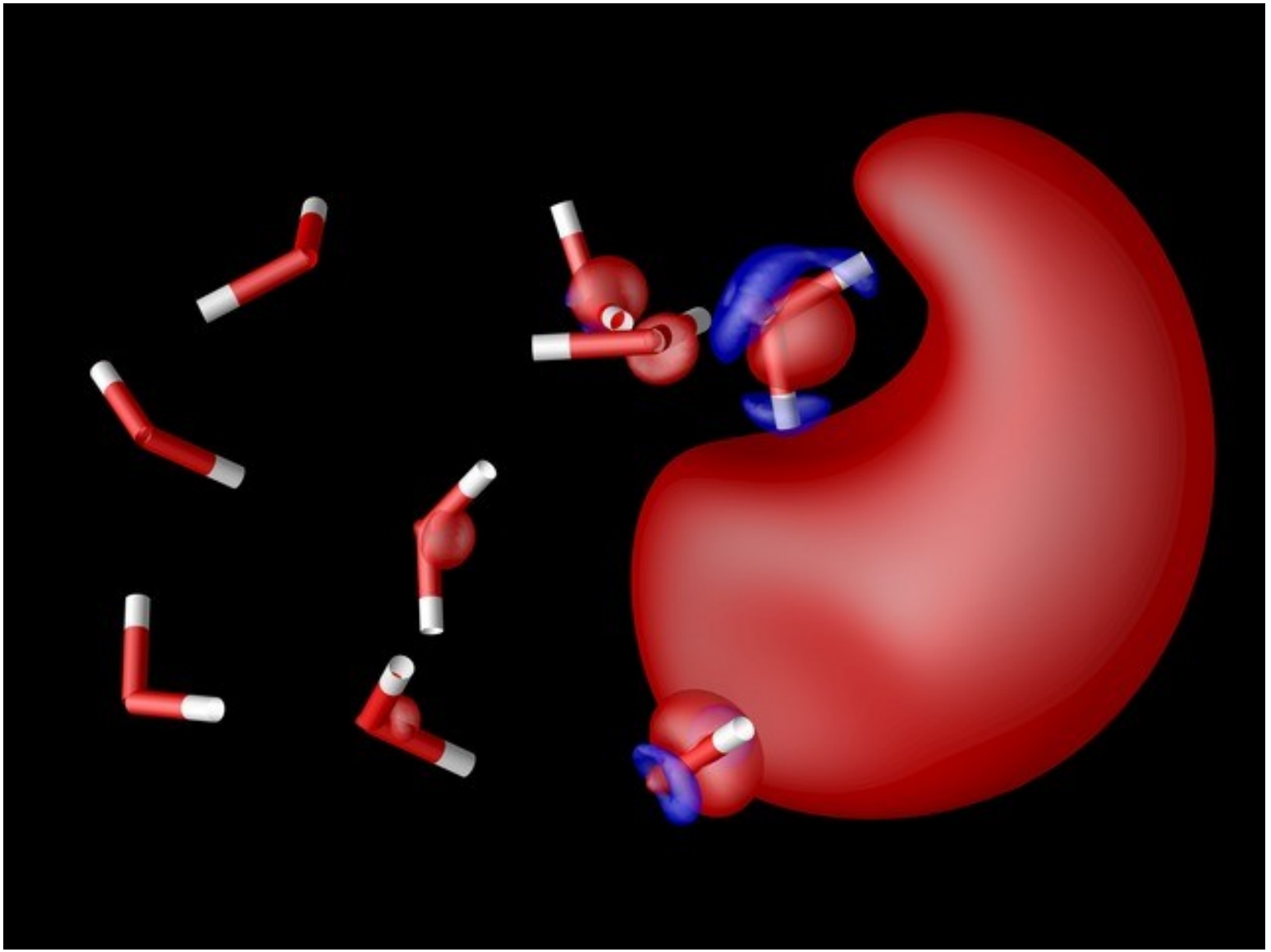
Different memory architecture, max of 11 worker threads

Tests with Intel TBB suggest it is vastly more scalable

Current molecular capabilities

- Hartree-Fock and DFT (LDA, GGA, Hybrid)
  - Energies
  - Derivatives
  - Frequencies (Bischoff)
  - Linear response (excited states, dynamic polarizability)
  - Essentially complete basis – no problems with gauge and Hellmann-Feynman theorem applies directly (with a little care
- In progress
  - Quadratic response (Sundahl)
    - Raman, hyper-Raman, excited state forces
  - Relativistic Hamiltonians (Anderson)

Molecular Electronic Structure



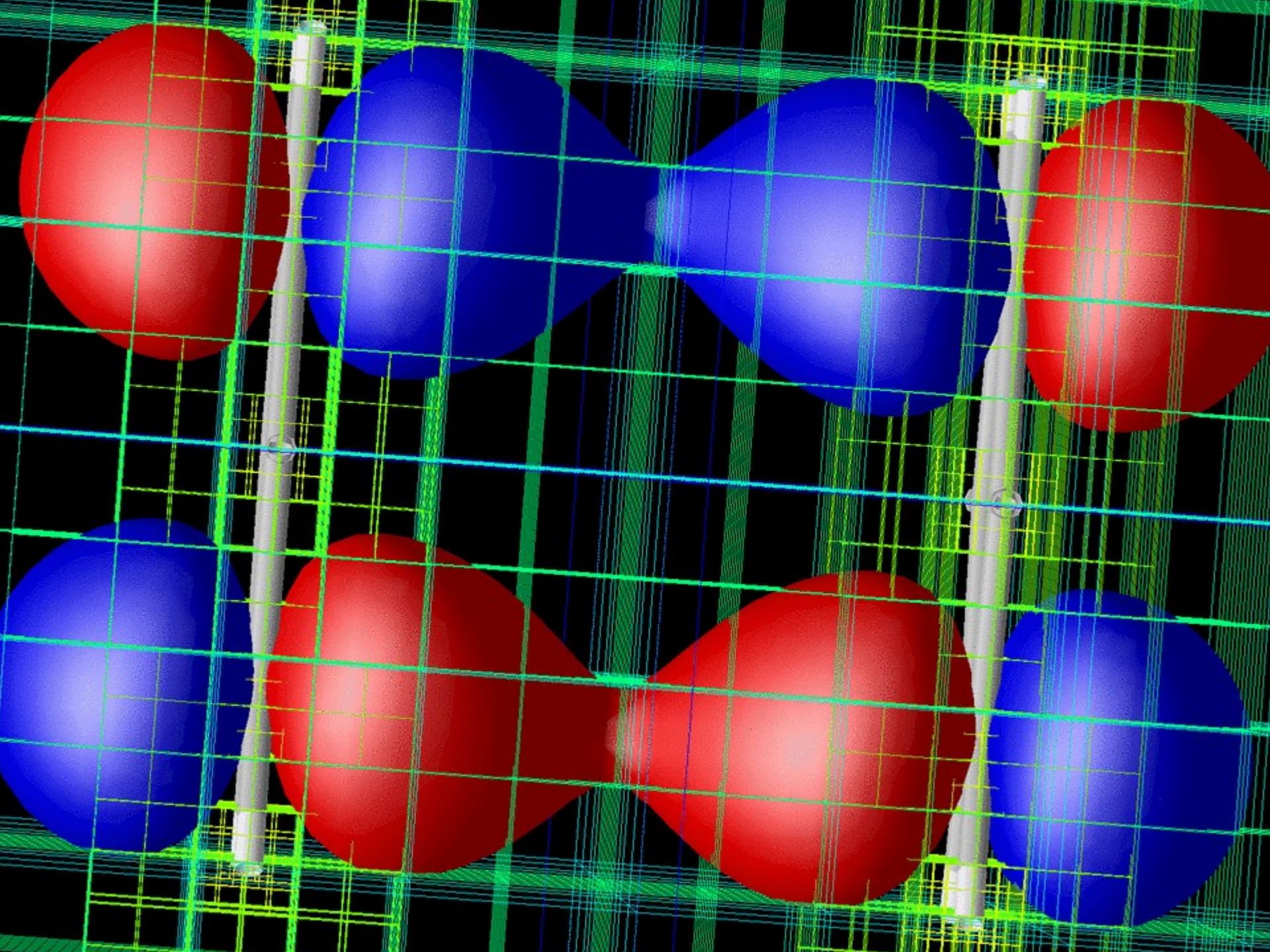
Energy and gradients

ECPs coming (Sekino, Kato)

Response properties (Vasquez and Sekino)

Still not as functional as previous Python version

Spin density of solvated electron



Path to linear scaling HF & DFT

- Need speed and precision
  - Absolute error cost $O\left(N \ln N / \epsilon\right)$
  - Relative error cost $O\left(N \ln 1 / \epsilon\right)$
- Coulomb potential
- HF potential
- Orbital update
- Orthogonalization and or diagonalization
- Linear response properties

HF Exchange (T. Yanai)

- HF or exact exchange
  - Features in the most successful XC functionals

$$\hat{K} f(x) = \sum_{i}^{\text{occupied}} n_i \phi_i(x) \int dy \frac{\phi_i(y) f(y)}{|x-y|}$$

  - Invariant to unitary rotation of occupied states with same occupation number
  - Localize the orbitals – only O(1) products but potential is still global
  - Compute potential only where orbital non-zero
    - Cost to apply to all orbitals circa O(N)

Orbital update

- Directly solve for localized orbitals that span space of occupied eigenfunctions
  - Rigorous error control from MRA refinement
  - Never construct the eigenfunctions
  - Update only diagonal multipliers
    - Off diagonal from localization process

$$\phi_i(x) = -(\hat{T} - \zeta)^{-1}\left((V + \zeta)\phi_i - \sum_j^{\text{occupied}} \phi_j(x)\epsilon_{ji}\right)$$

Inner products

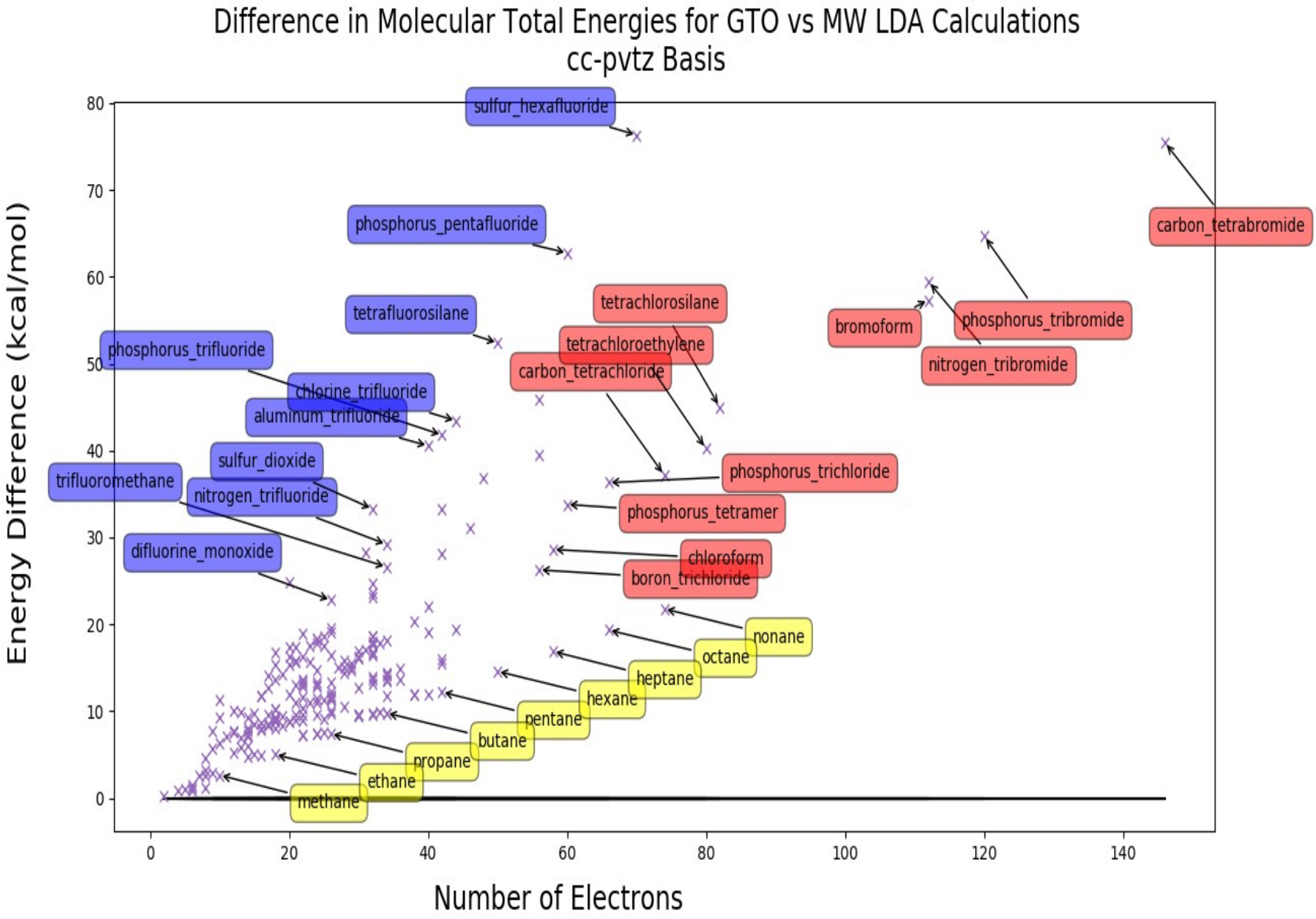
- The most expensive term for plane wave codes leading to cost O(N^2 M)
- Much less expensive in MRA basis

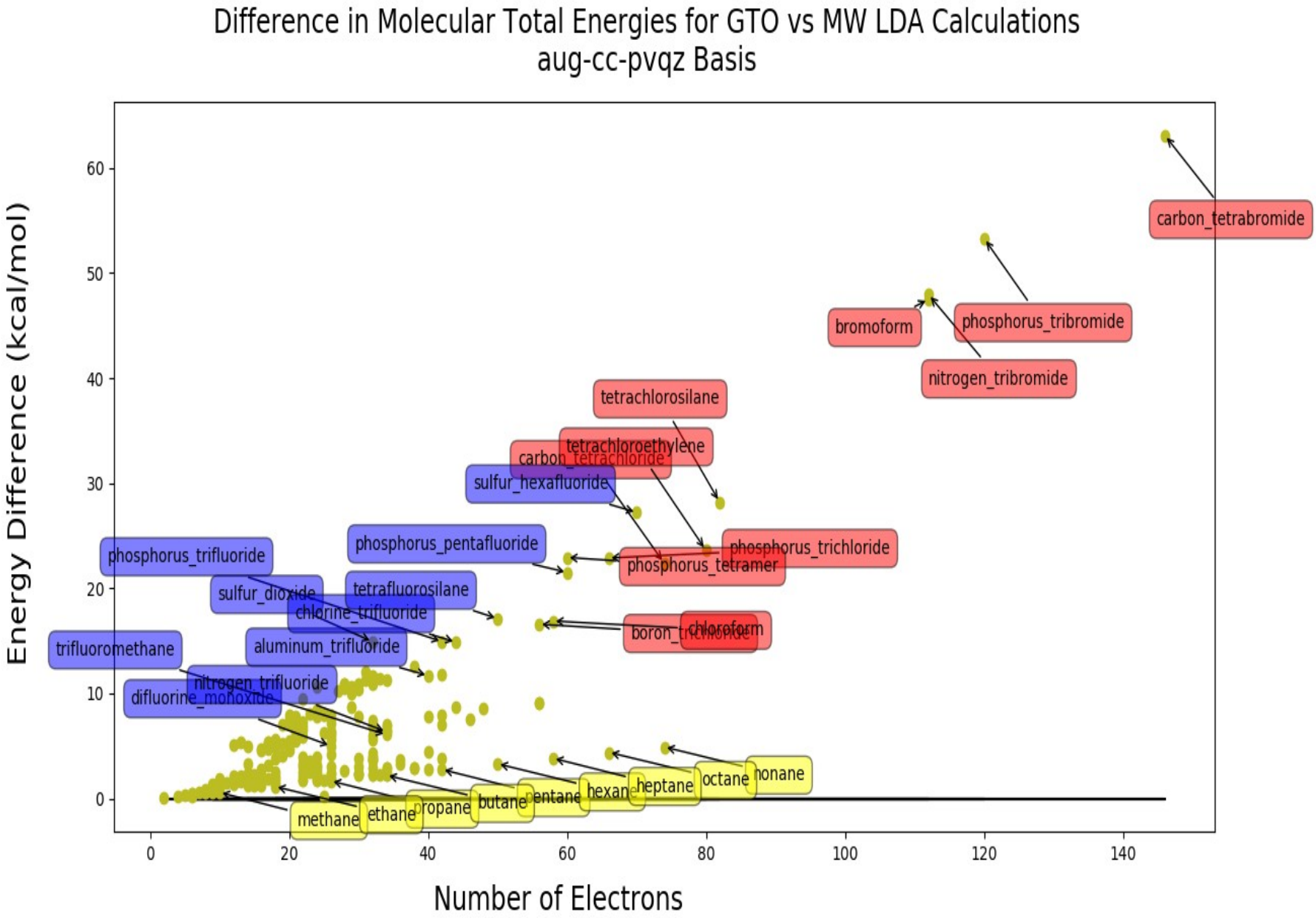
$$\langle f(x)|g(x)\rangle = s_f^{00}.s_g^{00} + \sum_{n=0}^{2^n-1}\sum_{l=0} d_f^{nl}.d_g^{nl}$$

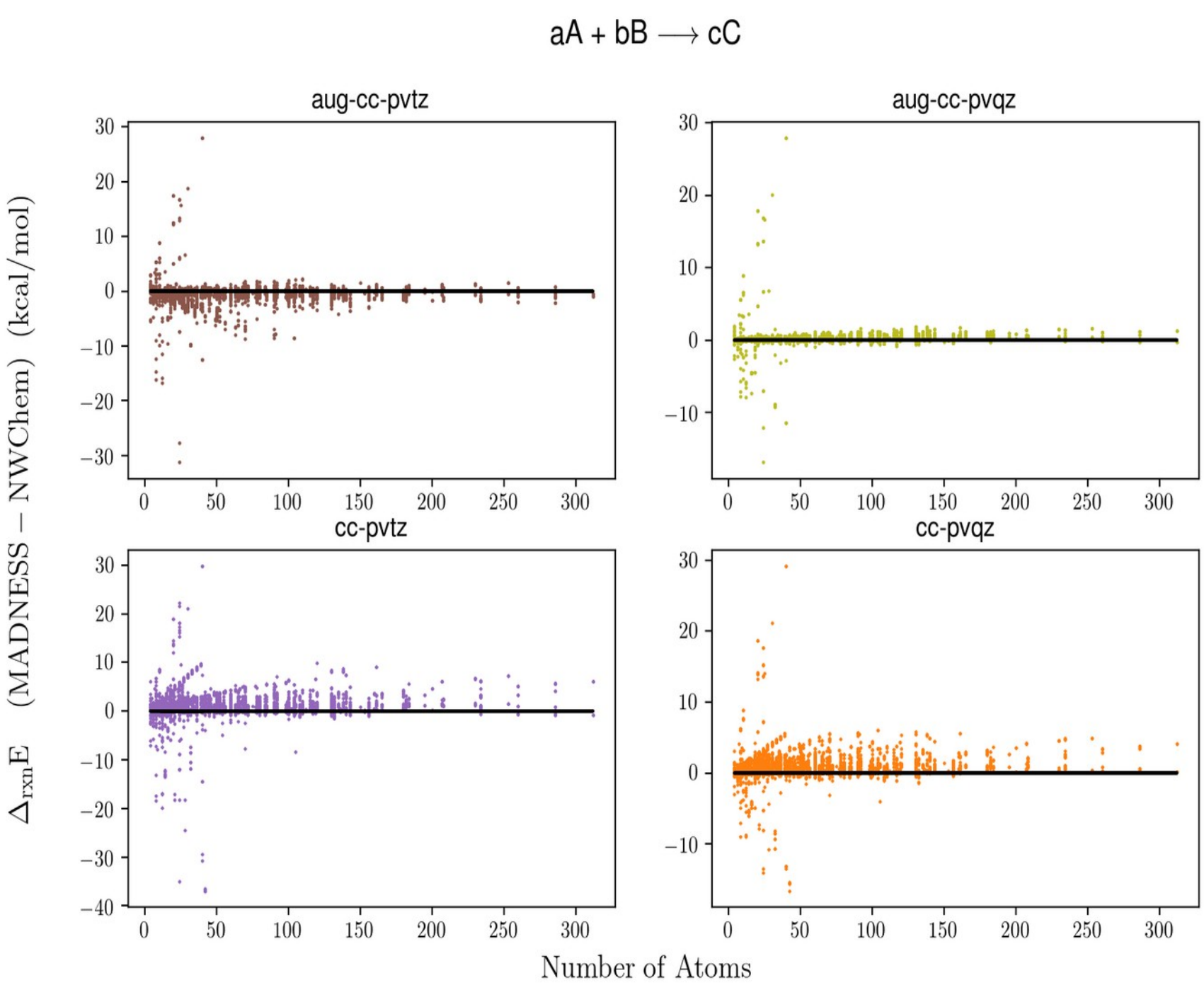
  - Orthogonal basis from local adaptive refinement implies zero/reduced work if
    - Functions do not overlap
    - Functions locally live at different length scales

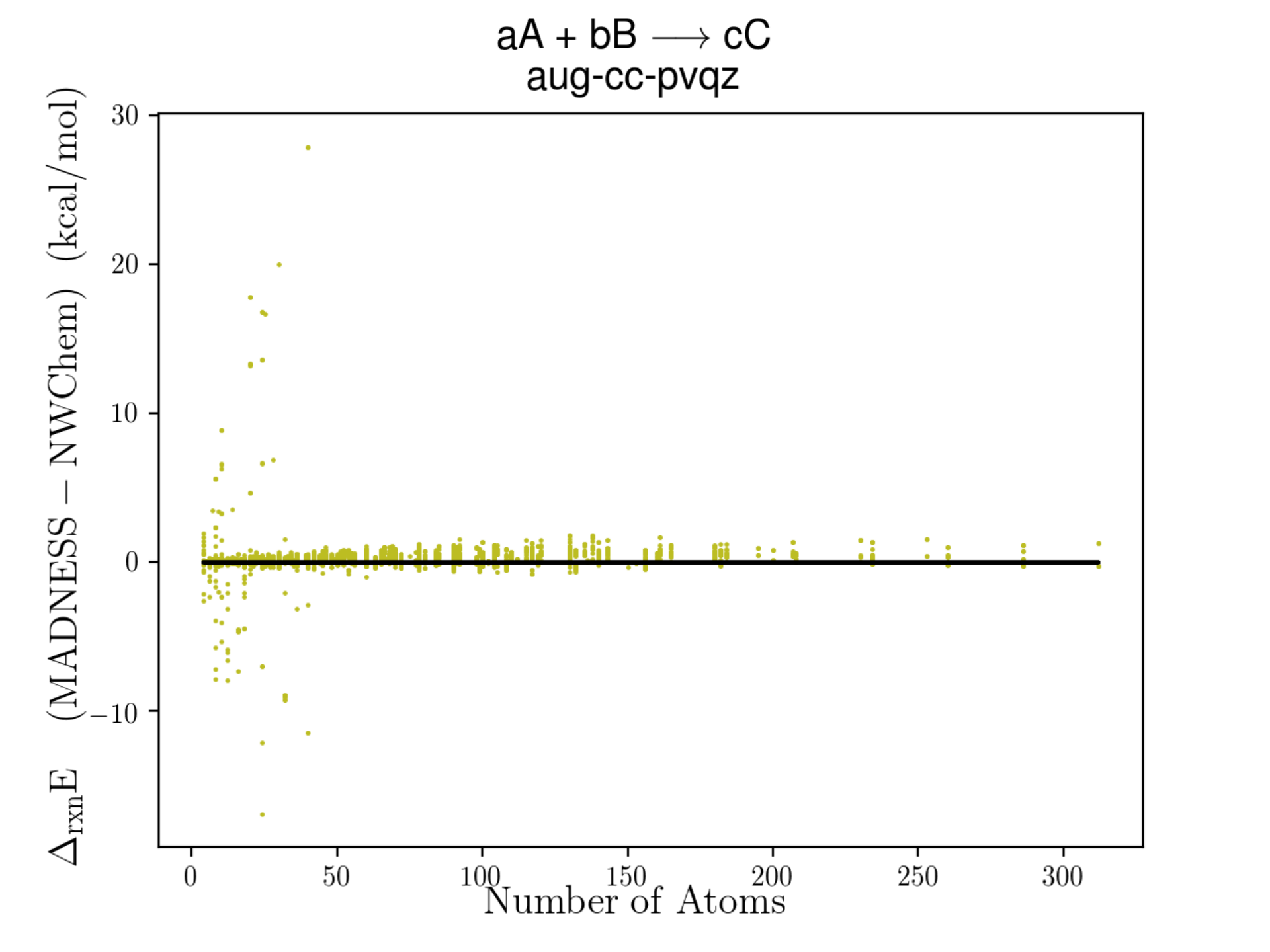
Comparing MRA and Gaussian basis results

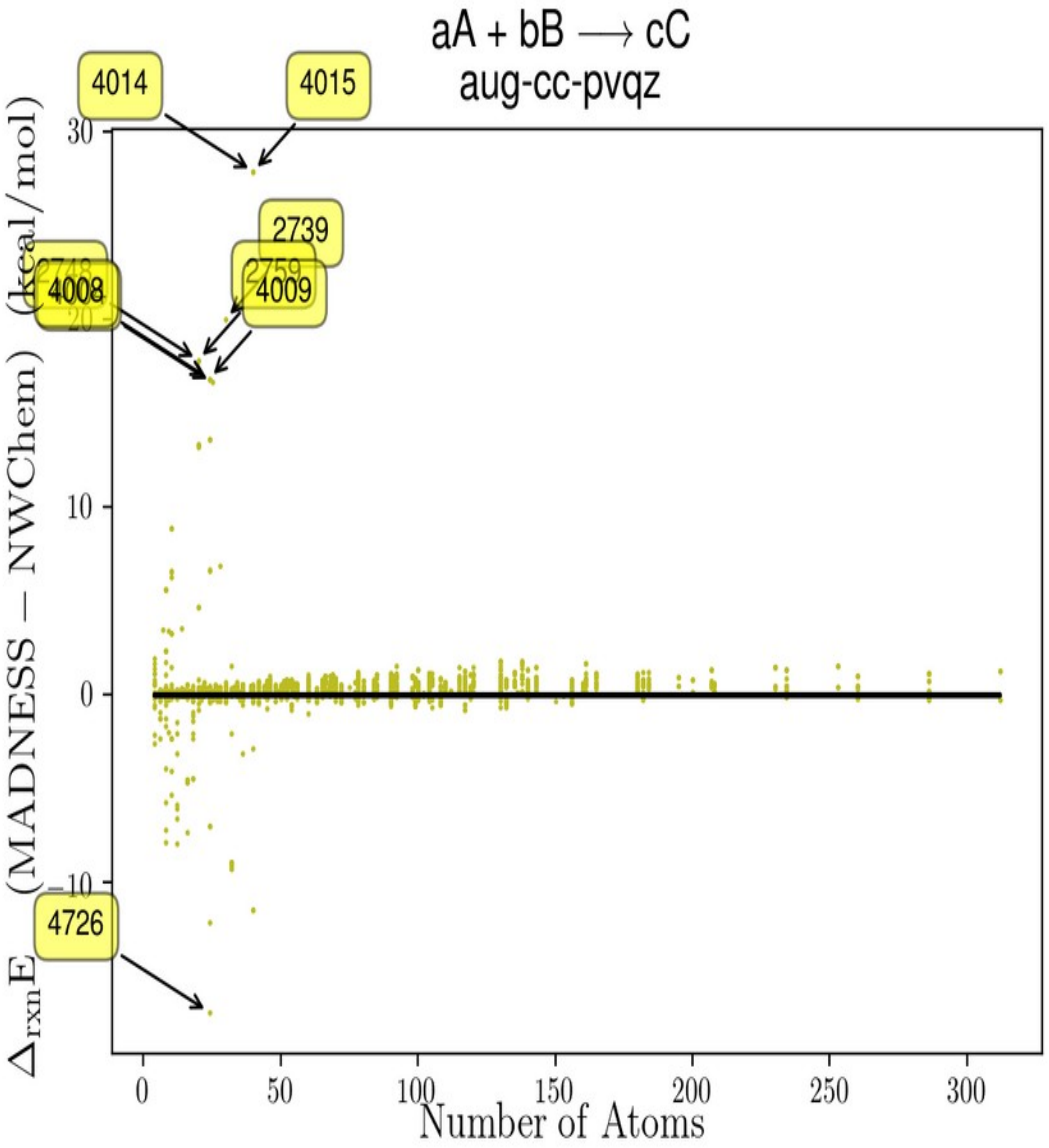
Emily Gentles (U. Ark), Colin Bunner (U. MN), Joel Anderson, Bryan Sundahl


Difference in Molecular Total Energies for GTO vs MW LDA Calculations
cc-pvtz Basis
Energy Difference (kcal/mol)
Number of Electrons
sulfur_hexafluoride
phosphorus_pentafluoride
tetrafluorosilane
phosphorus_trifluoride
chlorine_trifluoride
aluminum_trifluoride
sulfur_dioxide
trifluoromethane
nitrogen_trifluoride
difluorine_monoxide
tetrachlorosilane
tetrachloroethylene
carbon_tetrachloride
phosphorus_trichloride
phosphorus_tetramer
chloroform
boron_trichloride
bromoform
phosphorus_tribromide
nitrogen_tribromide
carbon_tetrabromide
nonane
octane
heptane
hexane
pentane
butane
propane
ethane
methane
0
10
20
30
40
50
60
70
80
0
20
40
60
80
100
120
140

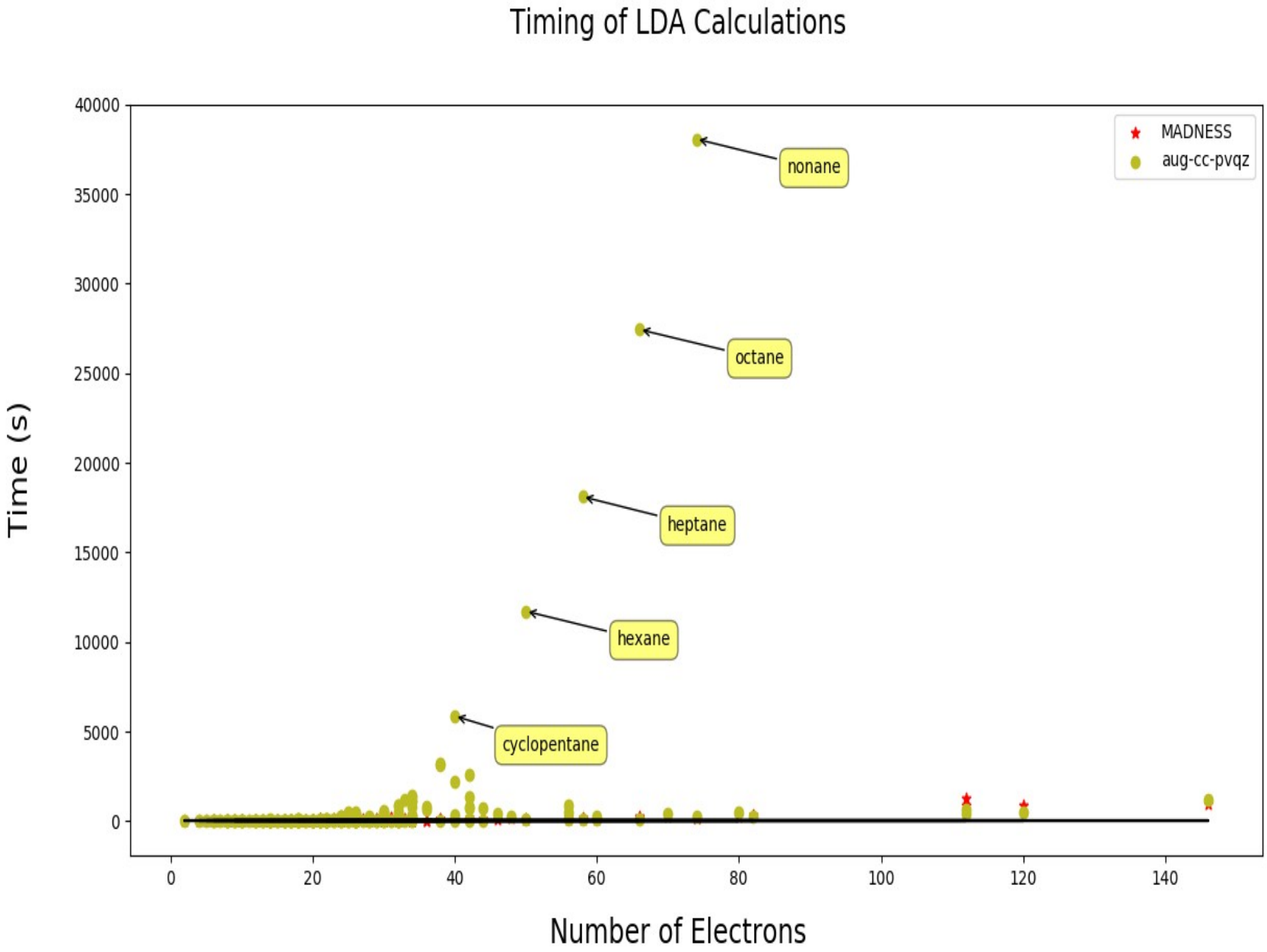

Difference in Molecular Total Energies for GTO vs MW LDA Calculations
aug-cc-pvqz Basis
Energy Difference (kcal/mol)
Number of Electrons
0
10
20
30
40
50
60
0
20
40
60
80
100
120
140
carbon_tetrabromide
bromoform
phosphorus_tribromide
nitrogen_tribromide
tetrachlorosilane
tetrachloroethylene
carbon_tetrachloride
sulfur_hexafluoride
phosphorus_pentafluoride
phosphorus_trichloride
phosphorus_tetramer
phosphorus_trifluoride
sulfur_dioxide
tetrafluorosilane
chlorine_trifluoride
boron_trichloride
chloroform
trifluoromethane
aluminum_trifluoride
nitrogen_trifluoride
difluorine_monoxide
nonane
octane
heptane
hexane
pentane
butane
propane
ethane
methane

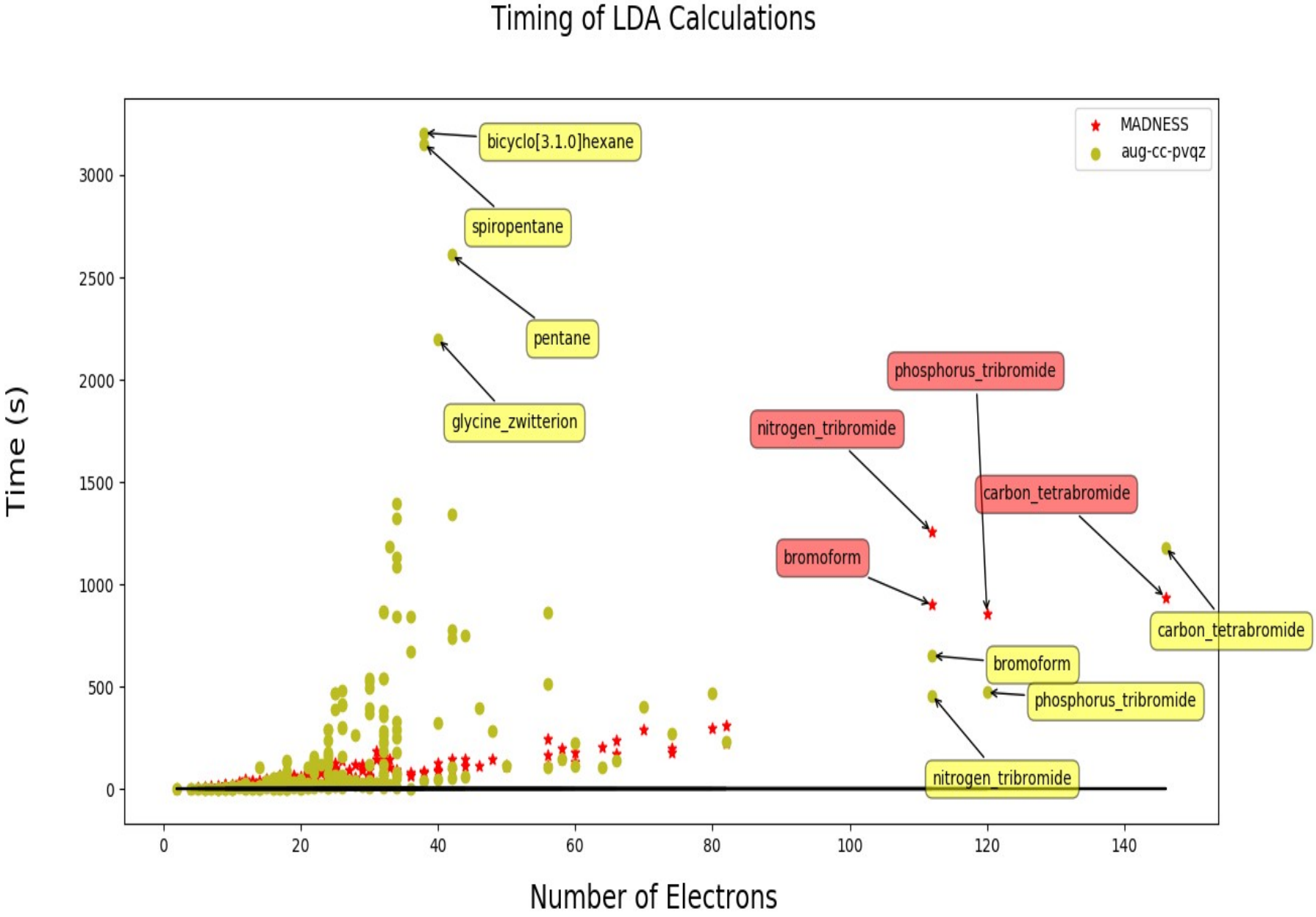

aA + bB ⟶ cC
aug-cc-pvtz
aug-cc-pvqz
cc-pvtz
cc-pvqz
Δ_{rxn}E (MADNESS − NWChem) (kcal/mol)
Number of Atoms

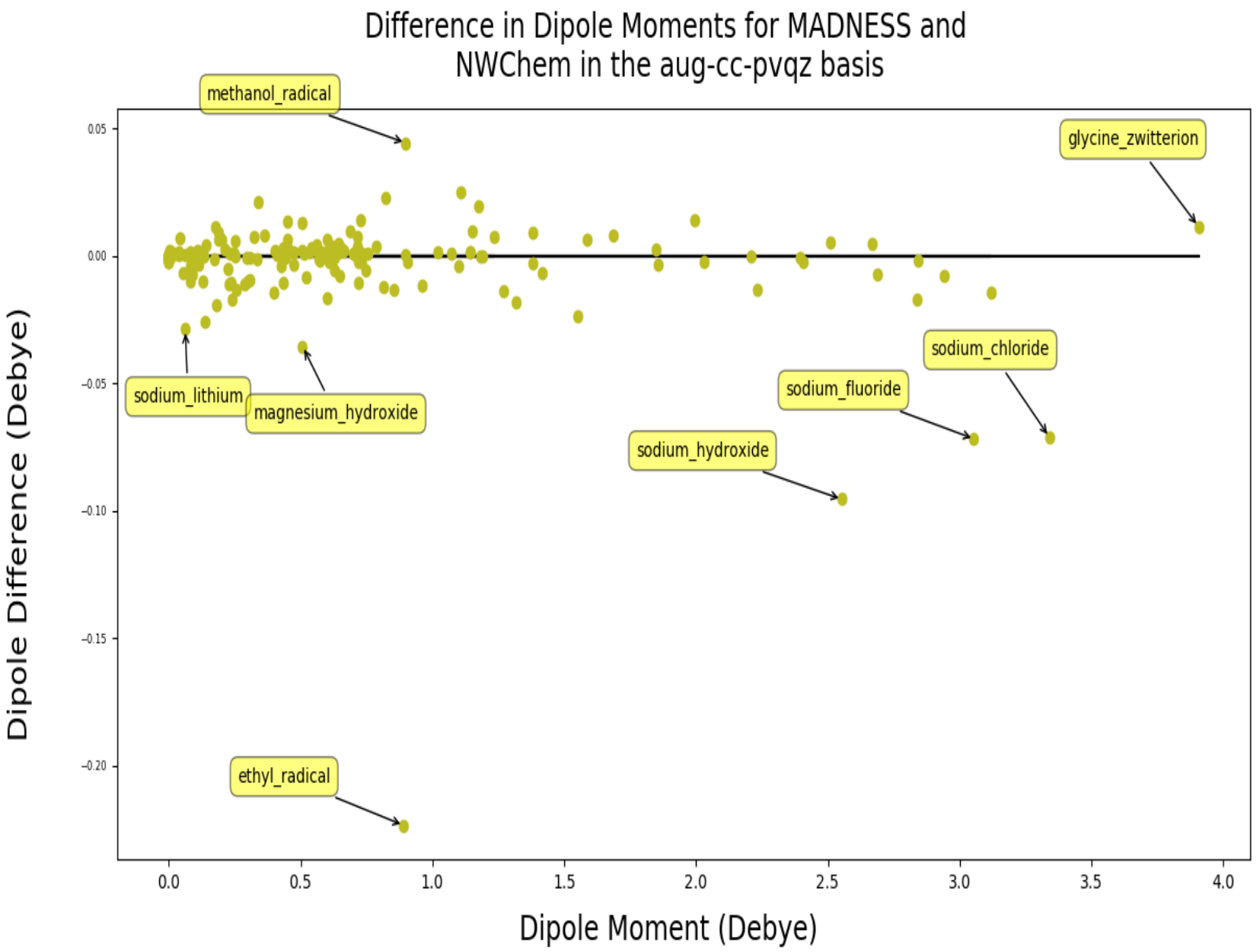

aA + bB ⟶ cC
aug-cc-pvqz
$\Delta_{rxn}E$ (MADNESS − NWChem) (kcal/mol)
30
20
10
0
−10
0
50
100
150
200
250
300
Number of Atoms



2739: 3 ethanol + ozone -->
6 methanol_radical 20.0409 kcal/mol
2748: 2 ethanol_radical + hydrogen_peroxide -->
4 methanol_radical 17.8414 kcal/mol
2759: 2 ethyl_radical + hydrogen_peroxide -->
4 methanol_radical 17.8414 kcal/mol
4004: 3 methanol + acetate_anion -->
5 methanol_radical 16.662 kcal/mol
4008: 3 methanol_radical + propene -->
3 ethanol_radical 16.837 kcal/mol
4009: 3 methanol_radical + propene -->
3 ethyl_radical 16.837 kcal/mol
4014: 5 methanol_radical + cyclopentane -->
5 ethanol_radical 27.890 kcal/mol
4015: 5 methanol_radical + cyclopentane -->
5 ethyl_radical 27.890 kcal/mol
4726: 5 sulfur_diatomic + 2 sulfur_hexafluoride -->
12 monosulfur_monofluoride -16.883 kcal/mol

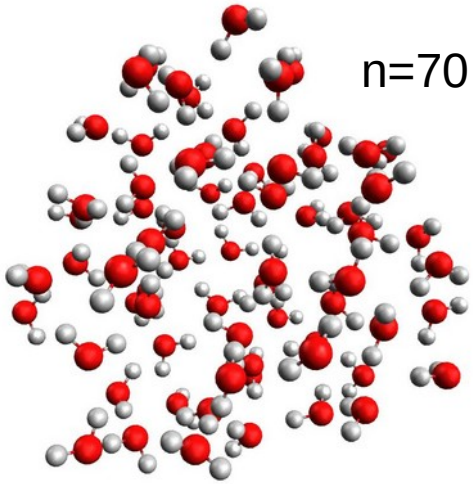

Timing of LDA Calculations
Time (s)
Number of Electrons
MADNESS
aug-cc-pvqz
nonane
octane
heptane
hexane
cyclopentane
40000
35000
30000
25000
20000
15000
10000
5000
0
0
20
40
60
80
100
120
140

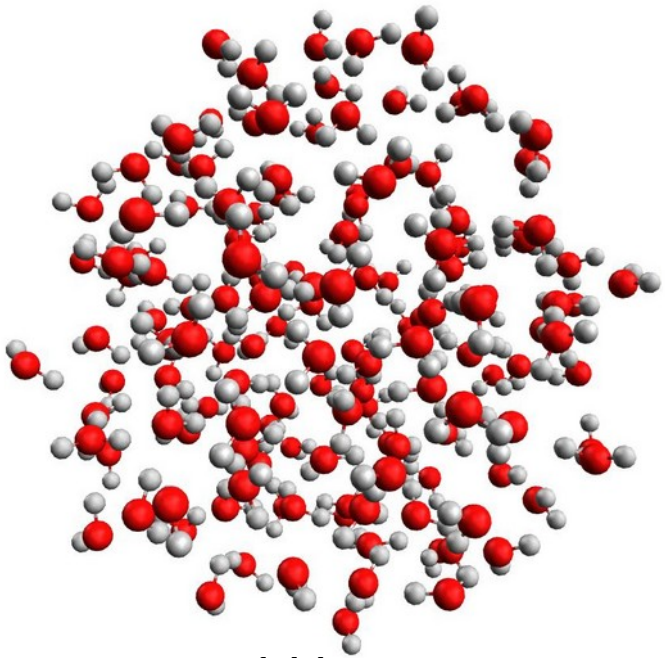

Timing of LDA Calculations
MADNESS
aug-cc-pvqz
bicyclo[3.1.0]hexane
spiropentane
pentane
glycine_zwitterion
phosphorus_tribromide
nitrogen_tribromide
carbon_tetrabromide
bromoform
carbon_tetrabromide
bromoform
phosphorus_tribromide
nitrogen_tribromide
Time (s)
Number of Electrons
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100
120
140

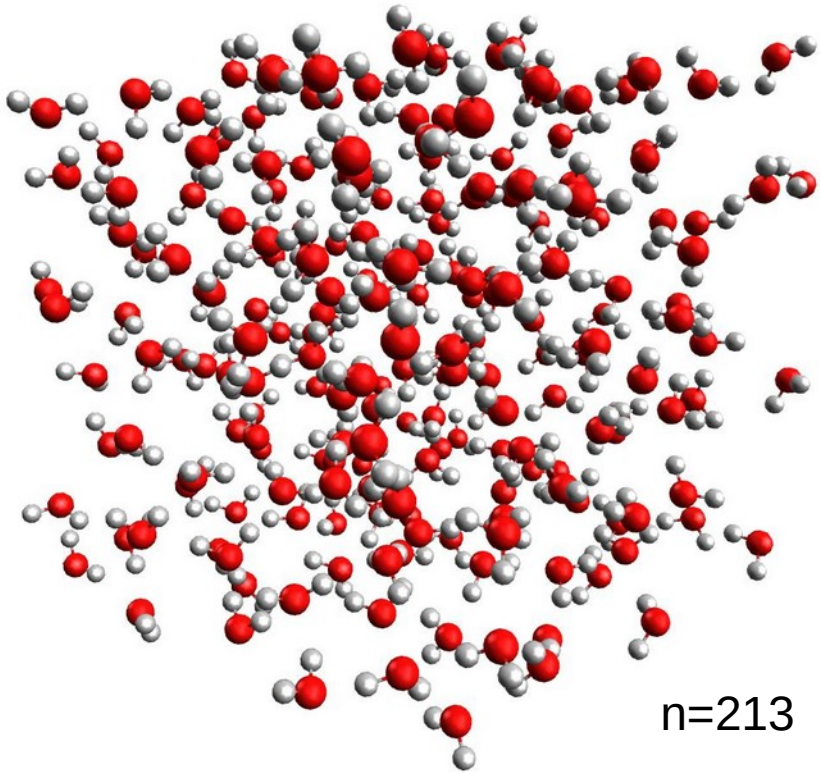

Difference in Dipole Moments for MADNESS and NWChem in the aug-cc-pvqz basis
methanol_radical
glycine_zwitterion
sodium_chloride
sodium_fluoride
sodium_hydroxide
sodium_lithium
magnesium_hydroxide
ethyl_radical
Dipole Difference (Debye)
Dipole Moment (Debye)
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Water clusters

- Classical STP MD simulation with cube of 213 waters
  - Smaller spherical clusters cut from interior







Scaling – LDA $(H_2O)_n$

n	Total	V*psi	Apply G	Tmat	Vmat	Ortho g	Localize	Deriv
70	256	2.0	8.7	7.8	0.3	0.7	2.7	1.9
144	744	4.2	20.9	49.2	0.9	2.9	11.9	5.5
213	1537	7.3	30.8	79.4	2.2	5.7	26.2	8.8
α	1.9	1.4	1.0	1.2	2.3	1.7	2.0	1.2

$t(n) = a n^{\alpha}$ α fitted to points n=144,213

Times in seconds running on 120 nodes, dual socket Intel Haswell, 2GHz, Infiniband FDR 40Gbit
Total is full time for convergence+derivatives; component times from last full iteration

TDDFT and CIS
T. Yanai with N.C. Handy

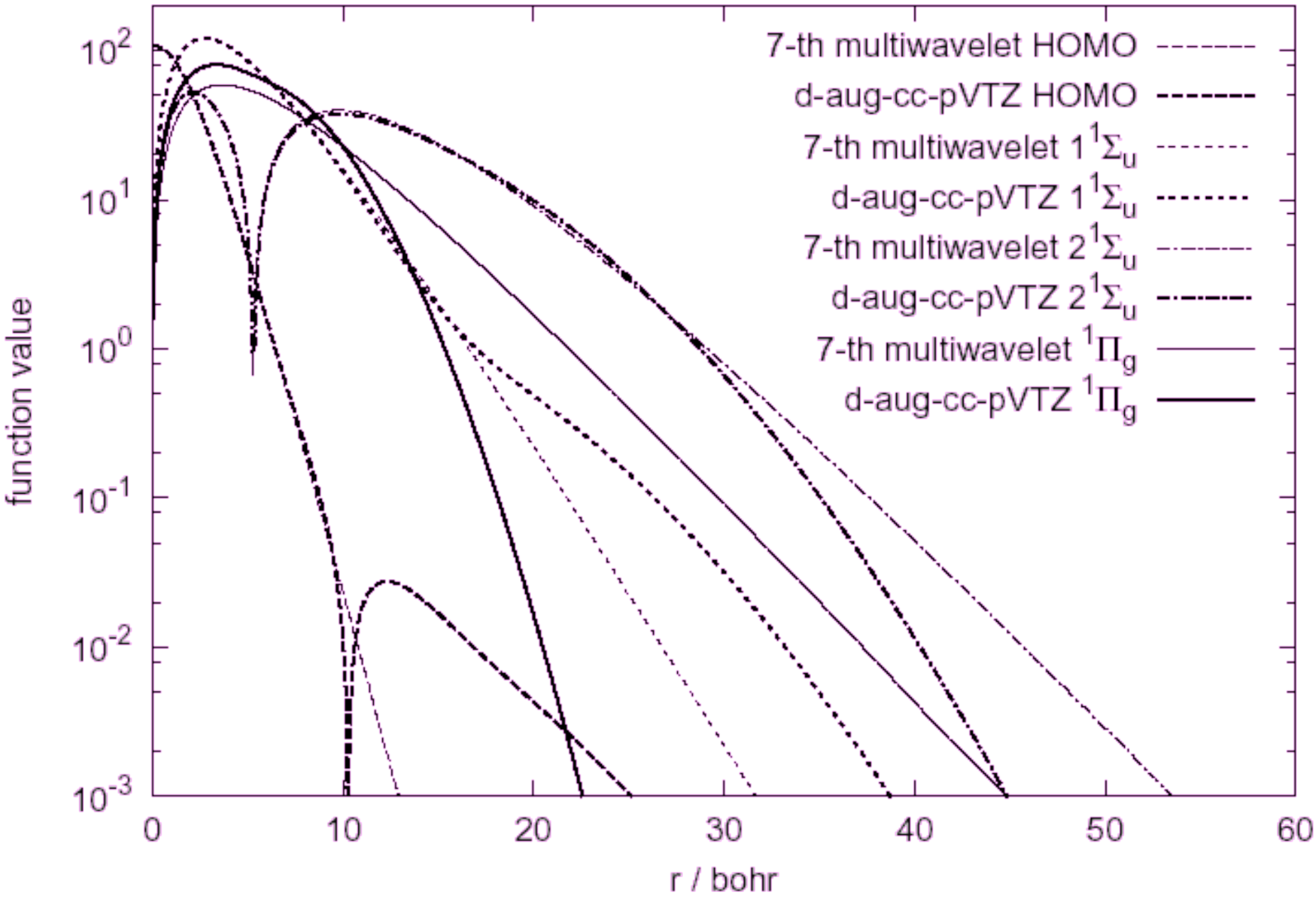
- Solve directly for the orbital response

$$\left(1-\hat{\rho}^{0}\right)\left[\left(\hat{F}^{0}-\epsilon_{p}^{0}\right)x_{p}\left(r\right)+\left\{\frac{\partial\hat{g}}{\partial\rho}\left[\rho^{0}\right]*\left(\sum_{i}^{occ}x_{i}\left(r\right)\phi_{i}^{\dagger}\left(r'\right)+\sum_{i}^{occ}\phi_{i}\left(r\right)y_{i}^{\dagger}\left(r'\right)\right)\right\}\phi_{p}\left(r\right)\right]$$
$$=\omega\, x_{p}\left(r\right),$$

$$\left(1-\hat{\rho}^{0}\right)^{\dagger}\left[\left(\hat{F}^{0}-\epsilon_{p}^{0}\right)^{\dagger}y_{p}\left(r\right)+\left\{\frac{\partial\hat{g}}{\partial\rho}\left[\rho^{0}\right]*\left(\sum_{i}^{occ}x_{i}\left(r\right)\phi_{i}^{\dagger}\left(r'\right)+\sum_{i}^{occ}\phi_{i}\left(r\right)y_{i}^{\dagger}\left(r'\right)\right)\right\}^{\dagger}\phi_{p}\left(r\right)\right]$$
$$=-\omega\, y_{p}\left(r\right),$$

  - Neglect y for CIS or Tamm-Dancoff

H_2 HOMO and CIS excited states



		$1^3\Sigma_u$	$1^3\Sigma g$	$^3\Pi_u$	$2^3\Sigma_u$	$2^3\Sigma_g$	$^3\Pi_g$
HF	**aug-cc-pVTZ**	**9.5520**	**11.958**	**12.81**	**14.36**	**16.98**	**17.74**
	aug-cc-pVQZ	**9.5529**	**11.961**	**12.59**	**14.29**	**16.32**	**16.79**
	d-aug-cc-pVQZ	**9.5523**	**11.959**	**12.26**	**14.15**	**14.49**	**14.81**
	k=7, r< $3*10^{-4}$	**9.55178**	**11.95491**	**12.26**	**14.11**	**14.48**	**14.66**
	k=9, r< $3*10^{-6}$	**9.55176**	**11.95490**	**12.26**	**14.11**	**14.48**	**14.66**
LSDA	**aug-cc-pVTZ**	**9.94**	**10.65**	**12.46**	**12.40**	**15.76**	**16.37**
	aug-cc-pVQZ	**9.95**	**10.61**	**12.07**	**12.10**	**14.85**	**15.22**
	d-aug-cc-pVQZ	**9.93**	**10.31**	**10.94**	**10.72**	**11.19**	**11.98**
LSDA(AC)	**k=7, r< $3*10^{-4}$**	**10.53**	**12.26**	**12.35**	**14.11**	**14.45**	**14.53**
HCTH	**aug-cc-pVTZ**	**10.19**	**10.79**	**12.74**	**12.71**	**15.85**	**16.30**
	aug-cc-pVQZ	**10.19**	**10.76**	**12.28**	**12.42**	**14.89**	**15.14**
	d-aug-cc-pVQZ	**10.18**	**10.52**	**11.12**	**10.95**	**11.36**	**12.23**
HCTH(AC)	**k=7, r< $3*10^{-4}$**	**10.87**	**12.40**	**12.50**	**14.10**	**14.44**	**14.50**

H_2 low-lying triplet excitation energies in eV

Table 6: The total, HOMO (ϵ_h) energies, and ionization potential (I) (in Hartree) of C_2H_4.

	k=7, r(MO)$< 3 \times 10^{-5}$ a)				Tozer et al. b)
	LSDA	HCTH	CAM-B3LYP	PBE0	HCTH
Total energy	-77.863 098	-78.603 861	-78.575 697	-78.519 234	-78.578 26
ϵ_h	-0.255 545	-0.246 448	-0.340 295	-0.289 974	...
I	0.404 218	0.388 123	0.387 648	0.385 352	0.390
Total energy (AC)	-77.862 891	-78.603 593	-78.575 653	-78.519 176	-78.578 18
ϵ_h (AC)	-0.400 493	-0.384 269	-0.385 994	-0.383 516	...

a) The asymptotic correction uses $X = 3.0, Y = 4.0$ for eq. (16).

b) Ref. [6]. The basis: augmented Sadlej basis sets.

	k=7, r(MO)$< 3 \times 10^{-5}$ [a] [b]				Tozer et al.[c]		
	LSDA	HCTH	CAM-B3LYP	PBE0	HCTH	CASPT2[d]	Exptl.[d]
$^1B_{1u}$	9.686	9.288	9.343	9.278	9.32	9.31	9.33
$^1B_{2u}$	9.287	8.937	8.946	8.898	9.04	9.18	9.05
$^1B_{3u}$	9.270	9.057	9.107	9.038	8.95	9.03	8.90
$^1B_{3u}$	8.949	8.691	8.688	8.661	8.70	8.66	8.62
$^3B_{3u}$	8.895	8.583	8.639	8.586	8.64	8.57	8.57
^1A_g	8.737	8.393	8.384	8.338	8.33	8.40	8.28
^3A_g	8.487	8.071	8.128	8.035	8.15	8.26	8.15
$^1B_{1u}$	8.404	8.322	8.262	8.289	7.61	8.40	8.0
$^1B_{2g}$	8.161	7.907	7.926	7.903	7.77	7.95	7.90
$^1B_{1g}$	8.117	7.890	7.831	7.838	7.78	7.85	7.80
$^3B_{1g}$	8.072	7.812	7.790	7.802	7.76	7.80	7.79
$^1B_{3u}$	7.429	7.280	7.198	7.217	7.16	7.17	7.11
$^3B_{3u}$	7.332	7.190	7.101	7.081	7.10	7.05	6.98
$^3B_{1u}$	4.807	4.468	4.103	4.064	4.33	4.39	4.36
Error	0.34	0.11	0.10	0.10	0.07	0.09	
Max	0.46	0.32	0.26	0.30	0.39	0.40	

C_2H_4 excitation energies with asymptotically corrected potentials

Table 9: The excitation energies (in eV) of benzene.

	k=7, r(MO)< 3×10^{-5} [a) b)]			Handy et al. / HCTH [c)]		
	LSDA	HCTH	CAM-B3LYP	6-31G*	TZ2P	Exptl./CASPT2* [d)]
Valence excitations						
1^3B_{1u}	4.408	4.019	3.595	4.11	4.02	3.94
1^3E_{1u}	4.757	4.651	4.781	4.78	4.66	4.76
1^1B_{2u}	5.276	5.283	5.489	5.44	5.28	4.90
1^3B_{2u}	5.006	4.977	5.088	5.15	4.98	5.60
1^1B_{1u}	6.045	6.006	6.158	6.24	6.02	6.20
1^1E_{1u}	6.895	6.912	7.012	7.00	6.94	6.94
Rydberg excitations						
1^1E_{1g}	6.529	6.404	6.504	6.15	6.24	6.334
1^1A_{2u}	6.885	6.992	7.221	6.81	6.92	6.932
1^1E_{2u}	6.980	6.980	7.109	6.79	6.90	6.953
1^1A_{1u}	7.004	6.990	7.014	6.81	6.91	6.99*
2^1E_{1u}	7.426	7.360	7.297	7.34	7.24	7.41
1^1B_{1g}	7.783	7.546	7.702	7.39	7.50	7.460
1^1B_{2g}	7.811	7.561	7.715	7.41	7.52	7.460
2^1E_{1g}	7.614	7.451	7.539	7.41	7.52	7.535
1^1E_{2g}	8.071	7.729	7.874	7.66	7.76	7.81
2^1A_{1g}	8.085	7.747	7.924	7.63	7.80	7.81
1^1A_{2g}	8.079	7.746	7.877	7.63	7.78	7.81

Mean absolute deviations						
Valence	0.27	0.24	0.26	0.21	0.22	
Rydberg	0.17	0.06	0.14	0.13	0.06	
total	0.21	0.12	0.18	0.16	0.12	
Maximum absolute deviation						
Valence	0.59	0.62	0.59	0.54	0.62	
Rydberg	0.35	0.10	0.29	0.18	0.17	
total	0.59	0.62	0.59	0.54	0.62	

a) The computed energies are converged within 10^{-3} eV.

b) The asymptotic correction uses $X = 3.0$ and $Y = 4.0$ for eq. (16).

c) The asymptotic correction uses $X = 3.5$ and $Y = 4.7$ for eq. (16). The basis sets are augmented with the double spd (0,01, 0.04) set.

d) Ref. [26].

Mean abs. error of Gaussian results relative to MRA

augmented 6-31g* 0.14 eV

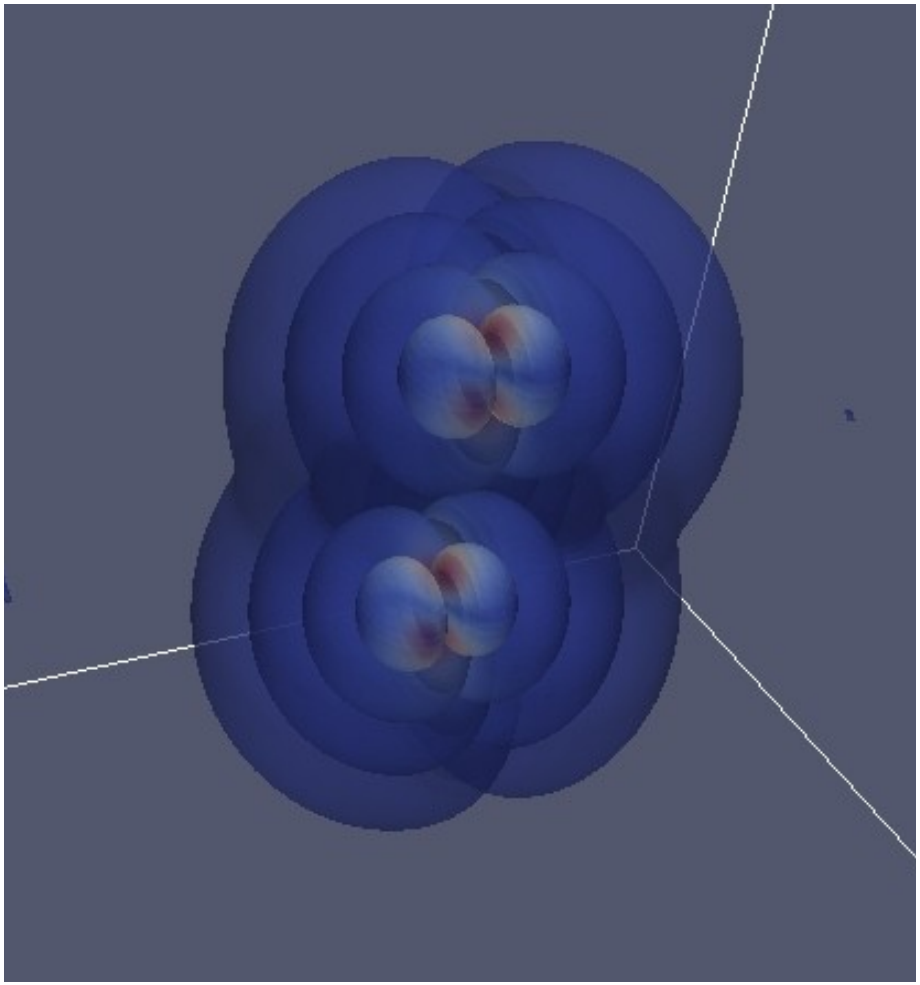
augmented TZ2P 0.05 eV

augmented TZ2P 0.01 eV (valence only)

Nuclear physics

J. Pei, G.I. Fann, Y. Ou,
W. Nazarewicz
UT/ORNL

- DOE UNDEF
- Nuclei & neutron matter
- ASLDA
- Hartree-Fock Bogliobulov
- Spinors
- Gamov states



Imaginary part of the seventh eigen function
two-well Wood-Saxon potential

Nuclear physics

J. Pei, G.I. Fann, W. Thornton
W. Nazarewicz
UT/ORNL

UNEDF Deformed SLDA in 3-D

deformation in an external trap with aspect ratio of 1/16

- DOE UNDEF
- Nuclei & neutron matter
- ASLDA
- Hartree-Fock Bogliobulov
- Spinors
- Gamov states

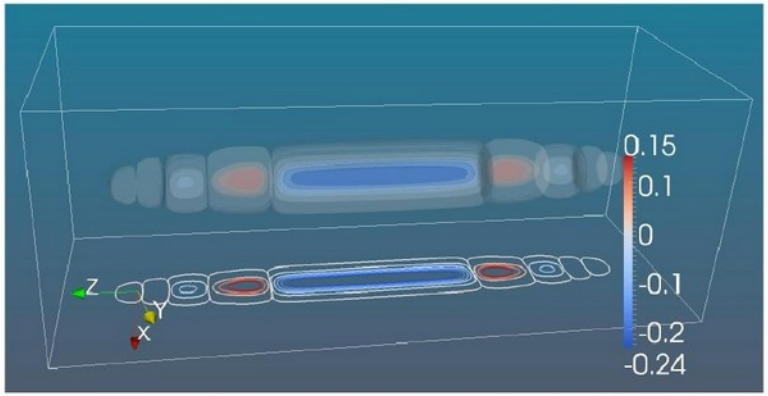
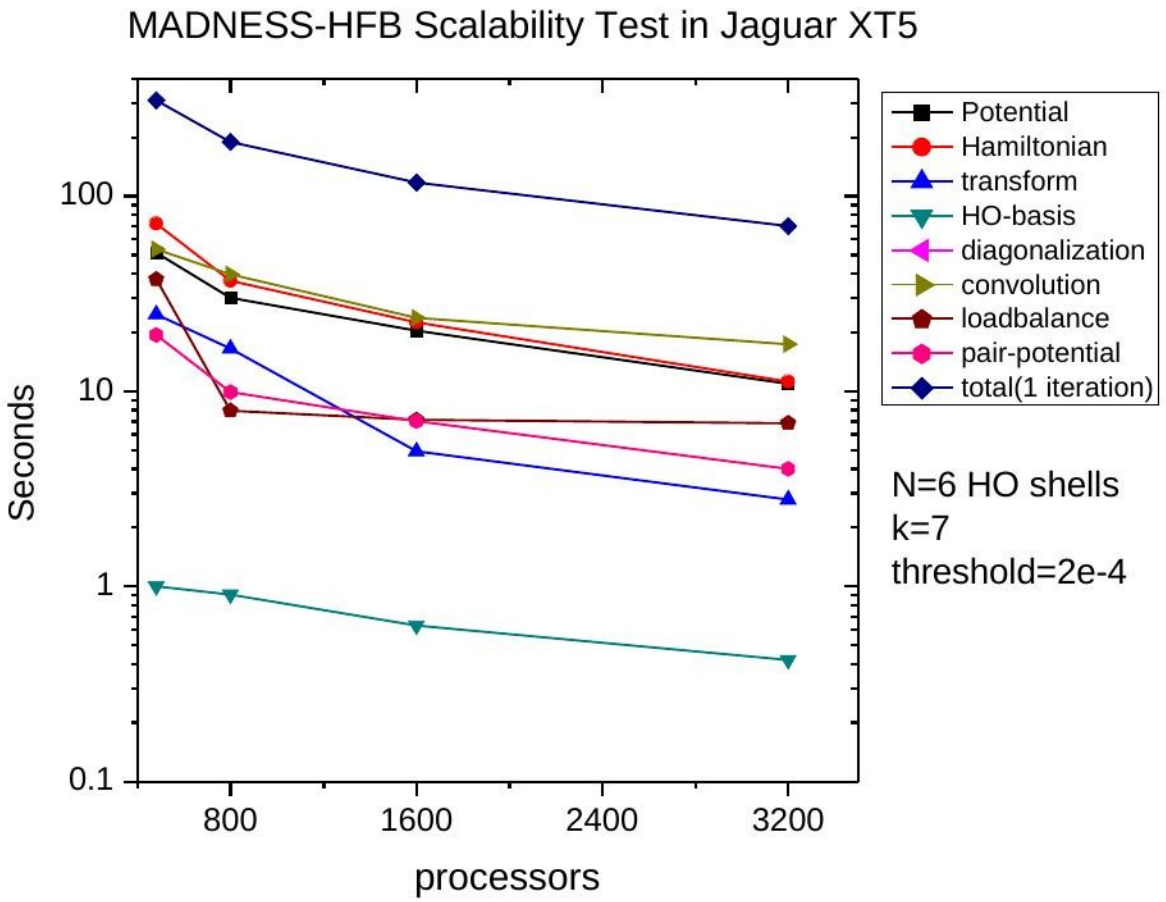


Figure 4. Pairing density $\kappa(x, y, z)$, calculated by MADNESS-HFB for the elongated trap wit $\eta = 16$. The box scale in view is x[-12, 12], y[-12, 12] and z[-32, 32].

UNEDF Solving HFB in Production



Realizable scaling even for small problems in 3-D

New HFB algorithm from UNEDF

Superfluidity: pairing entered

$$\begin{pmatrix} h(\boldsymbol{r}) - \mu_\uparrow & \Delta(\boldsymbol{r}) \\ \Delta^\dagger(\boldsymbol{r}) & -h(\boldsymbol{r}) + \mu_\downarrow \end{pmatrix} \begin{pmatrix} u_i(\boldsymbol{r}) \\ v_i(\boldsymbol{r}) \end{pmatrix} = E_i \begin{pmatrix} u_i(\boldsymbol{r}) \\ v_i(\boldsymbol{r}) \end{pmatrix}$$

Particle number condition

$$N_\uparrow = \sum_{-E_{cut}<E_i<0} \int |u_i(\mathbf{r})|^2$$

$$N_\downarrow = \sum_{0<E_i<E_{cut}} \int |v_i(\mathbf{r})|^2$$

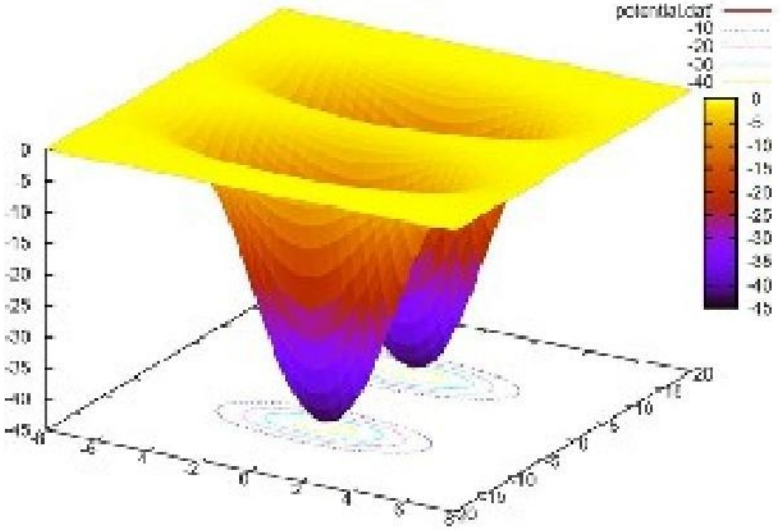
- Initial guess Wavefunction (HO tensor prod)
- Loop
  - Construct Hamiltonian (new, adapative representation) for HFB
  - Diagonalization
  - Solve Associated Lippmann-Schwinger Equation (new part from UNEDF)
- Iteration until convergence

SLDA density functional:

$$\mathcal{E}(\mathbf{r}) = \alpha \frac{\tau_c(\mathbf{r})}{2} + \beta \frac{3(3\pi^2)^{2/3} n^{5/3}(\mathbf{r})}{10} + g_{\mathit{eff}}(\mathbf{r}) |\nu_c(\mathbf{r})|^2 + V_{ext}(\mathbf{r}) n(\mathbf{r}),$$

UNEDF Two-cosh potential test

Benchmark comparison of multiwavelets with HO, B-spline



Precision Test with spin-orbit

State No.	Ω^π	HO N_{sh}=20	HO N_{sh}=30	B-spline h=0.6	Wavelets
1	$1/2^+$	−22.23916	−22.24008	−22.24011	−22.24011
2	$1/2^-$	−22.23816	−22.23995	−22.23998	−22.23998
3	$1/2^+$	−9.43145	−9.43659	−9.43663	−9.43662
4	$3/2^-$	−9.42314	−9.43199	−9.43203	−9.43202
5	$3/2^+$	−9.42561	−9.43078	−9.43081	−9.43080
6	$1/2^-$	−9.41931	−9.42783	−9.42788	−9.42788
7	$1/2^+$	−8.77250	−8.77825	−8.77828	−8.77828
8	$1/2^-$	−8.76475	−8.77380	−8.77384	−8.77383
9	$1/2^+$	−1.70727	−1.72405	−1.72506	−1.72516
10	$1/2^-$	−1.49222	−1.52490	−1.52675	−1.52693

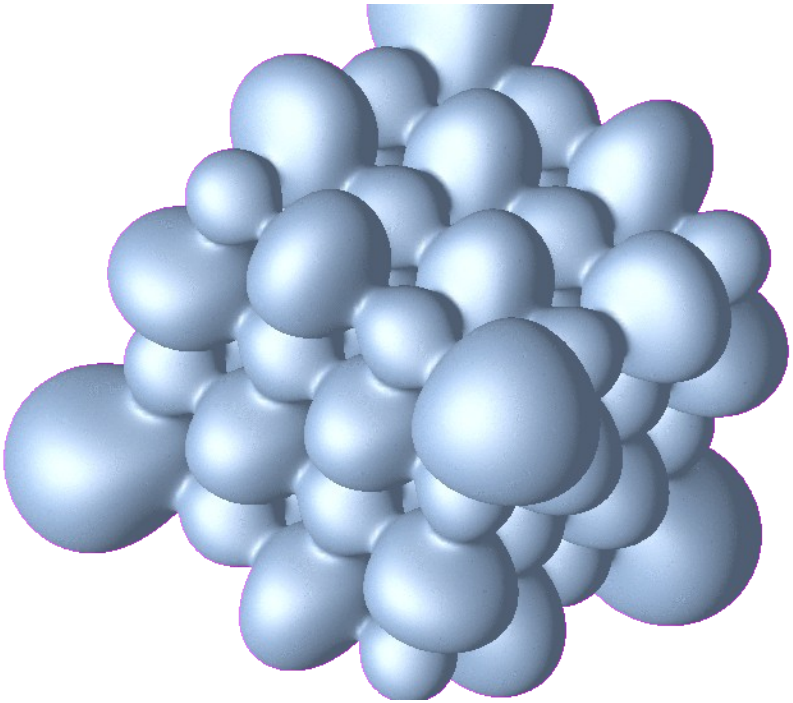
- Similar to nuclear fission problem

$$V_{so} = -\sqrt{-1}\lambda_0 \left(\frac{h}{2mc} \right)^2 \nabla V \bullet (\sigma \times \nabla)$$

$$h = -\frac{\hbar^2}{2m}\nabla^2 + V_{2\cosh}(x, y, z) + V_{so}(x, y, z)$$

Solid-state electronic structure

- Thornton, Eguiluz and Harrison (UT/ORNL)
  - NSF OCI-0904972: Computational chemistry and physics beyond the petascale
- Full band structure with LDA and HF for periodic systems
- In development: hybrid functionals, response theory, post-DFT methods such as GW and model many-body Hamiltonians via Wannier functions



Coulomb potential isosurface in LiF

Lattice sums
(Thornton, Beylkin, Harrison)

- Reduce range of integral to unit cell

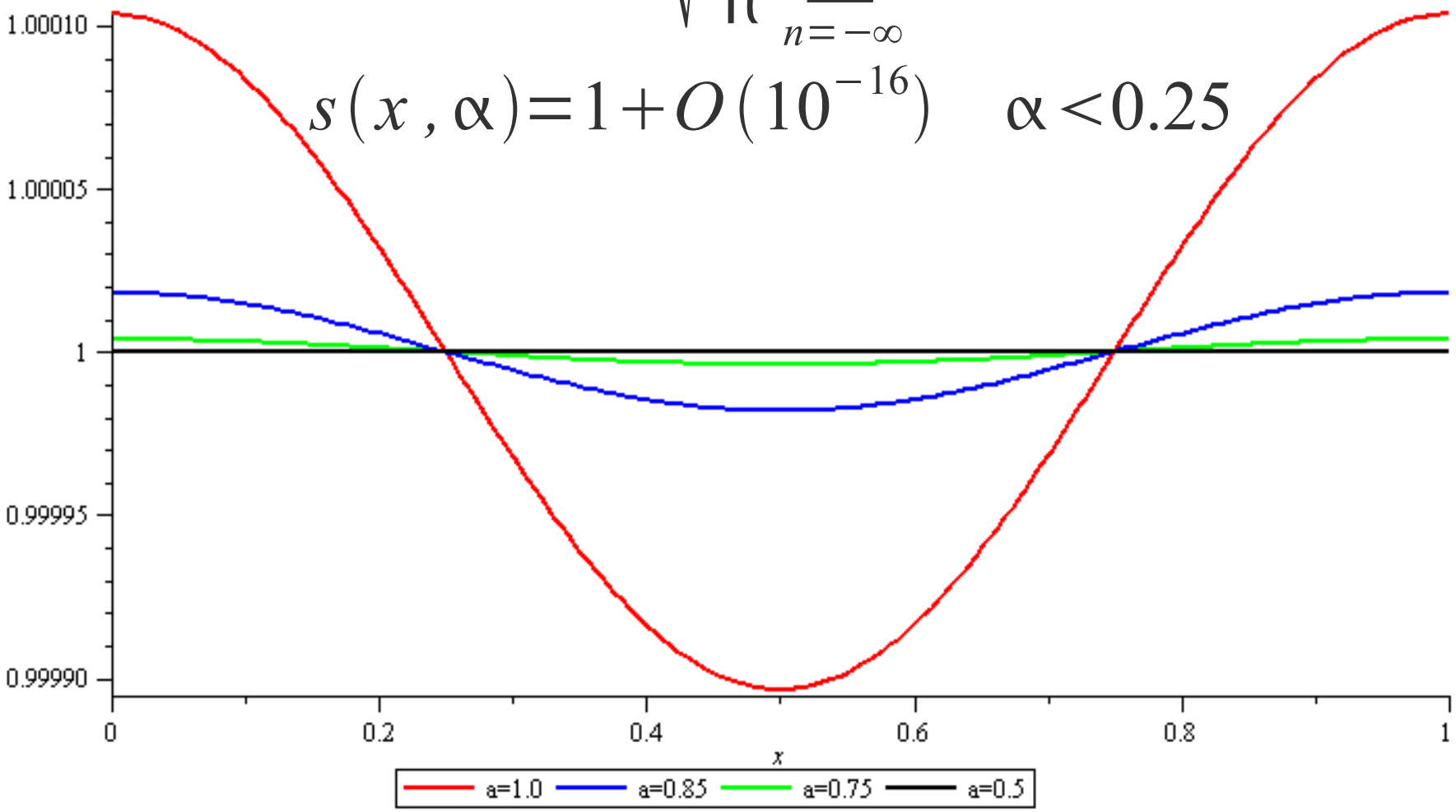
$$\int K(r-s)\sum_R f(s+R)\,ds = \int_\Omega f(s)\sum_R K(r-s+R)\,ds$$

- Expand kernel in Gaussians $e^{-\alpha x^2}$
- Diffuse components $\alpha < 0.25$
  - Trapezoidal rule on lattice points exact to machine precision for sufficiently diffuse Gaussians
  - Hence replace with constant (G=0)
- Real space sum out to $n = 6.1/\sqrt{\alpha} \leqslant 12.2$
  - Fast since is 1D due to separated representation

Lattice sum of Gaussians

$$s(x,\alpha)=\sqrt{\frac{\alpha}{\pi}}\sum_{n=-\infty}^{\infty} e^{-\alpha(x-n)^2}$$

$$s(x,\alpha)=1+O(10^{-16}) \quad \alpha<0.25$$



Lattice sums - II

- Fine scale structure inside any box, or molecule inside large box
  - Cost of periodic same as free space inside volume
  - Additional cost just at coarse scale and cell faces
- Non-orthogonal axes increases rank of expansion
- HF exchange with/without screening
  - Similar, but with additional phase factor
  - Presently implementing approach of Spencer (PRB 77, 193110, 2008)
  - Periodic code presently not solving for localized orbitals

Periodic Kohn Sham

- Solve for Bloch states

$$\psi_{nk}(r)=e^{ik.r}u_{nk}(r) \quad \text{with} \quad u_{nk}(r)=u_{nk}(r+R)$$

- Kohn Sham equations

$$\left(-\frac{1}{2}(\nabla+ik)^2+V\right)u_{nk}=\epsilon_{nk}u_{nk}$$

$$u_{nk}=-\left(-\frac{1}{2}\nabla^2-\epsilon\right)^{-1}\left(V+\frac{k^2}{2}-ik.\nabla\right)u_{nk}$$

$$u_{nk}=-\left(-\frac{1}{2}(\nabla+ik)^2-\epsilon\right)^{-1}V\,u_{nk}$$

The latter form of the integral equation converges most robustly

Early Results

Figure 4.3: Resulting eigen-spectrum of model LiF crystal. Eigenvalues are in atomic units.

k-points	n	Elk	MADNESS	% diff.
(0,0,0)	0	-1.57371E+000	-1.57365E+000	0.0044148
	1	-7.56372E-001	-7.53451E-001	0.3877002
	2	-4.12984E-002	-4.13546E-002	0.1357660
	3	-4.12978E-002	-4.13546E-002	0.1371776
	4	-4.12964E-002	-4.13505E-002	0.1307800
(0,0,½)	0	-1.57240E+000	-1.57233E+000	0.0042320
(0,½,0)	1	-7.41498E-001	-7.38472E-001	0.4098059
(½,0,0)	2	-1.44584E-001	-1.44611E-001	0.0189691
	3	-2.78905E-002	-2.80731E-002	0.6507416
	4	-2.78887E-002	-2.80704E-002	0.6474858
(0,½,½)	0	-1.57127E+000	-1.57120E+000	0.0042692
(½,0,½)	1	-7.25052E-001	-7.21941E-001	0.4309148
(½,½,0)	2	-1.28549E-001	-1.28610E-001	0.0473275
	3	-1.28548E-001	-1.28607E-001	0.0458906
	4	-1.86473E-002	-1.88499E-002	1.0751777
(½,½,½)	0	-1.57024E+000	-1.57017E+000	0.0043158
	1	-7.11624E-001	-7.08333E-001	0.4646281
	2	-9.40811E-002	-9.40824E-002	0.0014286
	3	-9.40805E-002	-9.40824E-002	0.0020186
	4	-9.40790E-002	-9.40790E-002	0.0000099

Thornton, William Scott, "Electronic Excitations in YTiO3 using TDDFT and electronic structure using a multiresolution framework," PhD diss., University of Tennessee, 2011. http://trace.tennessee.edu/utk_graddiss/1134

Time evolution

Vence, Harrison, Krstic, Jia, Fann (UT/ORNL)

- Multiwavelet basis not optimal
  - Not strongly band limited; explicit methods unstable
    * DG introduces flux limiters, we use filters
- Semi-group approach
  - Split into linear and non-linear parts

$$\dot{u}(x,t) = \hat{L}u + N(u,t)$$

$$u(x,t) = e^{\hat{L}t}u(x,0) + \int_0^t e^{\hat{L}(t-\tau)}N(u,\tau)\,d\tau$$

- Trotter-Suzuki methods

$$e^{A+B} = e^{A/2}\,e^{B}\,e^{A/2} + O(\|[[A,B],A]\ldots\|)$$

  - Time-ordered exponentials
  - Chin-Chen gradient correction (JCP 114, 7338, 2001)

Exponential propagator

- Imaginary time Schrodinger equation
  - Propagator is just the heat kernel

$$\left(-\frac{1}{2}\nabla^2 + V(x)\right)\psi(x,t) = \dot{\psi}(x,t)$$

$$\psi(x,t) \simeq e^{\nabla^2 t/4} e^{-Vt} e^{\nabla^2 t/4} \psi(x,0)$$

$$e^{\nabla^2 t/2} f(x) = \frac{1}{\sqrt{2\pi t}} \int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{2t}} f(y)\, dy$$

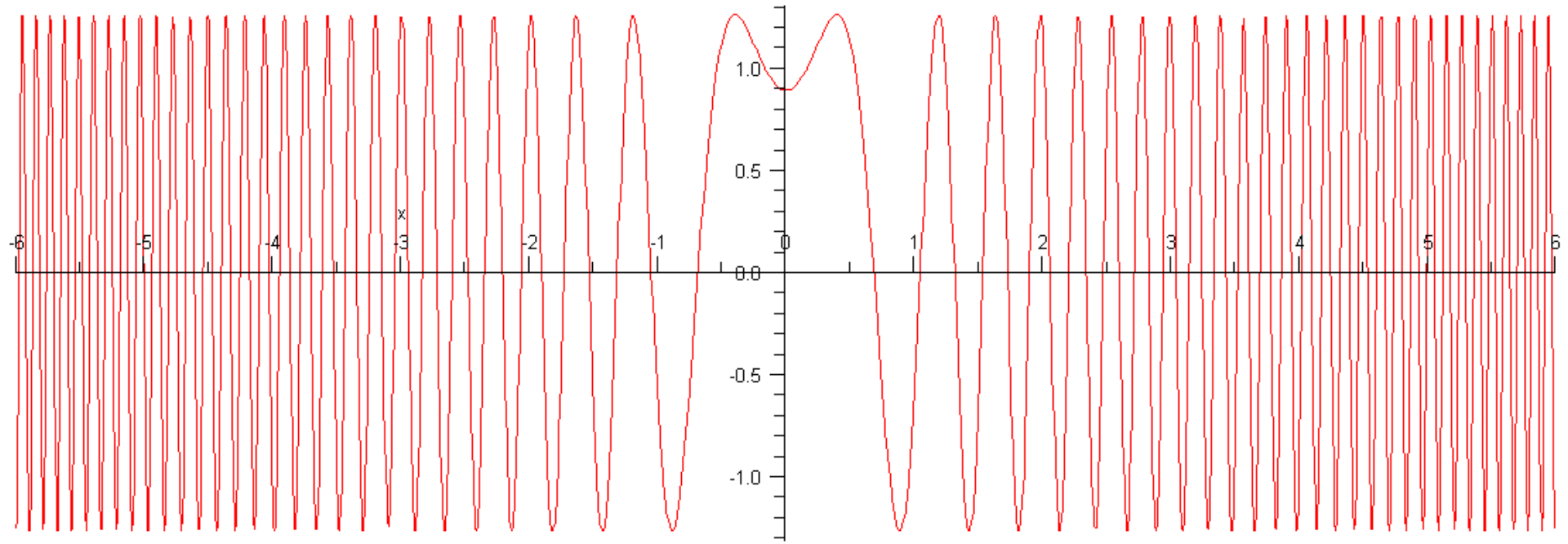
$$\lim_{t \to \infty} \psi(x,t) = \psi_0(x)$$

  - Wrap in solver to accelerate convergence

Exponential propagator

- Free-particle propagator in real time

$$\psi(x,t) = e^{i\nabla^2 t/2}\psi(x,0) = \frac{1}{\sqrt{2\pi i t}}\int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{2it}}\psi(y,0)\,dy$$

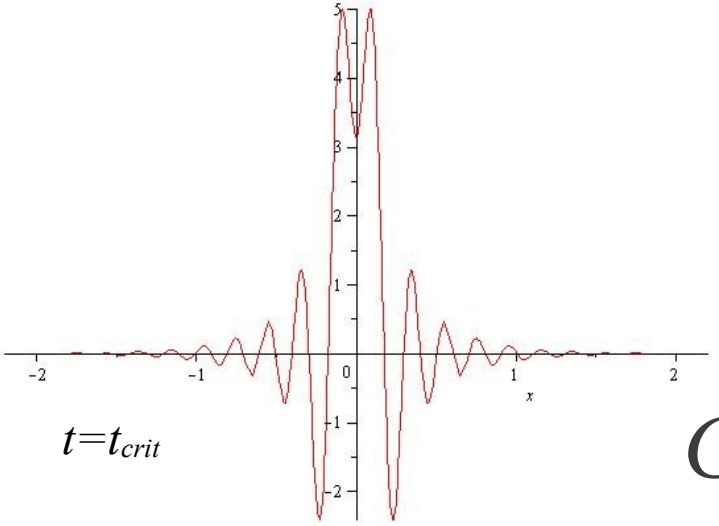


Exponential propagator

- Combine with projector onto band limit

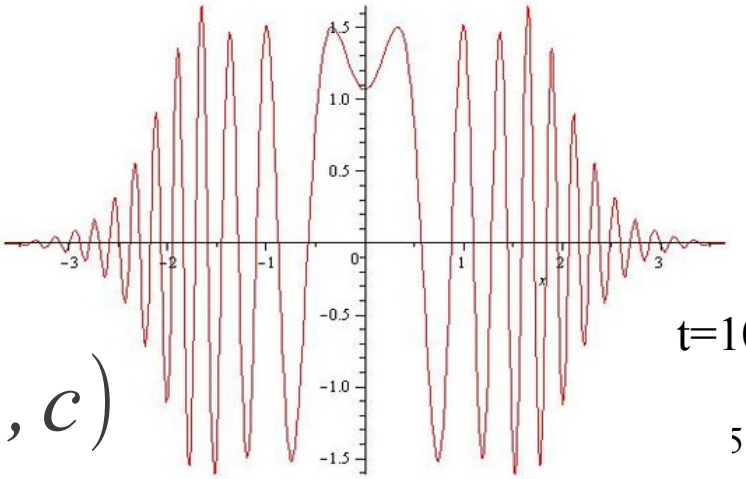
$$\hat{G}_0(k,t,c) = e^{-i\frac{k^2 t}{2}}\left(1+(k/c)^{30}\right)^{-1}$$

$$h = \frac{\pi}{c} \qquad t_{crit} = \frac{2h^2}{\pi}$$

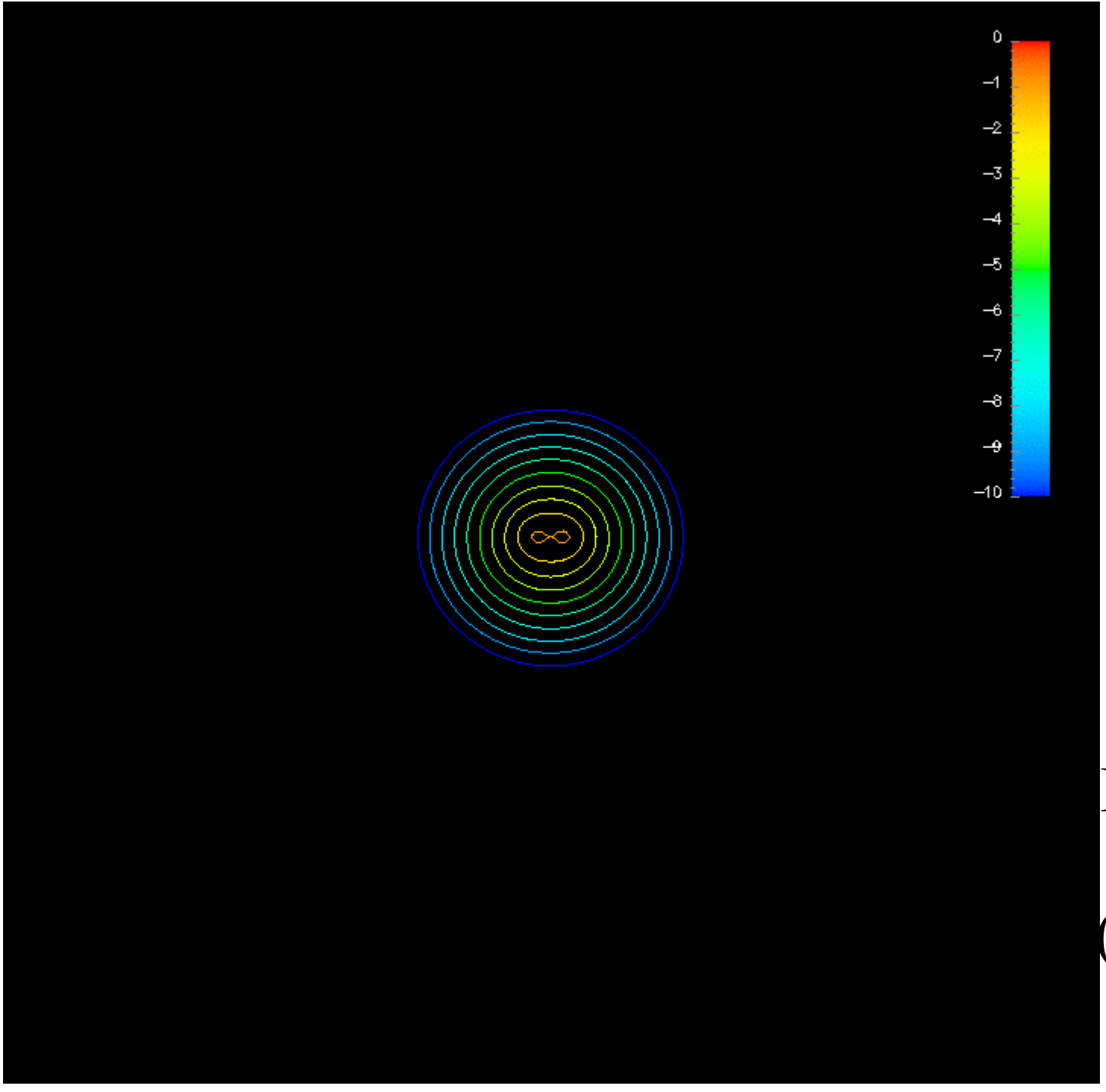


$t=t_{crit}$

$G_0(x,t,c)$



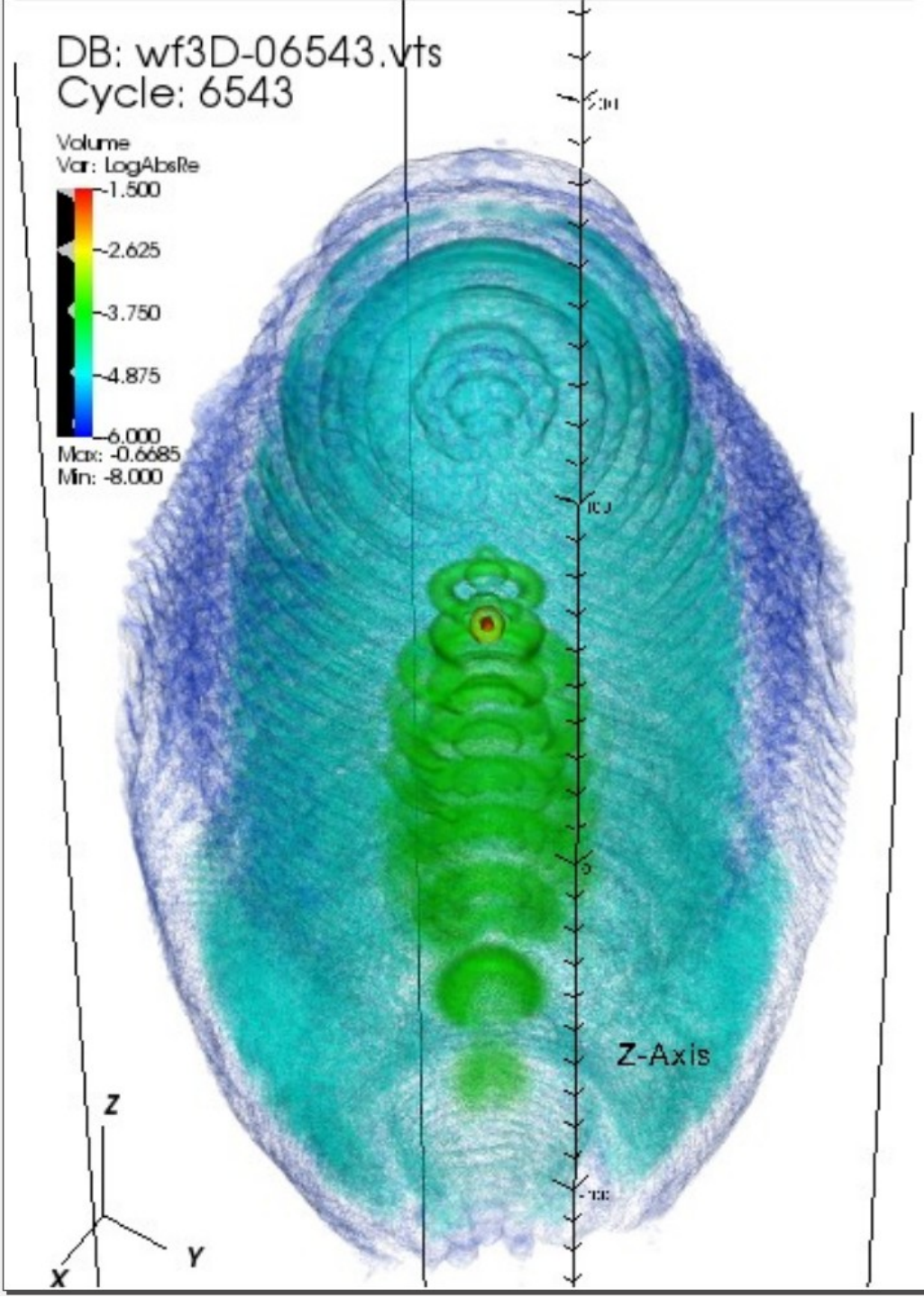
$t=10*t_{crit}$



Time dependent electronic structure

Vence, Krstic, Harrison UT/ORNL

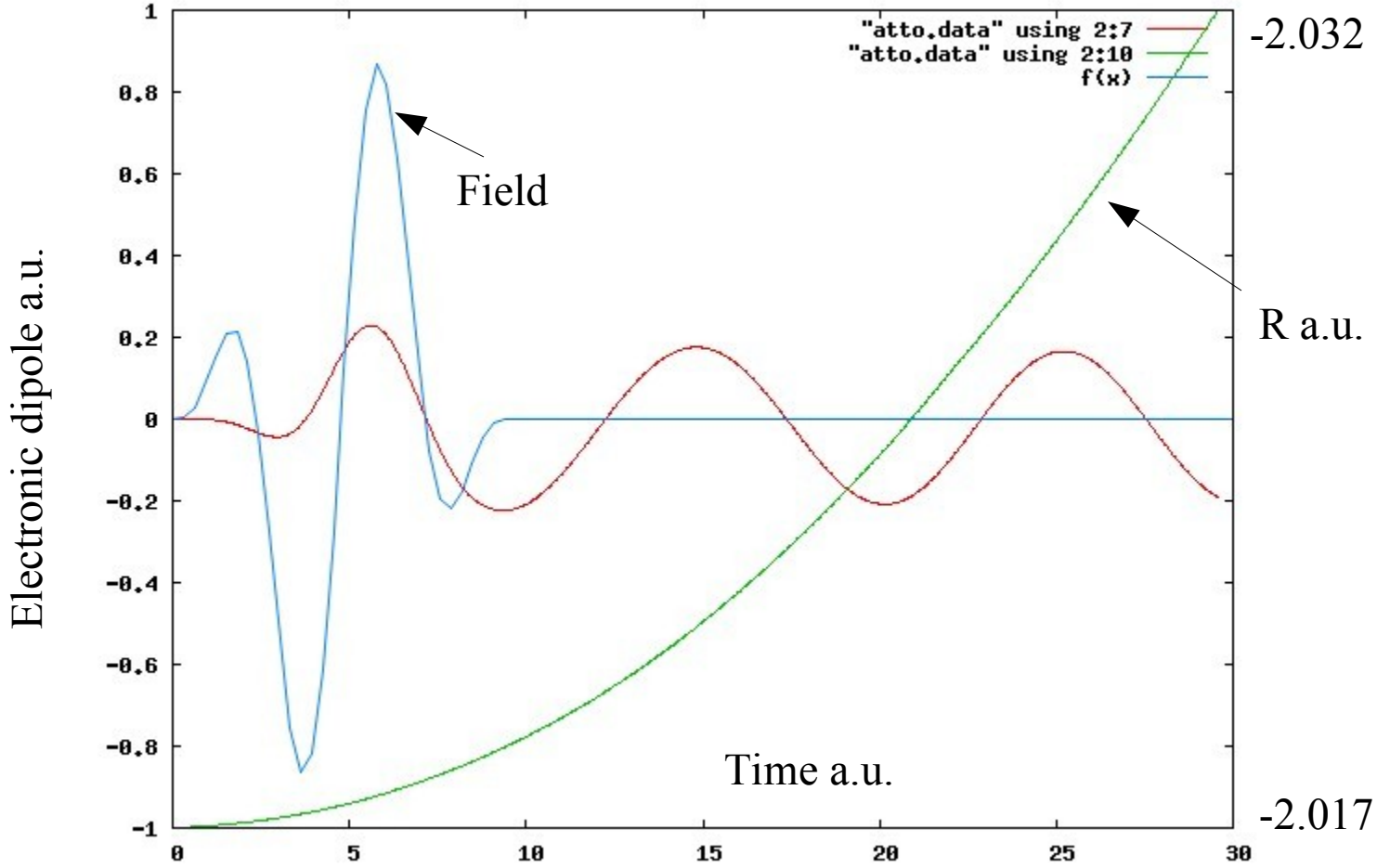
H_2^+ molecule in laser field (fixed nuclei)



H-atom
IR pulse

Dynamics of H_2^+ in laser

- 4D – 3 electronic + internuclear coordinate
  - First simulation with quantum nuclei and non-collinear field (field below is transverse)



Interior boundary conditions I

- Prototype equation

$$\nabla^2 u(x) = f(x) \quad x \in \Omega_D$$
$$u(x) = d(x) \quad x \in \partial\Omega_D$$

- Diffuse boundary approximation

$$\nabla^2 u(x) - \alpha(\epsilon) S_\epsilon(u(x) - d(x)) = f(x) \quad x \in \Omega \supset \Omega_D$$

ϵ width of diffuse layer

$$\lim_{\epsilon \to 0^+} u_\epsilon = u$$

$$S_\epsilon(x) = \frac{1}{\epsilon\sqrt{2\pi}} \exp\left(\frac{-s(x)^2}{2\epsilon^2}\right)$$

$s(x)$ signed minimum distance from surface

M. Reuter, J. Hill and R.J. Harrison, "Solving PDEs in Irregular Geometries with Multiresolution Methods I: Embedded Dirichlet Boundary Conditions," Computer Physics Communications, 183,1-7 (2012)

Interior boundary conditions II

- Recast as integral equation
  - Solved using GMRES

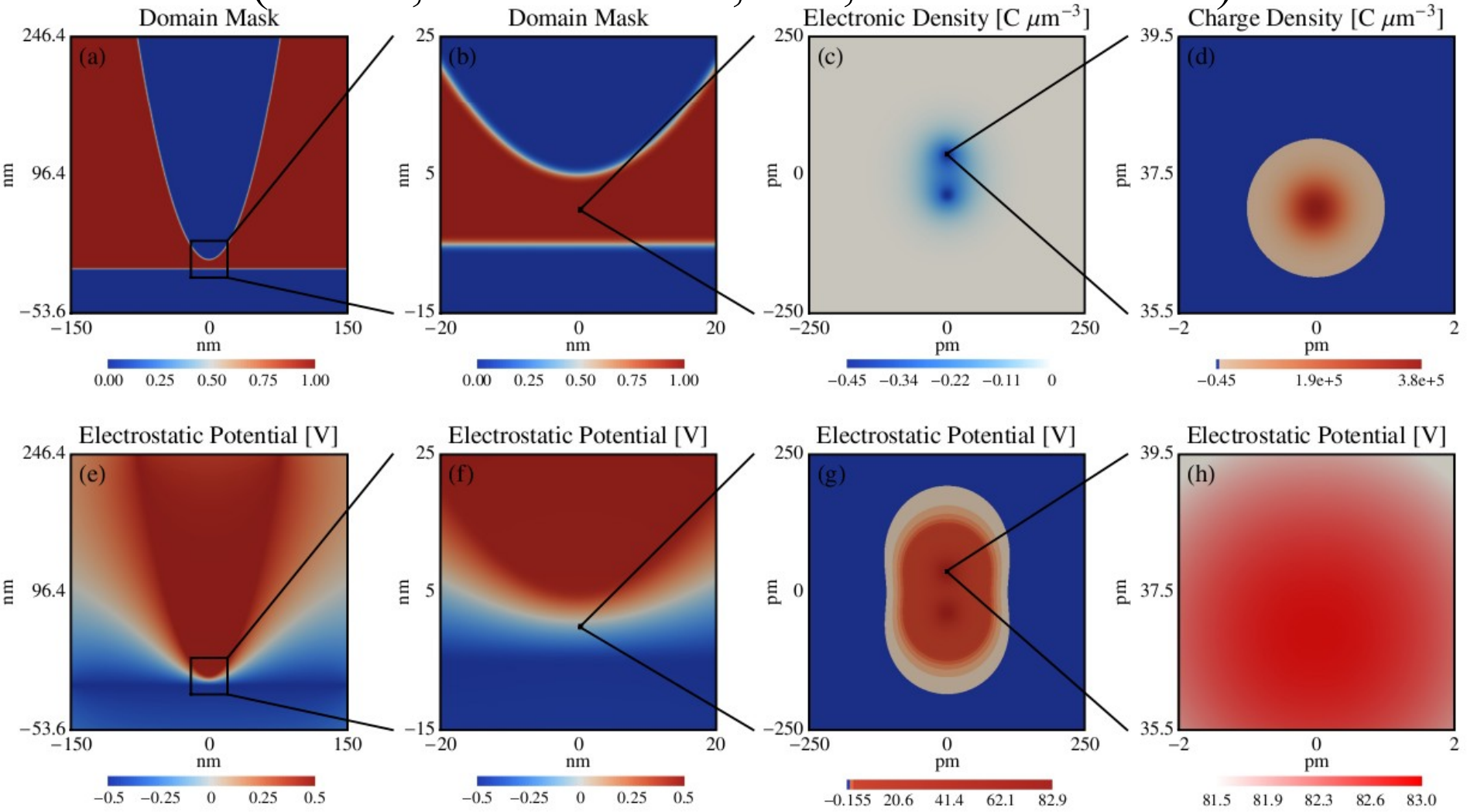
$$u_\epsilon(x) = \int_\Omega d^3x' \, G(x, x') \left[\alpha(\epsilon) S_\epsilon(x') \left(d(x') - u_\epsilon(x') \right) - C_\epsilon(x') f(x') \right]$$

$$C_\epsilon(x) = \frac{1}{2}\left[1 - \mathrm{erf}\left(\frac{s(x)}{\epsilon\sqrt{2}} \right) \right] \quad \text{characteristic function} \quad |\nabla C| = S$$

- Analysis of simple test cases and numerical tests suggest best choice is
  - Global first order convergence
  - (suggestions in older literature sub-optimal)

Nanoscale photonics

(Reuter, Northwestern; Hill, Harrison ORNL)



Diffuse domain approximation for interior boundary value problem; long-wavelength Maxwell equations; Poisson equation; Micron-scale Au tip 2 nm above Si surface with H2 molecule in gap – 10^7 difference between shortest and longest length scales.

Dielectric media (J. Fosso Tande)

- For implicit solvation want to solve Posson equation in presence of dielectric medium
  - Also want Poisson-Boltzmann

$$\nabla . (\epsilon \nabla u) = -4\pi\rho$$

$$\nabla^2 u = -\frac{4\pi}{\epsilon}\rho - \frac{\nabla \epsilon}{\epsilon} . \nabla u$$

Effective surface charge

- Dielectric defined via characteristic function

$$\epsilon(r) = \epsilon_0 C(r) + \epsilon_1 (1 - C(r)) \quad \text{or} \quad \epsilon(r) = \epsilon_1 \exp\left(C(r) \log \frac{\epsilon_0}{\epsilon_1}\right)$$

(Gygi and later Skylaris define via level set)

$$C(r) = \begin{cases} 1 & \text{inside} \\ 0 & \text{outside} \end{cases}$$

Use smooth approximation switching over finite distance

$$V = \int C(r)\, dV \qquad A = \int |\nabla C|\, dV$$

Dielectric media in external electric field (Neuman boundary conditions)

- Provided GF have only free space or periodic Dirichlet boundary conditions
  - Replace Neuman conditions by separately treating asymptotic form

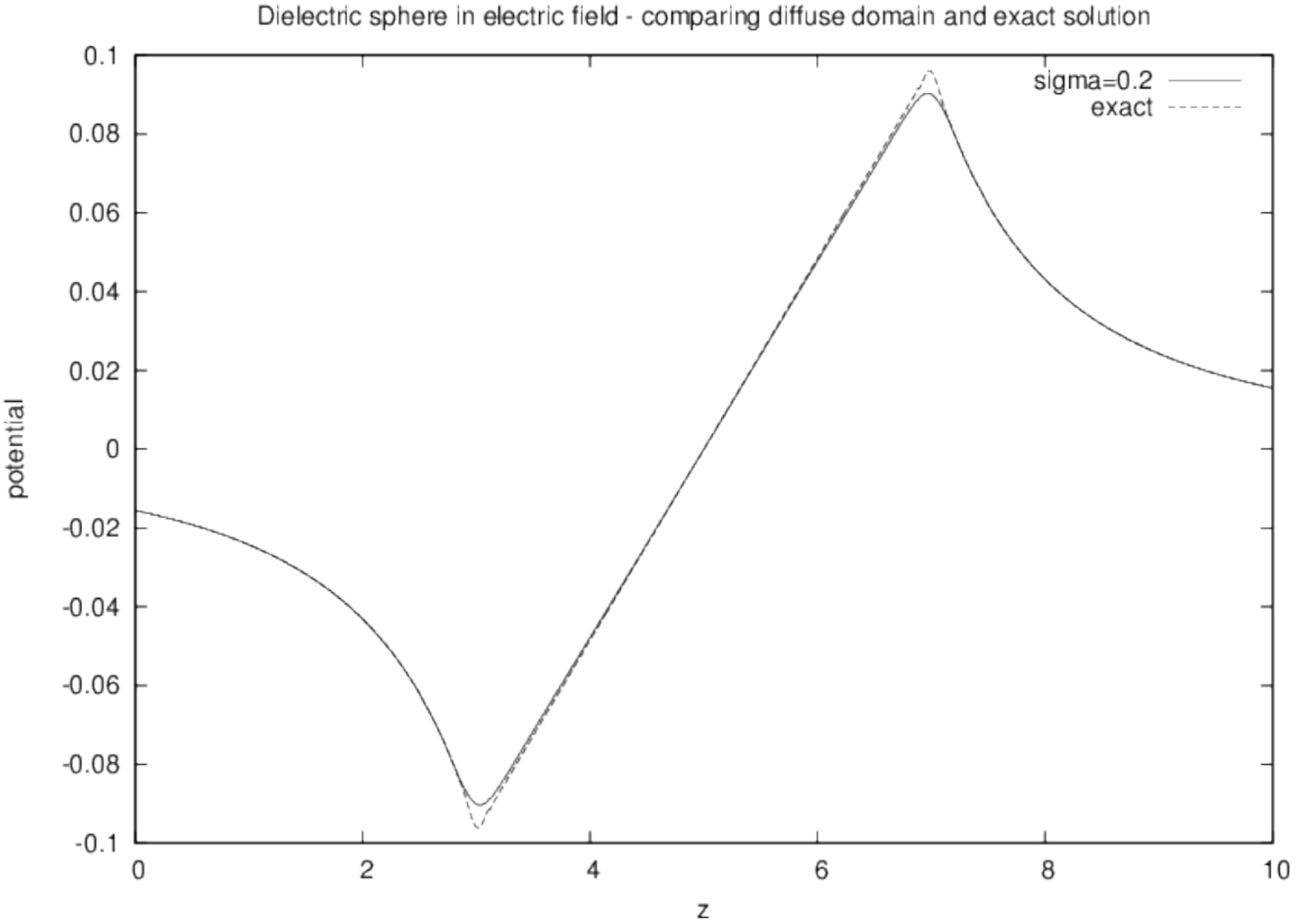
E = external field

$$v(x) = u(x) - E \,.\, r \qquad u(r \to \infty) = 0$$

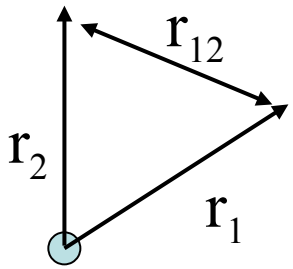
$$\nabla \,.\, (\epsilon \nabla v) = -4\pi v$$

$$\Rightarrow \quad \nabla^2 u = -\frac{4\pi}{\epsilon}\rho + \frac{\nabla \epsilon}{\epsilon} \,.\, (E - \nabla u)$$

Total electric field


Dielectric sphere in electric field - comparing diffuse domain and exact solution
sigma=0.2
exact
potential
z
0.1
0.08
0.06
0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
0
2
4
6
8
10

Electron correlation (6D)



- All defects in mean-field model are ascribed to electron correlation
- Singularities in Hamiltonian imply for a two-electron atom

$$\Psi(r_1, r_2, r_{12}) = 1 + \frac{1}{2} r_{12} + \cdots \quad \text{as} \quad r_{12} \to 0$$

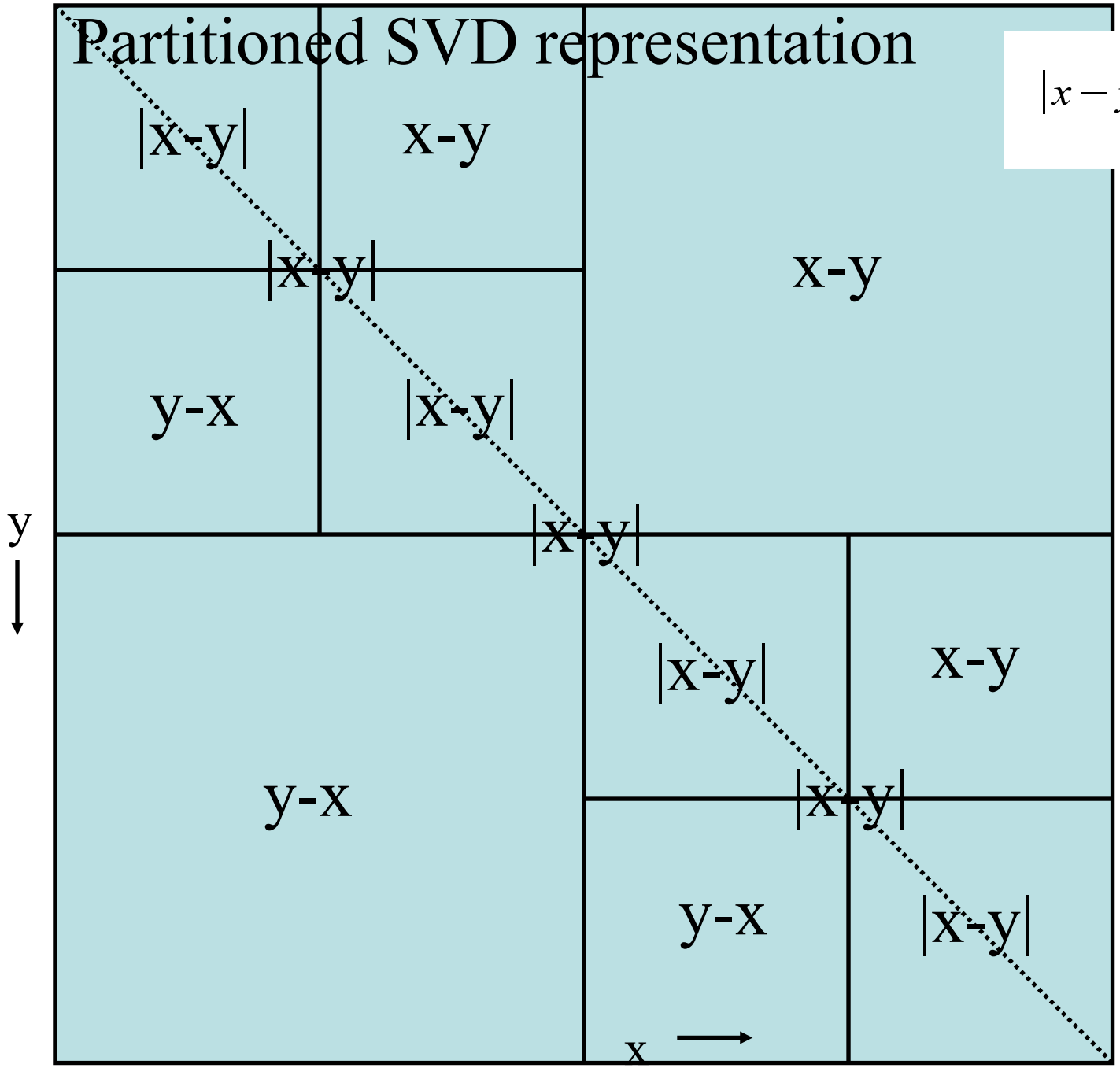
- Include the inter-electron distance in the wavefunction
  - E.g., Hylleraas 1938 wavefunction for He

$$\Psi(r_1, r_2, r_{12}) = \exp(-\xi(r_1 + r_2))(1 + a\, r_{12} + \cdots)$$

  - Potentially very accurate, but not systematically improvable, and (until recently) not computationally feasible for many-electron systems
- Configuration interaction expansion – slowly convergent

$$\Psi(r_1, r_2, \ldots) = \sum_i c_i \left| \Phi_1^{(i)}(r_1) \Phi_2^{(i)}(r_2) \ldots \right|$$

Partitioned SVD representation



$$|x-y| = \sum_{\mu=1}^{r} f_\mu(x)\, g_\mu(y)$$

r = separation rank

In 3D, ideally must be one box removed from the diagonal

Diagonal box has full rank

Boxes touching diagonal (face, edge, or corner) have increasingly low rank

Away from diagonal $r = O(-\log \varepsilon)$

Preliminary results for He atom (Yanai, 2005)

	Variational E	ΔE	residual
HF	-2.861 61		
Iter. 0	-2.871 08		0.414 73
1	-2.894 92	-0.023 84	0.017 28
2	-2.900 43	-0.005 51	0.007 94
3	-2.902 18	-0.001 75	0.003 84
4	-2.902 88	-0.000 70	0.002 02
5	-2.903 20	-0.000 32	0.001 25
6	-2.903 39	-0.000 20	0.000 91
…	…	…	…
12	-2.903 73	-0.000 04	0.000 36
13	-2.903 73	+0.000 004	0.000 32
14	-2.903 77	-0.000 04	0.000 28

exact	-2.903 724	(E(HF)=-2.861 68)
Hylleraas (6 terms)	-2.903 24	
Löwdin and Redei	-2.895 4	
cc-pV6Z	-2.903 48 (FCI)	(E(HF)= -2.861 67)

Computational details:

- 5-th order multiwavelets
- Wavelet threshold: 2×10^{-5}
- SVD threshold: 2×10^{-6}
- Exponential correlation factor

Perturbative wavefunction:

- Maximum refinement: n=4

- Memory: 132M in full partitioned SVD form

~10GB without SVD

Energy is variational (small non-variational is just truncation err)

Local separated approximations

- Density matrix
  - Local rank corresponds to number of products of localized occupied orbitals with significant variation in that volume
  - Conventional concept: local natural orbitals
- Pair correlated wave function
  - Away from diagonal, local rank corresponds to number of products of localized low-lying virtual orbitals with significant variation
  - Conventional concept: pair natural orbitals

Computing in separated representations

$$f_{j_1,\ldots,j_d} = \sum_{l=1}^{M} \sigma_l \prod_{i=1}^{d} \left[f_i^{(l)} \right]_{j_i} + O(\epsilon)$$

- Operating on or combing tensors inflates the rank – must eventually reduce closer to optimal
- Pair wave function separated by particles
  - Just 2 directions
  - Rank-revealing Gram-Schmidt, or reconstruct+SVD
- For many dimensions no ideal solution
  - Alternating least squares, other (Beylkin)

Douglas Kroll Hess 2nd order

- Some definitions

$$E_0(p) = \sqrt{p^2 c^2 + m^2 c^4}$$

$$A(p) = \sqrt{\frac{E_0(p) + mc^2}{2\,E_0(p)}}$$

$$\bar{P}(p) = \frac{c}{E_0(p) + mc^2}$$

$$\boldsymbol{P}(\boldsymbol{p}) = \bar{P}(p)\,\boldsymbol{p}$$

$$E_1 = A\left(V + \boldsymbol{P}.V\,\boldsymbol{P}\right)A$$

Douglas Kroll Hess 2nd order

The DKH2 one-electron Hamiltonian is

$$H_{DKH2} = E_0 - mc^2 + E_1 + \frac{1}{2}\left(\boldsymbol{W}_1 . \boldsymbol{O}_1 + \boldsymbol{O}_1 . \boldsymbol{W}_1\right)$$

where $\boldsymbol{W_1}$ is the solution to

$$\boldsymbol{W}_1 E_0 + E_0 \boldsymbol{W}_1 = \boldsymbol{O}_1$$

$$\boldsymbol{O}_1 = A\left[\boldsymbol{P}, V\right]A$$

Douglas Kroll Hess 2nd order

The momentum space kernels are

$$\boldsymbol{O}_1(\boldsymbol{p},\boldsymbol{p}') = A(p)\left(\boldsymbol{P}(\boldsymbol{p}) - \boldsymbol{P}(\boldsymbol{p}')\right)A(p')V_{ext}(\boldsymbol{p}-\boldsymbol{p}')$$

and

$$\boldsymbol{W}_1(\boldsymbol{p},\boldsymbol{p}') = \frac{A(p)\left(\boldsymbol{P}(\boldsymbol{p}) - \boldsymbol{P}(\boldsymbol{p}')\right)A(p')V_{ext}(\boldsymbol{p}-\boldsymbol{p}')}{E_0(p)+E_0(p')}$$

noting that in these expressions all entities are not operators (i.e., are just numbers or vectors of numbers).

Need for separable kernels

The kernel of W_1 is not easily transformed back into real space since p and p' are coupled through both the denominator and potential.

However, a separated representation of the denominator that can be analytically back transformed will solve the problem.

Hence, also need separable real space kernels for *A(p)*, *exp(-α E0(p))*, and *Pbar(p)*

Separated form of reciprocal

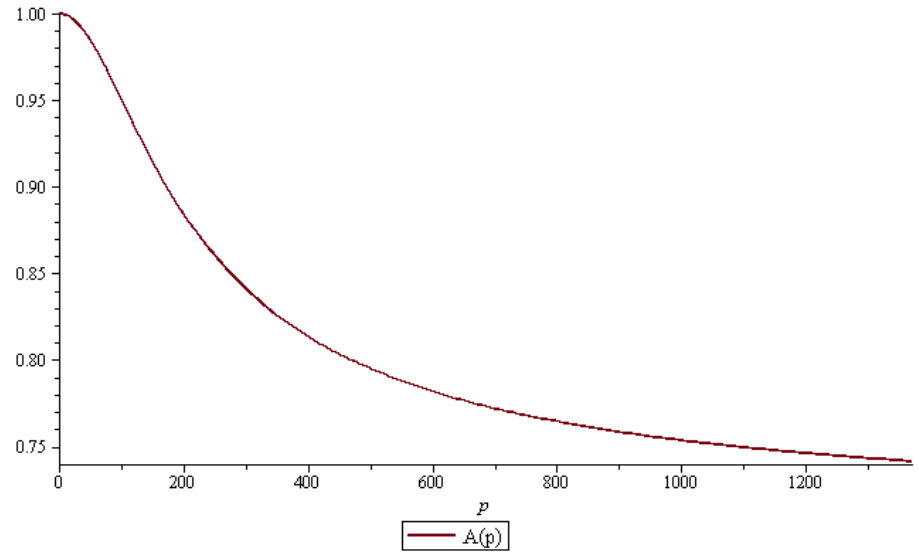
- Quadrature: $\frac{1}{x} = \int_0^\infty \exp(-t\,x)\,dt = \sum_{\mu=1}^{M} \omega_i \exp(-x\,t_\mu)$
  Hard to control error
  over desired range

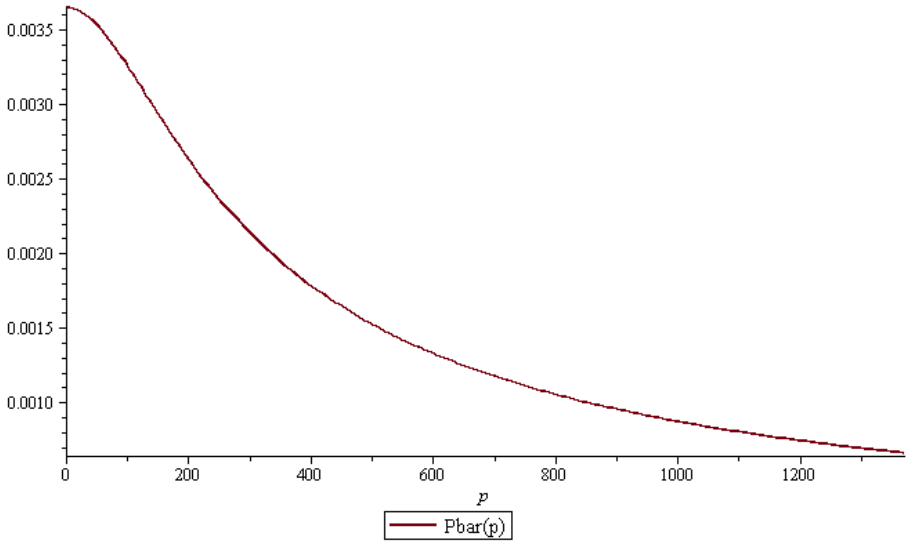
- Approximation
  $$\frac{1}{x} = \sum_{\mu=1}^{M} \omega_i \exp(-x\,t_\mu) + O(\epsilon) \qquad x \in [a, b]$$
  Better control and more general kernels
- Beylkin, Hackbush (/www.mis.mpg.de/scicomp/EXP_SUM)

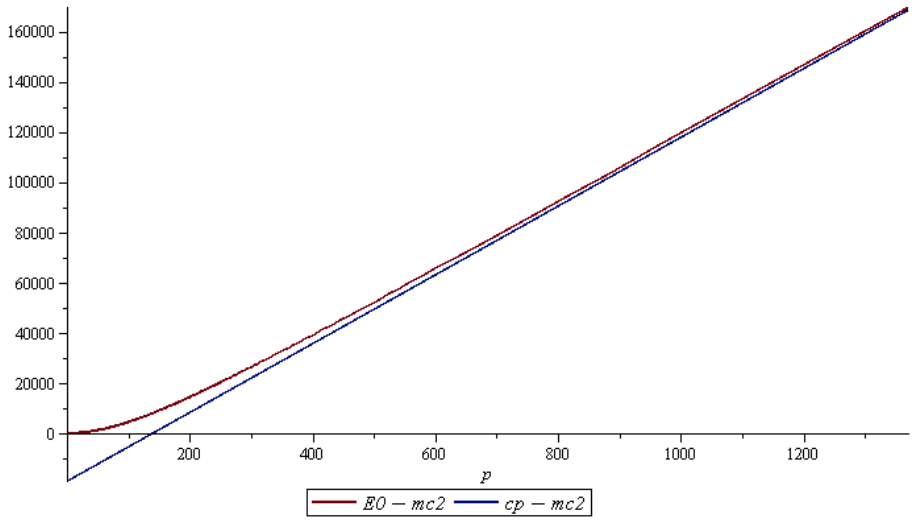
Asymptotic properties

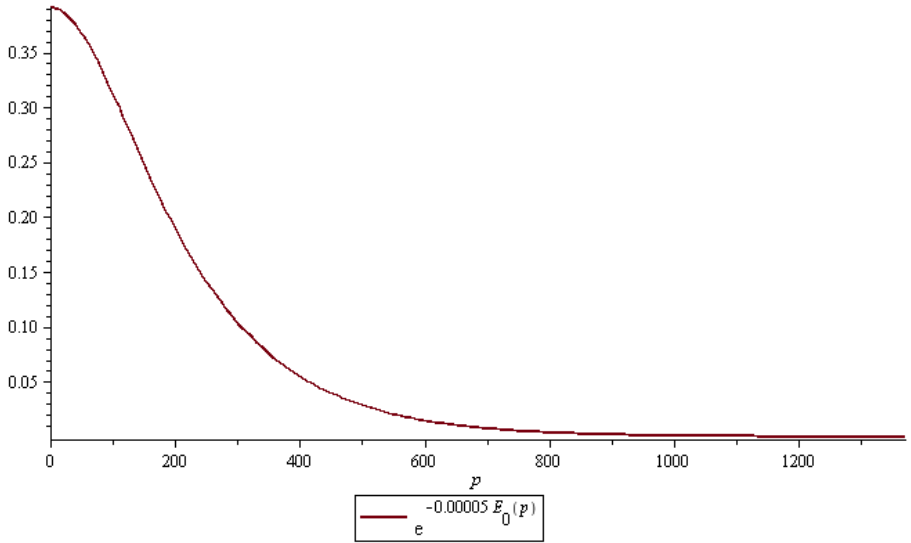
Operator	Small p	Large p
$E_0(p)$	$mc^2+\frac{p^2}{2m}+O(p^4)$	$pc+\frac{m^2c^3}{2p}+O(p^{-3})$
$A(p)$	$1-\frac{p^2}{8m^2c^2}+O(p^4)$	$\frac{1}{\sqrt{2}}+\frac{mc}{p2\sqrt{2}}+O(p^{-2})$
$\bar{P}(p)$	$\frac{1}{2mc}-\frac{p^2}{8m^3c^3}+O(p^4)$	$\frac{1}{p}-\frac{mc}{p^2}+O(p^{-3})$

Functional forms



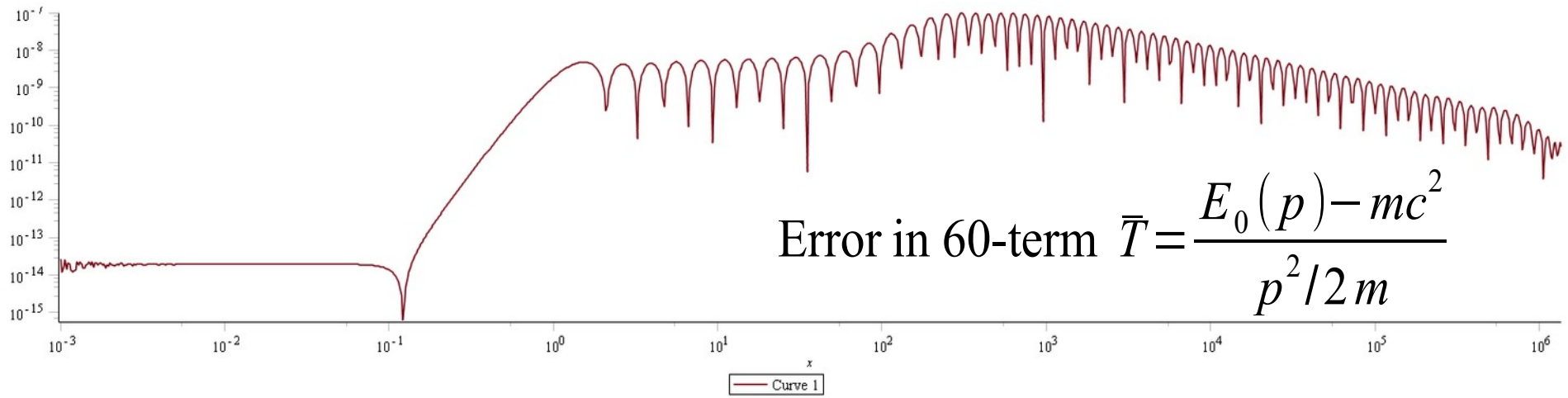






Approach 1 – Brute force for initial test

- Weighted least squares fit to Gaussians in Fourier space
  - Weight function p^2
  - Weight accurate reproduction of value and first four derivatives at p=0, and to bias towards +ve coeffs



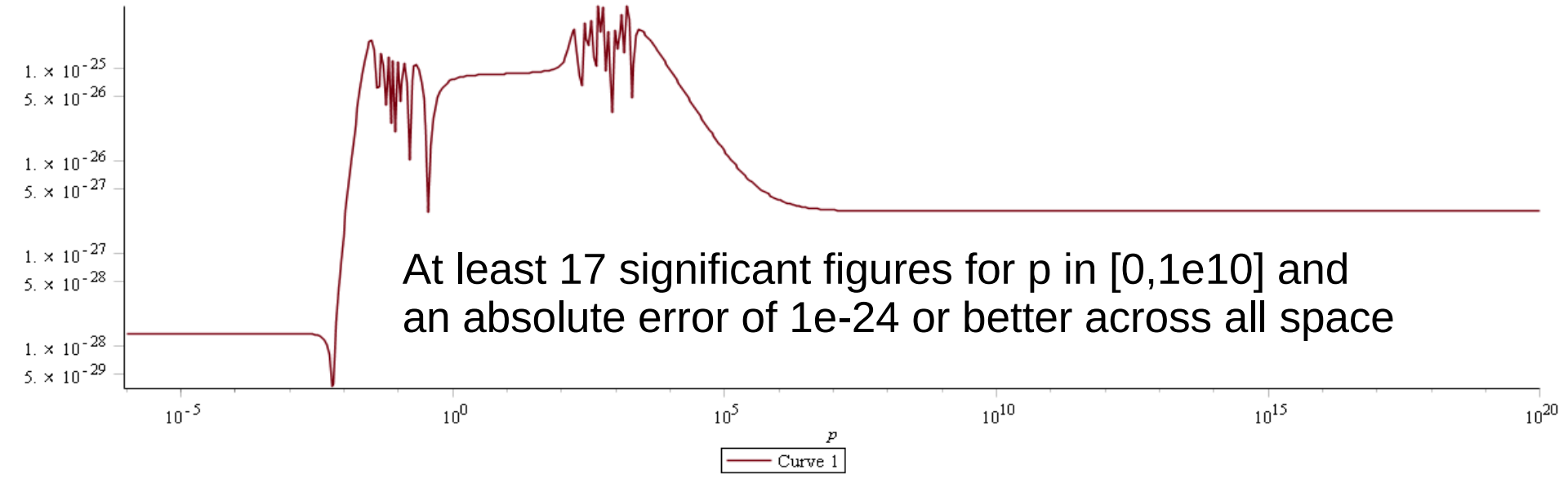
Error in 60-term $\bar{T} = \dfrac{E_0(p) - mc^2}{p^2/2m}$

Approach 2 – Construct highly accurate Gaussian approximations

- Subtract off asymptotic decay to very high precision

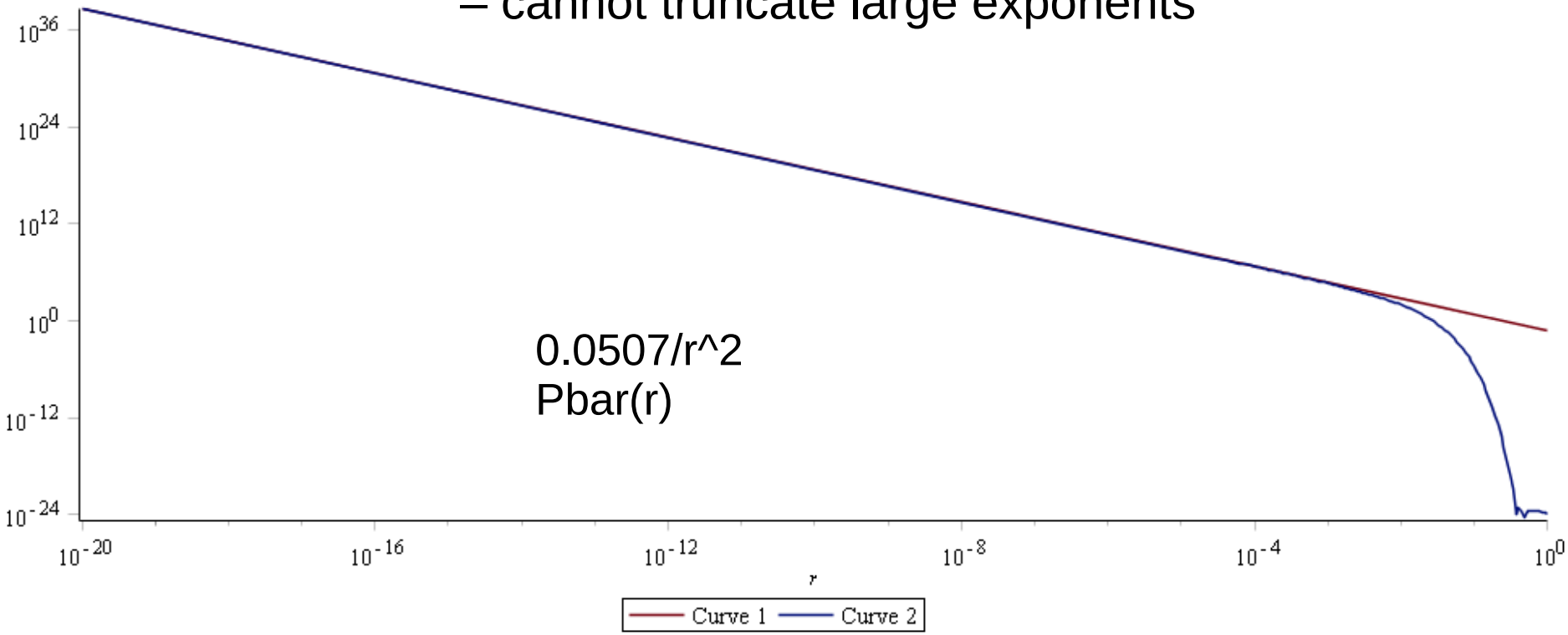
$$r^{-n} = \frac{2^{-n}}{\Gamma(n/2)} \int_0^\infty e^{-\frac{r^2}{4t}} t^{-1-n/2}\, dt$$

- Fit smooth localized remainder to very high precision using previous brute force approach



Pbar in real space

Highly singular kernels require careful application – cannot truncate large exponents



Approach 3: Analytic transform

- Analytic real space representations that produce accurate Gaussian approximations (most are not closed form)

$$\overline{P}\left(\|\mathbf{x}\|\right) = \frac{1}{2^{3/2}} \int_{-\infty}^{\infty} e^{-\frac{1}{4}e^{t}\|\mathbf{x}\|^{2}} \left(\frac{1}{\sqrt{\pi}} e^{-a^{2}e^{-t}+t} - ae^{\frac{1}{2}t} \operatorname{erfc}\left(ae^{-t/2}\right) \right) dt, \quad a = mc,$$

$$A\left(\|\mathbf{x}\|\right) = \frac{(2\pi)^{3/3}}{\sqrt{2}} \delta\left(\mathbf{x}\right) + \frac{a^{3}}{4} \int_{-\infty}^{\infty} e^{-\frac{1}{4}a^{2}\|\mathbf{x}\|^{2}e^{t}} \widetilde{w}\left(t\right) dt,$$

$$\widetilde{w}(t) = e^{-e^{-t}+t} \sum_{l=1}^{L} \frac{\beta_{l}\gamma_{l}}{1-\gamma_{l}e^{-\frac{1}{2}t}}.$$

Pbar agrees to at least double precision arithmetic with both approaches

Example

- Starting with (a=mc):

$$\bar{P}(p) = c(E_0(p) + mc^2)^{-1} = \frac{1}{\sqrt{||p||^2 + a^2} + a}$$

Using:

$$\frac{1}{r} = \int_0^\infty e^{-rs}\, ds$$

Example (cont)

- $$\frac{1}{\sqrt{||p||^2 + a^2} + a} = \int_0^\infty e^{-s\sqrt{||p||^2 + a^2}} e^{-as}\, ds$$

Then using:

$$e^{-xy} = \frac{x}{2\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-\frac{x^2 e^t}{4} - y^2 e^{-t} + \frac{t}{2}} dt$$

Where $x = s$ and $y = \sqrt{||p||^2 + a^2}$

Example (cont.)

- $$\frac{1}{\sqrt{\left|\left|p\right|\right|^2 + a^2} + a} = \int_{-\infty}^{\infty} \left(\int_0^{\infty} \frac{s}{2\sqrt{\pi}} e^{-\frac{s^2 e^t}{4}} e^{-as}\, ds \right) e^{-\left(\left|\left|p\right|\right|^2 + a^2\right)e^{-t} + \frac{t}{2}} dt$$

Evaluating the inner integral gives:

$$\bar{P}(\left|\left|p\right|\right|)$$

$$= \int_{-\infty}^{\infty} e^{-\left|\left|p\right|\right|^2 e^{-t}} \left(\frac{1}{\sqrt{\pi}} e^{-\frac{t}{2} - a^2 e^{-t}} - a e^{-t} \operatorname{erfc}(a e^{-\frac{t}{2}}) \right) dt$$

Operator	Momentum Space Representation	Real Space Represen

Operator	Momentum Space Representation	Real Space Representation [2]
$\bar{P}$	$c(E_0(p)+mc^2)^{-1}$	$\frac{1}{2^{3/2}}\int_{\mathbb{R}} e^{-\frac{1}{4}e^{t}\lVert x\rVert^2}\left(\frac{1}{\sqrt{\pi}}e^{-m^2c^2e^{-t}+2t}-mce^{\frac{3}{2}t}\mathrm{erfc}(mce^{-\frac{t}{2}})\right)dt$
A	$\sqrt{\frac{E_0(p)+mc^{\dagger}2}{2E_0(p)}}$	$\frac{a^3}{4}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l\gamma_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$
$\bar{P}A$	$c(E_0(p)+mc^2)^{-1}\sqrt{\frac{E_0(p)+mc^{\dagger}2}{2E_0(p)}}$	$\frac{a^2}{4}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$
$\bar{T}$	$\frac{E_0-mc^2}{p^2/2m}$	$\frac{a^3}{2^{\frac{3}{2}}}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$

Operator	Momentum Space Representation	Real Space Representation [2]
$\bar{P}$	$c(E_0(p)+mc^2)^{-1}$	$\frac{1}{2^{3/2}}\int_{\mathbb{R}} e^{-\frac{1}{4}e^{t}\lVert x\rVert^2}\left(\frac{1}{\sqrt{\pi}}e^{-m^2c^2e^{-t}+2t}-mce^{\frac{3}{2}t}\mathrm{erfc}(mce^{-\frac{t}{2}})\right)dt$
A	$\sqrt{\frac{E_0(p)+mc^{\dagger}2}{2E_0(p)}}$	$\frac{a^3}{4}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l\gamma_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$
$\bar{P}A$	$c(E_0(p)+mc^2)^{-1}\sqrt{\frac{E_0(p)+mc^{\dagger}2}{2E_0(p)}}$	$\frac{a^2}{4}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$
$\bar{T}$	$\frac{E_0-mc^2}{p^2/2m}$	$\frac{a^3}{2^{\frac{3}{2}}}\int_{\mathbb{R}} e^{-\frac{1}{4}a^2e^{t}\lVert x\rVert^2}\left[e^{-e^{-t}+t}\sum_{l=1}^{L}\frac{\beta_l}{1-\gamma_l e^{-\frac{1}{2}t}}\right]dt$

DK1 also don

Initial test with RK approximation

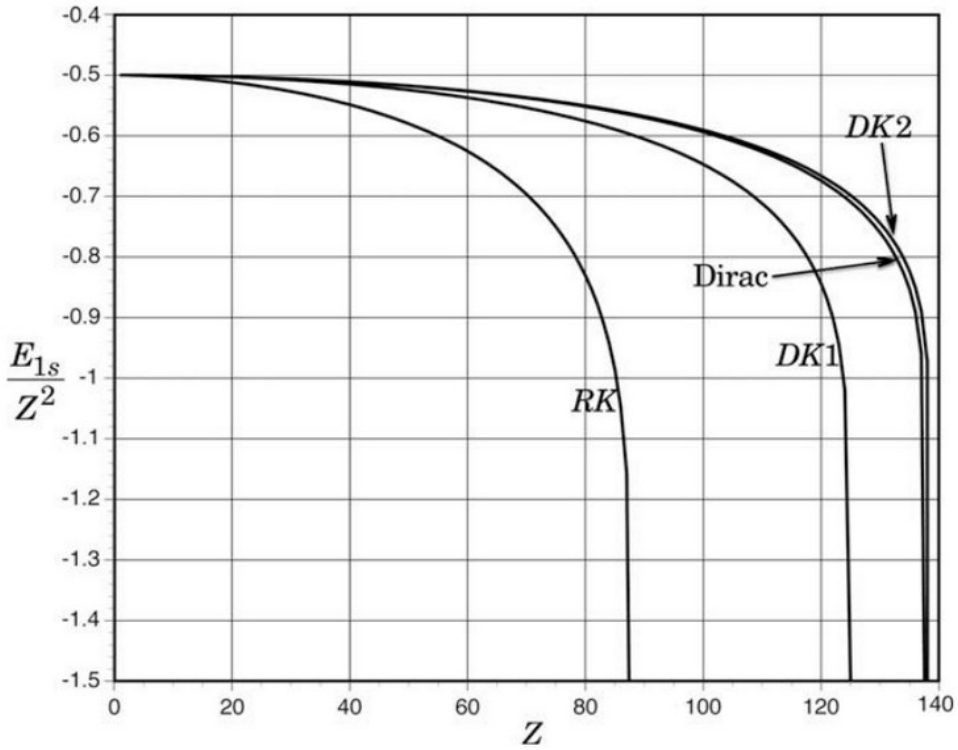
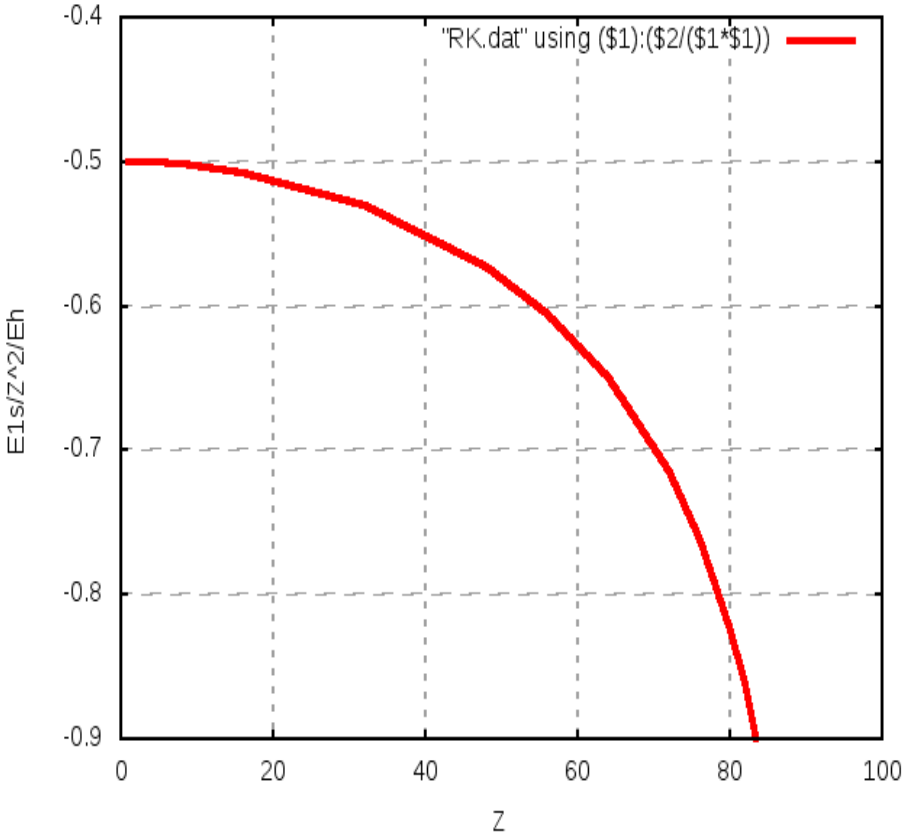


Fig. 1. Scaled 1s orbital energy for hydrogenlike ions for the Dirac and son quasirelativistic operators (RK: relativistic kinematics, *DK*1/2: first- and secon order Douglas–Kroll), obtained with a large basis set of 200 Gaussians. Data from higher-order Douglas–Kroll operators (not shown) is even closer to the Dirac line.



60 term approximation to Tbar

Wullen, Chem. Phys. 356 (2009) 199

Multiresolution Solution of the Dirac Equation

Joel Anderson, Robert Harrison,
Bryan Sundahl, Scott Thornton

The Schrödinger Equation

- $$\left(-\frac{1}{2}\Delta + V\right)\Psi = E\Psi$$

We solve this by rearranging it into a Helmholtz equation:

$$(-\Delta - 2E)\Psi = -2V\Psi$$

And convolution with the Green's function g(r) gives the fixed-point iteration:

$$\Psi(r) = g * (-2V\Psi) = \int \frac{e^{-|r-r_0|\sqrt{-2E}}}{4\pi|r-r_0|}(-2V(\mathrm{r}_0)\Psi(r_0))d^3r_0$$

The Dirac Equation

- $$H_D\psi = (c\boldsymbol{\alpha}\cdot\boldsymbol{p} + \beta c^2 + V)\psi = \mathrm{E}\psi$$

(Time independent, single particle, atomic units)

Here, ψ is a 4-vector describing the state of the system (called a 4-spinor), **p** is the 3D momentum operator, $\boldsymbol{\alpha}$ is a 3-vector containing the Pauli spin matrices (or some other compatible matrices), c = 137.0359895 is the speed of light, and V is the potential[1].

$$\alpha_k = \begin{bmatrix} 0 & \sigma_k \\ \sigma_k & 0 \end{bmatrix} \qquad \sigma_1 = I_2 \qquad \sigma_2 = \begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix} \qquad \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\beta = \begin{bmatrix} I_2 & 0 \\ 0 & -I_2 \end{bmatrix}$$

Dirac Green's Function

- It can be shown that[4]:

$$G(\boldsymbol{r}) = (H_D + E)g(\boldsymbol{r})$$

Where $g(\boldsymbol{r})$ is the Green's function used earlier in the nonrelativistic case.

So that we can write the fixed-point iteration:

$$\Psi = \int (H_D + E)g(r - r_0)(-V\Psi)\, d^3r_0$$

[4] Blackledge,J. On the Dirac Scattering Problem. Mathematica Aeterna, vol: 3, (7)

Initial Dirac results – Hyrdogenic atoms

small=1e-10 k=8 thresh=1e-6				
Z	E	Error	Exact Answer	Relative Error
1	-5.0000665993E-01	-3.3300000535E-09	-5.0000665660E-01	6.6599114422E-09
2	-2.0001065268E+00	-1.2699999719E-08	-2.0001065141E+00	6.3496616950E-09
4	-8.0017048164E+00	-4.6500000295E-08	-8.0017047699E+00	5.8112616788E-09
6	-1.8008635096E+01	-9.6999997368E-08	-1.8008634999E+01	5.3863048128E-09
8	-3.2027311391E+01	-1.3399999688E-07	-3.2027311257E+01	4.1839290161E-09
10	-5.0066742160E+01	-1.3400000398E-07	-5.0066742026E+01	2.6764274758E-09
20	-2.0107652137E+02	1.9899999870E-06	-2.0107652336E+02	9.8967296319E-09
30	-4.5552489048E+02	1.6630000061E-05	-4.5552490711E+02	3.6507334290E-08
40	-8.1780742505E+02	7.2779999982E-05	-8.1780749783E+02	8.8994048325E-08
60	-1.8956823665E+03	-1.0599999996E-05	-1.8956823559E+03	5.5916540886E-09
80	-3.5321921118E+03	3.8900000163E-05	-3.5321921507E+03	1.1012990943E-08
100	-5.9391948225E+03	5.6189999941E-04	-5.9391953844E+03	9.4608774934E-08
120	-9.7107628308E+03	2.0689400000E-02	-9.7107835202E+03	2.1305592857E-06
small=1e-10 k=10 thresh=1e-8				
30	-4.5552490745E+02	-3.3999998550E-07	-4.5552490711E+02	7.4639164663E-10
60	-1.8956823567E+03	-8.0000017988E-07	-1.8956823559E+03	4.2201172437E-10

High-level composition

- Close to the physics

$$E = \langle \psi | -\frac{1}{2}\nabla^2 + V | \psi \rangle + \int \psi^2(x) \frac{1}{|x-y|} \psi^2(y)\, dx\, dy$$

```
operatorT op = CoulombOperator(k, rlo, thresh);
functionT rho = psi*psi;
double twoe = inner(apply(op,rho),rho);
double pe = 2.0*inner(Vnuc*psi,psi);
double ke = 0.0;
for (int axis=0; axis<3; axis++) {
    functionT dpsi = diff(psi,axis);
    ke += inner(dpsi,dpsi);
}
double energy = ke + pe + twoe;
```

Let

$$\Omega = [-20, 20]^3$$

$$r = x \to \sqrt{x_0^2 + x_1^2 + x_2^2}$$

$$g = x \to \exp(-r(x))$$

$$v = x \to -r(x)^{-1}$$

In

$$\psi = \mathcal{F}\, g$$

$$\nu = \mathcal{F}\, v$$

$$S = \langle \psi | \psi \rangle$$

$$V = \langle \psi | \nu * \psi \rangle$$

$$T = \frac{1}{2} * \sum_{i=0}^{2} (\langle \nabla_i \psi | \nabla_i \psi \rangle)$$

print $S, V, T, \dfrac{T+V}{S}$

End

H atom Energy

H atom actual source

```
Let
  Omega = [-20, 20]^3
  r = x -> sqrt(x_0^2 + x_1^2 + x_2^2)
  g = x -> exp(-r(x))
  v = x -> -r(x)^-1
In
  psi = F g
  nu = F v
  S = < psi | psi >
  V = < psi | nu * psi >
  T = 1/2 * sum_i=0^2 < del_i psi | del_i psi >
  print S, V, T, (T + V)/S
End
```

Let

$$\Omega = [-20, 20]^6$$

$$r1 = x \to \sqrt{x_0^2 + x_1^2 + x_2^2}$$

$$r2 = x \to \sqrt{x_3^2 + x_4^2 + x_5^2}$$

$$r12 = x \to \sqrt{(x_0 - x_3)^2 + (x_1 - x_4)^2 + (x_2 - x_5)^2}$$

$$g = x \to \left(1 + \frac{1}{2} * r12\,(x)\right) * \exp\left(-2 * (r1\,(x) + r2\,(x))\right)$$

$$v = x \to -\frac{2}{r1\,(x)} - \frac{2}{r2\,(x)} + \frac{1}{r12\,(x)}$$

In

$$\psi = \mathcal{F}\, g$$

$$\nu = \mathcal{F}\, v$$

$$S = \langle \psi | \psi \rangle$$

$$V = \langle \psi | \nu * \psi \rangle$$

$$T = \frac{1}{2} * \sum_{i=0}^{5} (\langle \nabla_i \psi | \nabla_i \psi \rangle)$$

print $S, V, T, \dfrac{T+V}{S}$

End

He atom Hylleraas 2-term 6D

Let

$$\Omega = [-20, 20]^3$$

$$r = x \to \sqrt{x_0^2 + x_1^2 + x_2^2}$$

$$g = x \to \exp(-2 * r(x))$$

$$v = x \to -\frac{2}{r(x)}$$

In

$$\nu = \mathcal{F}\, v$$

$$\phi = \mathcal{F}\, g$$

$$\lambda = -1.0$$

for $i \in [0, 10]$

$$\phi = \phi * \|\phi\|^{-1}$$

$$V = \nu - \nabla^{-2}\left(4 * \pi * \phi^2\right)$$

$$\psi = -2 * \left(-2 * \lambda - \nabla^2\right)^{-1}(V * \phi)$$

$$\lambda = \lambda + \frac{\langle V * \phi | \psi - \phi \rangle}{\langle \psi | \psi \rangle}$$

$$\phi = \psi$$

print "iter", i, "norm", $\|\phi\|$, "eval", λ

end

End

He atom Hartree-Fock

Compose directly in terms of functions and operators

This is a Latex rendering of a program to solve the Hartree-Fock equations for the helium atom

The compiler also output a C++ code that can be compiled without modification and run in parallel

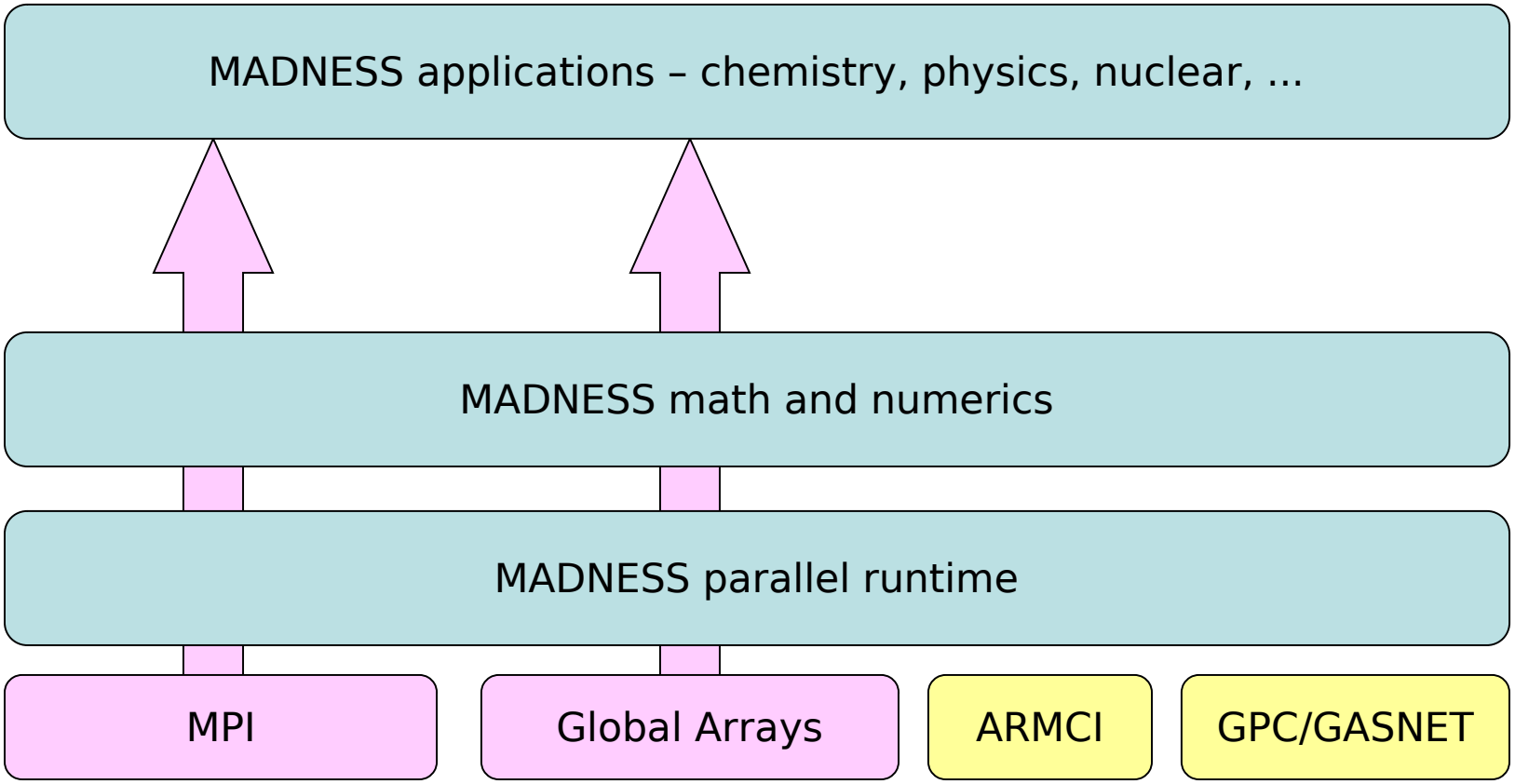
High-level composition

- Express ***ALL*** available parallelism without burdening programmer
  - Internally, MADNESS is looking after data and placement and scheduling of operations on individual functions
  - Programmer must express parallelism over multiple functions and operators
    - But is *not* responsible for scheduling or placement
    - However, efficiency often demands that the programmer provide info about where data should be located

Current Issues

- Load balancing
  - Currently a heuristic not a performance model
  - Work stealing prototype not in production code
- Too much parallelism!
  - Memory usage; need more than simple throttle
- Manual closures/continuations
  - Enables us to stay with standard C++
  - User-space threading helps but need S2S tools
- I/O
  - HDF5 seems right choice but it’s so complicated!
- Sequential kernels
  - Compiler generated code too slow by ~3x

MADNESS architecture



Intel Thread Building Blocks optional target for intranode runtime

Runtime Objectives

- Scalability to 1+M processors ASAP
- Runtime responsible for
  - scheduling and placement,
  - managing dependencies & hiding latency
- Compatible with existing models (MPI, GA)
- Borrow successful concepts from Cilk, Charm++, Python, HPCS languages

Why a new runtime?

- MADNESS computation is irregular & dynamic
  - 1000s of dynamically-refined meshes changing frequently & independently (to guarantee precision)

- Because we wanted to make MADNESS itself easier to write not just the applications using it
  - We explored implementations with MPI, Global Arrays, and Charm++ and all were inadequate

- MADNESS is helping drive
  - One-sided operations in MPI-3, DOE projects in fault tolerance, ...

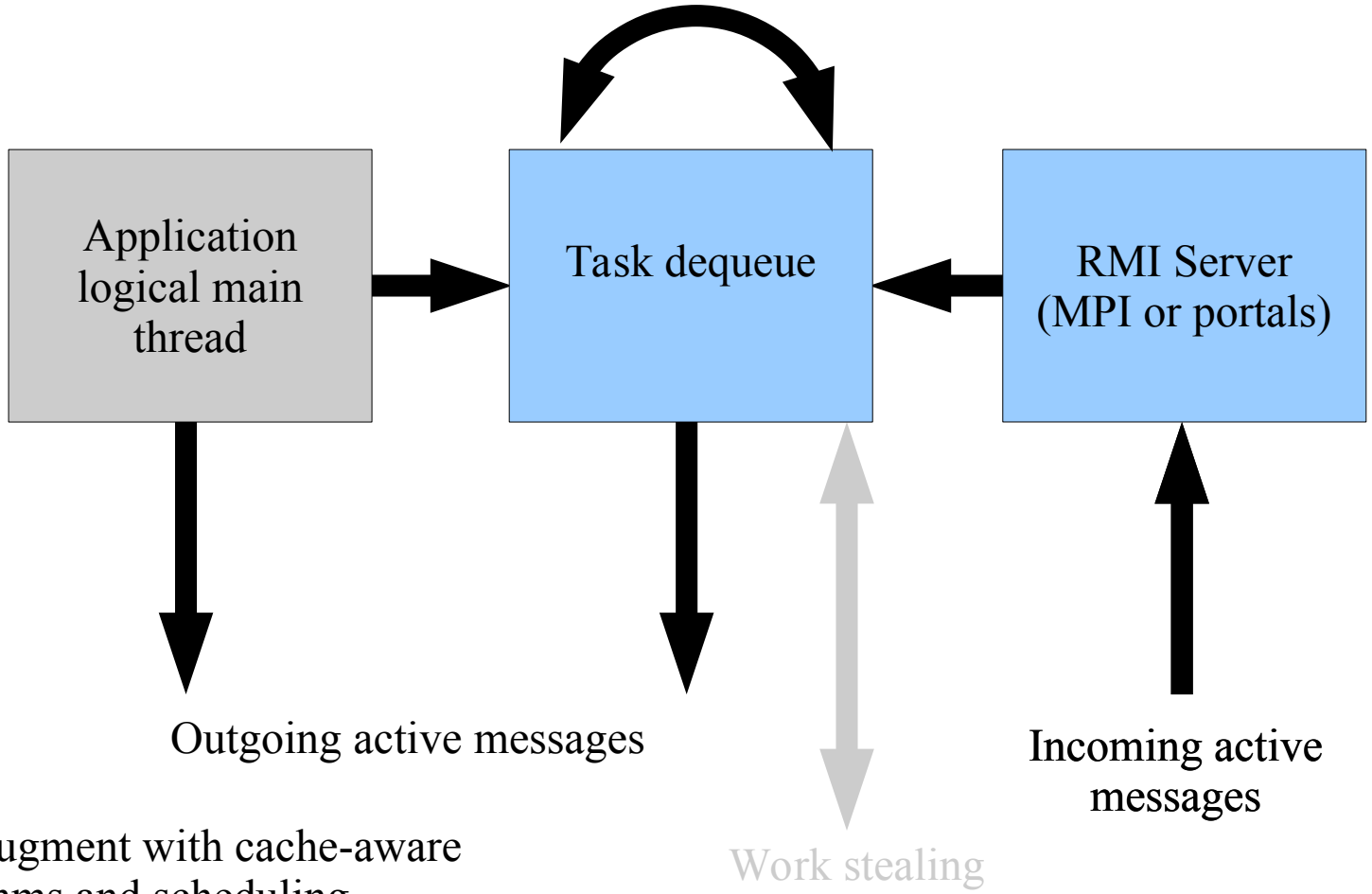
Key runtime elements

- Futures for hiding latency and automating dependency management
- Global names and name spaces
- Non-process centric computing
  - One-sided messaging between objects
  - Retain place=process for MPI/GA legacy compatibility
- Dynamic load balancing
  - Data redistribution, work stealing, randomization

Runtime layers

- Remote method invocation (RMI)
- Shared thread pool to execute ready tasks
- Futures for hiding latency and automating dependency management
- World – parallel execution environment
  - Wraps MPI communicator
  - Independent task queue for each world
  - Asynchronous global ops
  - Active messages
- Global names (WorldObject)
  - With APIs to send messages/tasks between nodes
- Global name spaces (WorldContainer)

Multi-threaded architecture



Must augment with cache-aware algorithms and scheduling

RMI

- Not exposed to users
- Handler routines have this type

```
typedef void (*rmi_handlerT)
    (void* buf, size_t nbyte);
```

- Send an asynchronous message

```
RMI::Request
  RMI::isend(const void* buf, size_t nbyte,
             int dest, rmi_handlerT func,
             unsigned int attr=0)
```

- Can probe/wait for completion

Active message interface

```
Class AmArg {
public:
    unsigned char* buf() const;
    ProcessID get_src() const;
    World* get_world() const;
    Archive& operator&(const T& t) const;
    Archive& operator&(T& t) const;
};
typedef void (*am_handlerT) (const AmArg&);

void send(ProcessID dest, am_handlerT op,
          const AmArg* arg, int attr);
```

Tasks

- Basic API takes pointer to object derived from

```
class TaskInterface {
public:
    virtual void run();
}
```

- User implements run function and adds task to queue (that takes ownership of ptr) using
  - Can add priority

```
world.taskq.add(pointer_to_task)
```

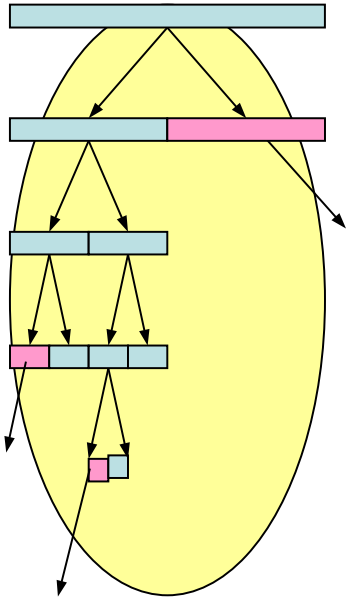
Futures

- Result of an asynchronous computation
  - Cilk, Java, HPCLs, C++0x
- Hide latency due to communication or computation
  -
- Management of dependencies
  - Via callbacks

```
int f(int arg);
ProcessId me, p;

Future<int> r0=task(p, f, 0);
Future<int> r1=task(me, f, r0);

// Work until need result

cout << r0 << r1 << endl;
```

Process “me” spawns a new task in process “p” to execute `f(0)` with the result eventually returned as the value of future `r0`. This is used as the argument of a second task whose execution is deferred until its argument is assigned. Tasks and futures can register multiple local or remote callbacks to express complex and dynamic dependencies.

Virtualization of data and tasks



Future:
- MPI rank
- probe()
- set()
- get()

Task:
- Input parameters
- Output parameters
- probe()
- run()
- get()

```
Future Compress(tree):
      Future left = Compress(tree.left)
      Future right = Compress(tree.right)
      return Task(Op, left, right)

Compress(tree)
Wait for all tasks to complete
```

Benefits: Communication latency & transfer time largely hidden
Much simpler composition than explicit message passing
Positions code to use "intelligent" runtimes with work stealing
Positions code for efficient use of multi-core chips
Locality-aware and/or graph-based scheduling

Global Names

- Objects with global names with different state in each process
  - C.f. shared[threads] in UPC; co-Array
- Non-collective constructor; deferred destructor
  - Eliminates synchronization

```
class A : public WorldObject<A>
{
  int f(int);
};
ProcessID p;
A a;
Future<int> b =
  a.task(p,&A::f,0);
```

A task is sent to the instance of a in process p. If this has not yet been constructed the message is stored in a pending queue. Destruction of a global object is deferred until the next user synchronization point.

Global Namespaces

- Specialize global names to containers
  - Hash table, arrays, ...
- Replace global pointer (process+local pointer) with more powerful concept
-
- User definable map from keys to “owner” process

```
class Index;  // Hashable
class Value {
  double f(int);
};

WorldContainer<Index,Value> c;
Index i,j;  Value v;
c.insert(i,v);
Future<double> r =
  c.task(j,&Value::f,666);
```

A container is created mapping indices to values.

A value is inserted into the container.

A task is spawned in the process owning key `j` to invoke `c[j].f(666)`.

Namespaces are a large part of the elegance of Python and success of Charm++ (chares+arrays)

```
#define WORLD_INSTANTIATE_STATIC_TEMPLATES
#include <world/world.h>
using namespace madness;
class Foo : public WorldObject<Foo> {
    const int bar;
public:
    Foo(World& world, int bar) : WorldObject<Foo>(world), bar(bar)
                {process_pending();}

    int get() const {return bar;}
};
int main(int argc, char** argv) {
    MPI::Init(argc, argv);
    madness::World world(MPI::COMM_WORLD);

    Foo a(world,world.rank()), b(world,world.rank()*10)

    for (ProcessID p=0; p<world.size(); p++) {
        Future<int> futa = a.send(p,&Foo::get);
        Future<int> futb = b.send(p,&Foo::get);
        // Could work here until the results are available
        MADNESS_ASSERT(futa.get() == p);
        MADNESS_ASSERT(futb.get() == p*10);
    }
    world.gop.fence();
    if (world.rank() == 0) print("OK!");
    MPI::Finalize();
}
```

Figure 1: Simple client-server program implemented using WorldObject.

```
#define WORLD_INSTANTIATE_STATIC_TEMPLATES
#include <world/world.h>

using namespace std;
using namespace madness;

class Array : public WorldObject<Array> {
   vector<double> v;
public:
   /// Make block distributed array with size elements
   Array(World& world, size_t size)
      : WorldObject<Array>(world), v((size-1)/world.size()+1)
   {
      process_pending();
   };

   /// Return the process in which element i resides
   ProcessID owner(size_t i) const {return i/v.size();};

   Future<double> read(size_t i) const {
      if (owner(i) == world.rank())
         return Future<double>(v[i-world.rank()*v.size()]);
      else
         return send(owner(i), &Array::read, i);
   };

   Void write(size_t i, double value) {
      if (owner(i) == world.rank())
         v[i-world.rank()*v.size()] = value;
      else
         send(owner(i), &Array::write, i, value);
      return None;
   };
};
```

```
int main(int argc, char** argv) {
   initialize(argc, argv);
   madness::World world(MPI::COMM_WORLD);

   Array a(world, 10000), b(world, 10000);

   // Without regard to locality, initialize a and b
   for (int i=world.rank(); i<10000; i+=world.size()) {
      a.write(i, 10.0*i);
      b.write(i,  7.0*i);
   }
   world.gop.fence();

   // All processes verify 100 random values from each array
   for (int j=0; j<100; j++) {
      size_t i = world.rand()%10000;
      Future<double> vala = a.read(i);
      Future<double> valb = b.read(i);
      // Could do work here until results are available
      MADNESS_ASSERT(vala.get() == 10.0*i);
      MADNESS_ASSERT(valb.get() ==  7.0*i);
   }
   world.gop.fence();

   if (world.rank() == 0) print("OK!");
   finalize();
}
```

Complete example program illustrating the implementation and use of a crude, block-distributed array upon the functionality of `WorldObject`.

Serialization

- Convert an object (and containers thereof) into a serial stream of bytes
  - Conceptually based on Boost serialization
  - Symmetric & operator for input/outut
  - ar & x & y & z;
  - Asymmetric operators also (<<, >>)
- Fundamental types and containers
- User types provide supported through both intrusive and non-intrusive methods
- Used to send objects between processes, to disk

Example symmetric seralization

```
class A {
    float a;
public:
    A(float a = 0.0) : a(a) {}
    template <class Archive>
    inline void serialize(Archive& ar) {ar & a;}
};
A a(99.0), b;
BinaryFstreamOutputArchive oar(ofile);
oar & a; // saves a to file
...
BinaryFstreamInputArchive iar(ifile);
iar & b; // loads b from file
```

C++ templates automate many things

- Wrapping calls to functions or lambdas or object member functions inside a task
- Managing dependencies by connecting futures and tasks through the dependency interface (callbacks)
- Assisting programmers to avoid explicit use of futures as arguments to functions
- (de)Serializing arguments to remote tasks
- Templates don’t add additional runtime overhead (compile time)

Abstraction Overheads

- If you are careful you win
  - *Increased performance and productivity*
  - This is the lesson of Global Arrays, Charm++, …
- Creating, executing, reaping a local, null task – 350ns (100K tasks, 3GHz Core2, Pathscale 3.0, -Ofast) dominated by new/delete
- Chain of 100K dependent tasks with the result of a task as the unevaluated argument of the previous task
  - ~1 us per task
- Creating a remote task adds overhead of inter-process communication which is on the scale of 5us (Cray XT).
  - Aggregation can reduce this.
- Switching between user-space threads <20ns

Success with tasks

- Can readily compose parallel versions of many complex and irregular algorithms
- Can obtain good performance
  - On multi-threaded CPUs
  - With attention to details
  - Especially effective for irregular work loads and complex, even dynamic, dependencies

Problems with tasks without both compiler+intelligent runtime

- Manual continuation passing - everywhere code may block need to partition into separate tasks (zillions of them!)
- Rigid decisions made about granularity and decomposition
- Hard to automatically aggregate many small tasks
- Can make portability from CPU to GPU worse
- Resource management – bounding buffers and not drowning/starving in parallelism: partially manage with task generators, data flow, task priorities
- Efficient execution challenging esp. use of memory hierarchy, scheduling of critical path
- Makes easy things harder (e.g., parallel for)
- Task specification often distant from task use - very effective for code obfuscation – partially fixed in modern C++
- Interoperability with just about everything else is painful

Some issues

- Excessive global barriers
  - Termination detection for global algorithms on distributed mutable data structures
- Messy, nearly redundant code expressing variants of algorithms on multiple trees
  - Need some templates / code generation
- Need efficient and easy way to aggregate data/work to exploit GPGPUs
- Efficient kernels for GPGPUs (single SM)
  - Non-square matrices, shortish loops – performance problem
- Switching between single-/multi-thread tasks
- Efficient multi-threaded code for thread units sharing L1 (e.g., BGQ, Xeon Phi)
- Multiple interoperable DSLs embedded in or generating general purpose language
- Need kitchen sink environment – full interoperability between runtimes, data structures, external I/O libraries, etc.

Summary

- MADNESS is a general purpose framework for scientific simulation
  - Conceived for the petascale to exascale era
  - Increases HPC productivity by reducing many sources of complexity
  - Deploys advanced math, numerics, and C/S

http://code.google.com/p/m-a-d-n-e-s-s

Summary

- MADNESS
  - High-level composition – functions and operators
  - Fast computation with guaranteed precision
  - Separated representations of operators (and functions)
- DFT code nearing production quality
  - GGAs, hybrid, TDDFT, derivatives, pseudopotentials
- Efficient execution on parallel computers
  - Cray XE, IBM BGQ, multithreading
- High-productivity parallel programming framework

http://code.google.com/p/m-a-d-n-e-s-s

Summary

- Exascale programming models
  - Resilience, Power, Performance, Productivity
  - Productivity is arguably the most important
  - Enable innovation and discovery at scale
- MADNESS and NWChem
  - Frameworks – places for disciplines to meet to leverage investments and expertise
  - Face different challenges in moving forward
- Data and computation are inseparable challenges

BROOKHAVEN NATIONAL LABORATORY

Stony Brook University

Summary

- We need radical changes in how we compose scientific S/W
  - Complexity at limits of cost and human ability
  - Need extensible tools/languages with support for code transformation not just translation
- Students need to be prepared for computing and data in 2020+ not as it was in 2000 and before
  - Pervasive, massive parallelism
  - Bandwidth limited computation and analysis
  - An intrinsically multidisciplinary activity

Summary

- MADNESS
  - High-level composition – functions and operators
  - Fast computation with guaranteed precision
  - Separated representations of operators (and functions)
- DFT code nearing production quality
  - GGAs, hybrid, TDDFT, derivatives, pseudopotentials
- Efficient execution on parallel computers
  - Cray XE, IBM BGQ, multithreading
- High-productivity parallel programming framework

Funding

- NSF ACI-1450344
- DOE: SciDAC, Office of Science divisions of Advanced Scientific Computing Research and Basic Energy Science, under contract DE-AC05-00OR22725 with Oak Ridge National Laboratory, in part using the National Center for Computational Sciences.
- DARPA HPCS2: HPCS programming language evaluation
- NSF CHE-0625598: Cyber-infrastructure and Research Facilities: Chemical Computations on Future High-end Computers
- NSF CNS-0509410: CAS-AES: An integrated framework for compile-time/run-time support for multi-scale applications on high-end systems
- NSF OCI-0904972: Computational Chemistry and Physics Beyond the Petascale

References

B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi, *"Adaptive Solution of Partial Differential Equations in Multiwavelet Bases,"* Journal of Computational Physics, v. 182, pp. 149-190, 2002

G.I. Fann, G. Beylkin, R.J. Harrison and K.E. Jordan, *"Singular operators in multiwavelet bases,"* IBM J. Res. Dev., 48 (2004) 161.

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 (2004) 11587.

T. Yanai, G.I. Fann, Z. Gan, R.J. Harrison, G. Beylkin, *"Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for Hartree-Fock and density functional theory,"* J. Chem. Phys., 121 (2004) 2866.

T. Yanai, R.J. Harrison and N.C. Handy, *"Multiresolution quantum chemistry in multiwavelet bases: time-dependent density functional theory with asymptotically corrected potentials in local density and generalized gradient approximations,"* Mol. Phys., 103 (2004) 403.

R. Harrison, G. Fann, Z. Gan, T. Yanai, S. Sugiki, A. Beste, and G. Beylkin, *"Multiresolution computational chemistry,"* J. Physics, Conference Series, 16, 243-246, 2005.

T. Yanai, R.J. Harrison, G.I. Fann and G. Beylkin, *"Multi-resolution quantum chemistry: linear response for excited states,"* J. Chem. Phys., submitted for publication, 2005.

G. Beylkin, R. Cramer, G. Fann and R. J. Harrison, *"Multiresolution separated representations of singular and weakly singular operators,"* Applied and Computational Harmonic Analysis, 23 (2007) 235-253

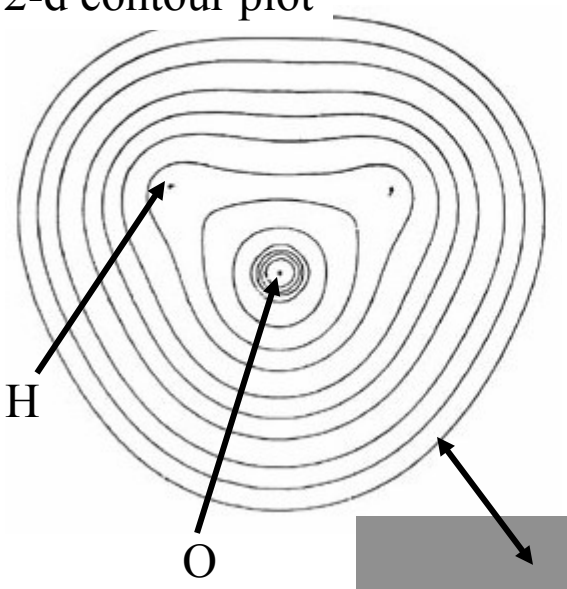
H. Sekino, Y. Maeda, T. Yanai, and R.J. Harrison, *"Basis set limit Hartree-Fock and density functional theory response property evaluation by multiresolution multiwavelet basis,"* J. Chem. Phys,129(3):034111 , 2008

G.I. Fann, J.C. Pei, R.J. Harrison ,J. Jia, J.C. Hill, M.J. Ou, W. Nazarewicz, W.A. Shelton and N. Schunck., *"Fast multiresolution methods for density functional theory in nuclear physics,"* Journal of Physics, 180, 012080, 2009

M. Reuter, J. Hill and R.J. Harrison, *"Solving PDEs in Irregular Geometries with Multiresolution Methods I: Embedded Dirichlet Boundary Conditions,"* Computer Physics Communications, submitted March 2010

Molecular orbitals of water

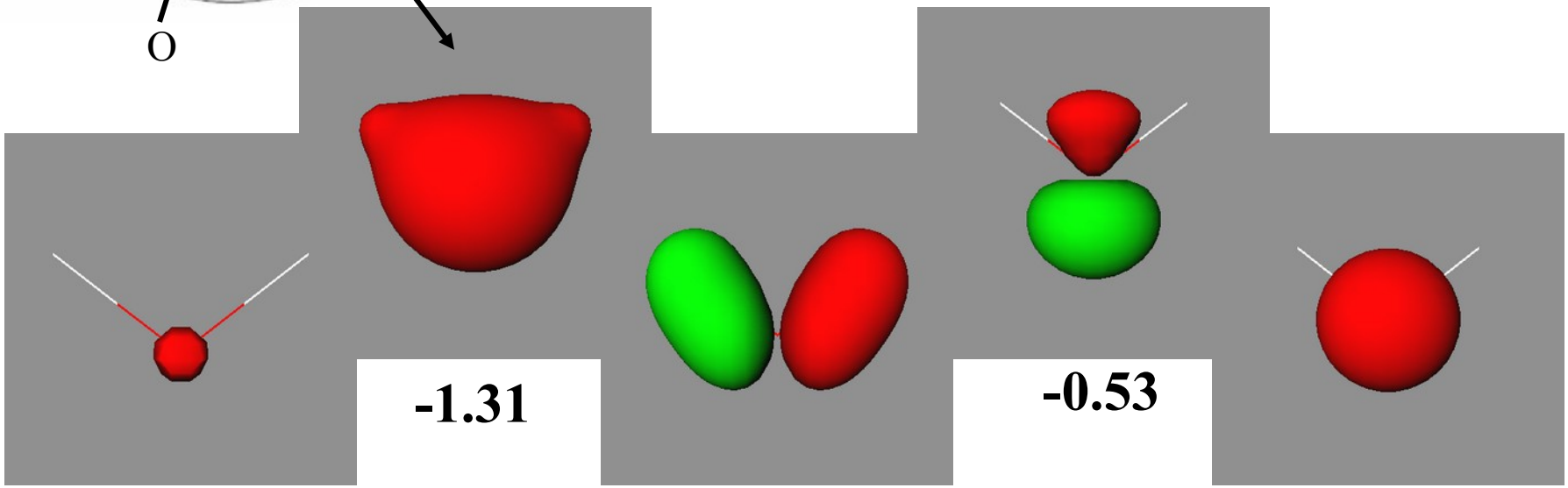
2-d contour plot



Iso-surfaces are 3-d contour plots – they show the surface upon which the function has a particular value

Water has 10 electrons (8 from oxygen, 1 from each hydrogen).

It is closed-shell, so it has 5 molecular orbitals each occupied with two electrons.



The energy of each orbital in atomic units

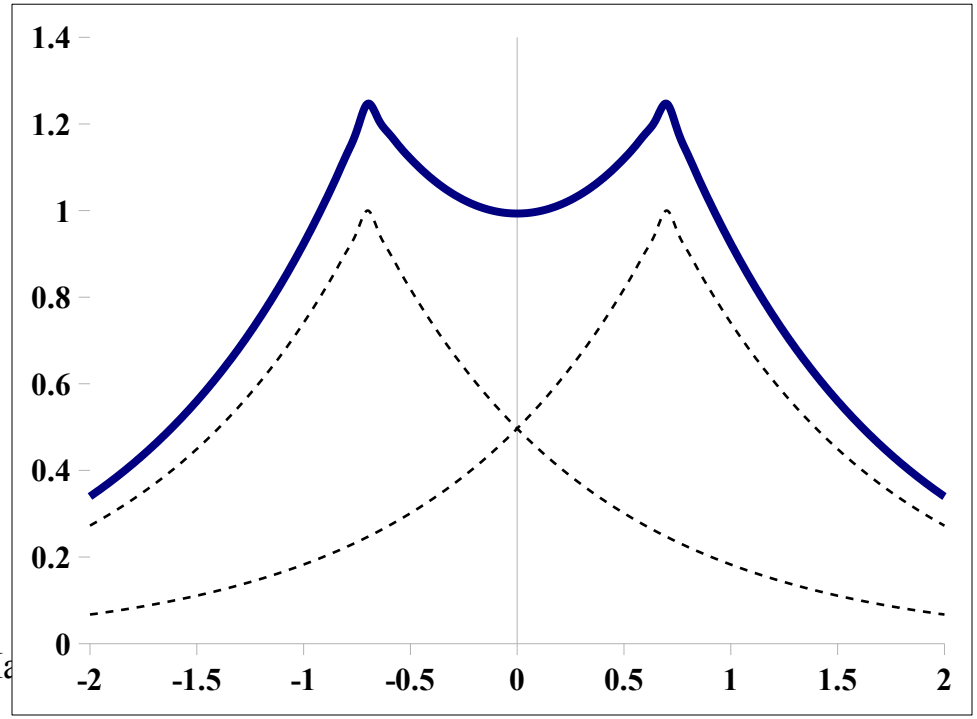
Linear Combination of Atomic Orbitals (LCAO)

- Molecules are composed of (weakly) perturbed atoms
  - Use finite set of atomic wave functions as the basis
  - Hydrogen-like wave functions are exponentials
- E.g., hydrogen molecule (H_2)

$$1s(r) = e^{-|r|}$$

$$\phi(r) = e^{-|r-a|} + e^{-|r-b|}$$

- Smooth function of molecular geometry
- MOs: cusp at nucleus with exponential decay



LCAO with Gaussian Functions

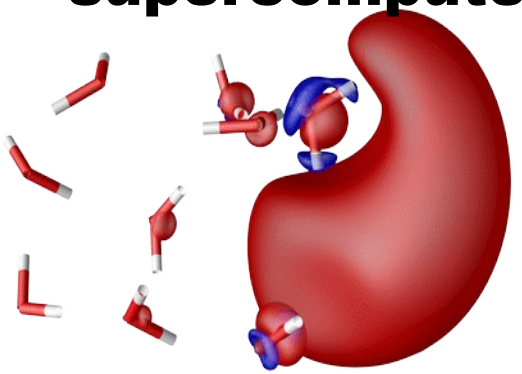
- Cannot compute integrals over exponential orbitals
- Boys (1950) noted that Gaussians are feasible
  - 6D integral reduced to 1D integrals which are tabulated once and stored (related to error function)
- Gaussian functions form a complete basis
  - With enough terms any radial function can be approximated to any precision using a linear combination of Gaussian functions

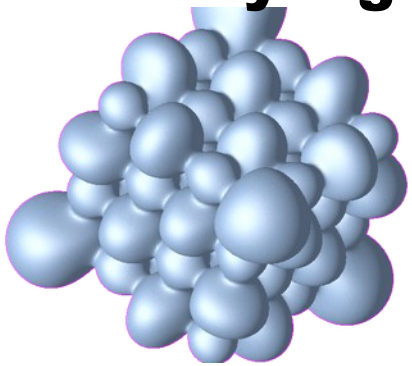
$$f(r) = \sum_{i=1}^{N} c_i e^{-a_i r^2} + O(\epsilon)$$

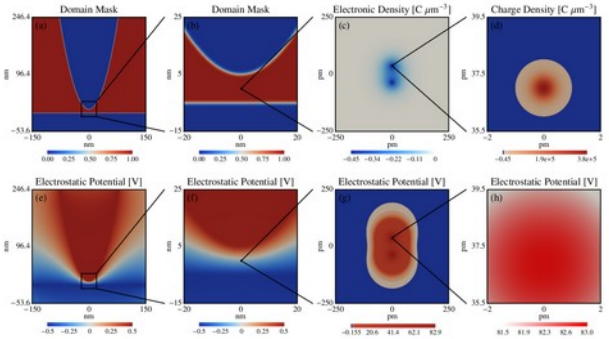
LCAO

- A fantastic success, but …
- Basis functions have extended support
  - causes great inefficiency in high accuracy calculations (functions on different centers overlap)
  - origin of non-physical density matrix
- Basis set superposition error (BSSE)
  - incomplete basis on each center leads to over-binding as atoms are brought together
- Linear dependence problems
  - accurate calculations require balanced approach to a complete basis on every atom
  - molecular basis can have severe linear dependence
- Must extrapolate to complete basis limit
  - unsatisfactory and not feasible for large systems

MADNESS deploys advanced math and C/S to enable robu solution of diverse physical problems on the largest supercomputers with a very high level of composition







Chemistry: coupled non-linear Schrodinger equations

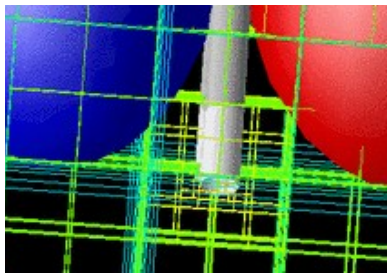
$$\phi_i(\boldsymbol{r}) = 2\left(\nabla^2 + 2E\right)^{-1}\hat{V}(\boldsymbol{r})\phi_i(\boldsymbol{r})$$

Solid state: band structure including lattice sums

$$f_{nk}(\boldsymbol{r}) = \sum_{\boldsymbol{R}} \int \frac{e^{-\mu|\boldsymbol{r}-\boldsymbol{r}'-\boldsymbol{R}|}}{|\boldsymbol{r}-\boldsymbol{r}'-\boldsymbol{R}|} u_{nk}(\boldsymbol{r}')\, d^3r'$$

Nanoscience: electrostat with 7 decades of length

$$u(r) = \int G(\boldsymbol{r},\boldsymbol{r}')\rho(\boldsymbol{r}')\, d^3r' + \oint_{\partial\Omega}\left[G(\boldsymbol{r},\boldsymbol{r}')\nabla' u(\boldsymbol{r}') - u(\boldsymbol{r}')\nabla' G(\boldsymbol{r},\boldsymbol{r}')\right].\, \boldsymbol{dS}$$



Adaptive discontinuous spectral element

$$f(x_1,\ldots,x_n) = \sum_{l=1}^{M} \sigma_l \prod_{i=1}^{d} f_i^{(l)}(x_i) + O(\epsilon)$$

Separated representations

$$V_n = V_0 + \left(V_1 - V_0\right) + \cdots + \left(V_n - V_{n-1}\right)$$

Multiresolution analys

Example tree in Haar basis

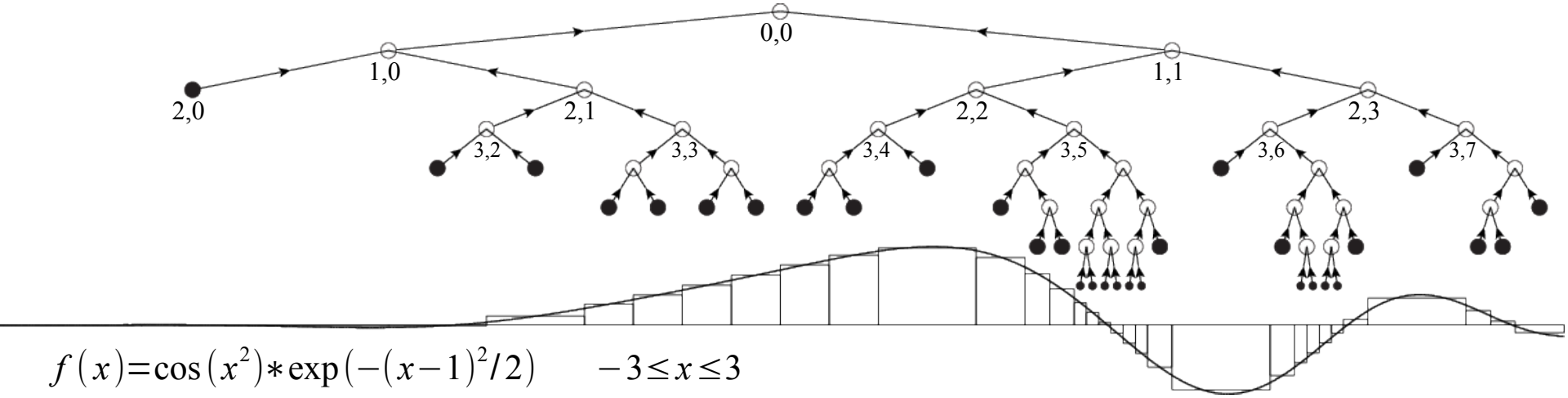
Haar basis is a piecewise constant (like a histogram)
- Not useful for real calculations but easy to visualize and of fundamental importance

Adaptive local refinement until local error measure is satisfied
- Smaller boxes where rate of change is high and value not negligible

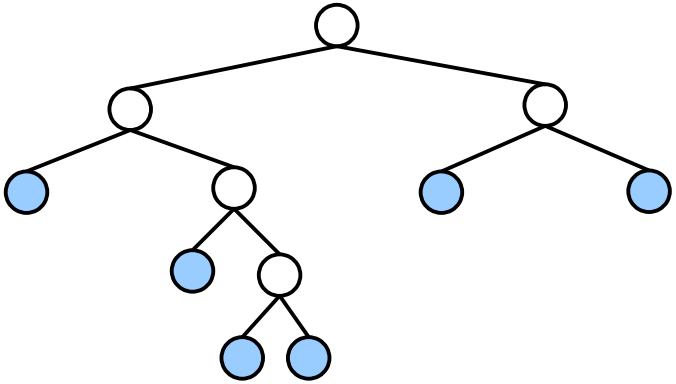
Conventional adaptive mesh corresponds to boxes

Construct tree connecting fine-scale to coarser-scale boxes

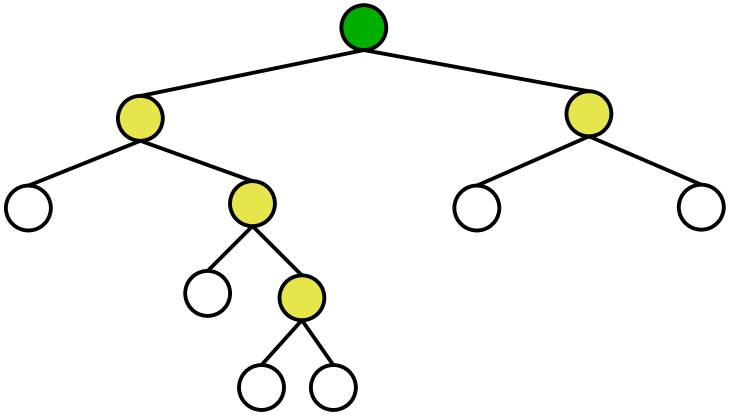
Boxes labeled with level ($n=0,1,\ldots$) and translation ($l=0,1,\ldots,2^n-1$)



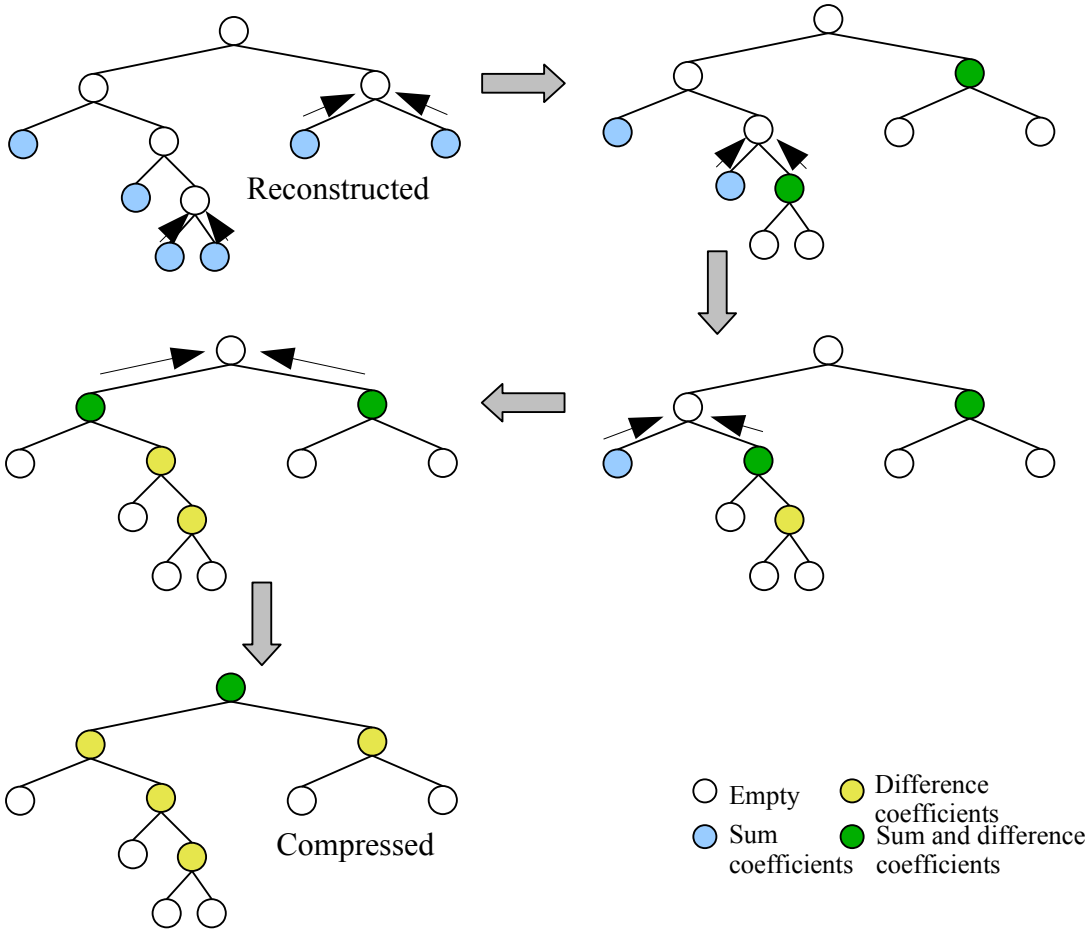
$$f(x)=\cos(x^2)*\exp(-(x-1)^2/2) \qquad -3\le x\le 3$$



Tree in **reconstructed** form. Scaling function (sum) coefficients at leaf nodes. Interior nodes empty.



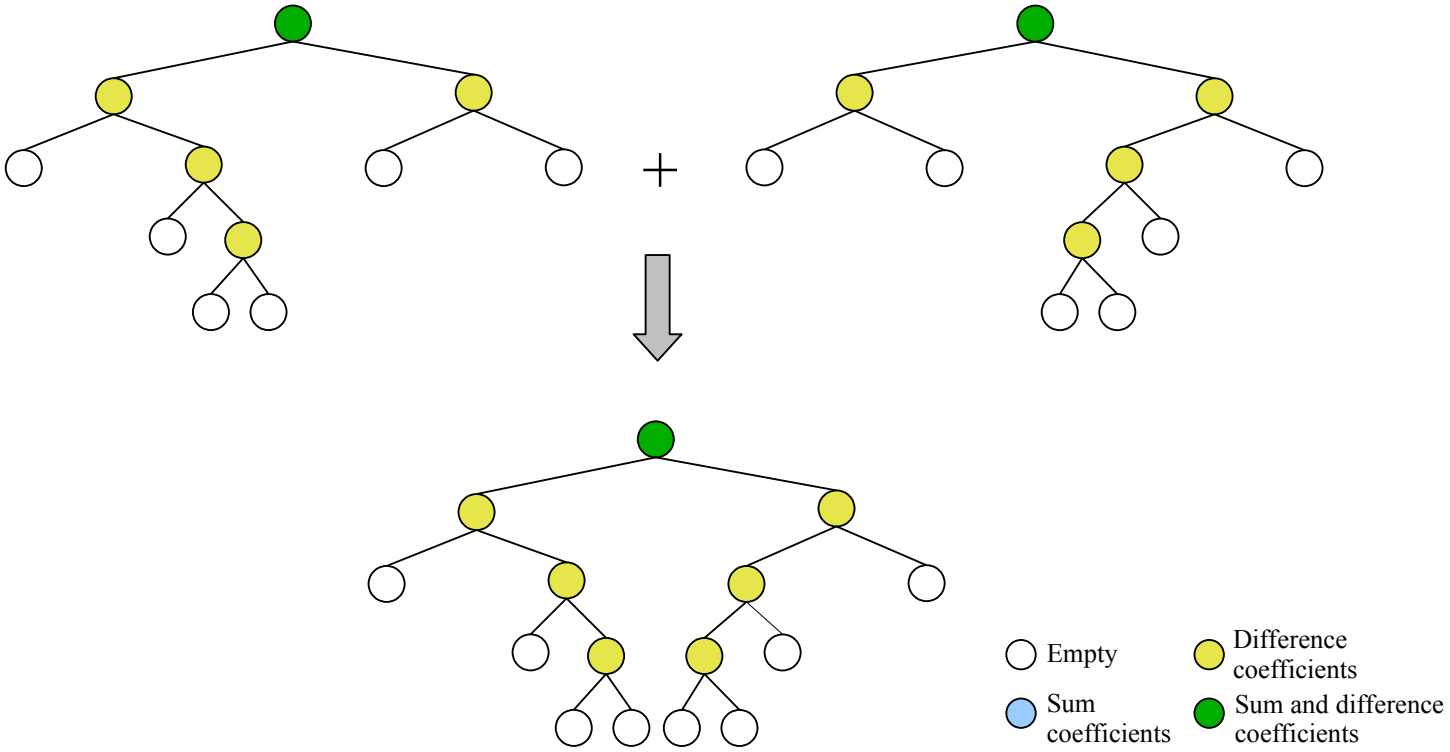
Tree in **compressed** form. Wavelet (difference) coefficients at interior nodes, with scaling functions coefficients also at root. Leaf nodes empty.



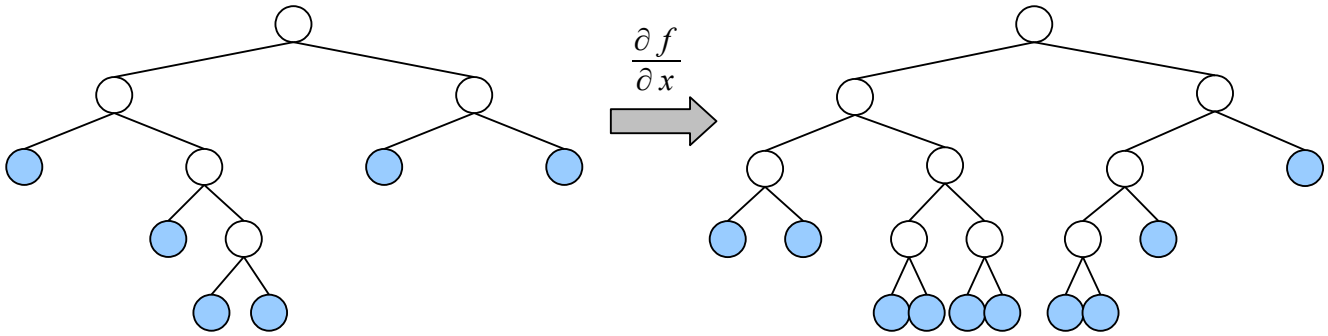
Compression algorithm. Starting from leaf nodes, scaling function (sum) coefficients are passed to parent. Parent "filters" the childrens' coefficients to produce sum and wavelet (difference) coefficients at that level, then passes sum coefficients to its parent.

Reconstruction is simply the reverse processes.

To produce the non-standard form the compression algorithm is run but scaling function coefficients are retained at the leaf and interior nodes.



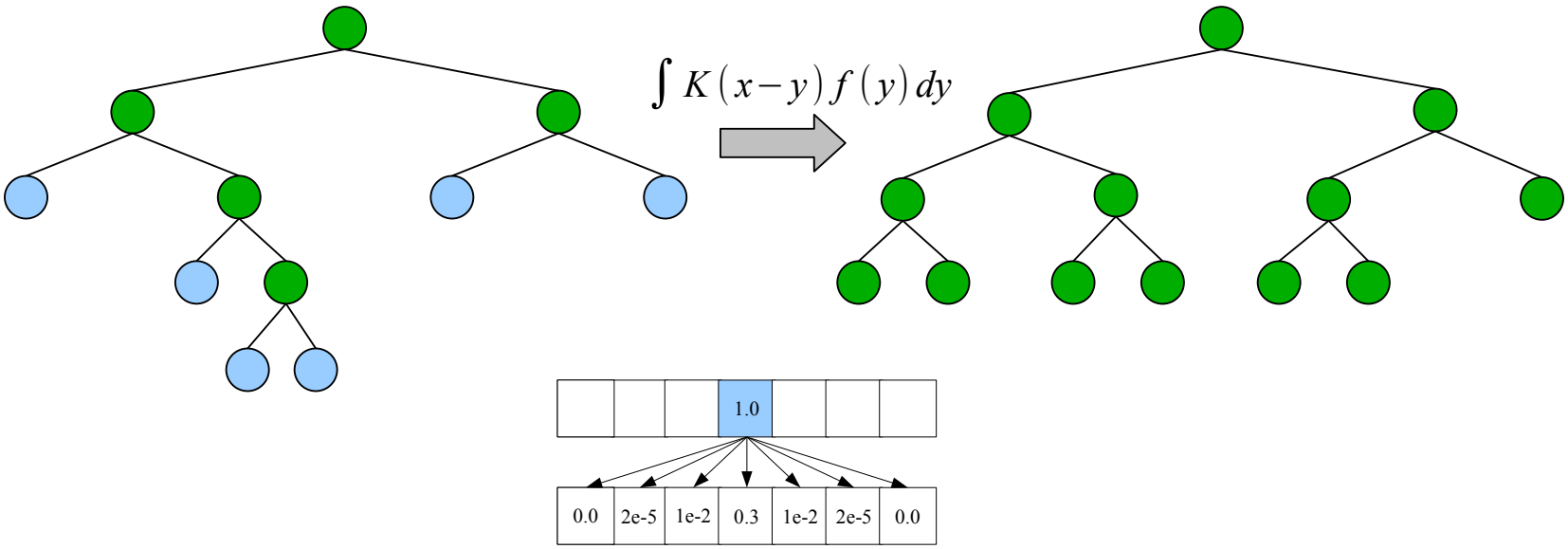
Addition is (most straightforwardly) performed in the compressed form. Coefficients are simply added with missing nodes being treated as if zero.



Differentiation (for simplicity here using central differences and Dirichlet boundary conditions) is applied in the scaling function basis. To compute the derivative of the function in the box corresponding to a leaf node, we require the coefficients from the neighboring boxes at the same level.

- If the neighboring leaf nodes exist, all is easy.
- If it exists at a higher level,we can make the coefficients by recurring down from the parent using the two-scale relation.
- If the neighbor exists at a finer scale, we must recur down until both neighbors are at the same level.

Hence, phrased as parallel computation on all leaf nodes , differentiation must search for neighbors in the tree at the same and higher levels, and may initiate computation at lower levels. It can also be phrased as a recursive descent of the tree, which can have advantages in reducing the amount of probes up the tree for parents of neighbors (esp. in higher dimensions).



Convolution The first step is to compress into non-standard form with scaling function and wavelet coefficients at each interior node. Then, we can independently compute the contribution of each box (node) to the result *at the same level of the tree*. Depending upon dimensionality, accuracy, and the kernel (K), we usually only need to compute the contributions of a box to itself and its immediate neighbors. The support (i.e., level of refinement) of the result is very dependent on the kernel. Here we consider convolution with a Gaussian (Green's function for the heat equation) which is a *smoothing* operator. After the computation is complete, we must sum down the tree to recover the standard form.

Hence, phrased as computation on all the nodes in non-standard form, convolution requires compression and reconstruction, and during the computation communicates across the tree at the same level to add results into neighboring boxes and up to connect new nodes to parents.