What's in the semantic gap?

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

harrisonrj@ornl.gov

National Science Foundation
WHERE DISCOVERIES BEGIN

SciDAC
Scientific Discovery through Advanced Computing

UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

Possible (probable?) HPC futures

High-end simulation is the most credible vehicle for accelerating the application of knowledge from basic science to the design of new energy technologies

Complexity constrains all of our ambitions (cost & feasibility).

We hear about the successes, but what about the failures?

What about the disciplines that are not computing?

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

- Growing intrinsic complexity of problem
- Complexity kills … sequential or parallel
  - Expressing concurrency at extreme scale
  - Managing the memory hierarchy
- Semantic gap (Colella)
  - Our equations are O(100) lines but
  - The program is O(100K) & growing
  - Why?

Wish list

- Eliminate gulf between theoretical innovation in small groups and realization on high-end computers
- Eliminate the semantic gap so that efficient parallel code is no harder than doing the math
- Enable performance-portable “code” that can be automatically migrated to future architectures
- Reduce cost at all points in the life cycle

- Much of this is pipe dream – but what can we aspire to?

Guy Steele’s example

- Natural language problem description

```
I will give you some text – please remove
white space and give me the list of words
```

- The wetware of an undistracted first grader is capable of running this two-line “program” with no more guidance

Missing ingredients

- Natural language problem description
  - Implicit & fuzzy definitions of verbs, nouns, ...
  - Incomplete without more context
    - E.g., what if an American first grader was given

وقال خليل إبراهيم في برنامج ضيف المنتصف على قناة الجزيرة اليوم إن "ما يحدث في الدوحة هو تزوير لأن الأطراف الرئيسية غير موجودة والحرب دائرة على الأرض وتمتد للمدن".

- No ordering specified – parallelism
- Does not include an <u>algorithm</u>

Khalil Ibrahim in the middle Guest on Al-Jazeera today, "what is happening in Doha is a fraud because the main parties do not exist and the war department on the ground and extends to cities"

Conventional solution

- Problem statement + brain → algorithm
- Algorithm + language + brain → program
- Computer + program + input → result
- The brain is
  - Expensive
  - Finite
  - Not growing exponentially

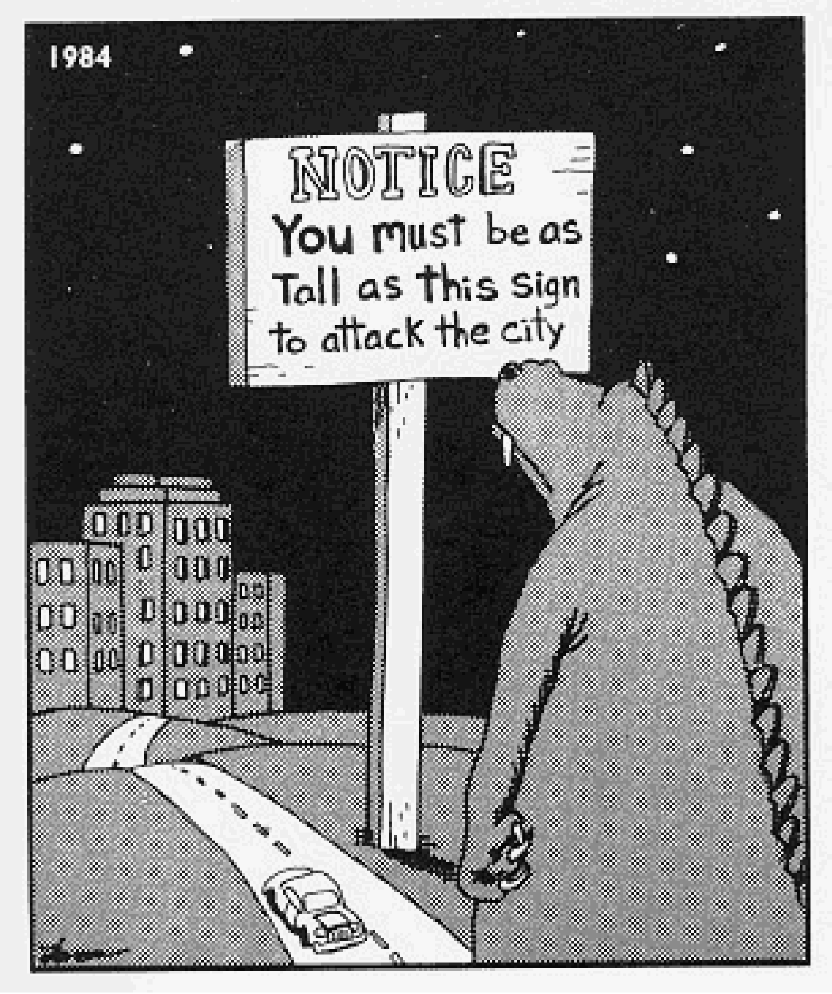
This is clearly Guy's brain
- Flashes of inspiration
- Blue aura of authority

Image from http://www.ucdmc.ucdavis.edu/welcome/features/20071017_Medicine_whitematter/Photos/head_and_brain.jpg

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, ...

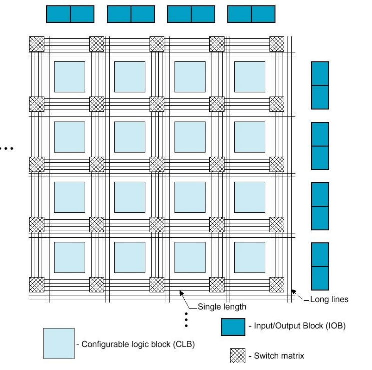


Scientific vs. WWW software

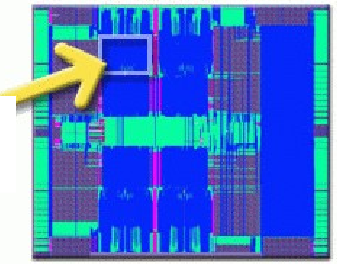
- Why are we not experiencing the same nearly exponential growth in functionality?
  - Level of investment or number of developers?
  - Lack of software interoperability and standards?
  - Competition not cooperation between groups?
  - Shifting scientific objectives?
  - Our problems are intrinsically harder?
  - Failure to embrace/develop higher levels for composing applications?
  - Differing impact of hardware complexity?

What is a processor in 2015?

- Probably not an x86
- E.g., Nvidia C2060 480 cores
  - ~1TF peak (single), ~0.5TF (double)
- LANL RoadRunner – hybrid
  - 1 PF of IBM Cell + Opteron
- Cyclops
  - 160 simple thread units per chip
  - no cache - S/W managed on-chip memory
- <u>FLOPs are cheap; bandwidth is expensive</u>



1 core



There is no escape

- Trickle-down computing
  - 2010's national supercomputer is
  - 2013's campus resource
  - 2016's group cluster
  - 2019's personal server
- Why petascale computing now?
  - Because you have large problems now
  - Because you want to prepare for the future

Back to the Gap – what did Guy’s brain do?

- Complete the <u>specification</u> with mathematical rigor
  - Including aspects of <u>representation</u>
- Provide an <u>algorithm</u>
- Whence came the algorithm?
  - Derived from the specification?
  - Instantly and unconsciously pattern matched against decades of prior experience?
  - Guy ... what’s your answer?
- Express the algorithm as Fortress code

Fortunately for scientific HPC

- Mathematical rigor is the norm
  - Which partially explains why some disciplines are late to the table
- Unfortunately, also the norm are neglect or ignorance of
  - difference between algebraic and floating-point numbers,
  - accuracy, stability, and other properties of common algorithms
  - parallel algorithms and programming

Frameworks

- NWChem
- MADNESS
- ChemES – chemistry end-station
- Frameworks
  - Increase productivity; hide complexity
  - Interface disciplines
  - Capture knowledge
  - Open HPC to wider community
  - Expensive, communal projects

Molecular Science Software Project

MS3
MOLECULAR SCIENCE SOFTWARE SUITE

ECCE
EXTENSIBLE COMPUTATIONAL CHEMISTRY ENVIRONMENT

NWCHEM
HIGH-PERFORMANCE COMPUTATIONAL CHEMISTRY SOFTWARE

GA TOOLS
PARALLEL COMPUTING LIBRARIES AND SOFTWARE TOOLS

PNNL
Yuri Alexeev,
Eric Bylaska,
Bert deJong,
Mahin Hackler,
Karol Kowalski,
Lisa Pollack,
Tjerk Straatsma,
Marat Valiev,
Theresa Windus

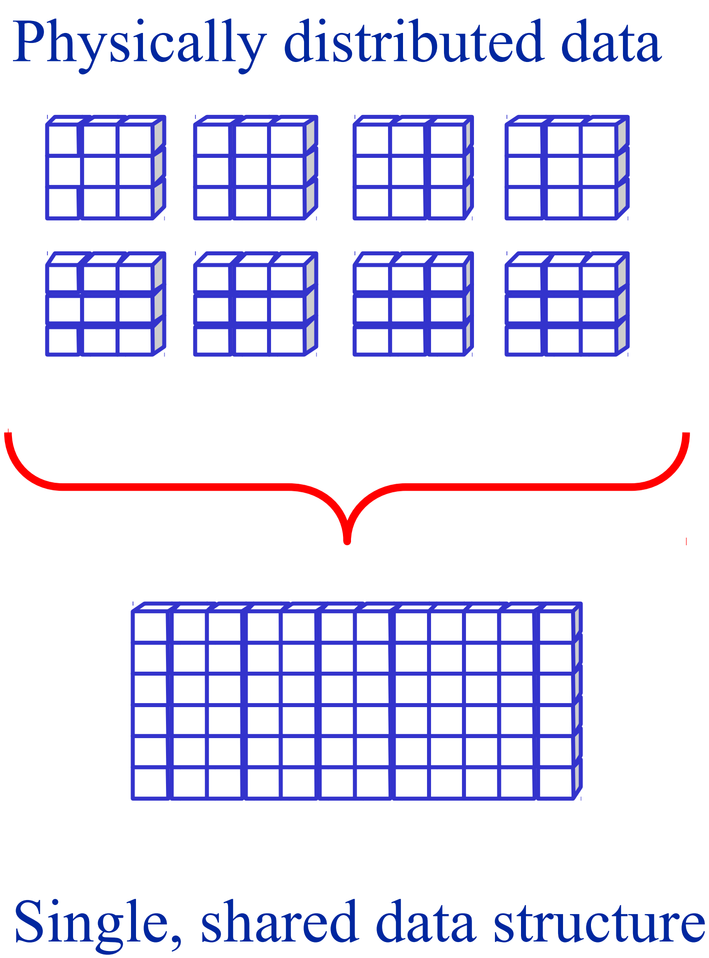
ORNL
Edo Apra,
Vincent Meunier
Robert Harrison

Gary Black,
Brett Didier,
Todd Elsenthagen,
Sue Havre,
Carina Lansing,
Bruce Palmer,
Karen Schuchardt,
Lisong Sun
Erich Vorpagel

Manoj Krishnan, Jarek Nieplocha,
Bruce Palmer, Vinod Tipparaju

http://www.emsl.pnl.gov/docs/nwchem/nwchem.html

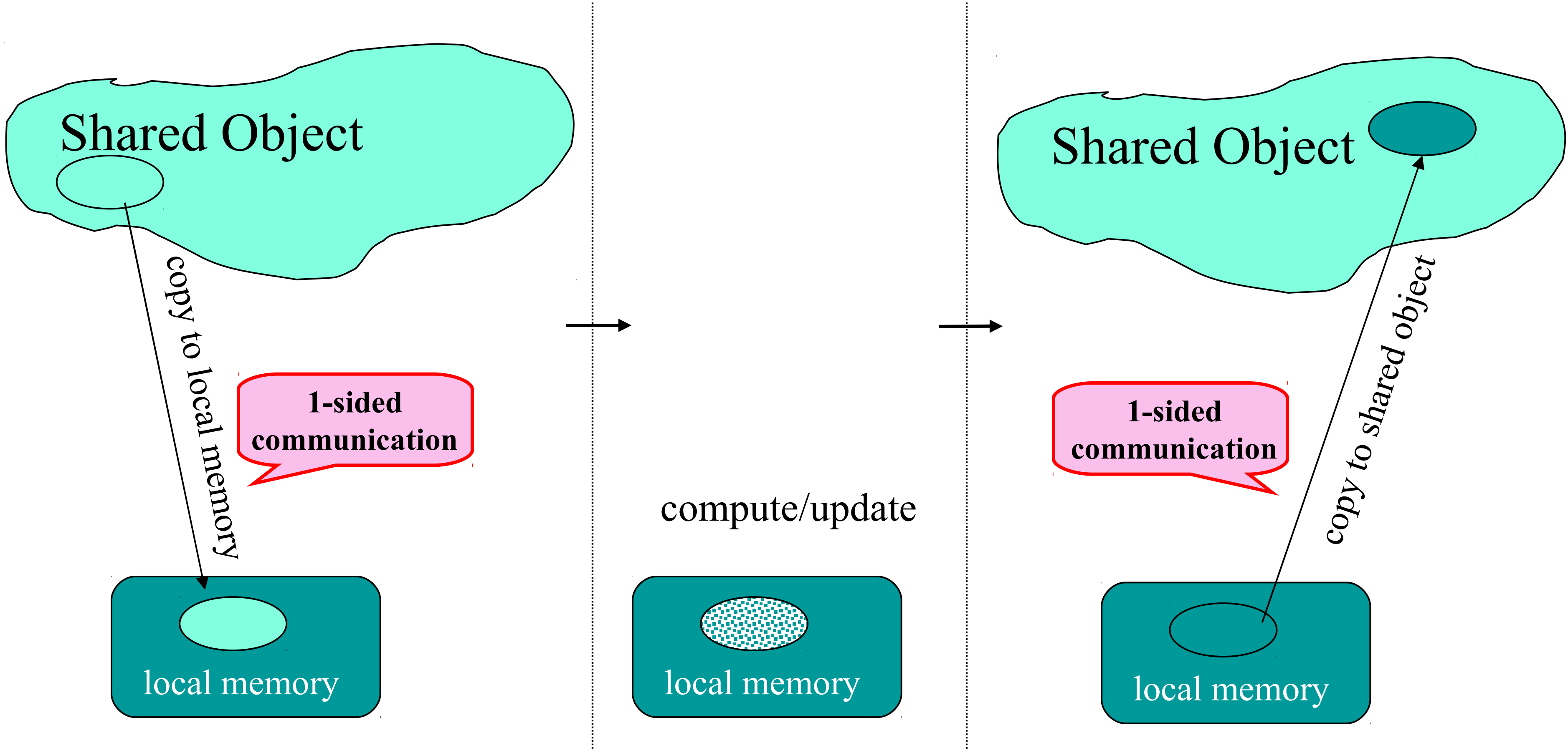
Global Arrays (technologies)



- Shared-memory-like model
  - Fast local access
  - NUMA aware and easy to use
  - MIMD and data-parallel modes
  - Inter-operates with MPI, …
- BLAS and linear algebra interface
- Ported to major parallel machines
  - IBM, Cray, SGI, clusters,...
- Originated in an HPCC project
- Used by most major chemistry codes, financial futures forecasting, astrophysics, computer graphics
- Supported by DOE
- A legacy of Jarek Nieplocha, PNNL

http://www.emsl.pnl.gov/docs/global/

Non-uniform memory access model of computation



Dynamic load balancing

While ((task = SharedCounter())< max)

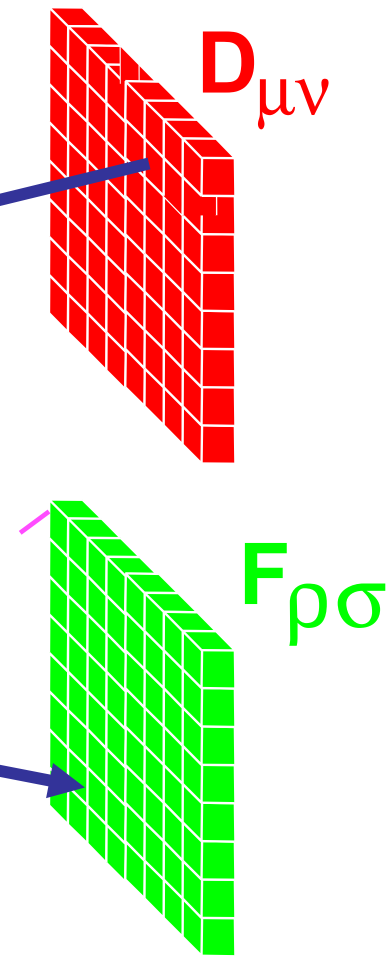
call ga_get( )

(do work)

call ga_acc( )

End while

Barrier()



More scalable structures easily composed but often not necessary.

Dead code

7 December 1969

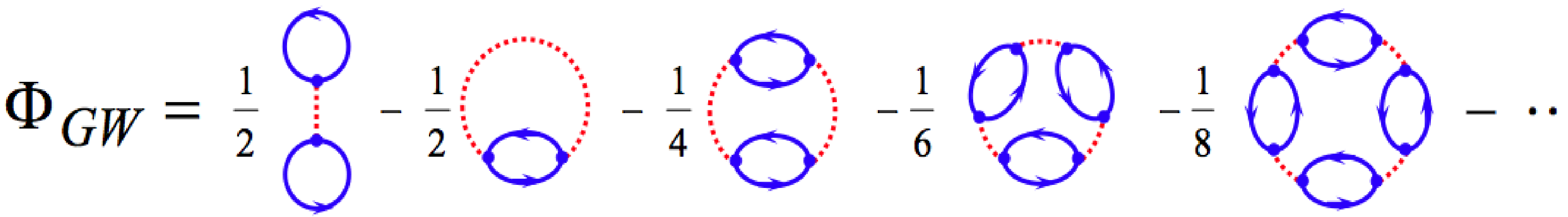
- Requires human labor
  - to migrate to future architectures, or
  - to exploit additional concurrency, or
  - ...
- By these criteria most extant code is dead
- Sanity check
  - How much effort is required to port to hybrid cpu+GPGPU?

Next generation ORNL NLCF

- ORNL has proposed a system to meet DOE's requirement for 20-40 PF of compute capability split between the Oak Ridge and Argonne LCF centers
- ORNL's proposed system will be based on accelerator technology
  - includes software development environment
- We plan to deploy the system in late 2011 with users getting access in 2012
- Watch this space for more details soon

http://www.nccs.gov/

The language of many-body physics



Hartree   Fock   Infinite chain of ***dressed*** electron-hole bubbles

CCSD Doubles Equation

```
hbar[a,b,i,j] == sum[f[b,c]*t[i,j,a,c],{c}] -sum[f[k,c]*t[k,b]*t[i,j,a,c],{k,c}] +sum[f[a,c]*t[i,j,c,b],{c}] -sum[f[k,c]*t[k,a]*t[i,j,c,b],{k,c}]
    -sum[f[k,j]*t[i,k,a,b],{k}] -sum[f[k,c]*t[j,c]*t[i,k,a,b],{k,c}] -sum[f[k,i]*t[j,k,b,a],{k}] -sum[f[k,c]*t[i,c]*t[j,k,b,a],{k,c}]
    +sum[t[i,c]*t[j,d]*v[a,b,c,d],{c,d}] +sum[t[i,j,c,d]*v[a,b,c,d],{c,d}] +sum[t[j,c]*v[a,b,i,c],{c}] -sum[t[k,b]*v[a,k,i,j],{k}]
    +sum[t[i,c]*v[b,a,j,c],{c}] -sum[t[k,a]*v[b,k,j,i],{k}] -sum[t[k,d]*t[i,j,c,b]*v[k,a,c,d],{k,c,d}] -sum[t[i,c]*t[j,k,b,d]*v[k,a,c,d],
    {k,c,d}] -sum[t[j,c]*t[k,b]*v[k,a,c,i],{k,c}] +2*sum[t[j,k,b,c]*v[k,a,c,i],{k,c}] -sum[t[j,k,c,b]*v[k,a,c,i],{k,c}]
    -sum[t[i,c]*t[j,d]*t[k,b]*v[k,a,d,c],{k,c,d}] +2*sum[t[k,d]*t[i,j,c,b]*v[k,a,d,c],{k,c,d}] -sum[t[k,b]*t[i,j,c,d]*v[k,a,d,c],{k,c,d}]
    -sum[t[j,d]*t[i,k,c,b]*v[k,a,d,c],{k,c,d}] +2*sum[t[i,c]*t[j,k,b,d]*v[k,a,d,c],{k,c,d}] -sum[t[i,c]*t[j,k,d,b]*v[k,a,d,c],{k,c,d}]
    -sum[t[j,k,b,c]*v[k,a,i,c],{k,c}] -sum[t[i,c]*t[k,b]*v[k,a,j,c],{k,c}] -sum[t[i,k,c,b]*v[k,a,j,c],{k,c}]
    -sum[t[i,c]*t[j,d]*t[k,a]*v[k,b,c,d],{k,c,d}] -sum[t[k,d]*t[i,j,a,c]*v[k,b,c,d],{k,c,d}] -sum[t[k,a]*t[i,j,c,d]*v[k,b,c,d],{k,c,d}]
    +2*sum[t[j,d]*t[i,k,a,c]*v[k,b,c,d],{k,c,d}] -sum[t[j,d]*t[i,k,c,a]*v[k,b,c,d],{k,c,d}] -sum[t[i,c]*t[j,k,d,a]*v[k,b,c,d],{k,c,d}]
    -sum[t[i,c]*t[k,a]*v[k,b,c,j],{k,c}] +2*sum[t[i,k,a,c]*v[k,b,c,j],{k,c}] -sum[t[i,k,c,a]*v[k,b,c,j],{k,c}]
    +2*sum[t[k,d]*t[i,j,a,c]*v[k,b,d,c],{k,c,d}] -sum[t[j,d]*t[i,k,a,c]*v[k,b,d,c],{k,c,d}] -sum[t[j,c]*t[k,a]*v[k,b,i,c],{k,c}]
    -sum[t[j,k,c,a]*v[k,b,i,c],{k,c}] -sum[t[i,k,a,c]*v[k,b,j,c],{k,c}] +sum[t[i,c]*t[j,d]*t[k,a]*t[l,b]*v[k,l,c,d],{k,l,c,d}]
    -2*sum[t[k,b]*t[l,d]*t[i,j,a,c]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[k,a]*t[l,d]*t[i,j,c,b]*v[k,l,c,d],{k,l,c,d}]
    +sum[t[k,a]*t[l,b]*t[i,j,c,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[j,c]*t[l,d]*t[i,k,a,b]*v[k,l,c,d],{k,l,c,d}]
    -2*sum[t[j,d]*t[l,b]*t[i,k,a,c]*v[k,l,c,d],{k,l,c,d}] +sum[t[j,d]*t[l,b]*t[i,k,c,a]*v[k,l,c,d],{k,l,c,d}]
    -2*sum[t[i,c]*t[l,d]*t[j,k,b,a]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,c]*t[l,a]*t[j,k,b,d]*v[k,l,c,d],{k,l,c,d}]
    +sum[t[i,c]*t[l,b]*t[j,k,d,a]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,k,c,d]*t[j,l,b,a]*v[k,l,c,d],{k,l,c,d}]
    +4*sum[t[i,k,a,c]*t[j,l,b,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,k,c,a]*t[j,l,b,d]*v[k,l,c,d],{k,l,c,d}]
    -2*sum[t[i,k,a,b]*t[j,l,c,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,k,a,c]*t[j,l,d,b]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,k,c,a]*t[j,l,d,b]*v[k,l,c,d],
    {k,l,c,d}] +sum[t[i,c]*t[j,d]*t[k,l,a,b]*v[k,l,c,d],{k,l,c,d}] +sum[t[i,j,c,d]*t[k,l,a,b]*v[k,l,c,d],{k,l,c,d}]
    -2*sum[t[i,j,c,b]*t[k,l,a,d]*v[k,l,c,d],{k,l,c,d}] -2*sum[t[i,j,a,c]*t[k,l,b,d]*v[k,l,c,d],{k,l,c,d}] +sum[t[j,c]*t[k,b]*t[l,a]*v[k,l,c,i],
    {k,l,c}] +sum[t[l,c]*t[j,k,b,a]*v[k,l,c,i],{k,l,c}] -2*sum[t[l,a]*t[j,k,b,c]*v[k,l,c,i],{k,l,c}] +sum[t[l,a]*t[j,k,c,b]*v[k,l,c,i],{k,l,c}]
    -2*sum[t[k,c]*t[j,l,b,a]*v[k,l,c,i],{k,l,c}] +sum[t[k,a]*t[j,l,b,c]*v[k,l,c,i],{k,l,c}] +sum[t[k,b]*t[j,l,c,a]*v[k,l,c,i],{k,l,c}]
    +sum[t[j,c]*t[l,k,a,b]*v[k,l,c,i],{k,l,c}] +sum[t[i,c]*t[k,a]*t[l,b]*v[k,l,c,j],{k,l,c}] +sum[t[l,c]*t[i,k,a,b]*v[k,l,c,j],{k,l,c}]
    -2*sum[t[l,b]*t[i,k,a,c]*v[k,l,c,j],{k,l,c}] +sum[t[l,b]*t[i,k,c,a]*v[k,l,c,j],{k,l,c}] +sum[t[i,c]*t[k,l,a,b]*v[k,l,c,j],{k,l,c}]
    +sum[t[j,c]*t[l,d]*t[i,k,a,b]*v[k,l,d,c],{k,l,c,d}] +sum[t[j,d]*t[l,b]*t[i,k,a,c]*v[k,l,d,c],{k,l,c,d}]
    +sum[t[j,d]*t[l,a]*t[i,k,c,b]*v[k,l,d,c],{k,l,c,d}] -2*sum[t[i,k,c,d]*t[j,l,b,a]*v[k,l,d,c],{k,l,c,d}]
    -2*sum[t[i,k,a,c]*t[j,l,b,d]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,c,a]*t[j,l,b,d]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,a,b]*t[j,l,c,d]*v[k,l,d,c],
    {k,l,c,d}] +sum[t[i,k,c,b]*t[j,l,d,a]*v[k,l,d,c],{k,l,c,d}] +sum[t[i,k,a,c]*t[j,l,d,b]*v[k,l,d,c],{k,l,c,d}] +sum[t[k,a]*t[l,b]*v[k,l,i,j],
    {k,l}] +sum[t[k,l,a,b]*v[k,l,i,j],{k,l}] +sum[t[k,b]*t[l,d]*t[i,j,a,c]*v[l,k,c,d],{k,l,c,d}] +sum[t[k,a]*t[l,d]*t[i,j,c,b]*v[l,k,c,d],
    {k,l,c,d}] +sum[t[i,c]*t[l,d]*t[j,k,b,a]*v[l,k,c,d],{k,l,c,d}] -2*sum[t[i,c]*t[l,a]*t[j,k,b,d]*v[l,k,c,d],{k,l,c,d}]
    +sum[t[i,c]*t[l,a]*t[j,k,d,b]*v[l,k,c,d],{k,l,c,d}] +sum[t[i,j,c,b]*t[k,l,a,d]*v[l,k,c,d],{k,l,c,d}] +sum[t[i,j,a,c]*t[k,l,b,d]*v[l,k,c,d],
    {k,l,c,d}] -2*sum[t[l,c]*t[i,k,a,b]*v[l,k,c,j],{k,l,c}] +sum[t[l,b]*t[i,k,a,c]*v[l,k,c,j],{k,l,c}] +sum[t[l,a]*t[i,k,c,b]*v[l,k,c,j],{k,l,c}]
    +v[a,b,i,j]
```

$$\bar{h}^{a\,b}_{i\,j} = \left\langle \begin{matrix} a\,b \\ i\,j \end{matrix} \middle| e^{-\hat{T}_1 - \hat{T}_2}\, \hat{H}\, e^{\hat{T}_1 + \hat{T}_2} \middle| 0 \right\rangle$$

The Tensor Contraction Engine: A Tool for Quantum Chemistry

Oak Ridge National Laboratory
David E. Bernholdt, Venkatesh Choppella, ***Robert Harrison***

Pacific Northwest National Laboratory
So Hirata

Louisiana State University
J Ramanujam

Ohio State University
Gerald Baumgartner, Alina Bibireata, Daniel Cociorva, Xiaoyang Gao, Sriram Krishnamoorthy, Sandhya Krishnan, Chi-Chung Lam, Quingda Lu, ***Russell M. Pitzer, P Sadayappan***, Alexander Sibiryakov

University of Waterloo
Marcel Nooijen, Alexander Auer

http://www.cis.ohio-state.edu/~gb/TCE/

Research at ORNL supported by the Laboratory Directed Research and Development Program. Research at PNNL supported by the Office of Basic Energy Sciences, U. S. Dept. of Energy. Research at OSU, Waterloo, and LSU supported by the National Science Foundation Information Technology Research Program

Pesky details of an incomplete spec.

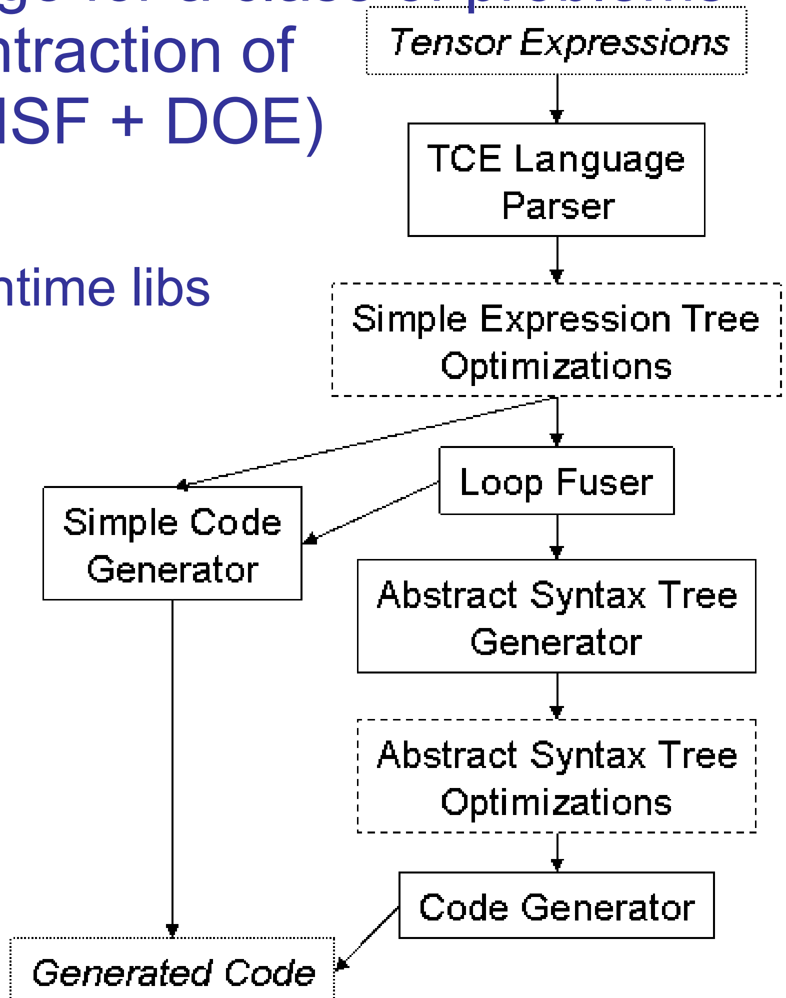
- Some tensors have symmetries w.r.t. index permutations $$\langle pq|rs\rangle = \langle ps|rq\rangle = \cdots$$
- Others have predictable block sparsity $$b_{2u} \times b_{3u} = b_{1g}$$
- Huge impact on memory use and algorithm cost
- ... one year later ...

Tensor Contraction Engine (TCE)

- Sadayappan et al. Proc. IEEE, 93, 2005
- High-level domain-specific language for a class of problems in chemistry/physics based on contraction of large multi-dimensional tensors (NSF + DOE)
- Specialized optimizing compiler
  - Produces F77+GA code, linked to runtime libs

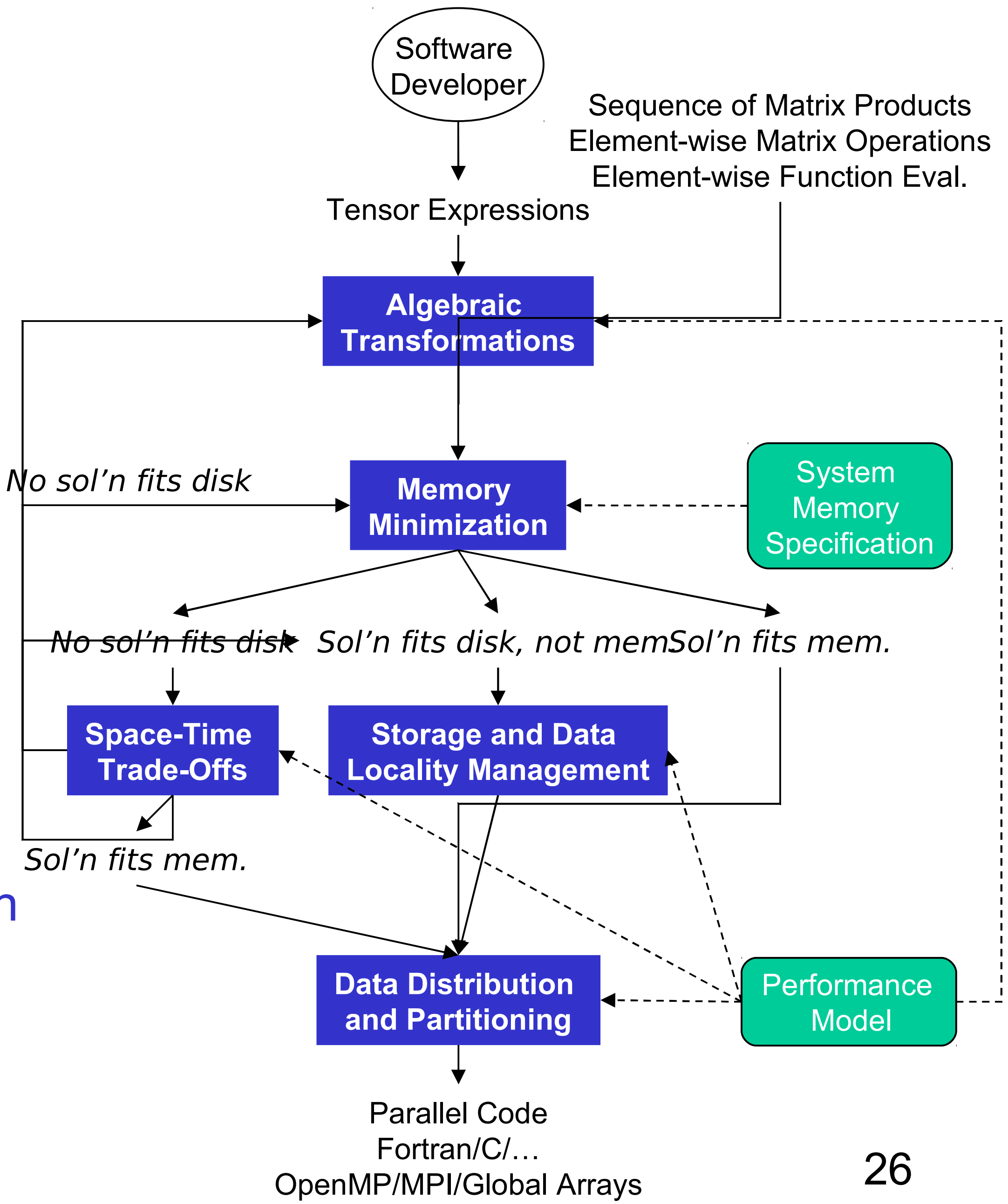
$$S_{abij} = \sum_{cefkl} A_{acik} B_{befl} C_{dfjk} D_{cdel}$$

```
range V = 3000;
range O = 100;

index a,b,c,d,e,f : V;
index i,j,k,l : O;

mlimit = 100GB;

procedure P(in A[V,V,O,O], in B[V,V,V,O],
            in C[V,V,O,O], in D[V,V,V,O],
            out S[V,V,O,O])=
begin
  S[a,b,i,j] == sum[ A[a,c,i,k] * B[b,e,f,l]
                   * C[d,f,j,k] * D[c,d,e,l],
                     {c,e,f,k,l}];
end
```



TCE Components

- Algebraic Transformations
  - Minimize operation count
- Memory Minimization
  - Reduce intermediate storage via loop fusion (LCPC'03)
- Space-Time Transformation
  - Trade-offs between storage and recomputation (PLDI'02)
- Data Locality Optimization
  - Optimize use of storage hierarchy via tiling (ICS'01, HiPC'03, IPDPS'04)
- Data Dist./Comm. Optimization
  - Optimize parallel data layout (IPDPS'03)
- Integrated System
  - (SuperComputing'02, Proc. IEEE 05)



Productivity of TCE

- The tensor contraction expressions for the higher members of the Coupled Cluster family of models can be generated relatively easily, but the effort to manually generate Fortran code is quite significant
- The code development time for other models of comparable complexity can be reduced from years to days/weeks
- More than 25 methods implemented $\rightarrow$ » 5 years to hand code

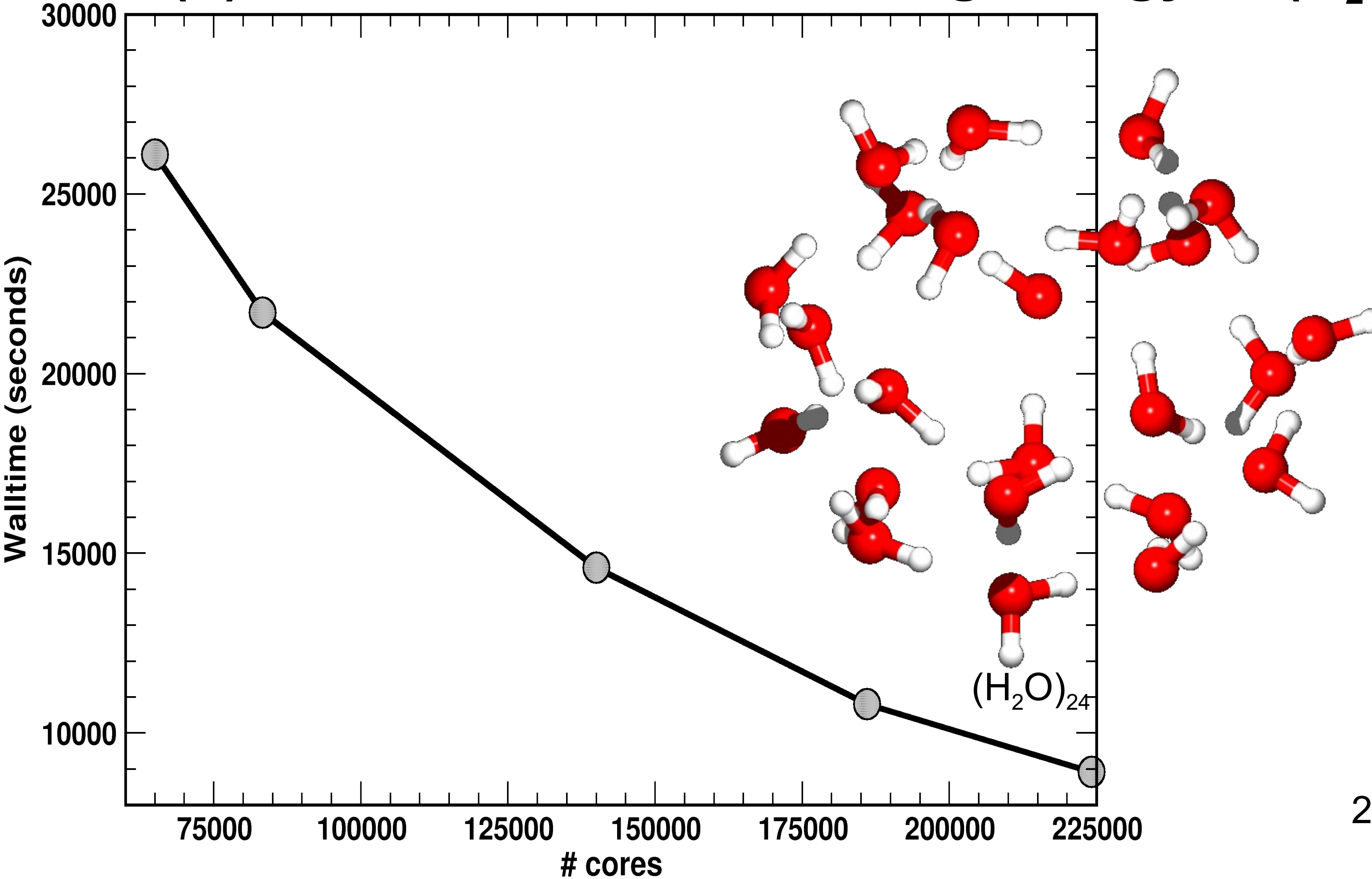
Theory	#Terms	#F77Lines	Year
CCD	11	3209	1978
CCSD	48	13213	1982
CCSDT	102	33932	1988
CCSDTQ	183	79901	1992

Result of first hand implementation of CCSDTQ

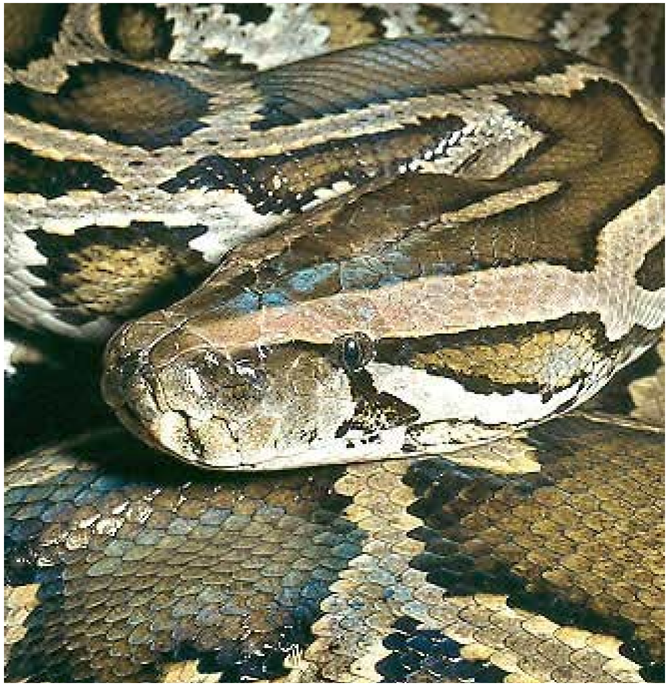
NWChem CCSD(T) – 1.31 PFLOP/s

E. Aprà, R.J. Harrison, W.A. deJong, A.P. Rendell, V. Tipparaju and R.M. Olson

CCSD(T) benchmark of the binding energy of $(H_2O)_{24}$



Python vs. Java



- The initial Python prototype written by chemists works but has lots of “issues” with memory, speed, ...
- The OSU TCE generates better code, respects bounds on memory use, but is written in Java by C/S graduate students
- And none of the chemists have a clue how it works and none of them know Java
- Guess which is in use

MADNESS

狂

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], Judy Hill[1], Jun Jia[1],
[1]Oak Ridge National Laboratory
[2]University of Tennessee, Knoxville

in collaboration with

Gregory Beylkin[4], Lucas Monzon[4],
Martin Mohlenkamp[5], and Hideo Sekino[6]
[4]University of Colorado
[5]Ohio University
[6]Toyohashi Technical University, Japan

harrisonrj@ornl.gov

National Science Foundation
WHERE DISCOVERIES BEGIN

SciDAC
Scientific Discovery through Advanced Computing

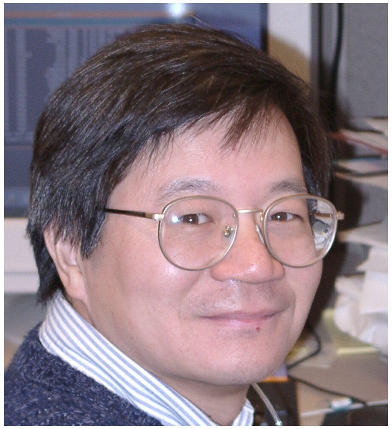
UT-BATTELLE
OAK RIDGE NATIONAL LABORATORY

Funding

- MADNESS started as a DOE SciDAC project and the majority of its support still comes from the DOE
- DOE SciDAC, 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

What is MADNESS?

- A general purpose numerical environment for reliable and fast scientific simulation
  - Applications already in nuclear physics, chemistry, atomic physics, material science, with investigations beginning in climate and fusion.
- A general purpose parallel programming environment designed for the petascale
  - Standard C++ with concepts from Cilk, Charm++, HPCS languages, with a multi-threaded runtime that dynamically manages task dependences, scheduling and provides global data view.
  - Compatible by design with existing applications

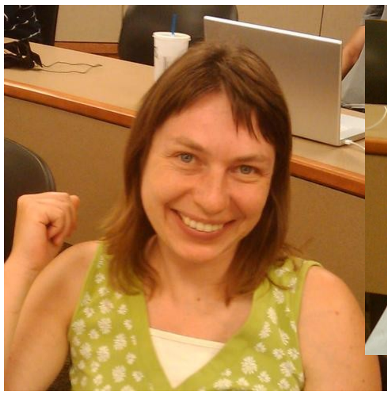

George Fann

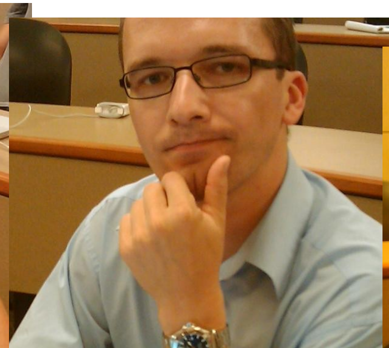

Judy Hill

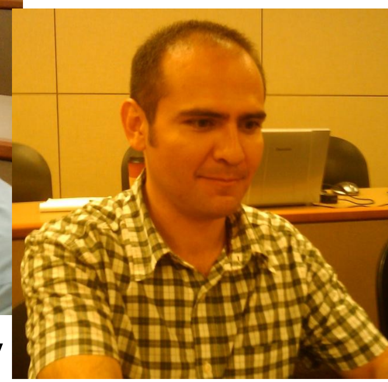

Gregory Beylkin


Rebecca Hartman-Baker

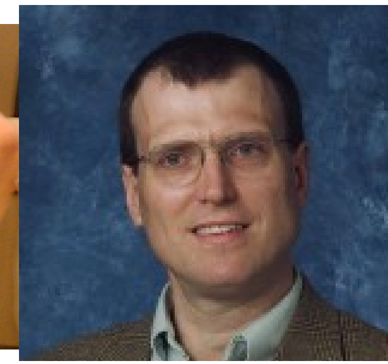
Jun Jia
Tetsuya Kato
Justus Calvin
J. Pei


Ariana Beste

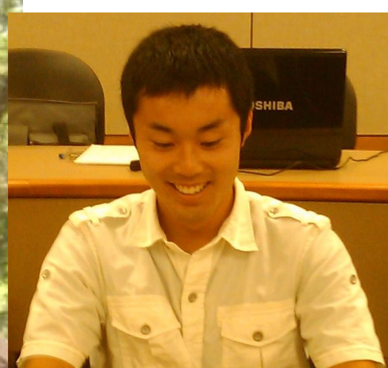

Eduard Valeyev

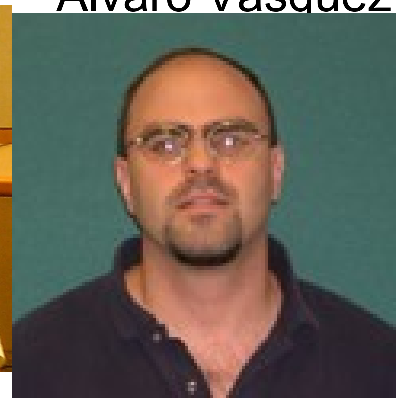

Alvaro Vasquez


Hideo Sekino


Robert Harrison


Nicholas Vence


Takahiro Ii


Scott Thornton


Matt Reuter

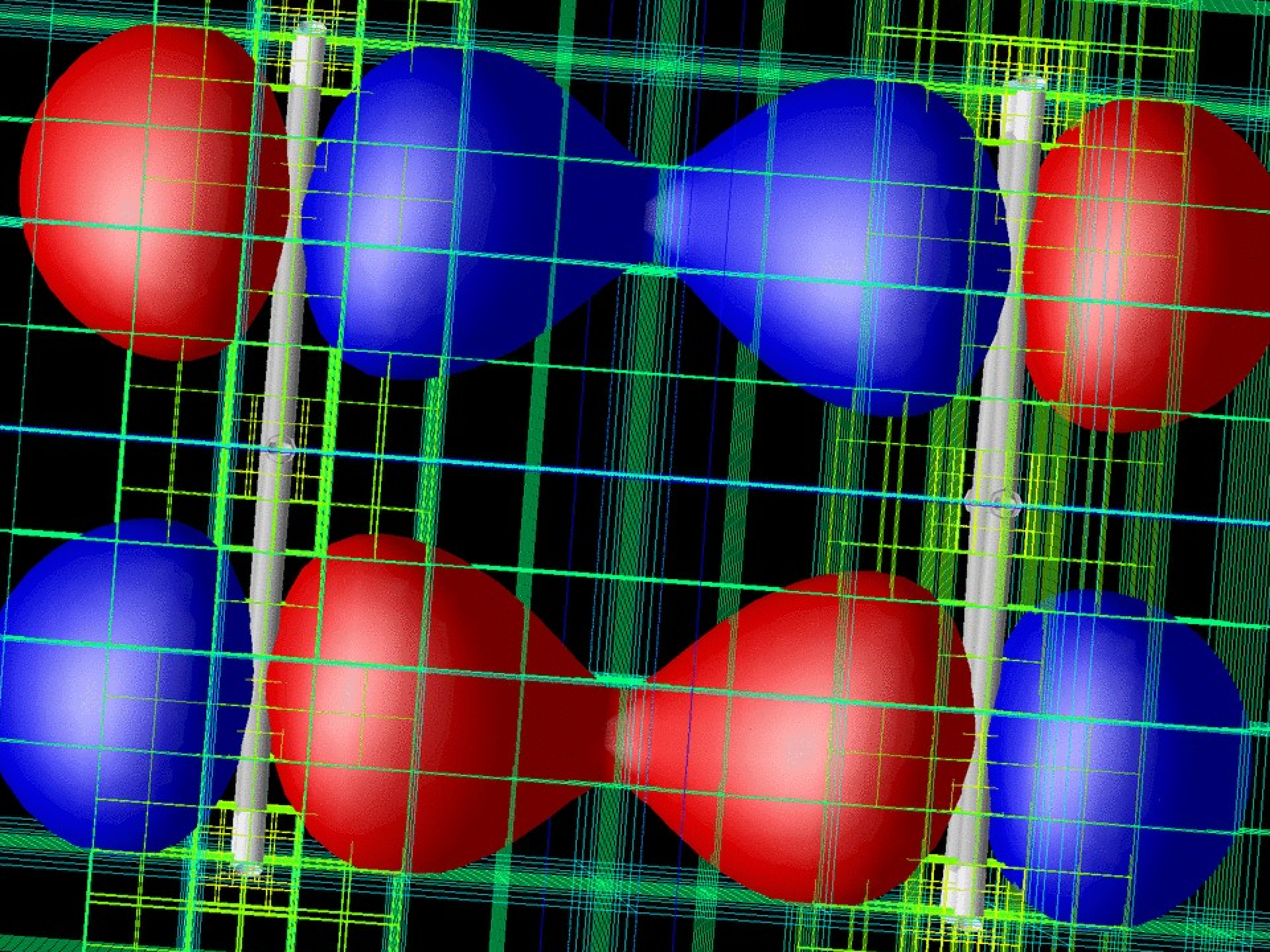

Paul Sutter

Why MADNESS

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

The math behind the MADNESS

- Discontinuous spectral element basis
  - High-order convergence ideally suited for modern computer technology
- Multi-resolution analysis for fast algorithms
  - Sparse representation of many integral operators
  - Precision guaranteed through adaptive refinement
- Separated representations of operators and functions
  - Enable efficient computation in many dimensions



Essential techniques for fast computation

- 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

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

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

- Low-operator rank

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

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

Integral Formulation

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

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

$$\begin{aligned}\Psi &= -2\left(-\nabla^2 - 2E\right)^{-1} V\Psi \\ &= -2G*\left(V\Psi\right)\end{aligned}$$

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

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

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)

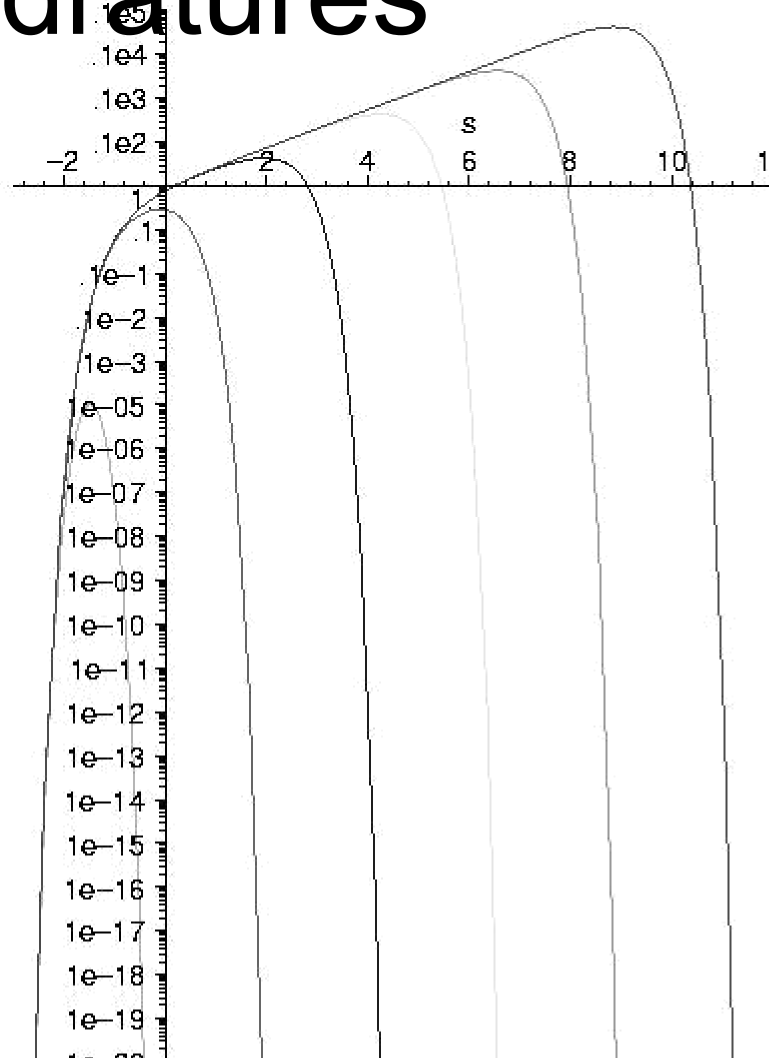
$$r_{ii',jj',kk'}^{n,l-l'} = \sum_{\mu=0}^{M} X_{ii'}^{n,l_x-l'_x} Y_{jj'}^{n,l_y-l'_y} Z_{kk'}^{n,l_z-l'_z} + O(\varepsilon)$$

  - 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 yet automated



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

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

High-level composition

- Close to the physics

$$E = \langle \psi | -\frac{1}{2}\nabla^2 + V | \psi \rangle + \langle \psi \psi | \psi \psi \rangle$$

```
operatorT G = CoulombOperator(k, rlo, thresh);
functionT rho = psi*psi;
double twoe = inner(G(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

$$\begin{aligned} \Omega &= [-20, 20]^3 \\ r &= x \rightarrow \sqrt{x_0^2 + x_1^2 + x_2^2} \\ g &= x \rightarrow \exp\left(-r\left(x\right)\right) \\ v &= x \rightarrow -r\left(x\right)^{-1} \end{aligned}$$

In

$$\begin{aligned} \psi &= \mathcal{F}\ g \\ \nu &= \mathcal{F}\ v \\ S &= \langle \psi | \psi \rangle \\ V &= \langle \psi | \nu * \psi \rangle \\ T &= \frac{1}{2} * \sum_{i=0}^{2} \left(\langle \nabla_i \psi | \nabla_i \psi \rangle\right) \end{aligned}$$

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

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

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

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

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

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

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

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

He atom Hartree-Fock

Let

$$\Omega = [-20, 20]^3$$
$$r = x \to \sqrt{x_0^2 + x_1^2 + x_2^2}$$
$$g = x \to \exp\left(-2 * r(x)\right)$$
$$v = x \to -\frac{2}{r(x)}$$

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

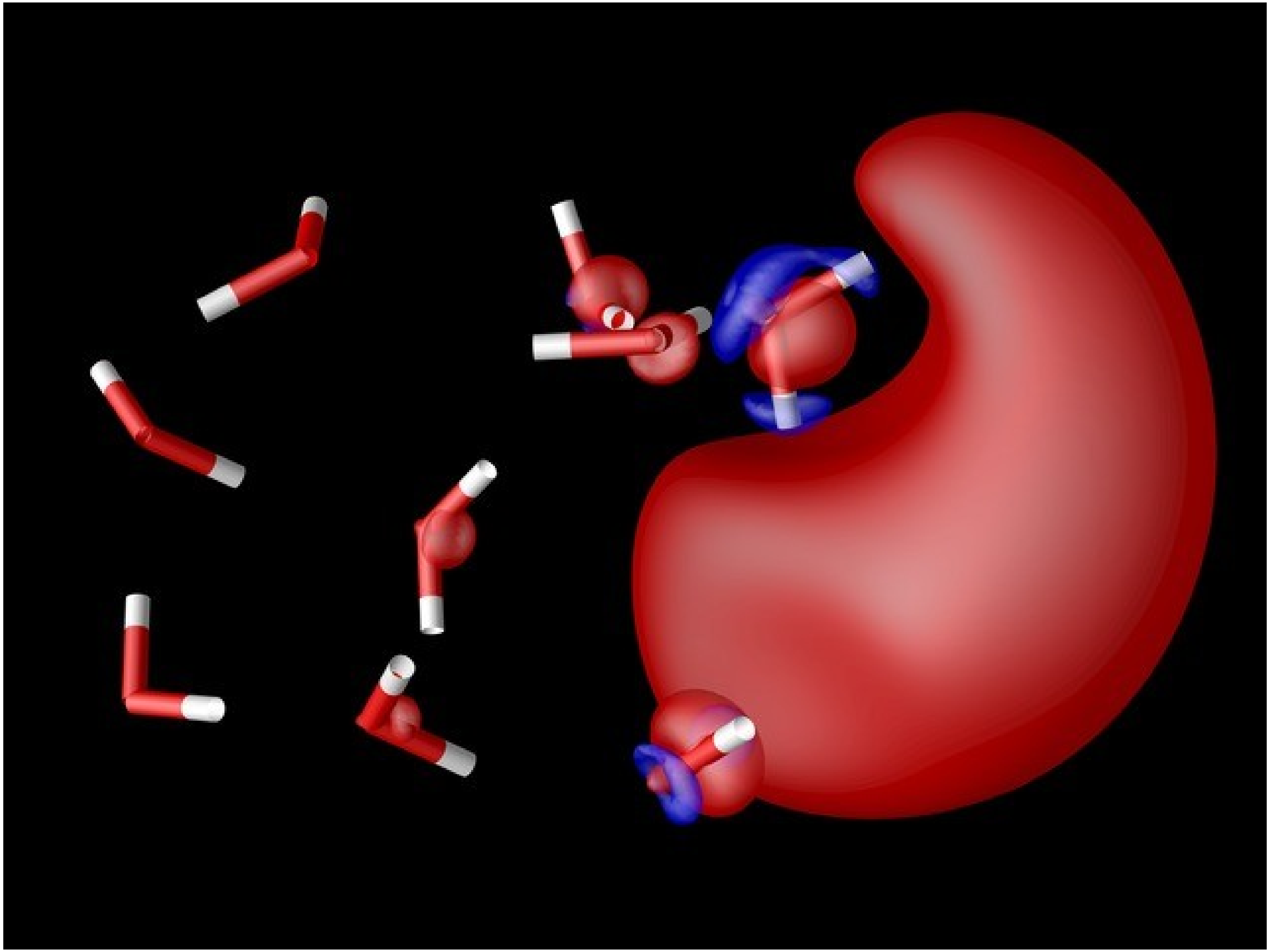
Hartree-Fock

- What I really wanted to type was

$$\min_{\phi} E[\phi] \quad \text{s.t.} \quad \|\phi\|_2 = 1$$

- But had to
  - Provide E or rather $dE/d\varphi$
  - Describe inexact-Newton algorithm with stopping criterion
  - Transform to integral representation for efficiency and accuracy
- Can automate some steps, c.f. Maple, Mathematica
  - But properties of computation in the underlying basis are crucial for accuracy and efficiency
  - So let's go back and ask why is this working ...

Molecular HF and DFT



Energy and gradients

ECPs coming (Sekino)

Response properties (Vasquez and Sekino)

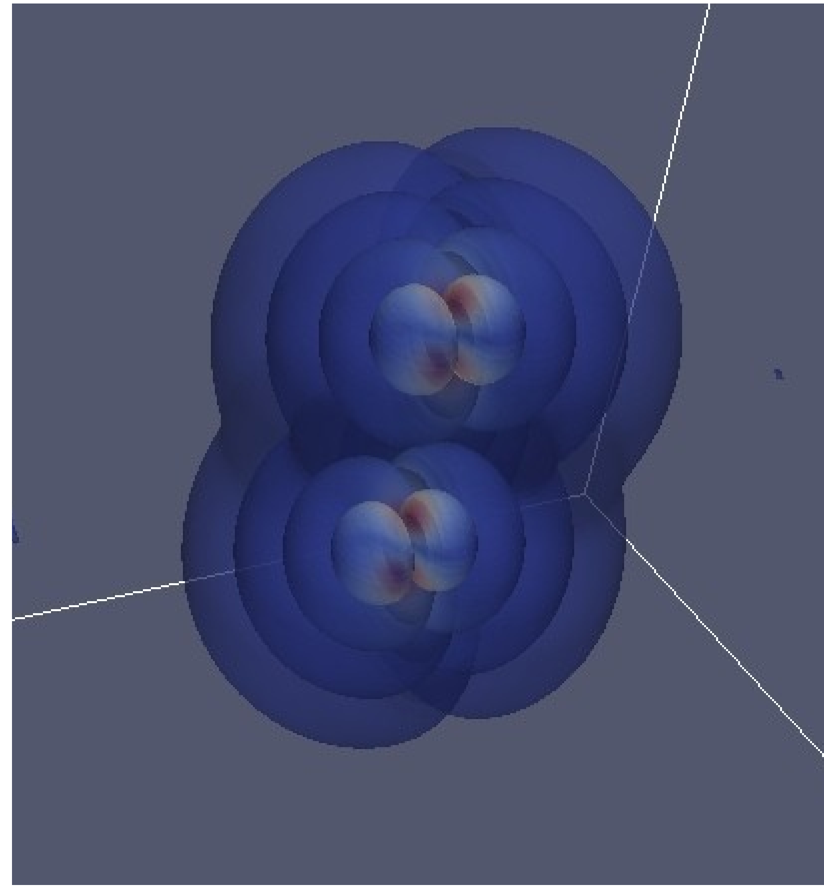
Still not as functional as previous python version

Spin density of solvated electron

Nuclear physics

Pei, Fann, Ou, Nazarewicz

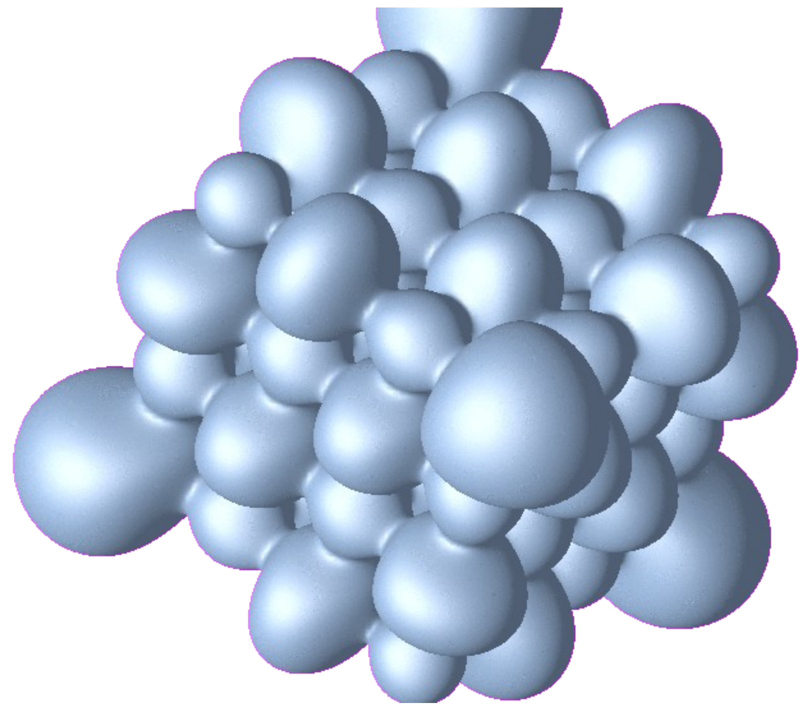
- DOE UNDEF
- Nuclei & neutron matter
- ASLDA
- Hartree-Fock Bogliobulov
- Spinors
- Gamov states



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

Solid-state physics

- Thornton, Eguiluz and Harrison (UT)
  – 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

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 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 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);
```

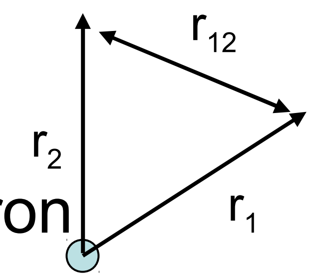
A container is created mapping indices to values.

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)

Electron correlation



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

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

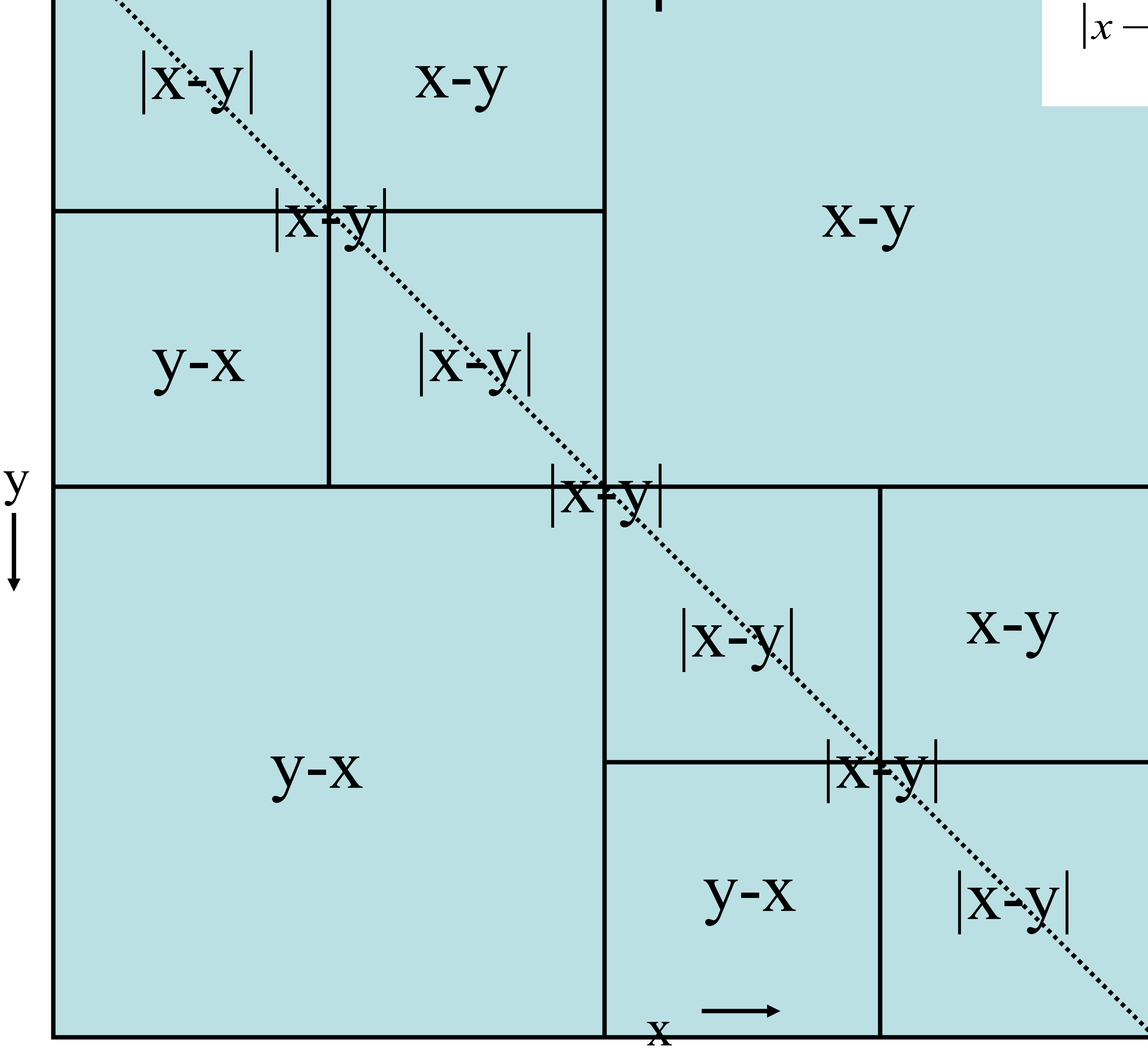
- Include the inter-electron distance in the wavefunction
  - E.g., Hylleraas 1938 wavefunction for He

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

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

	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 74 (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)

Preliminary results for He atom (Yanai, 2005)

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)

Summary

- Huge computational resources are rushing towards us
  - Tremendous scientific potential
  - Tremendous challenges in
    - Research,
    - Education, and
    - Community
- We need radical changes how we compose scientific S/W – and we need your help.
- UT and ORNL
  - Think of us if you have some good students looking for challenging graduate or postdoctoral study

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

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

- 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

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

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

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

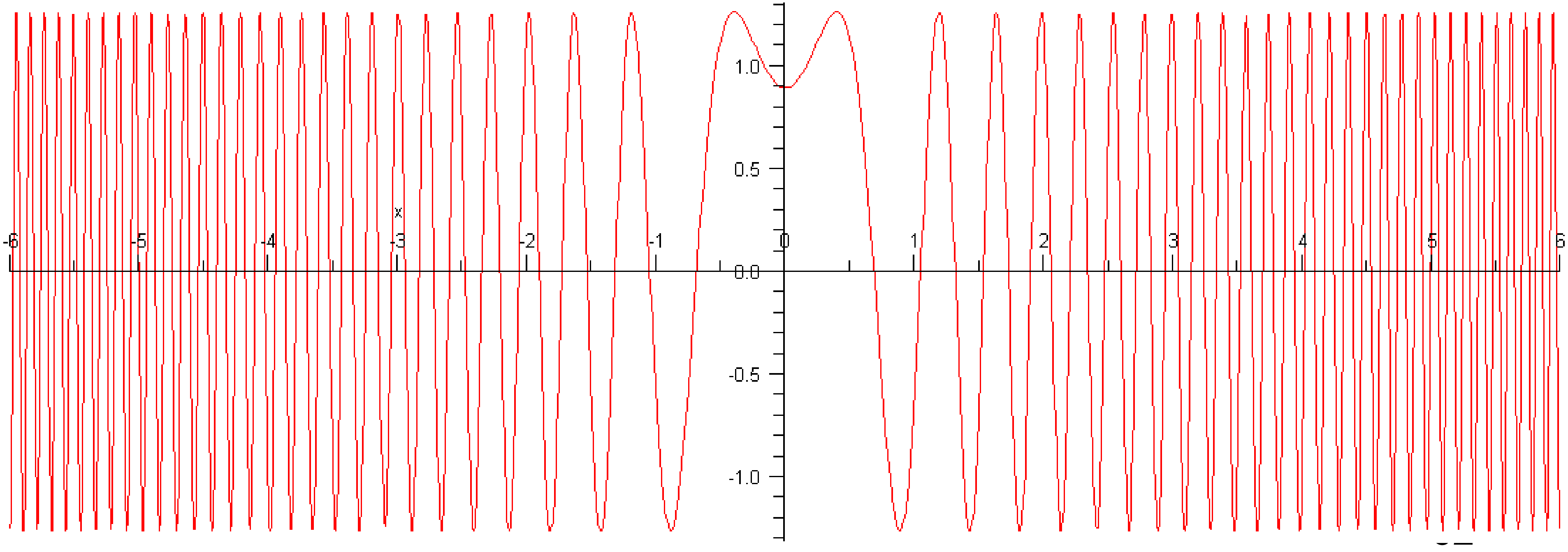
$$\lim_{t\to\infty}\psi(x,t)=\psi_0(x)$$

  - Wrap in solver to accelerate convergence

Exponential propagator

- Free-particle propagator in real time

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

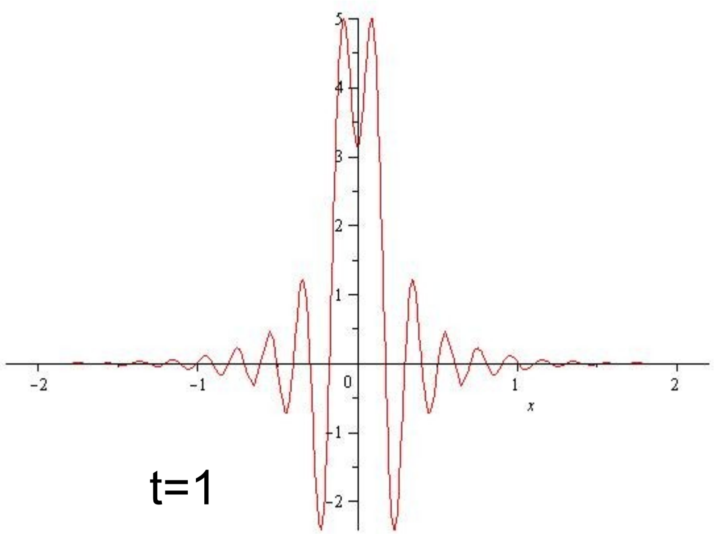


Exponential propagator

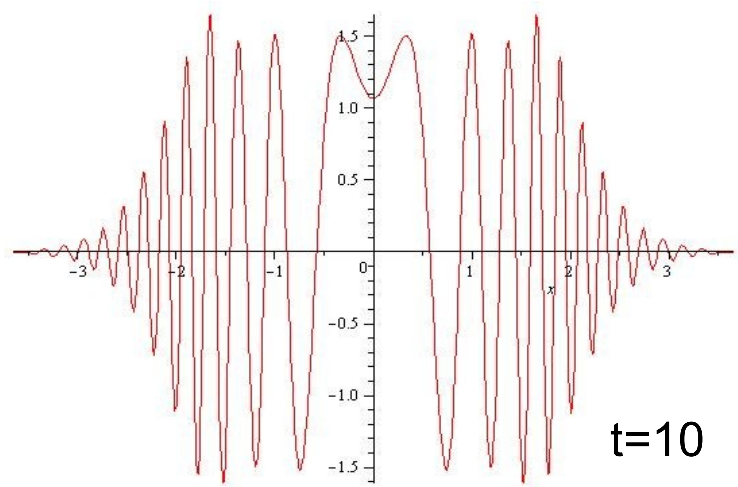
- Combine with projector onto band limit

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

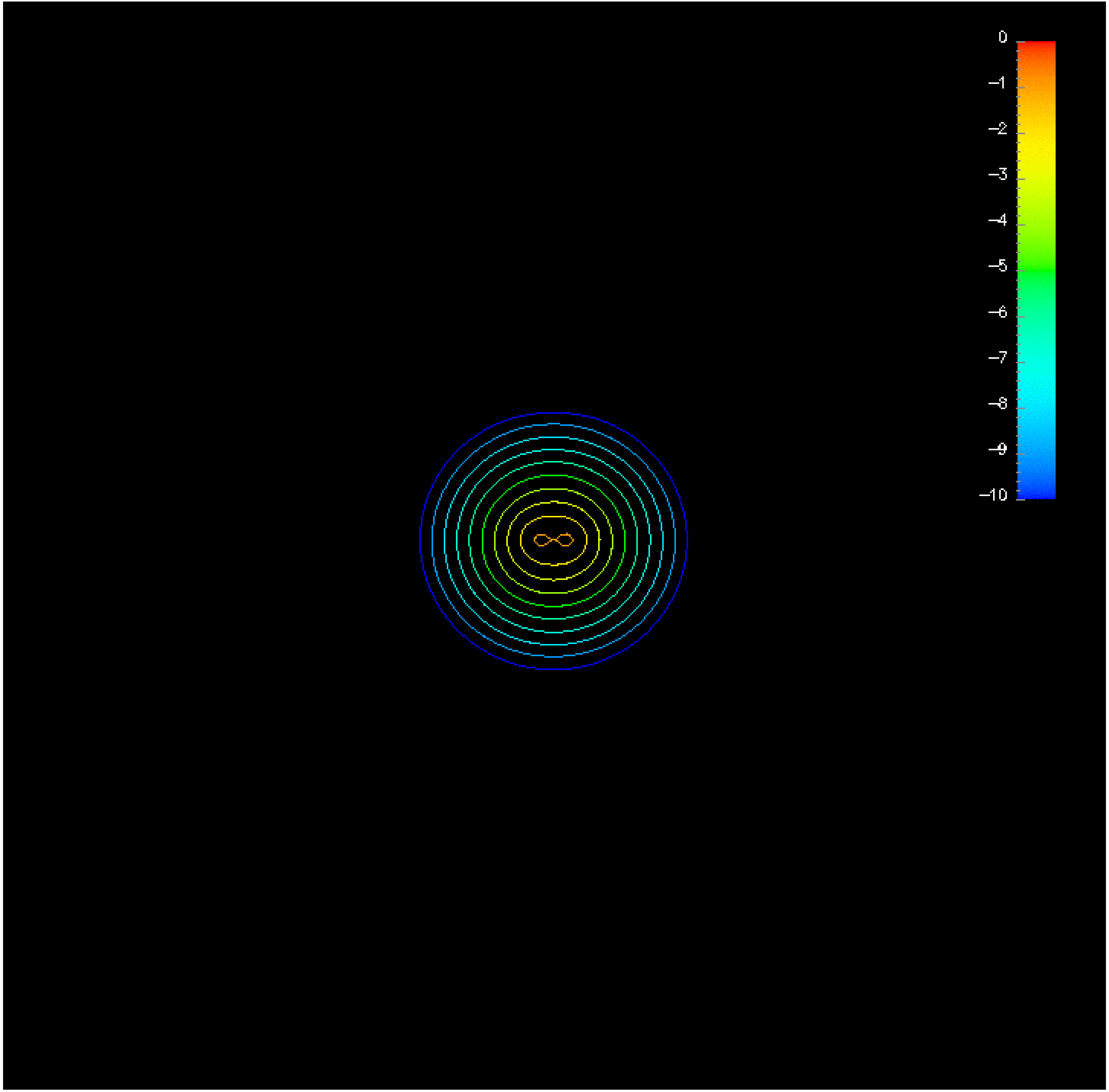
$$h = \frac{\pi}{c} \qquad t_{crit} = \frac{2h^2}{pi}$$



t=1



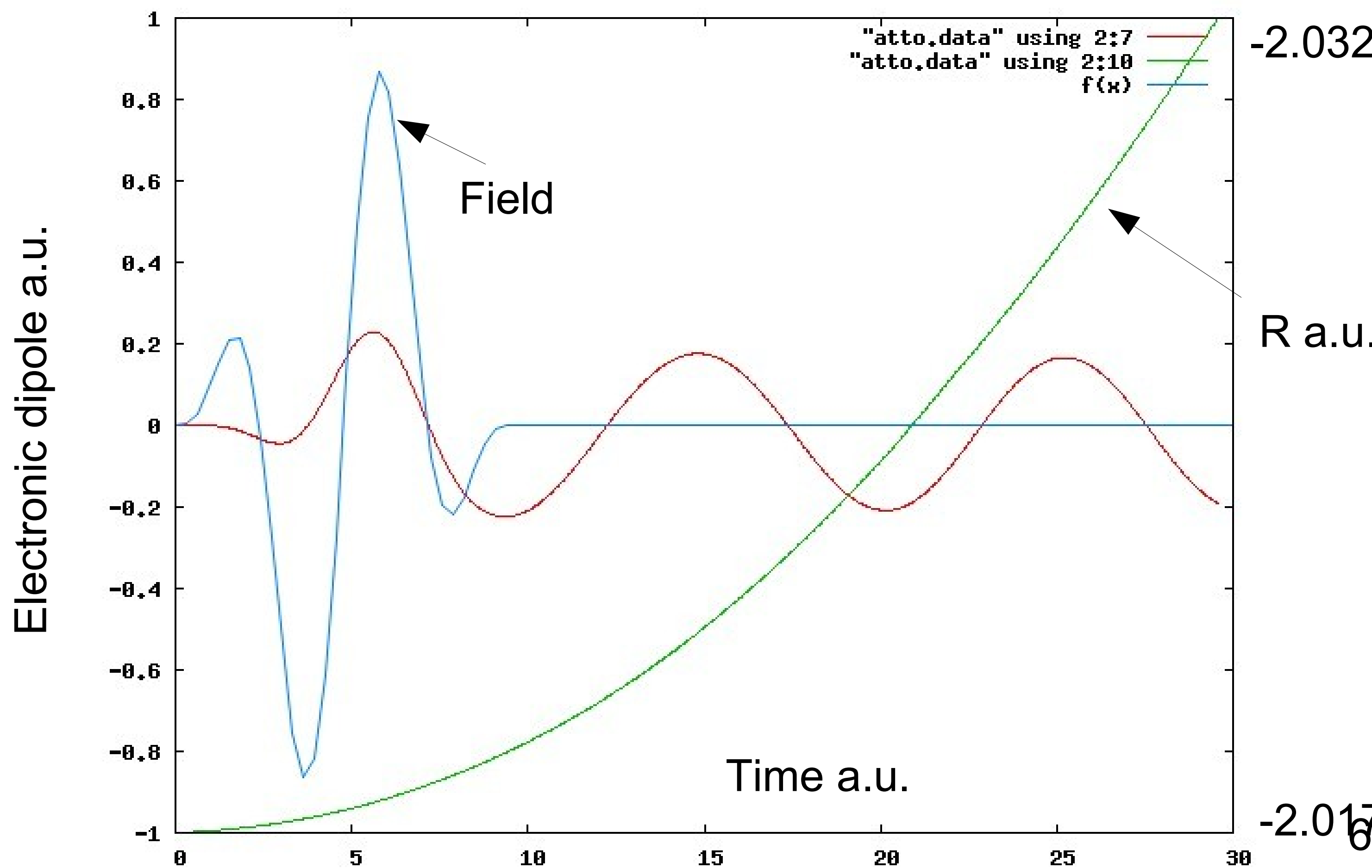
t=10



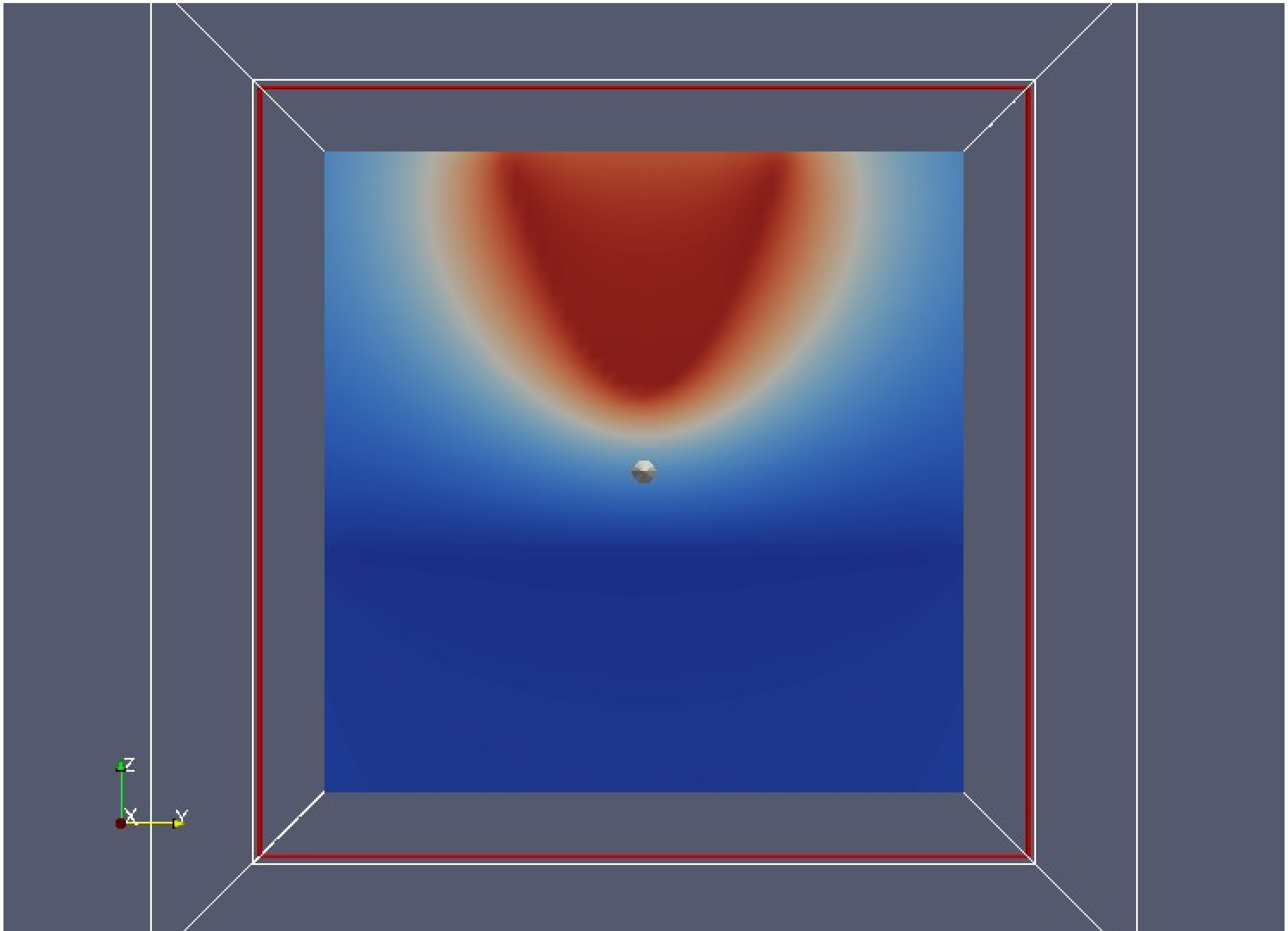
H2+ molecule in laser field (fixed nuclei)

Dynamics of H_2^+ in laser

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



Nanoscale phtonics (Matt Reuter, Northwestern)



Diffuse domain approximation for boundary value problem

Maxwell equations

Micron-scale Au tip 2nm above Si surface.

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, localization, diagonalization
- Linear response properties

Systolic loop parallel algorithm for localization and diagonalization – Ii, Sekino, Harrison

Summary

- MADNESS is a general purpose framework for scientific simulation
  - Conceived for the next (not the last) decade
  - Aims to makes scientific HPC more productive by reducing various sources of complexity
  - Deploys advanced numerical and C/S methods
- Multiple science applications at various levels of completeness

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