Chemistry beyond the petascale

Why is it necessary?
How do we get there?

Robert J. Harrison

harrisonrj@ornl.gov

robert.harrison@utk.edu

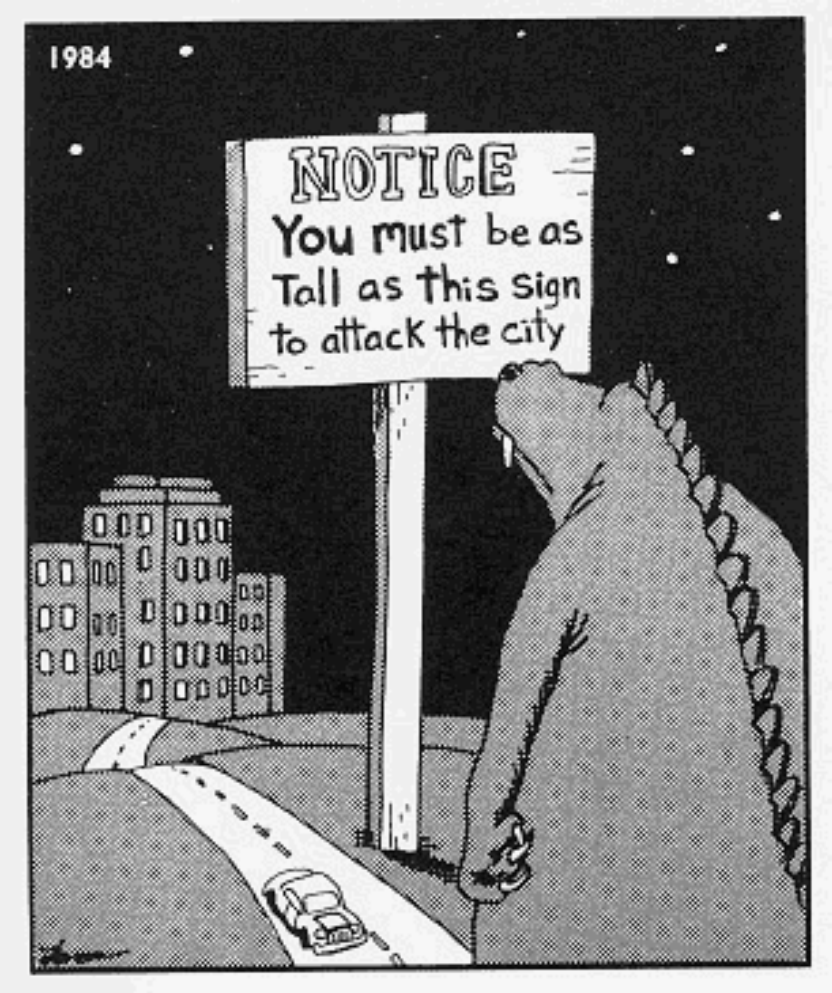
If you're not scared, you're not thinking big enough.

Robert J. Harrison
Oak Ridge National Laboratory,
University of Tennessee, Knoxville

Impact of sustained exponential growth

- We are only beginning to realize the transforming power of computing as an enabler of innovation and discovery.

- A characteristic of exponential growth is that we will make as much progress in the next doubling cycle as we've made since the birth of the field:
  - 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...



Computing now …

- The death of sequential computing
- Does anyone in the room still have a single cpu
  - Desktop computer?
  - Laptop?
  - Cell phone?

Computing in 2022 - I

- Looking back from 2007 to 1992
  - About 500x increase in desktop performance
    - 100MHz to 2x2 3GHz Core2
    - 30x from clock, 8x from parallelism
  - About 2500x increase in supercomputer speed
    - 100GF to 250TF
    - 30x from clock, 40x from parallelism

Computing in 2022 - II

- Looking forward to 2022
  - Expect same performance increases
  - Almost entirely from increased parallelism
  - Custom devices with much higher speed
  - Memory and I/O hierarchy much deeper

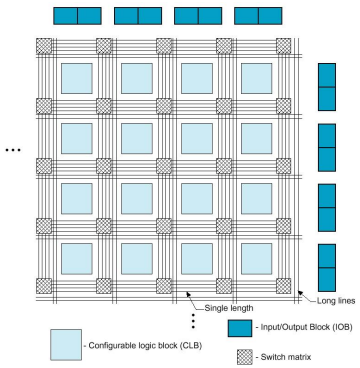
  - 20K * 2500 = 500M “processors”

O(1) programmers … O(10,000) nodes … O(100,000) processors … O(10,000,000) threads

- Complexity kills … sequential or parallel
- Expressing/managing concurrency at the petascale
  - It is too trite to say that the parallelism is in the physics
  - Must express and discover parallelism at more levels
  - Low level tools (MPI, Co-Array Fortran, UPC, …) don't discover parallelism or hide complexity or facilitate abstraction
- Management of the memory hierarchy
  - Memory will be deeper ; less uniformity between vendors
  - Need tools to automate and manage this, even at runtime

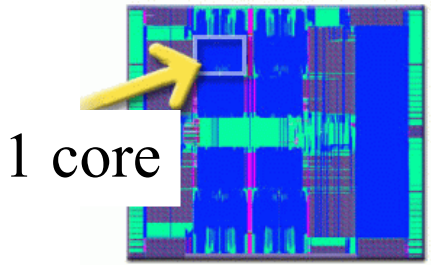
Other technologies

- Field programmable
  gate arrays – multi TOP/s now



- General purpose
  graphical processor unit – 1TFLOP/s now

- Highly threaded devices



- FLOPs are cheap; bandwidth is expensive

The way forward demands a change in paradigm

- by us chemists, the funding agencies, and the supercomputer centers

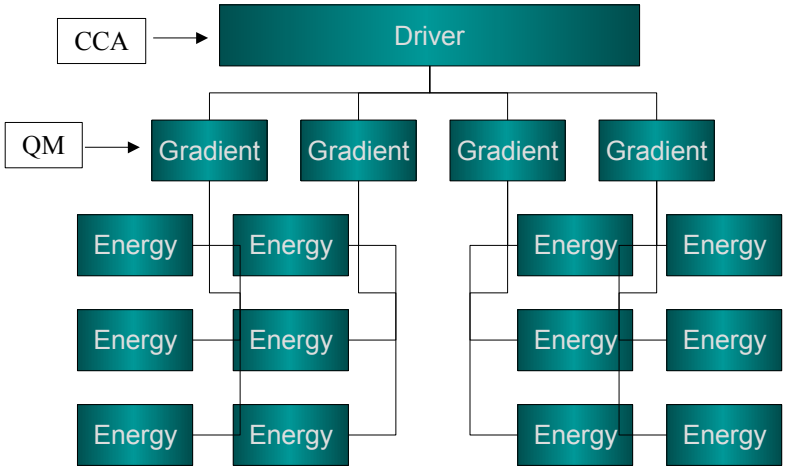
- A communal effort recognizing the increased cost and complexity of code development for modern theory at the petascale

- Re-emphasizing basic and advanced theory and computational skills in undergraduate and graduate education

Computational Chemistry Endstation

International collaboration spanning 7 universities and 6 national labs

- Led out of UT/ORNL
- Focus
  - Actinides, Aerosols, Catalysis
- ORNL Cray XT, ANL BG/L



TL Windus

Capabilties:

- Chemically accurate thermochemistry
  - Many-body methods required
- Mixed QM/QM/MM dynamics
  - Accurate free-energy integration
  - Simulation of extended interfaces
- Families of relativistic methods

Participants:

- Harrison, UT/ORNL
- Sherrill, GATech
- Gordon, Windus, Iowa State / Ames
- Head-Gordon, U.C. Berkeley / LBL
- Crawford, Valeev, VTech.
- Bernholc, NCSU
- (Knowles, U. Cardiff, UK)
- (de Jong, PNNL)
- (Shepard, ANL)
- (Sherwood, Daresbury, UK)

MADNESS

Multiresolution
Adaptive
Numerical
Scientific
Simulation

Multiresolution Adaptive Numerical Scientific Simulation

Ariana Beste[1], George I. Fann[1], Robert J. Harrison[1,2], Rebecca Hartman-Baker[1], Jun Jia[1], Shinichiro Sugiki[1]

[1]Oak Ridge National Laboratory

[2]University of Tennessee, Knoxville

In collaboration with

Gregory Beylkin[4], Fernando Perez[4], Lucas Monzon[4], Martin Mohlenkamp[5] and others

[4]University of Colorado

[5]Ohio University

harrisonrj@ornl.gov

The DOE funding

- This work is funded by the U.S. Department of Energy, the divisions of Advanced Scientific Computing Research and Basic Energy Science, Office of Science, under contract DE-AC05-00OR22725 with Oak Ridge National Laboratory. This research was performed in part using
  - resources of the National Energy Scientific Computing Center which is supported by the Office of Energy Research of the U.S. Department of Energy under contract DE-AC03-76SF0098,
  - and the Center for Computational Sciences at Oak Ridge National Laboratory under contract DE-AC05-00OR22725 .

SciDAC

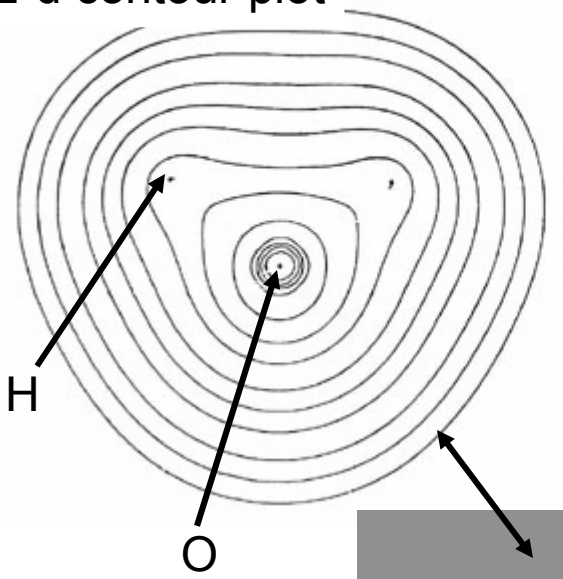
Scientific Discovery through Advanced Computing

Multiresolution chemistry objectives

- Scaling to 1+M processors ASAP
- Complete elimination of the basis error
  - One-electron models (e.g., HF, DFT)
  - Pair models (e.g., MP2, CCSD, …)
- Correct scaling of cost with system size
- General approach
  - Readily accessible by students and researchers
  - Higher level of composition
  - Direct computation of chemical energy differences
- New computational approaches
  - *Fast algorithms with guaranteed precision*

Molecular orbitals of water

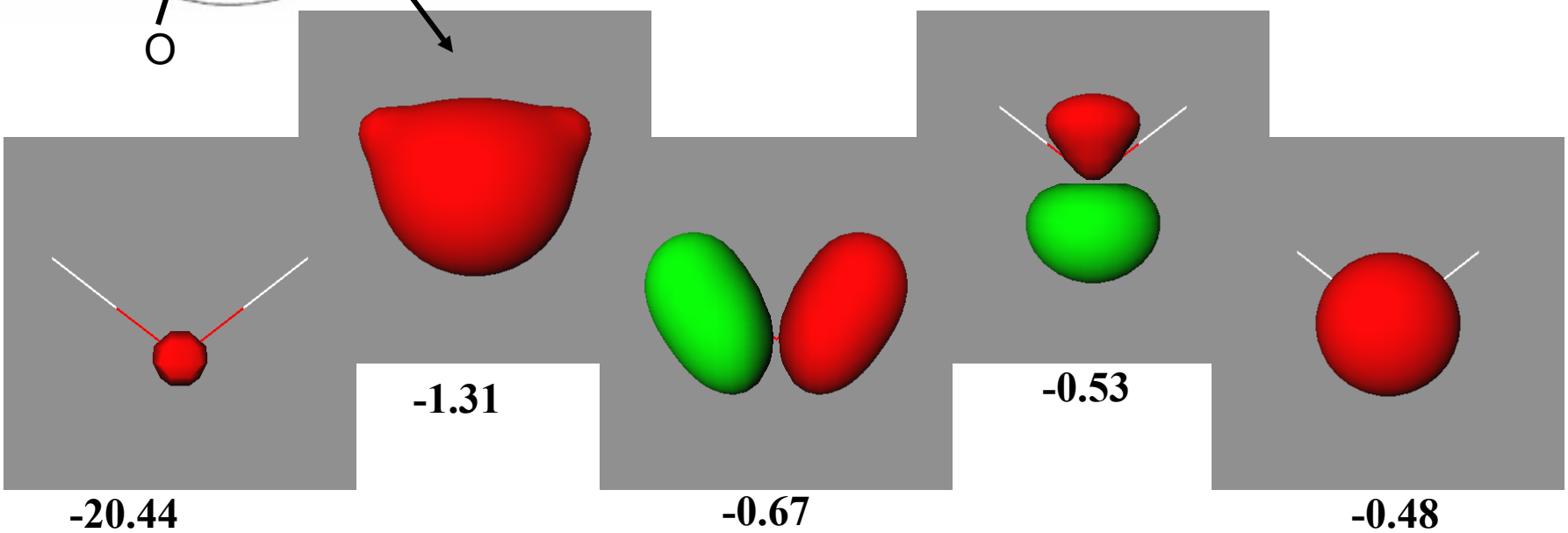
2-d contour plot



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.



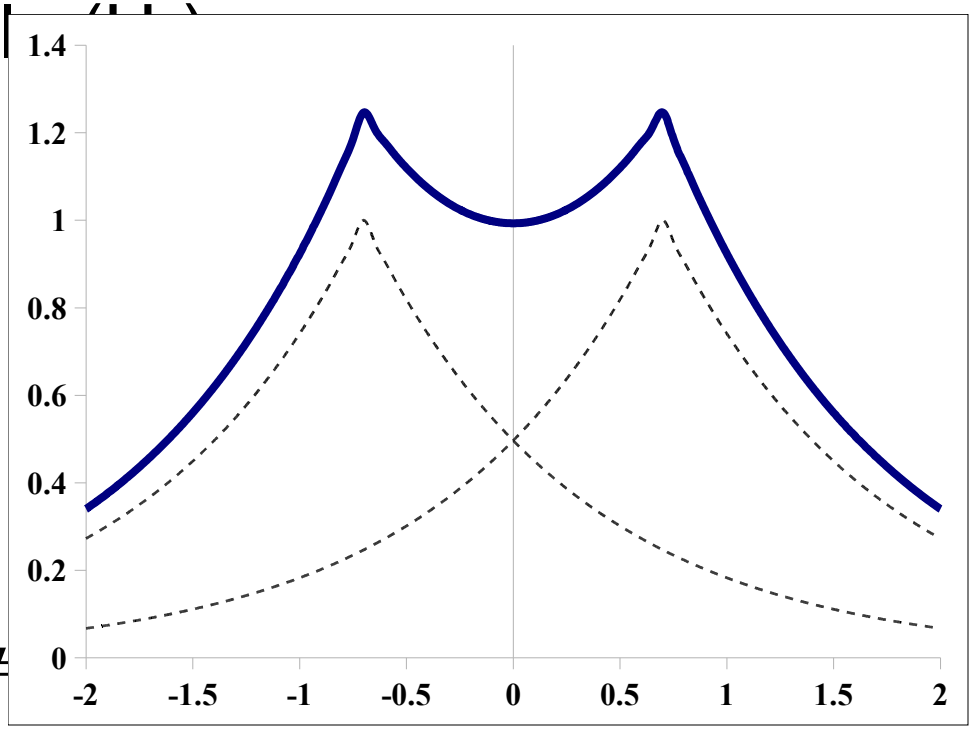
The energy of each orbital in atomic units

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)

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

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



- 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

Essential techniques for fast computation

- Multiresolution

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

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

- Low-separation rank

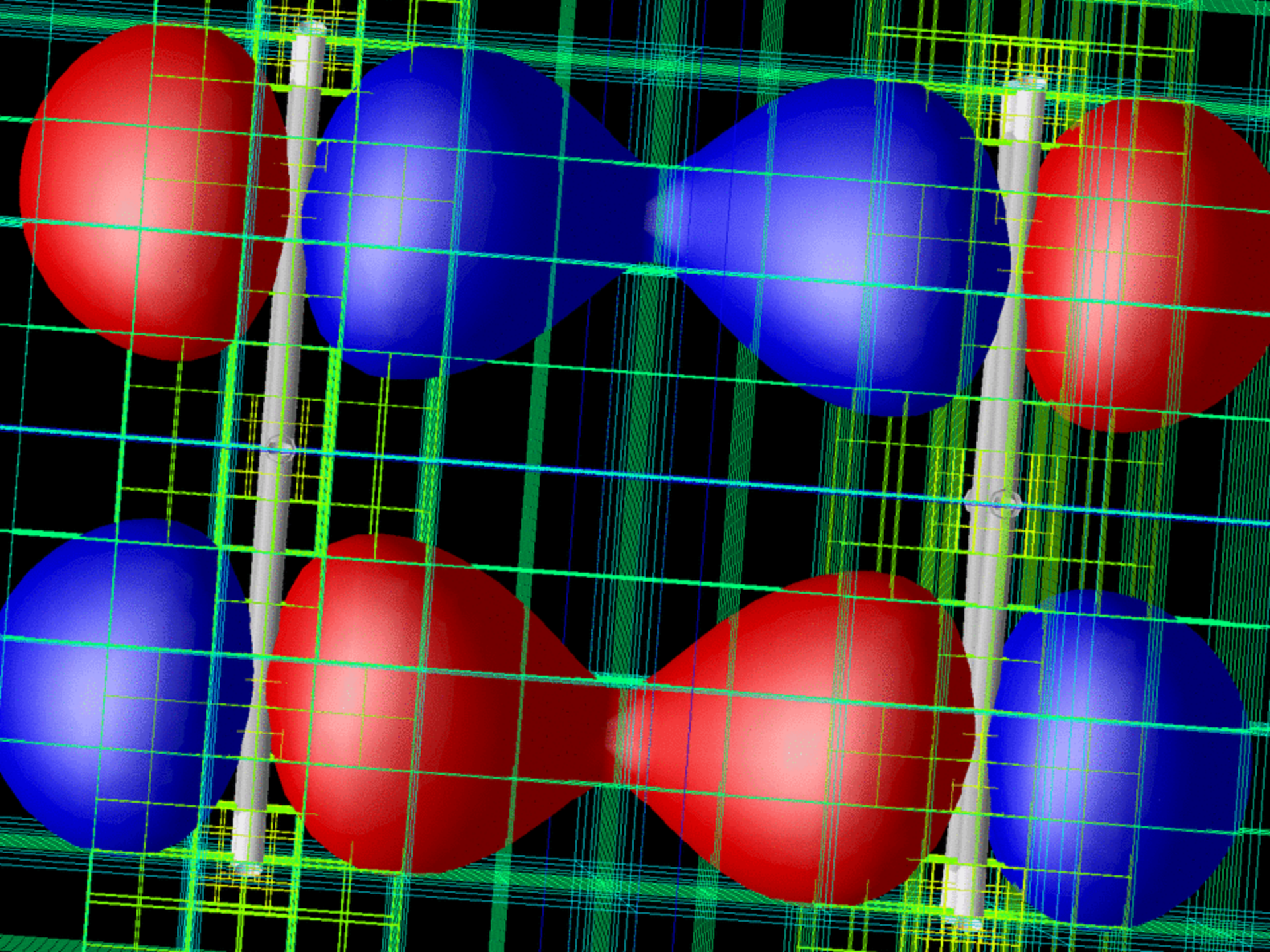
$$f(x_1, \ldots, x_n) = \sum_{l=1}^{M} \sigma_l \prod_{i=1}^{d} f_i^{(l)}(x_i) + O(\epsilon)$$

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

- Low-operator rank

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

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



Please forget about wavelets

- They are not central
- Wavelets are a convenient basis for spanning V_n-V_{n-1} and 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.

Integral Formulation

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

$$\left(-\frac{١}{٢}\nabla^{٢}+V\right)\Psi = E\Psi$$

$$\Psi = -٢\left(-\nabla^{٢}-٢E\right)^{-١}V\Psi$$

$$= -٢G*\left(V\Psi\right)$$

$$\left(G*f\right)(r) = \int ds\,\frac{e^{-k|r-s|}}{٤\pi|r-s|}f(s) \;\text{ in ٣D ; } k^{٢} = -٢E$$

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

Separated form for integral operators

$$T * f = \int ds\, K(r-s) f(s)$$

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

$$r_{ii',jj',kk'}^{n,l-l'} = \sum_{\mu=1}^{M} X_{ii'}^{n,l_x-l'_x} Y_{jj'}^{n,l_y-l'_y} Z_{kk'}^{n,l_z-l'_z} + O(\epsilon)$$

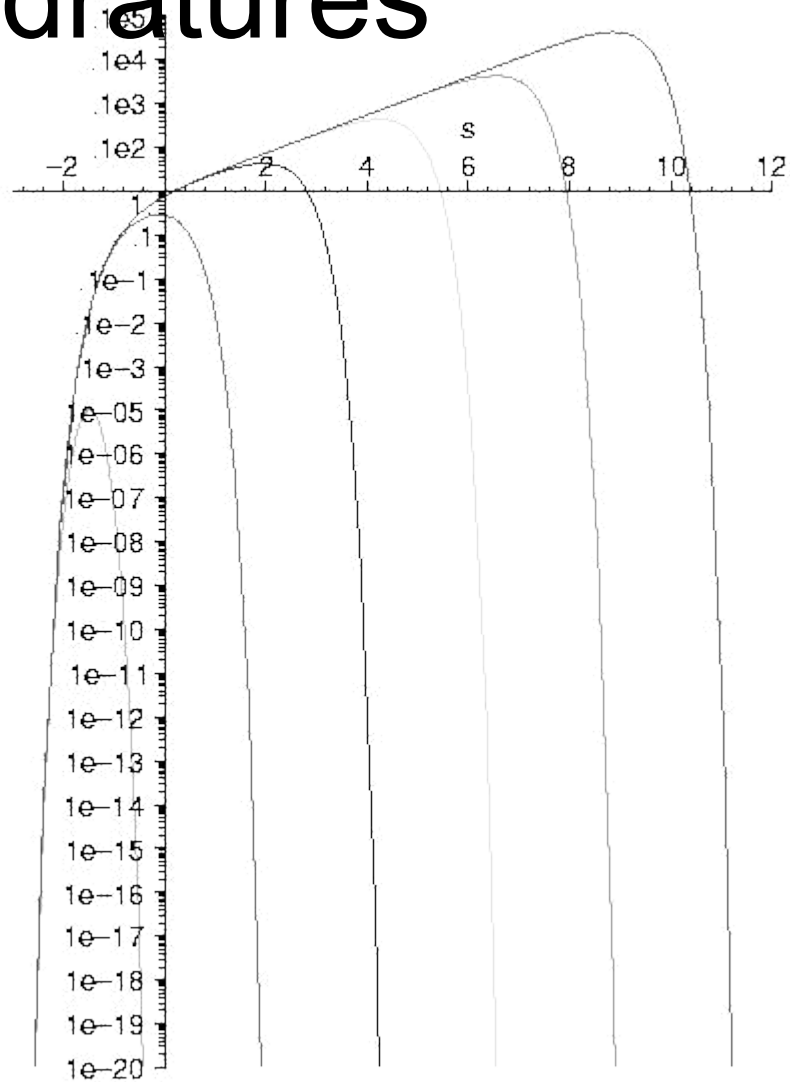
  - Only need compute 1D transition matrices (X,Y,Z)
  - SVD the 1-D operators (low rank away from singularity)
  - Apply most efficient choice of low/full rank 1-D operator
  - Even better algorithms not yet implemented

Accurate Quadratures

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

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

- Trapezoidal quadrature
  - Geometric precision for periodic functions with sufficient smoothness
- Beylkin & Monzon
  - Further reductions, but not automatic



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

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

Applications under active development

- DFT & HF for electrons
  - Energies, gradients, spectra, non-linear optical properties, Raman intensities (Harrison, Sekino, Yanai)
  - Molecules & periodic systems (Eguilez and Thornton)
- Atomic and molecular physics
  - Exact dynamics of few electron systems in strong fields (Krstic and Vence), MCSCF for larger systems
- Nuclear structure
  - G. Fann, et al.
- Preliminary studies in fusion and climate

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

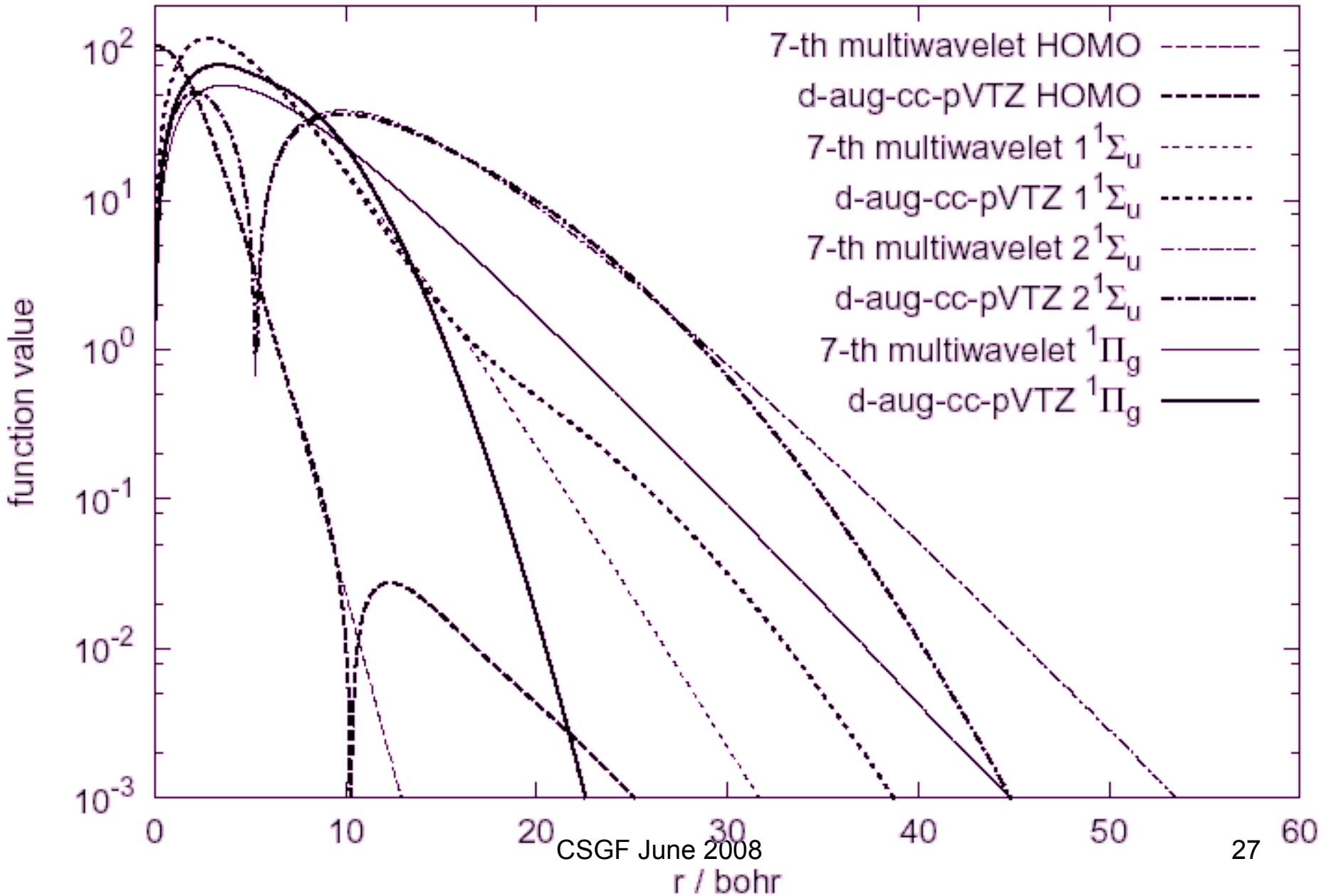
- Solve directly for the orbital response

$$\left(1-\hat{\rho}^0\right)\left[\left(\hat{F}^0-\epsilon_p^0\right)x_p\left(r\right)+\left\{\frac{\partial\hat{g}}{\partial\rho}\left[\rho^0\right]*\left(\sum_i^{occ}x_i\left(r\right)\phi_i^\dagger\left(r'\right)+\sum_i^{occ}\phi_i\left(r\right)y_i^\dagger\left(r'\right)\right)\right\}\phi_p\left(r\right)\right]$$
$$=\omega\, x_p\left(r\right),$$

$$\left(1-\hat{\rho}^0\right)^\dagger\left[\left(\hat{F}^0-\epsilon_p^0\right)^\dagger y_p\left(r\right)+\left\{\frac{\partial\hat{g}}{\partial\rho}\left[\rho^0\right]*\left(\sum_i^{occ}x_i\left(r\right)\phi_i^\dagger\left(r'\right)+\sum_i^{occ}\phi_i\left(r\right)y_i^\dagger\left(r'\right)\right)\right\}^\dagger\phi_p\left(r\right)\right]$$
$$=-\omega\, y_p\left(r\right),$$

  – Neglect *y* for CIS or Tamm-Dancoff

H_2 HOMO and CIS excited states



Time evolution

- Multiwavelet basis not optimal
  - Not strongly band limited
  - Explicit methods very 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
  - Time-ordered exponentials
  - Chin-Chen gradient correction (JCP 114, 7338, 2001)

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

Exponential propagator

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

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

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

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

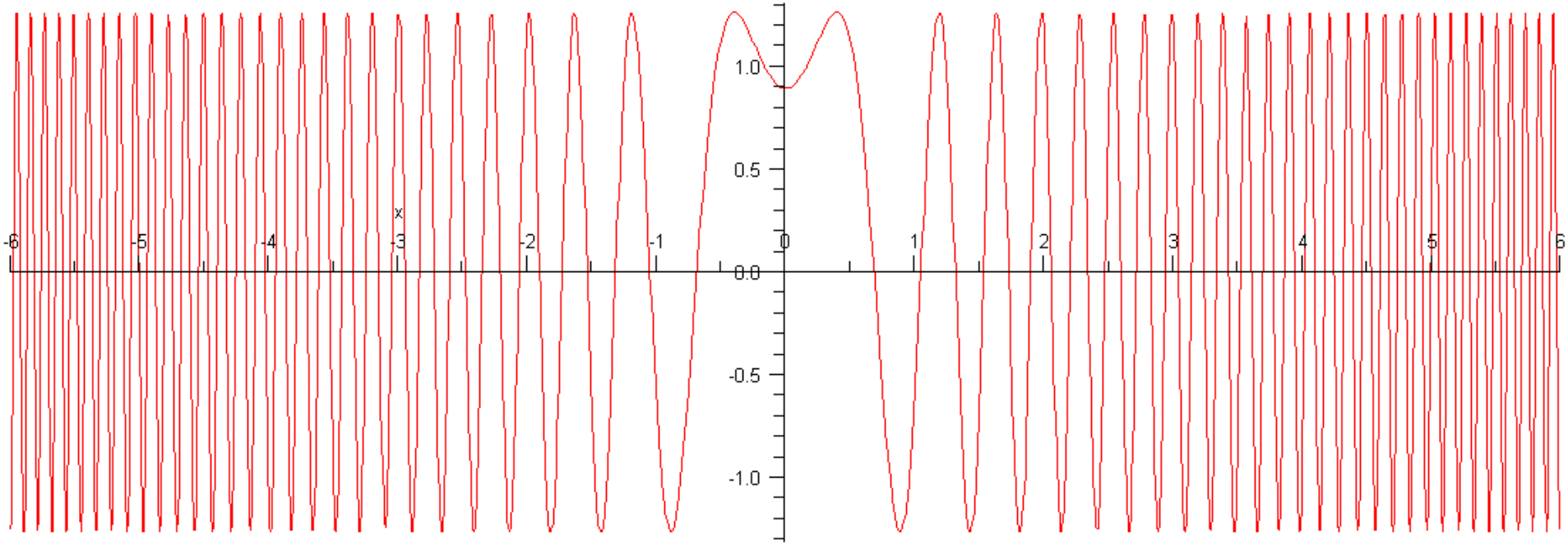
$$\lim_{t\to\infty} \psi(x,t) = \psi_0(x)$$

  - Wrap in solver to accelerate convergence

Exponential propagator

- Free-particle propagator in real time

$$\psi(x,t)=e^{i\nabla^{۲}t/۲}\psi(x,۰)=\frac{۱}{\sqrt{۲\pi i t}}\int_{-\infty}^{\infty}e^{-\frac{(x-y)^2}{2it}}\psi(y,0)dy$$

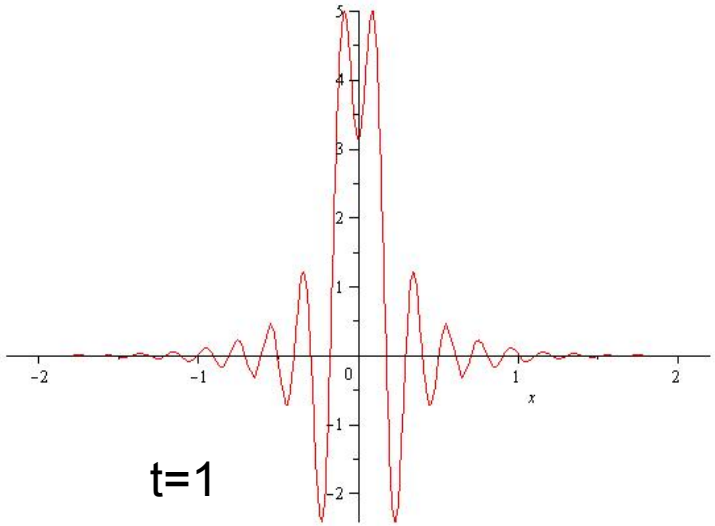


Exponential propagator

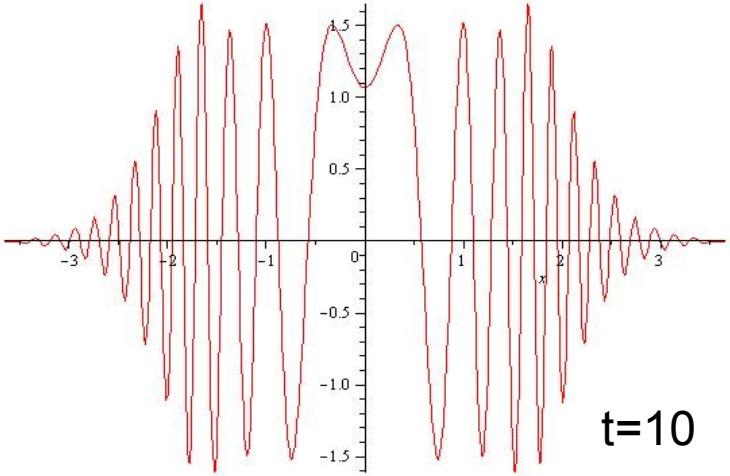
- Combine with projector onto band limit

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

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



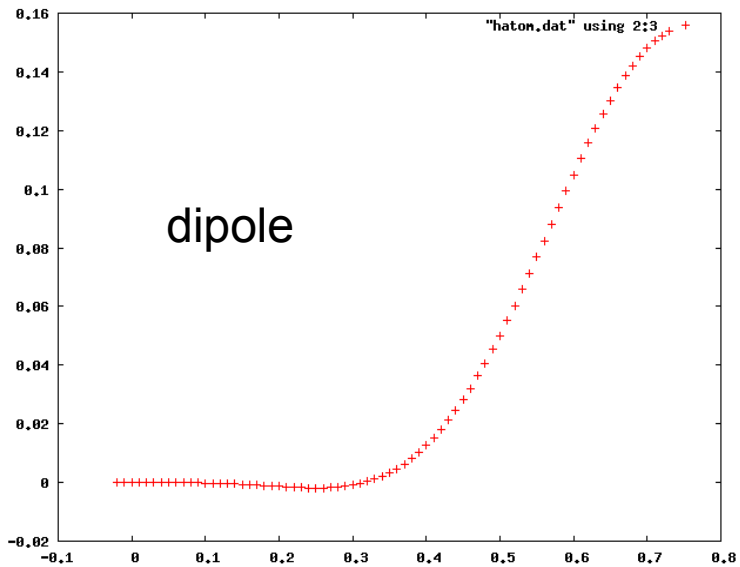
t=1

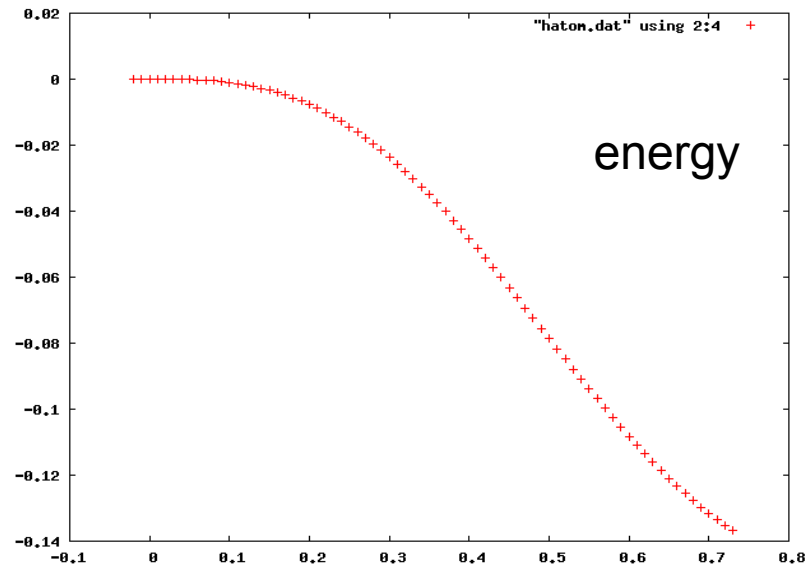


t=10

H-atom in laser field

- One electron – A still interesting test case
  - E.g., high-harmonic generation
  - With P. Krstic and N.E. Vence
- Preparing for T2O runs
  - Lie propagator much faster and stable

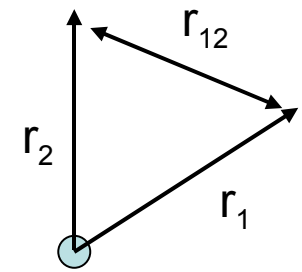




Path to linear scaling HF & DFT

- 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 exchange potential
- Orbital update
- Orthogonalization and or diagonalization
- Linear response properties

Electron correlation



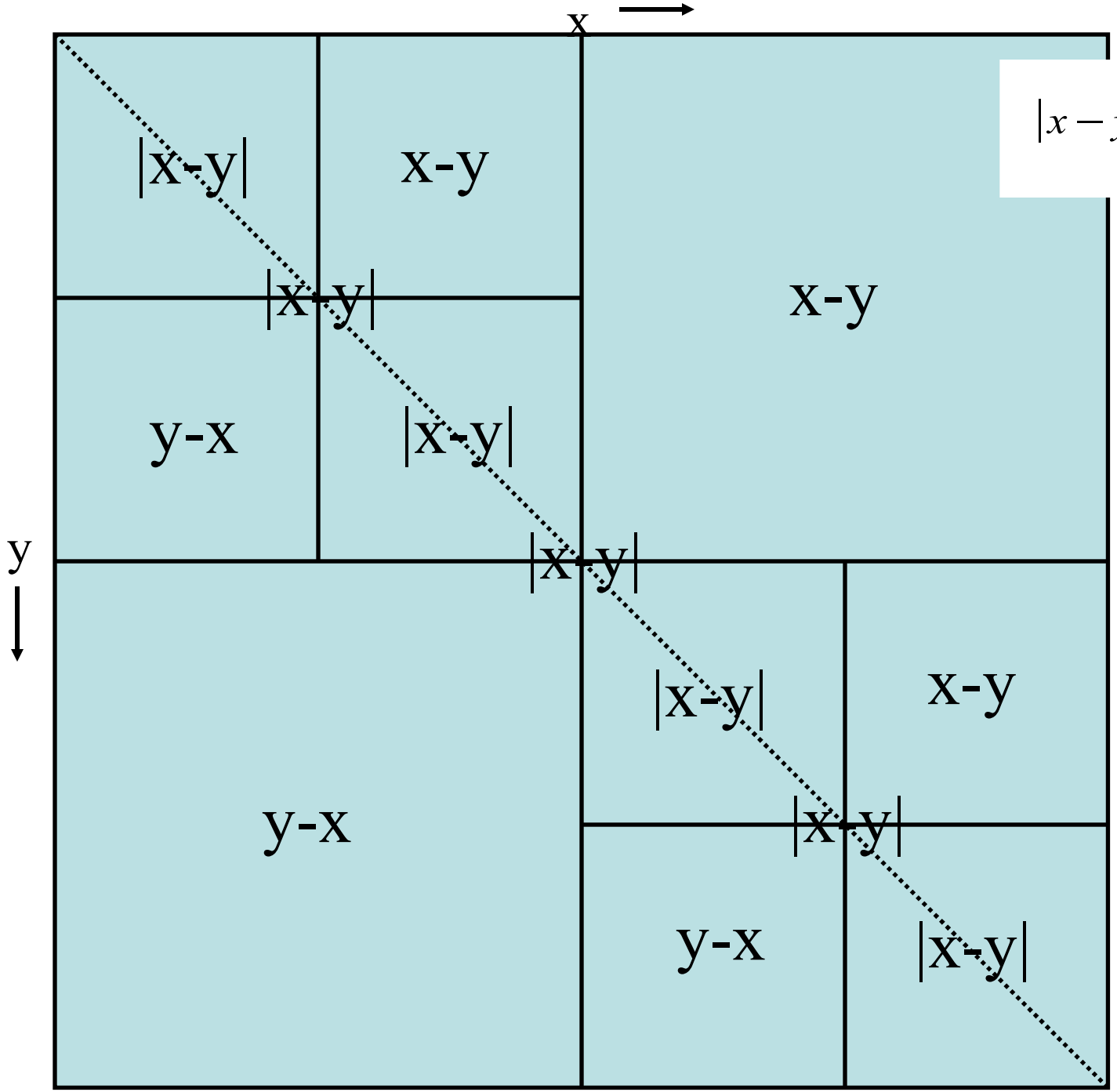
- All defects in the mean-field model are ascribed to electron correlation
- Consideration of singularities in the Hamiltonian imply that for a two-electron singlet atom (e.g., He)

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

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

$$\Psi(r_1, r_2, r_{12}) = e^{-\zeta(r_1 + r_2)}(1 + a r_{12} + \ldots)$$

  - Potentially very accurate, but not systematically improvable, and (until recently) not computationally feasible for many-electron systems



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

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 $r = O(-\log \varepsilon)$

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;
```

High-level composition

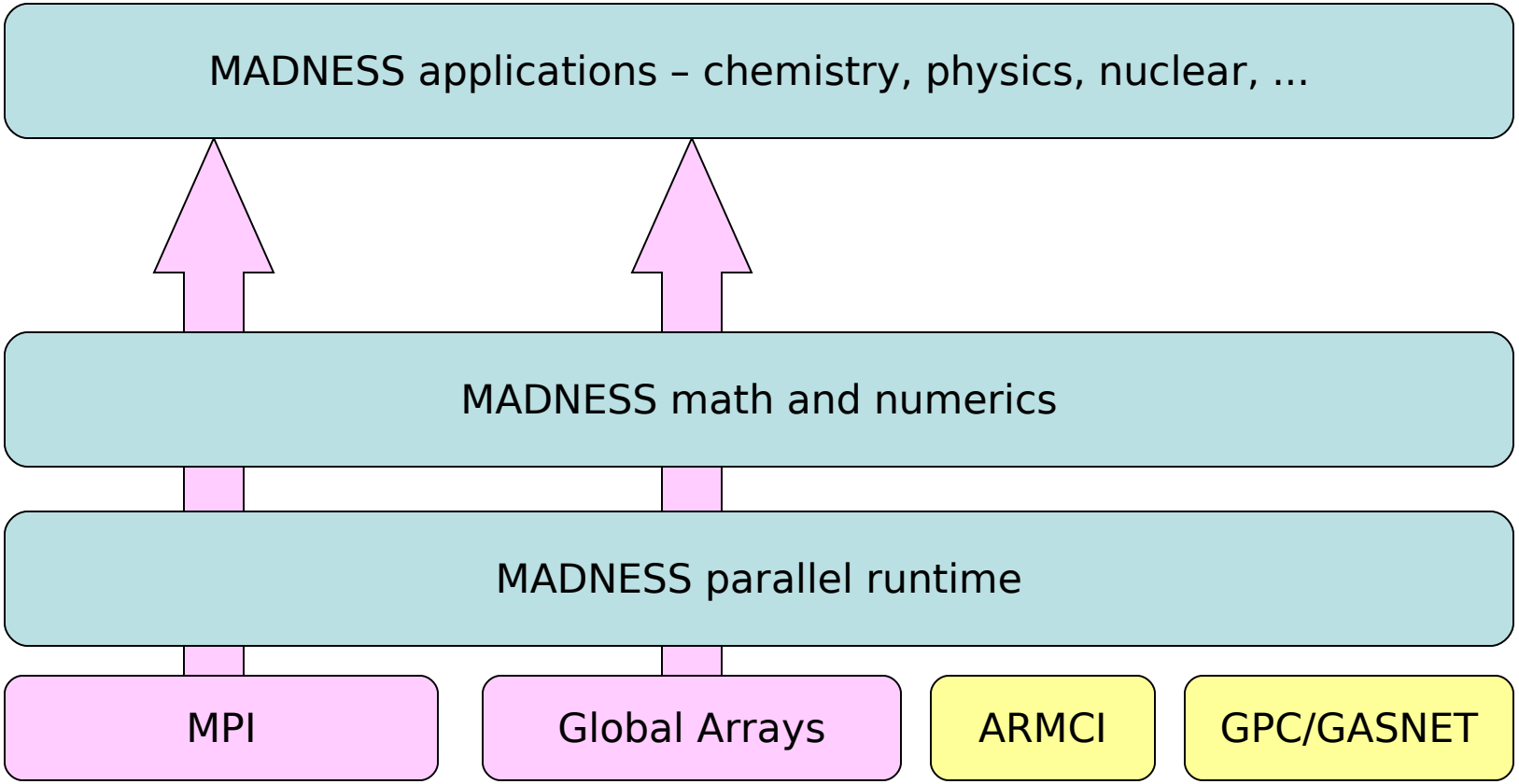
- Express ***ALL*** available parallelism without burdening programmer
  - Internally, MADNESS is looking after data and placement and scheduling of operations on individual functions
  - Programmer must express parallelism over multiple functions and operators
    - But is *not* responsible for scheduling or placement

High-level composition

- E.g., make the matrix of KE operator
  - All scalar operations include optional fence
    - E.g., functionT scale(const functionT& f, T scale, bool fence=true)
  - Internally, operations on vectors schedule all tasks with only one fence

```
Tensor<double>
kinetic_energy_matrix(World& world,
                      const vector<functionT>& v) {
  int n = v.size();
  Tensor<double> r(n,n);
  for (int axis=0; axis<3; axis++) {
     vector<functionT> dv = diff(world,v,axis);
     r += inner(world, dv, dv);
  }
  return r.scale(0.5);
}
```

MADNESS architecture



Runtime Objectives

- Scalability to 1+M processors ASAP
- Runtime responsible for
  - scheduling and placement,
  - managing data dependencies,
  - hiding latency, and
  - Medium to coarse grain concurrency
- Compatible with existing models
  - MPI, Global Arrays
- Borrow successful concepts from Cilk, Charm++, Python
- Anticipating next gen. languages

Key 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
- Dynamic load balancing
  - Data redistribution, work stealing, randomization

Futures

- Result of an asynchronous computation
  - Cilk, Java, HPCLs
- 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.

Global Namespaces

- Specialize global names to containers
  - Hash table done
  - Arrays, etc., planned
- 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)

Summary

- Huge computational resources are rushing towards us
  - Tremendous scientific potential
  - Tremendous challenges
    - Research
    - Education
    - Community
- UT and ORNL are at the very center
  - Think of us when you want something fun and challenging to do

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 $O(N^2 M)$
- Inexpensive in MRA basis

$$\langle f | g \rangle = s_f^{٠٠} . s_g^{٠٠} + \sum_{n=٠} \sum_{l=٠}^{٢^n - ١} d_f^{nl} . d_g^{nl}$$

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