The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2220-17

15th International Workshop on Computational Physics and Materials Science: Total Energy and Force Methods

13 - 15 January 2011

Harnessing the power of new computer hardware for electronic structure calculations with the BigDFT code

Stefan Goedecker
University of Basel
Basel
Switzerland

Harnessing the power of new computer hardware for electronic structure calculations with the BigDFT code

Stefan.Goedecker@unibas.ch
http://www.unibas.ch/comphys/comphys

- Trends in computer architecture
- What are wavelets and their advantages
- Performance characteristics of the BigDFT code on various hardware platforms
- Capabilities and applications of the BigDFT code

Some unpleasant and pleasant facts of computer hardware development

Hardware:

- The clock cycle has saturated at around 3 GHz
- The number of transistors is still increasing acoording to Moore's law
- The number of cores per processor/node is increasing
- Memories are getting larger but not faster

Consequences:

- The speed of a serial program will not increase in the future
- The speed will only increase if the program can be parallelized
- Number of memory access operations becomes more important than number of floating point operations

Computing on massively parallel computers with multicore nodes

Two possibilities

- Each core executes a MPI process
  - exisiting MPI programs can be used
  - The number of MPI processes is limited for a fixed problem size
  - There will frequently not be enough memory per MPI process
  - Communication requirements typically increase strongly with the number of MPI processes
- Each node executes an MPI process which will spawn several threads which run on the different cores of a processor
  - Requires an additional parallelization level (OpenMP for traditional CPU's, CUDA for GPU's) and possibly code restructuring for increased parallelism
  - The only way to obtain additional speed in the near future for fixed problem size

GPU's

Characteristics

- Multicore architecture with a huge number (about 500) of cores
- Nearly perfect hiding of latency: high memory bandwidth
- Relatively low clock speed (1 GHz)
- Very low transfer speed form CPU to GPU memory

Consequences

Higher speed can only be obtained if

- The program can use a few thousand threads
- Many consecutive operations for a given data set are executed on the GPU.

BigDFT is a program which solves the many-electron Schrödinger equation in the Kohn-Sham density functional approximation using a wavelet basis

European project: Luigi Genovese, Alexey Neelov, Stefan Goedecker, Thierry Deutsch, Alireza Ghasemi, Oded Zilberberg, Anders Bergman, Mark Rayson and Reinhold Schneider: Daubechies wavelets as a basis set for density functional pseudopotential calculations: J. Chem. Phys. **129**, 014109 (2008)

This wavelet basis set combines advantages of

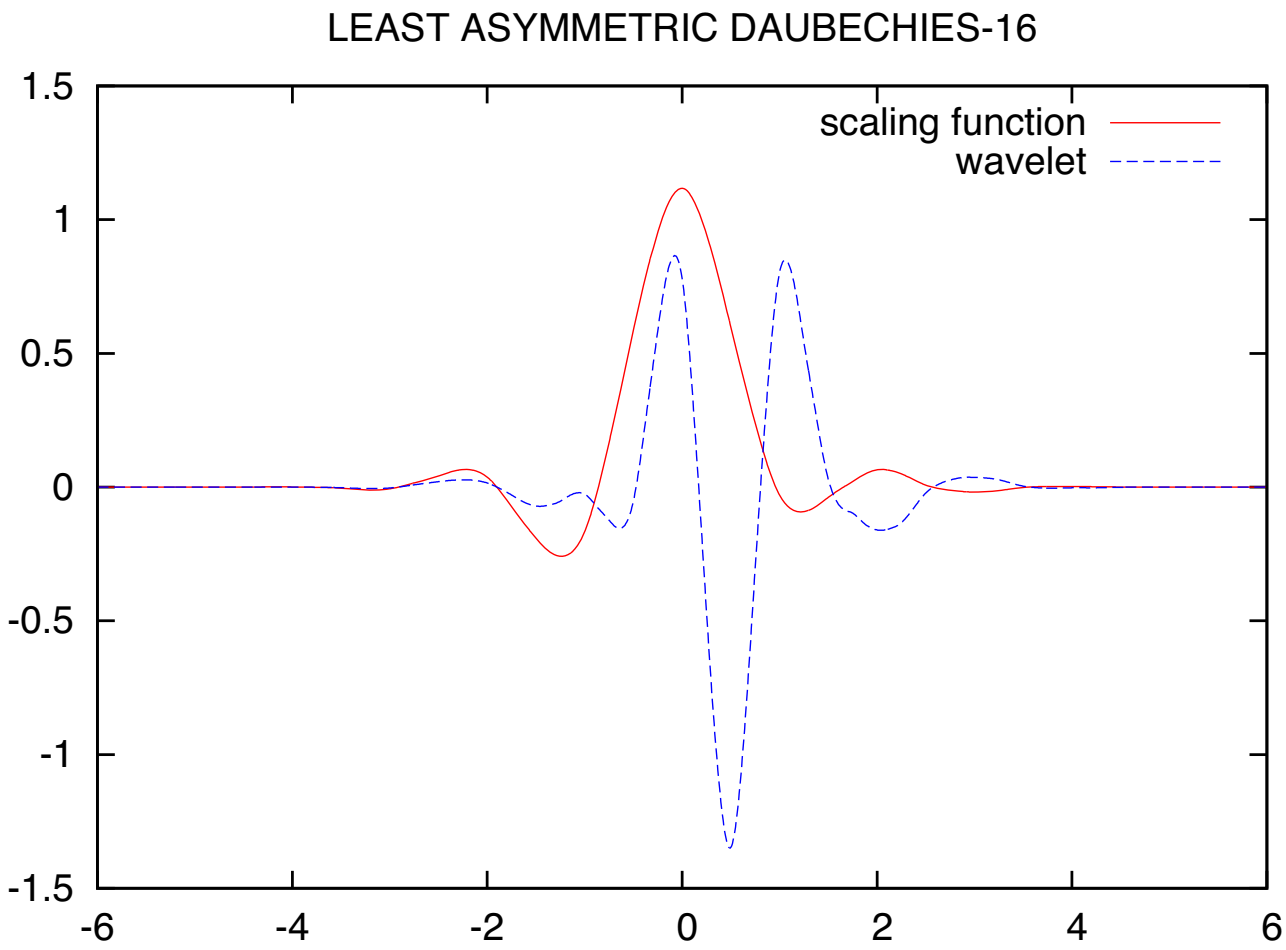
Plane waves:

- Systematic, orthogonal basis set
- Localization in Fourier space allows for efficient preconditioning techniques.

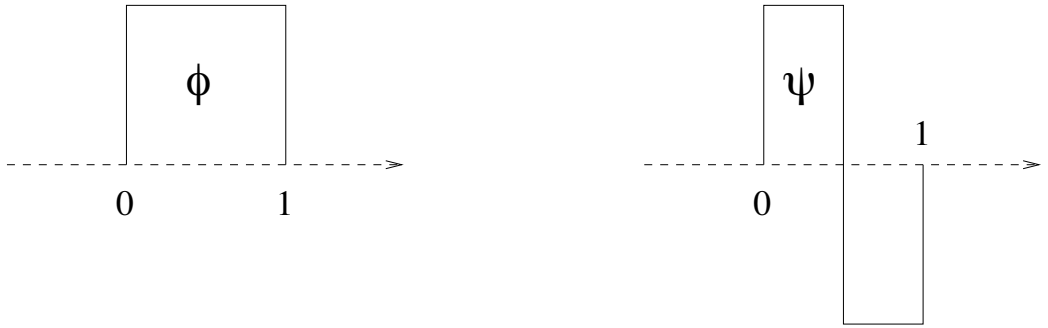
Gaussians:

- Localized in real space: well suited for molecules and other open structures.
- Adaptivity

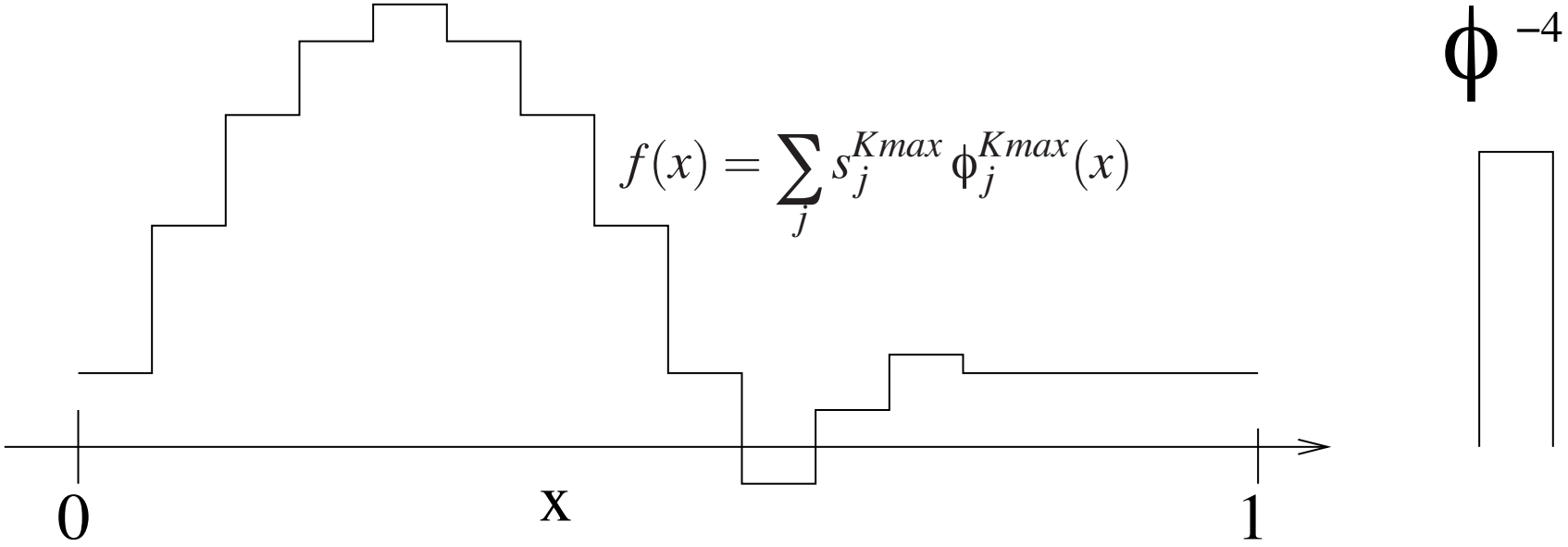
High order Daubechies wavelets are used to represent wavefunctions



Haar wavelet ψ and scaling function ϕ



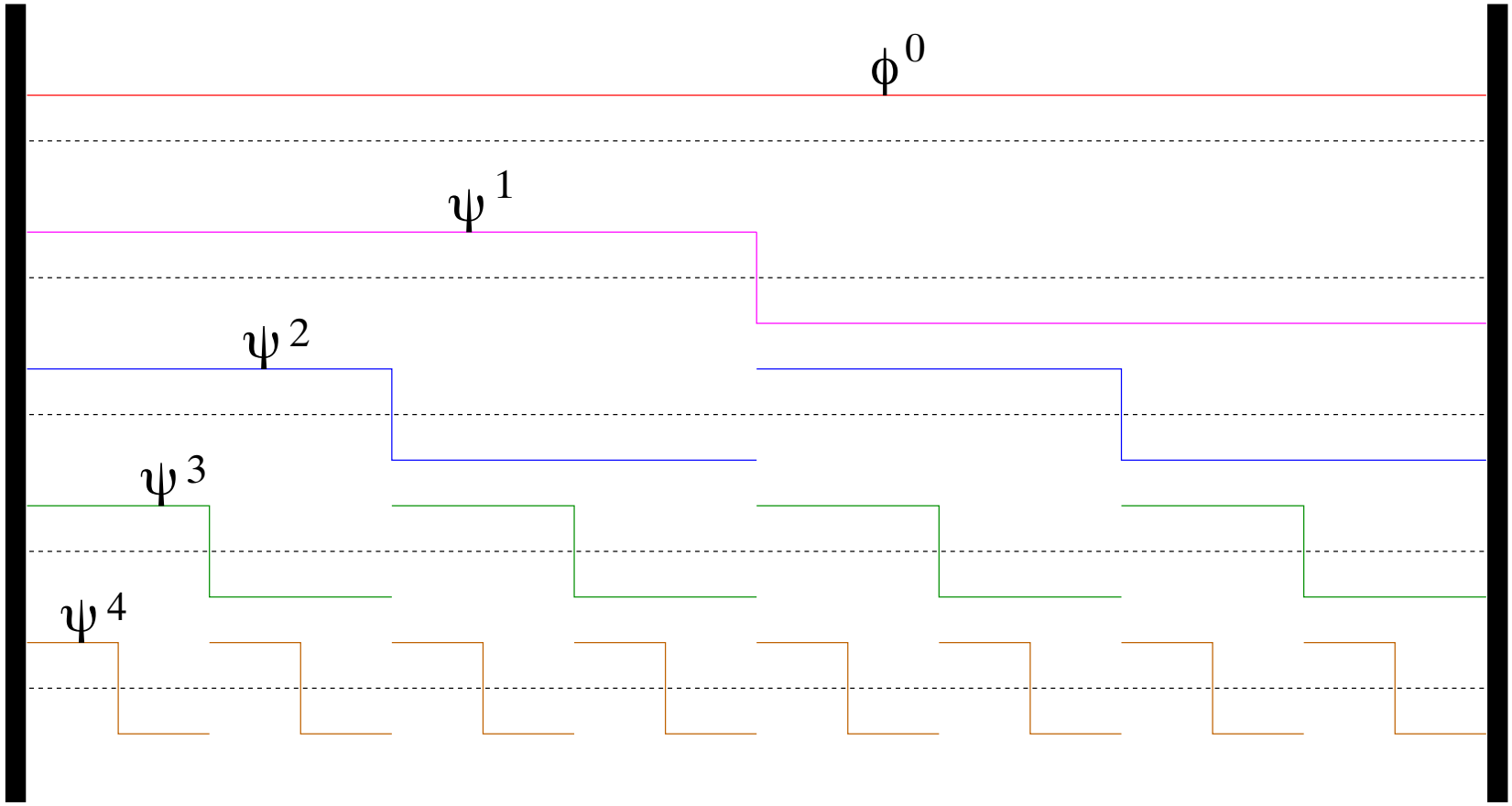
Scaling function representation



Refinement relations:

$$\phi(x) = \sum_{j=-m}^{m} h_j \, \phi(2x - j)$$

Haar wavelet basis set



Wavelet representation in general

$$f(x) = \sum_j s_j^{Kmin}\, \phi_j^{Kmin}(x) + \sum_{k=Kmin}^{Kmax} \sum_j d_j^k\, \psi_j^k(x)$$

Wavelet basis sets in three dimensions

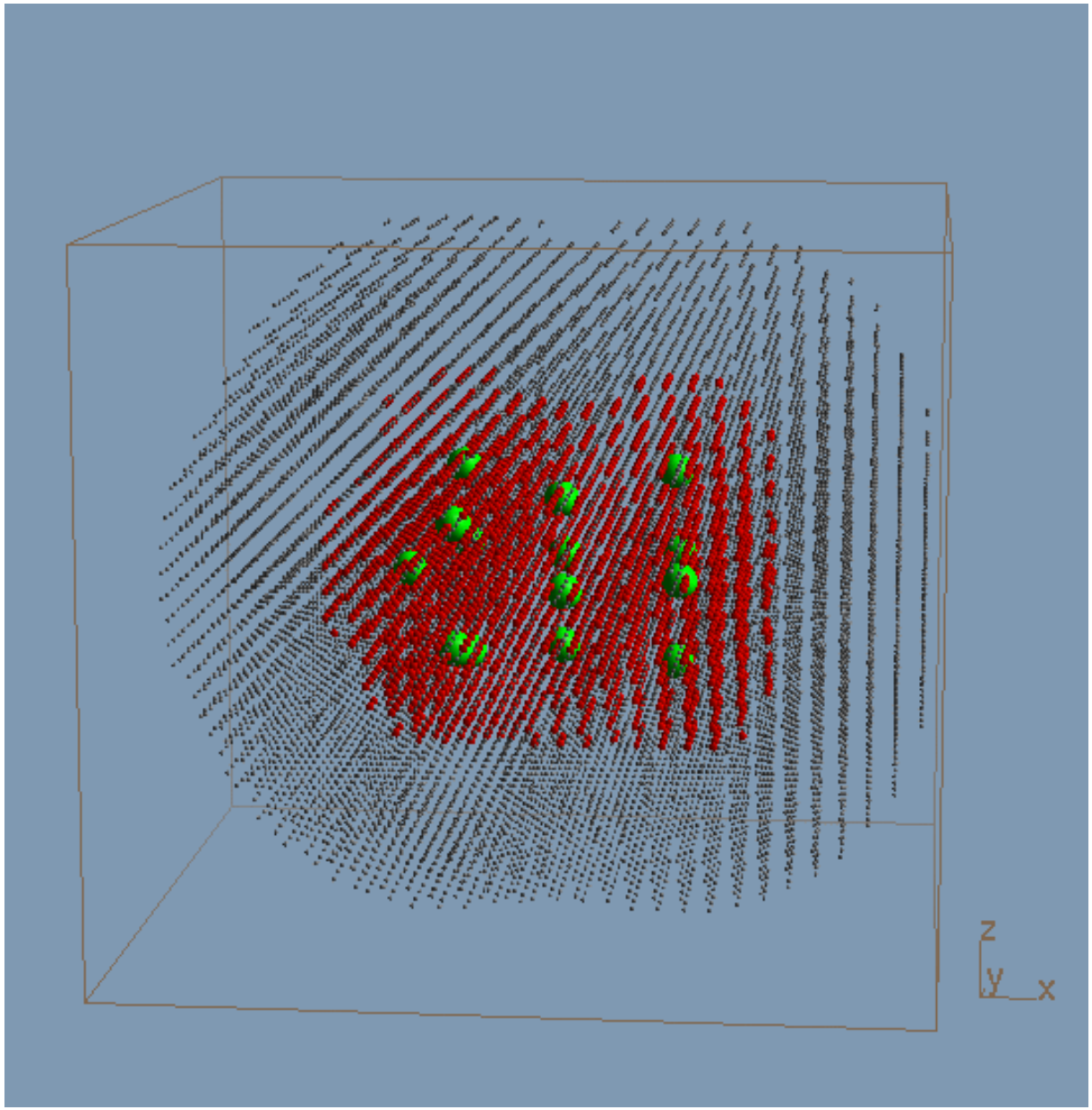
1 scaling function

7 wavelets

all are products of 1-dim scaling functions and wavelets

$$
\begin{aligned}
\phi_{i,j,k}(x,y,z) &= \phi(x-i)\phi(y-j)\phi(z-k) \\
\psi^1_{i,j,k}(x,y,z) &= \phi(x-i)\phi(y-j)\psi(z-k) \\
\psi^2_{i,j,k}(x,y,z) &= \phi(x-i)\psi(y-j)\phi(z-k) \\
\psi^3_{i,j,k}(x,y,z) &= \phi(x-i)\psi(y-j)\psi(z-k) \\
\psi^4_{i,j,k}(x,y,z) &= \psi(x-i)\phi(y-j)\phi(z-k) \\
\psi^5_{i,j,k}(x,y,z) &= \psi(x-i)\phi(y-j)\psi(z-k) \\
\psi^6_{i,j,k}(x,y,z) &= \psi(x-i)\psi(y-j)\phi(z-k) \\
\psi^7_{i,j,k}(x,y,z) &= \psi(x-i)\psi(y-j)\psi(z-k)
\end{aligned}
$$

Two resolution levels are used



Basic types of operations

- Application of the Kohn-Sham Hamiltonian on the wavefunctions: Operator approach
  - Kinetic energy
  - Separable pseudopotential energy
  - Potential energy
- Calculation of Hartree potential: exact solution of integral equation for all boundary conditions
- Orthogonalization and orthogonality constraints: Euclidean scalar products and rank one updates (BLAS)

Kinetic energy matrix elements between scaling functions

$$a_i = \int \phi(x) \frac{\partial^2}{\partial x^2} \phi(x-i) dx$$

Can be calculated exactly (Beylkin):

$$\begin{aligned}
a_i &= \int \phi(x) \frac{\partial^2}{\partial x^2} \phi(x-i) dx \\
&= \sum_{\nu,\mu} 2h_\nu h_\mu \int \phi(2x-\nu) \frac{\partial^2}{\partial x^2} \phi(2x-2i-\mu) dx \\
&= \sum_{\nu,\mu} 2h_\nu h_\mu 2^{2-1} \int \phi(y-\nu) \frac{\partial^2}{\partial y^2} \phi(y-2i-\mu) dy \\
&= \sum_{\nu,\mu} h_\nu h_\mu 2^2 \int \phi(y) \frac{\partial^2}{\partial y^l} \phi(y-2i-\mu+\nu) dy \\
&= \sum_{\nu,\mu} h_\nu h_\mu \, 2^2 \, a_{2i-\nu+\mu}
\end{aligned}$$

We thus have to find the eigenvector $\mathbf{a}$ associated with the eigenvalue of 2^{-2},

$$\sum_j A_{i,j} \, a_j = \left(\frac{1}{2}\right)^2 a_i$$

where the matrix $A_{i,j}$ is given by

$$A_{i,j} = \sum_{\nu,\mu} h_\nu h_\mu \, \delta_{j,2i-\nu+\mu}$$

Operator approach: Applying the kinetic energy operator results in a convolution with the filter a

$$(K\Psi)_i = \sum_j \Psi_j a_{i-j}$$

In the 3-dim case applying the kinetic energy operator requires a 3-dim convolution with a filter that is a product of 3 1-dim filters. The operation count is $3\,N_1N_2N_3L$ where L is the lenght of the filter.

The potential energy

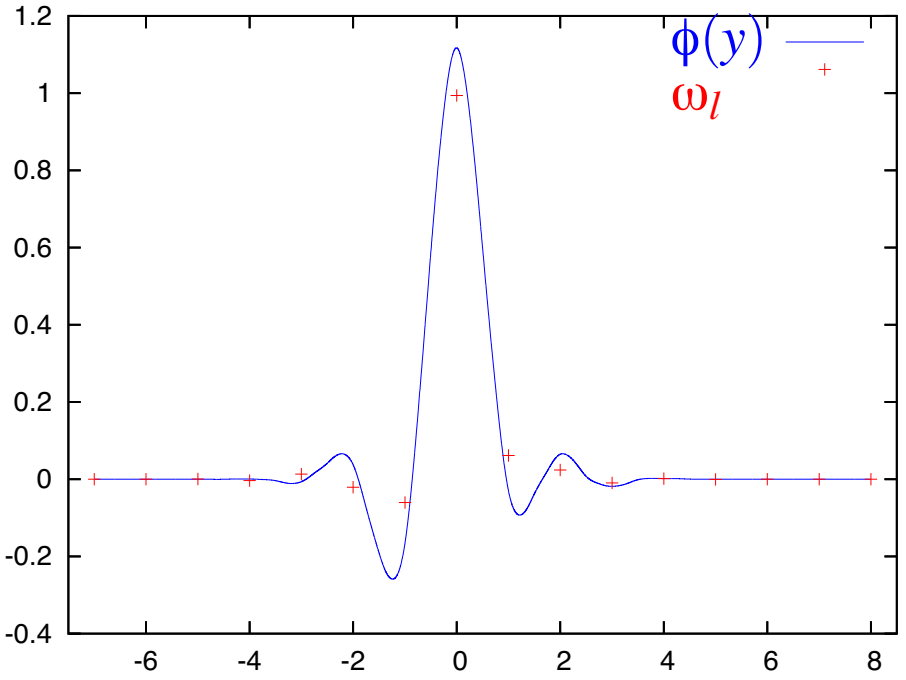
Problem: Daubechies are not smooth enough for conventional integration schemes.

The "magic filter" method

A. Neelov and S. Goedecker, J. of. Comp. Phys. **217**, *312-339 (2006)*

We do not calculate the values of Ψ on grid points, but we calculate the values $\bar{\Psi}_i$ that represent best Ψ in a neighborhood around the grid point

$$\bar{\Psi}_i = \sum_j \Psi_j \omega_{i-j} \qquad ; \qquad E_{pot} = \sum_i \bar{\Psi}_i V_i \bar{\Psi}_i$$



total energy convergence rate: h^{14}

Solution of Poisson's equation for free boundary conditions

L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, G. Beylkin, J. Chem. Phys. 125 (2006)

Given the values of the charge density on a regular grid, $\rho_{i,j,k}$, the continuous charge distribution is represented in terms of interpolating scaling functions

$$\rho(\mathbf{r}) = \sum_{i,j,k} \rho_{i,j,k} \phi(x-i)\phi(y-j)\phi(z-k)$$

The moments of the discrete and continuous charge distributions $\rho_{i,j,k}$ and $\rho(\mathbf{r})$ are identical

$$\sum_{i,j,k} i^{l_1} j^{l_2} k^{l_3} \rho_{i,j,k} = \int d\mathbf{r}\, x^{l_1} y^{l_2} z^{l_3} \rho(\mathbf{r}) \tag{1}$$

if $l_1, l_2, l_3 < m$, where m is the order of the scaling functions. The potential at a grid point i_1, i_2, i_3 is given by

$$\begin{aligned} V_{i_1,i_2,i_3} &= \sum_{j_1,j_2,j_3} \rho_{j_1,j_2,j_3} \int \frac{\phi(x'-j_1)\phi(y'-j_2)\phi(z'-j_3)}{|r_{j_1,j_2,j_3} - r'|} dr' \\ &= \sum_{j_1,j_2,j_3} \rho_{j_1,j_2,j_3} K_{i_1-j_1,i_2-j_2,i_3-j_3} \end{aligned}$$

The above convolution can be calculated rapidly with Fourier methods

The integral $K_{i-ip,j-jp,k-kp}$ becomes a sum of separable terms if $1/r$ is expanded in a sum of Gaussians

$$\frac{1}{r} \approx \sum_{k=1}^{89} \omega_k e^{-p_k r^2}$$

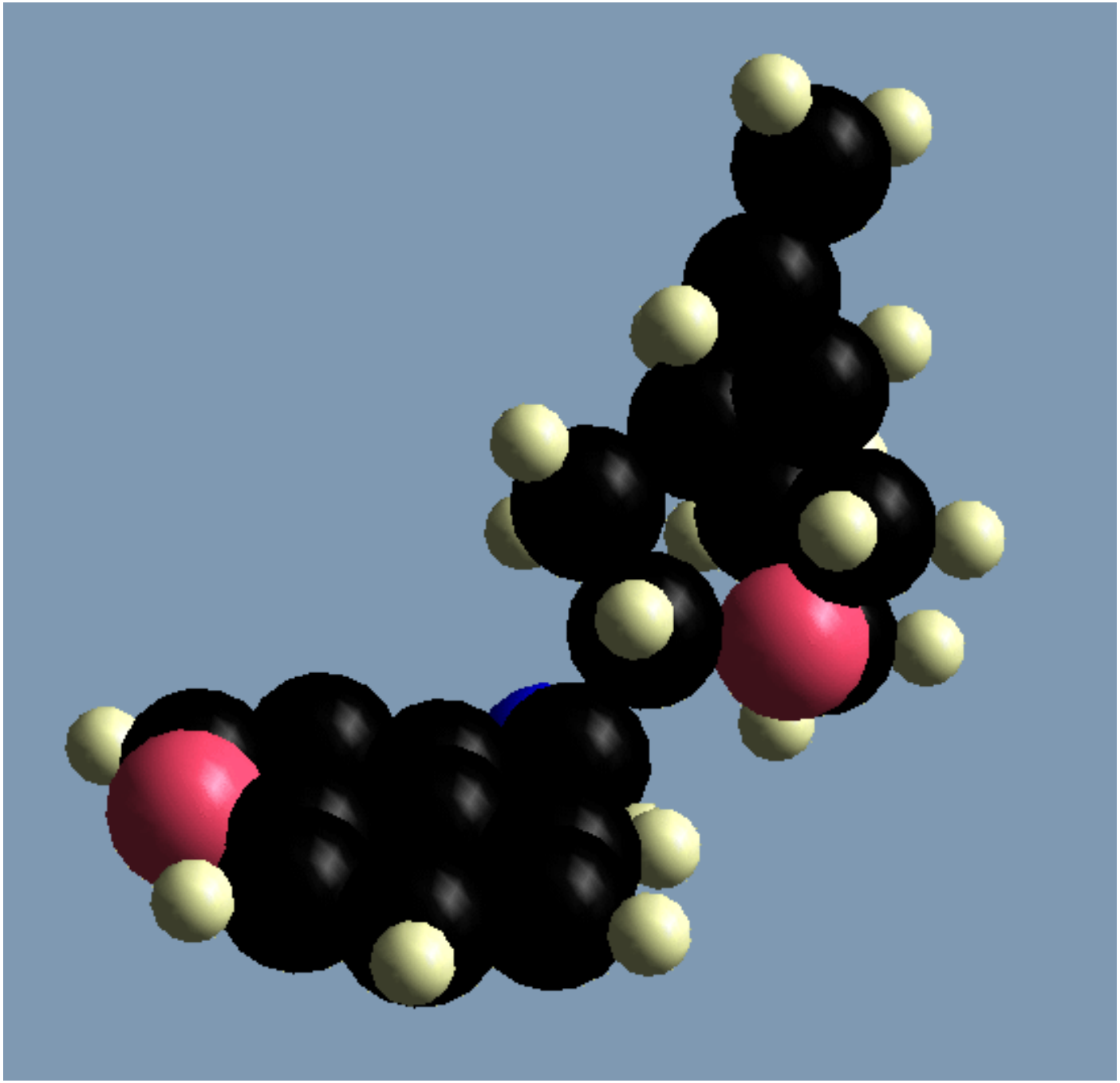
$$K_{j_1,j_2,j_3} = \sum_{k=1}^{89} \omega_k K_{j_1}(p_k) K_{j_2}(p_k) K_{j_3}(p_k)$$

We need to evaluate $89 \times \max(\{n_1, n_2, n_3\})$ integrals of the type

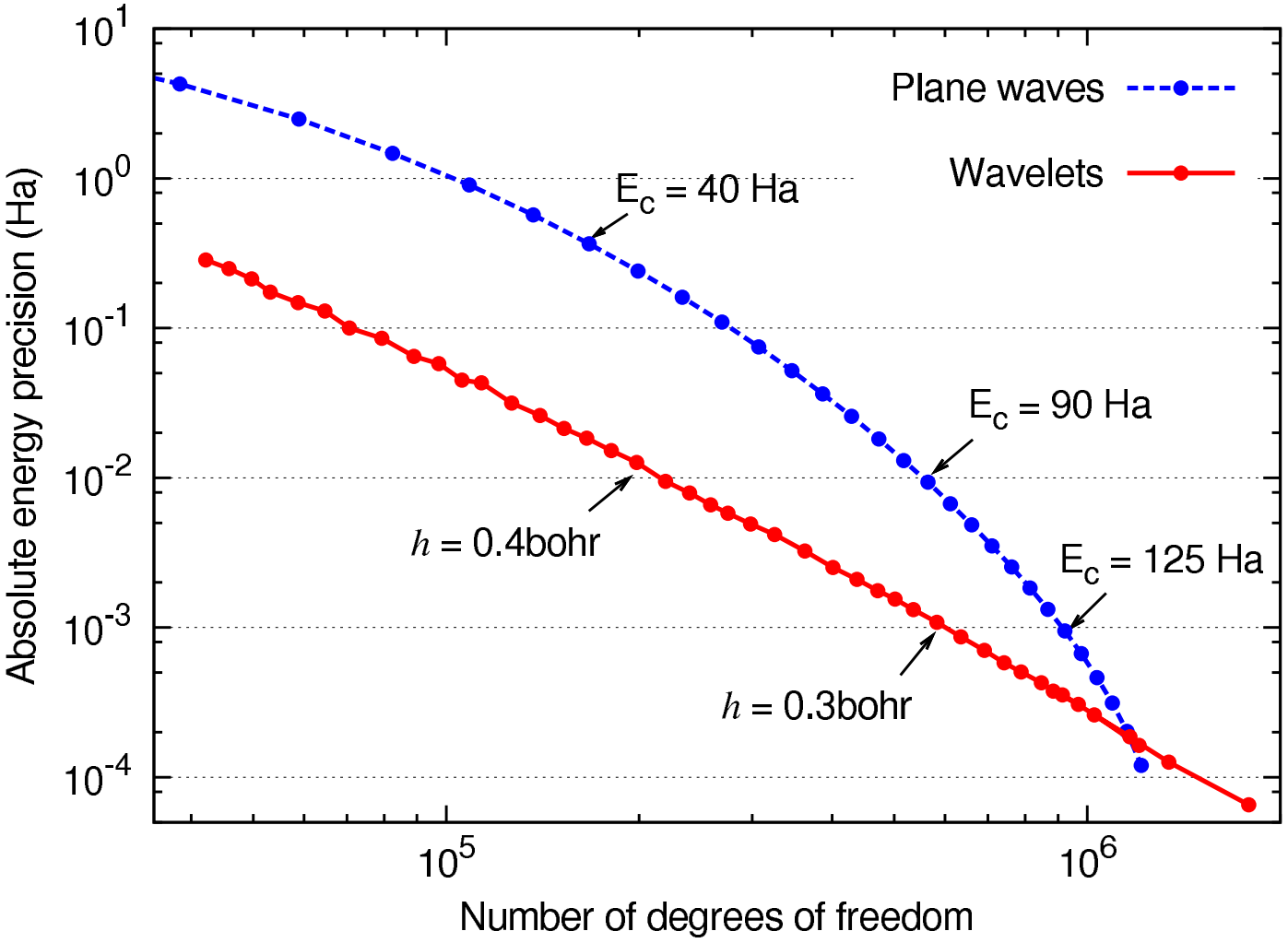
$$K_j(p) = \int \varphi_0(x) e^{-p(x-j)^2} dx$$

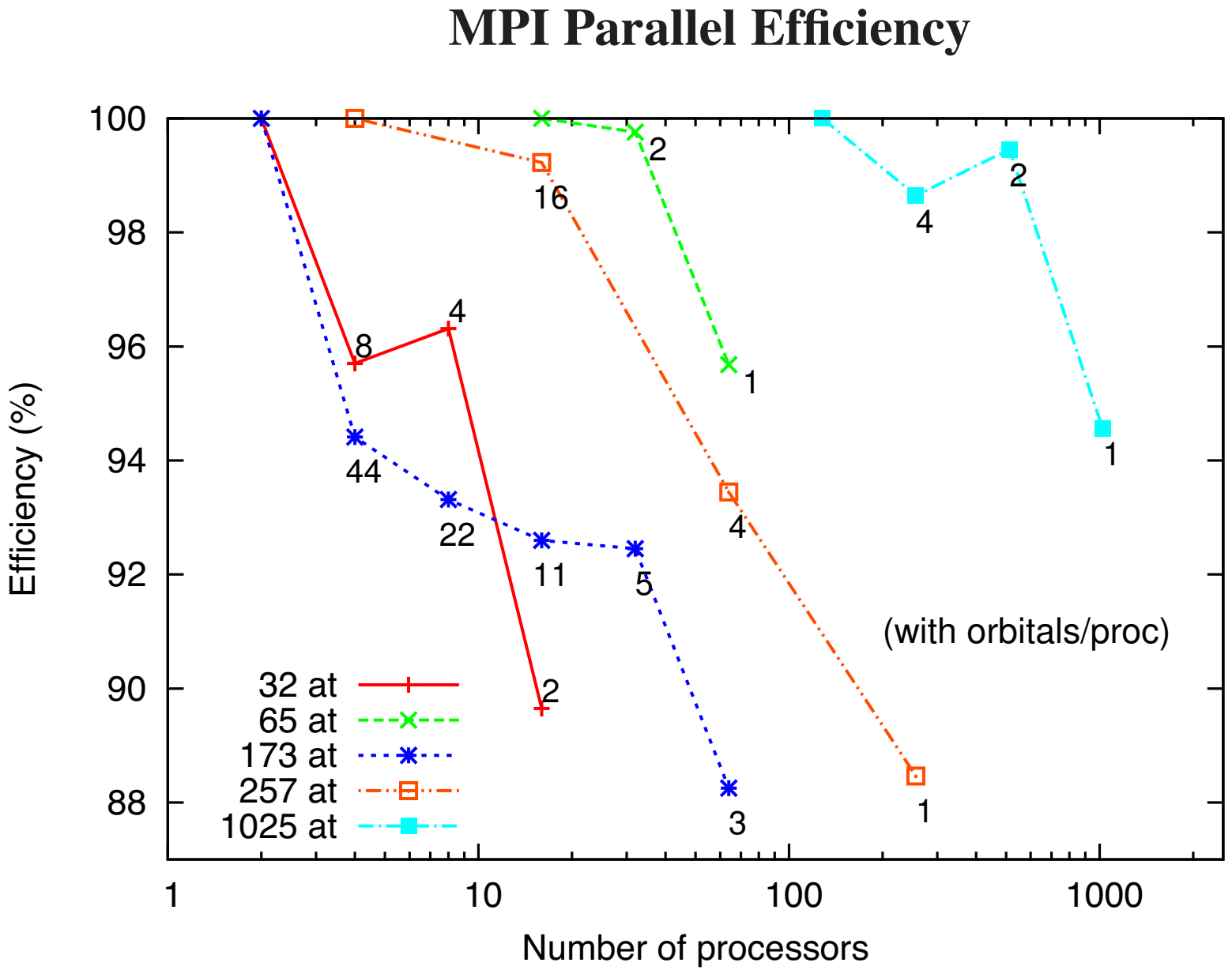
In this way all the integrals can easily be calculated up to machine precision

Performance Tests: Cinconidine



Performance Comparison with plane waves

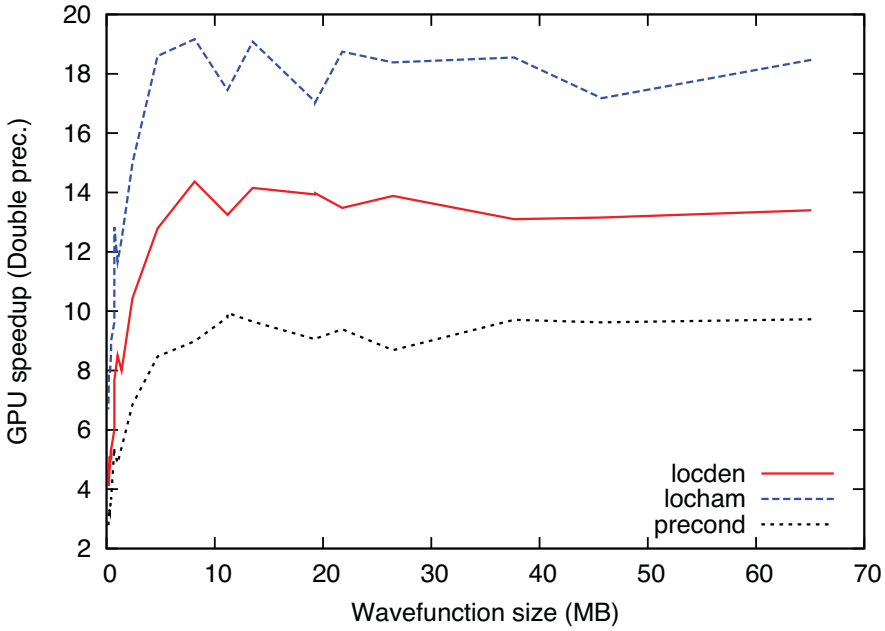



MPI Parallel Efficiency
Efficiency (%)
Number of processors
100
98
96
94
92
90
88
1
10
100
1000
32 at
65 at
173 at
257 at
1025 at
(with orbitals/proc)
8
4
2
44
22
11
5
3
16
4
1
2
1
4
2
1

Speedups obtained in BigDFT from multicores

- OpenMP on 6 core AMD Opteron: 5 in OpenMP parallelized subroutines, 4 overall
- CUDA on Nvidia Tesla GPU: 10 - 20 in CUDA parallelized subroutines, 6 overall

Speedups for the convolutions:



Total speedups for 32 water molecules:

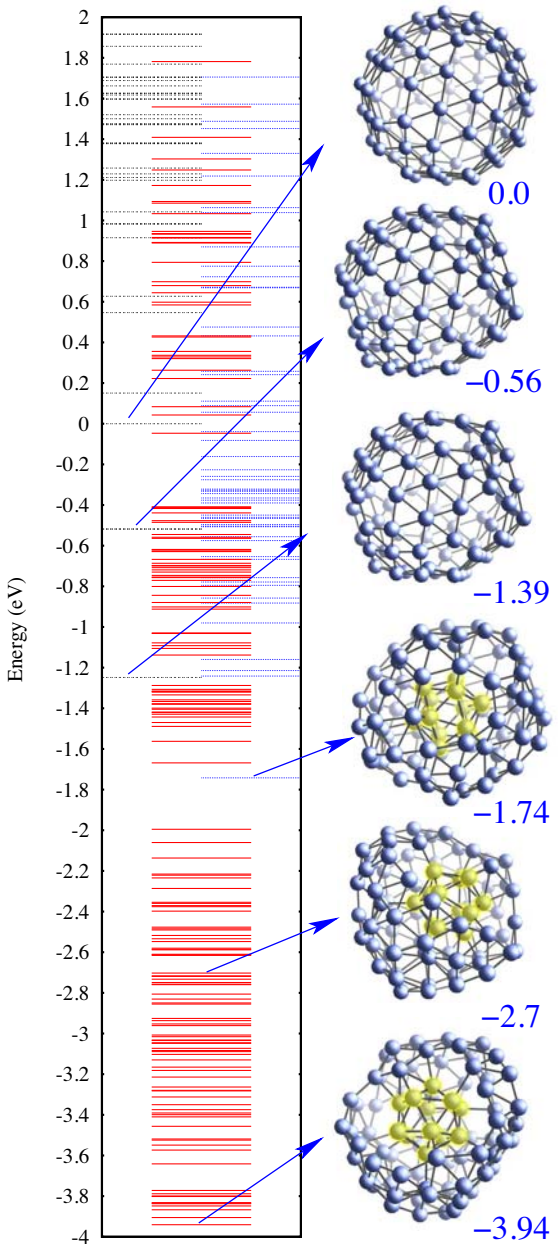
number of cores	serial	32*1	64*1	128*1	32*6=192	128*6=768	128*GPU
time in seconds (speedup)	180	7.2 (25)	3.8 (47)	2.0 (90)	1.5 (120)	.44 (410)	.30 (600)

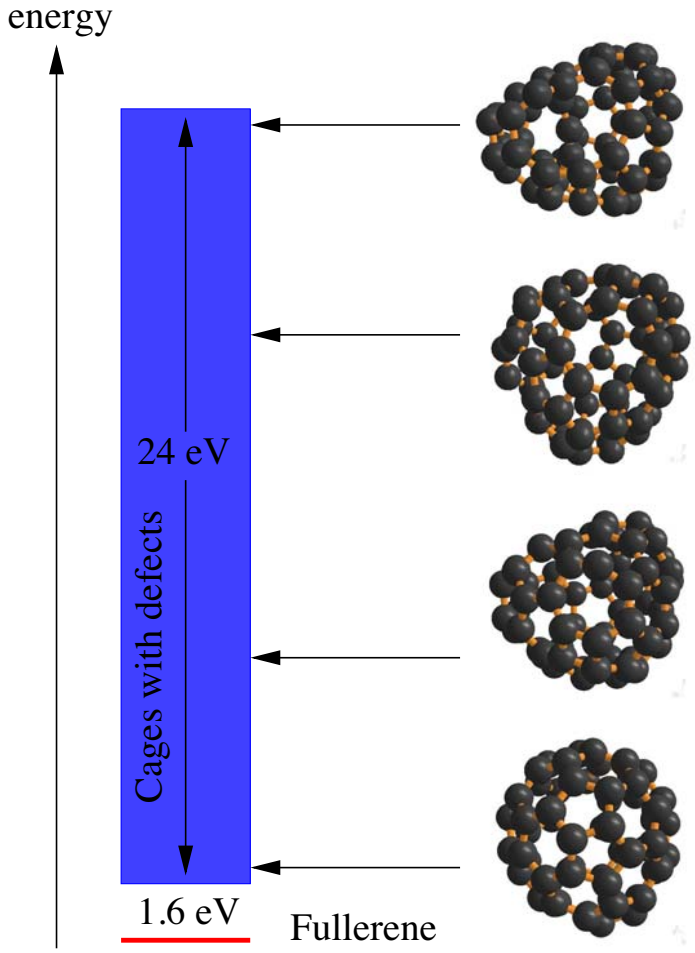
Capabilities of BigDFT

- Free, surface and periodic boundary conditions
- Local geometry optimizations (with constraints)
- Global geometry optimization
- Saddle point searches
- Vibrations
- Born Openheimer MD
- Excited states
- collinear and non-collinear magnetism
- All XC functionals of the LibXC library (LDA, GGA, hybrid functionals)
- Empirical van der Waals terms
- Also available from within the ABINIT package

Some applications of BigDFT

Global geometry optimization of large boron and carbon clusters





Studying the quality of force fields for proteins

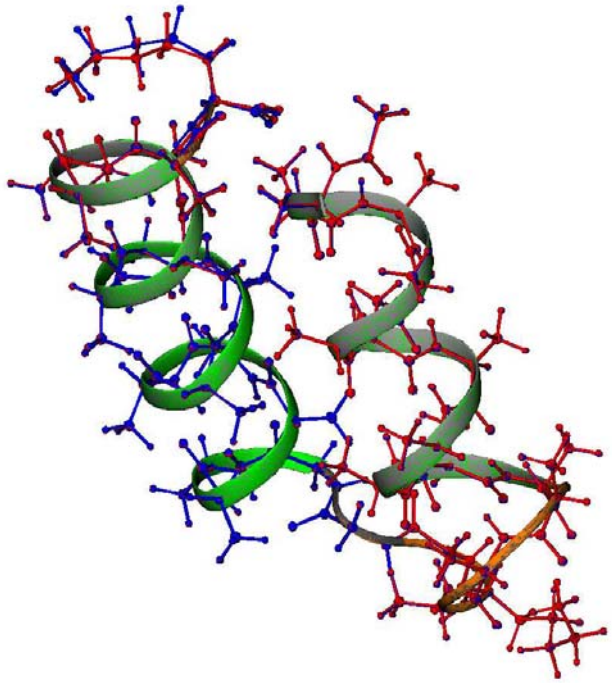
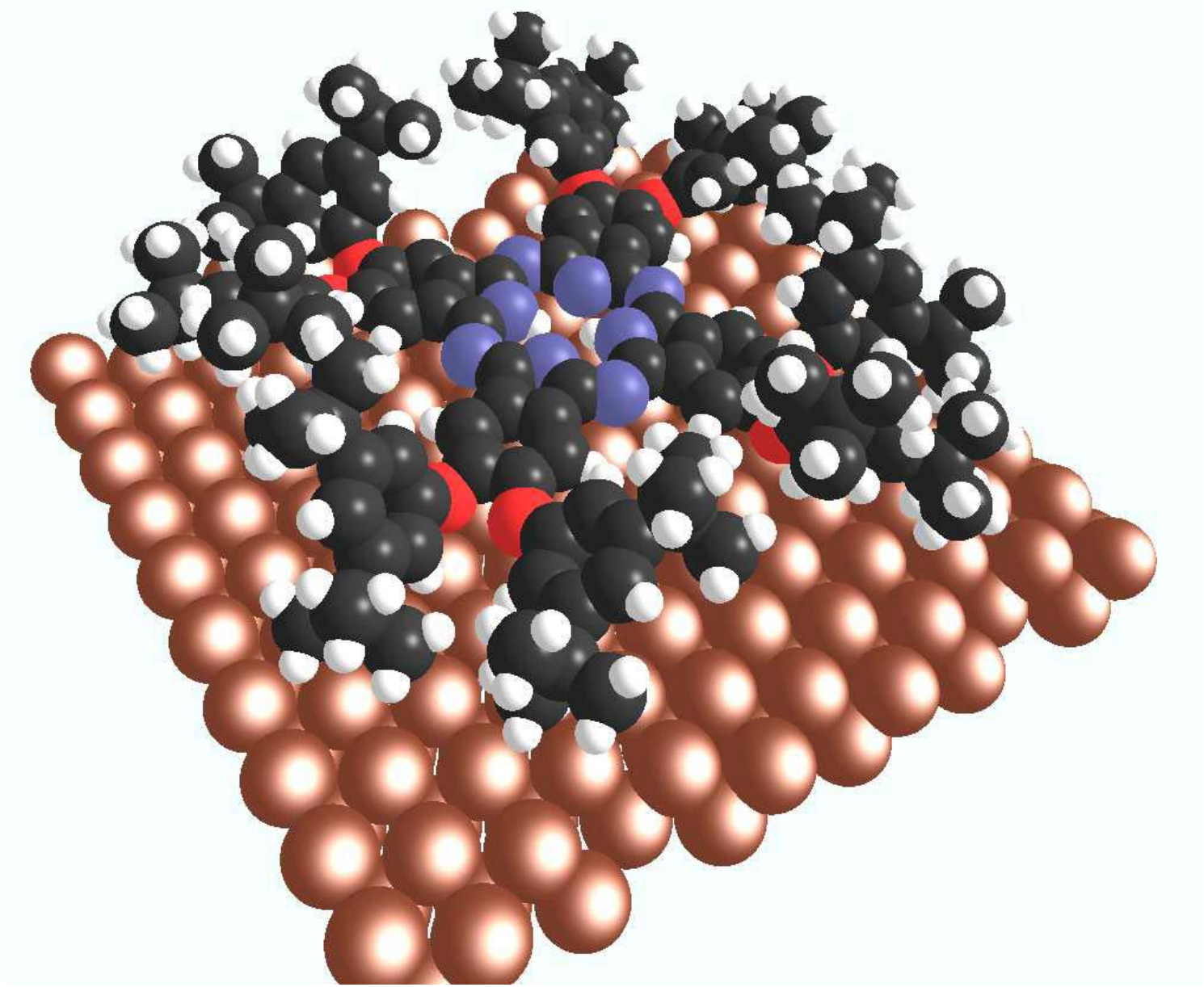


Figure 1: Two closely lying minima of OPLS forcefield having rms deviation $0.067\mathring{A}$ and energy difference $254.24 KJmol^{-1}$, which would eventually coalesce in the ab-initio geometry optimization(DFT-LDA).

Homo Lumo orbitals of a relaxed 350 atom molecule adsorbed on a Cu surface



Summary

The BigDFT electronic structure code:

- uses a systematic wavelet basis set which allows to converge rapidly to the correct result with arbitrary precision
- requires less memory and is faster than plane wave programs for molecular systems
- can calculate the electrostatic potential (Hartree potential) exactly for various boundary conditions
- gives very short time to solution on parallel machines due to its high parallel efficiency of either the MPI/OpenMP or MPI/CUDA implementation.

Main developers of version 1.4 (http://inac.cea.fr/L_Sim/BigDFT/):
Luigi Genovese, Stefan Goedecker, Damien Caliste, Brice Videau, Matthieu Ospici, Alessandro Mirone, Alexander Willand, Seyed Alireza Ghasemi, Max Amsler, Stephan Mohr, Thierry Deutsch, Eduardo Machado-Charry, Normand Mousseau