Large scale density functional electronic structure calculations in a systematic wavelet basis set

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

BIGDFT

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)

http://inac.cea.fr/L_Sim/BigDFT/

BigDFT is a program which solves the many-electron Schrödinger equation in the Kohn-Sham debsity functional approximation

We have to find a set of Kohn-Sham orbitals $\phi_i(\mathbf{r})$ wich minimizes the Kohn-Sham energy expression:

$$E = -\sum_{i=1}^{N} \frac{1}{2} \int \phi_i^*(\mathbf{r}) \nabla^2 \phi_i(\mathbf{r}) d\mathbf{r} + \int V_{en}(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \frac{\rho(\mathbf{r}) \rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' d\mathbf{r} + E_{xc}[\rho(\mathbf{r})] \quad (1)$$

where the charge density $\rho(\mathbf{r})$ is the sum over the square of all the occupied Kohn-Sham orbitals ϕ_i

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} \phi_i^*(\mathbf{r}) \phi_i(\mathbf{r}) \quad (2)$$

Need basis set $U_j(\mathbf{r})$

$$\phi_i(\mathbf{r}) = \sum_j C_{i,j} U_j(\mathbf{r})$$

A basis sets for electronic structure that combines the advantages of plane waves and Gaussians

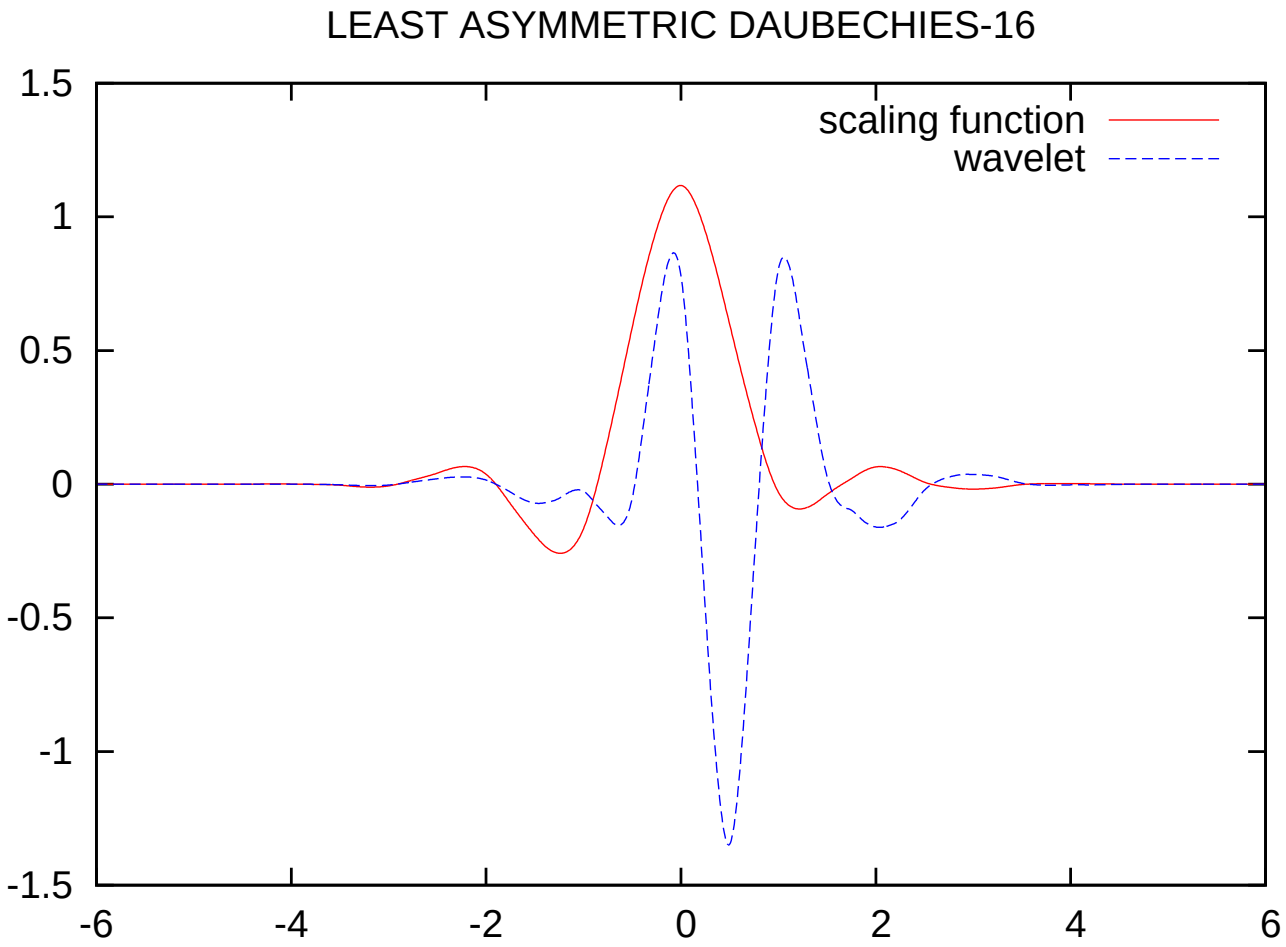
Plane waves:

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

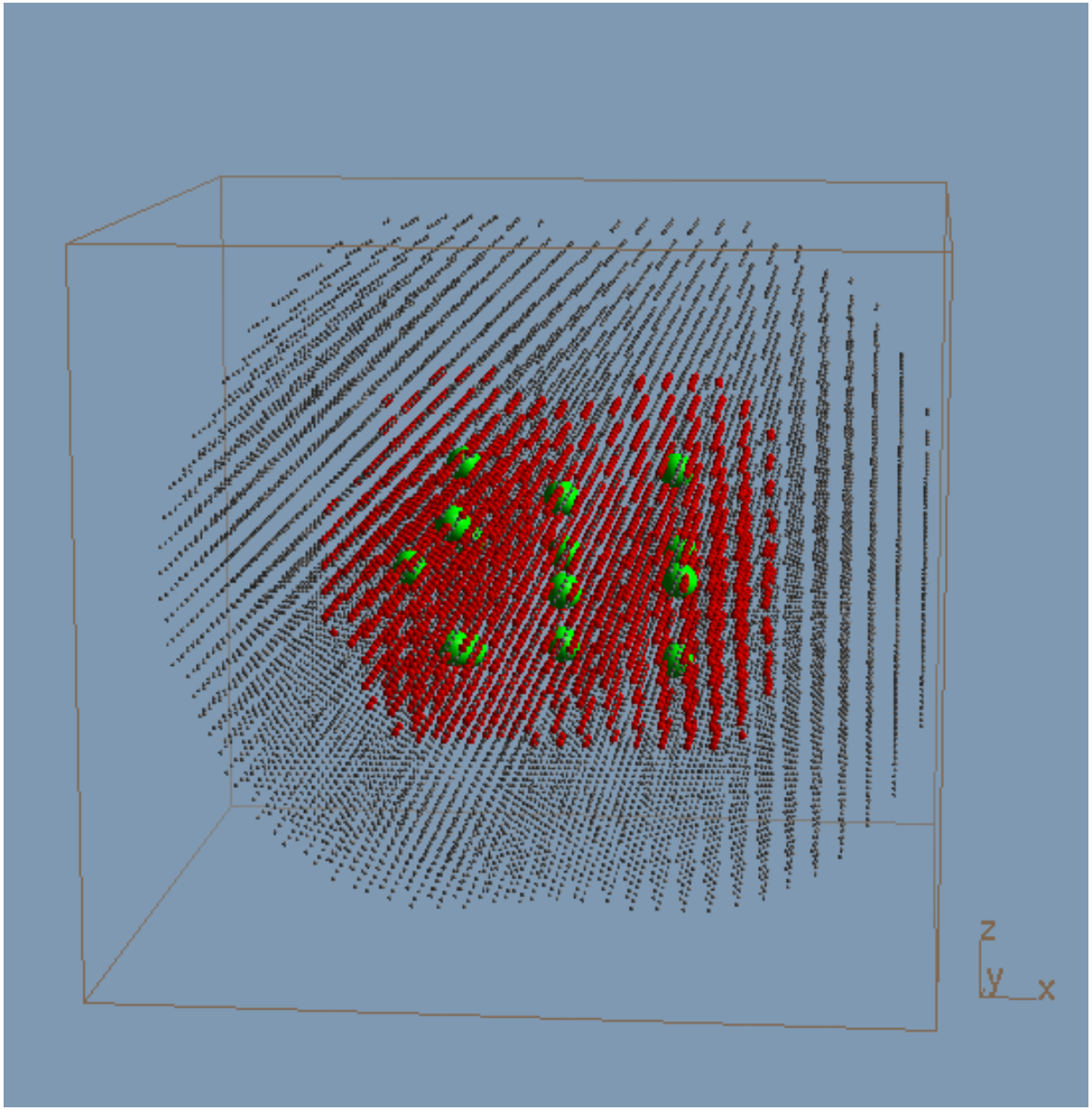
Gaussians:

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

High order Daubechies wavelets are used to represent wavefunctions



Only 2 resolution levels are used

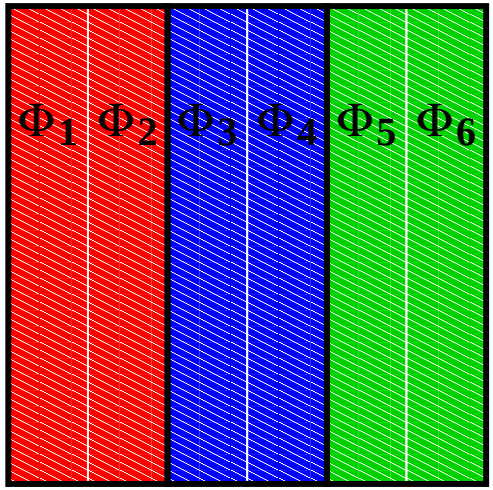


Capabilities of BigDFT (GNU GPL code in the ABINIT family)

- 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 (LDA, GGA, hybrid functionals) of the LIBXC library
- Can also be called from within the ABINIT package

MPI parallelization scheme

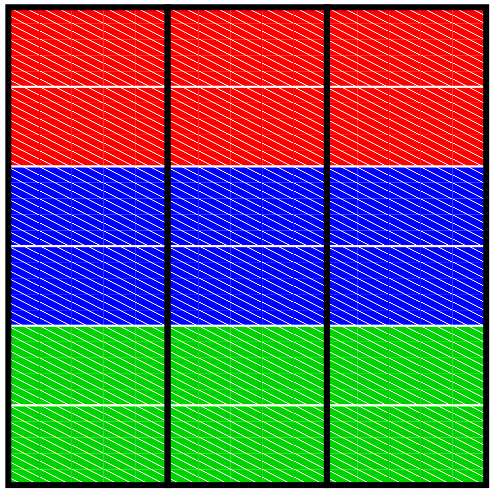
Only 2 (3) global communication steps are required per iteration in the wavefunction optimization for the orbitals



Data distribution scheme for Hamiltonian application

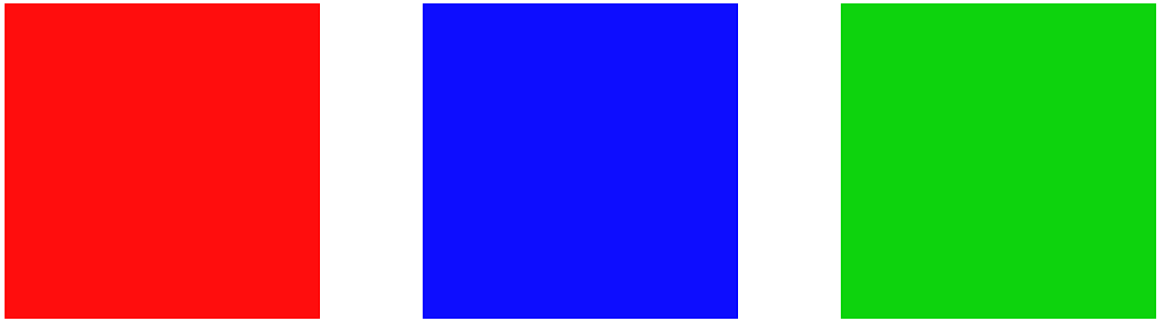
P1 P2 P3

MPI_ALLTOALL

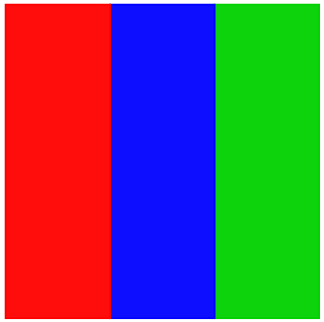


Data distribution scheme for orthogonalization

1 communication step for the charge density



MPI_ReduceScatter



MPI speedup

- Communication part is not latency limited, but only limited by the bisectional bandwidth
- Programming is much easier since most routines are just serial

About 10 percent of the elapsed time goes into communication if one MPI process treats one orbital on a computer with a very fast network or if one MPI process treats 2 orbitals on a Linux cluster.
Speedups of a few hundred can be achieved

Basic types of operations

- 3-dim convolutions: application of the kinetic and potential energy operators on the Kohn–Sham orbitals, preconditioning

$$y_{i1,i2,i3} = \sum_l c_l y_{i1+l,i2,i3}$$

$$y_{i1,i2,i3} = \sum_l c_l y_{i1,i2+l,i3}$$

$$y_{i1,i2,i3} = \sum_l c_l y_{i1,i2,i3+l}$$

  Can be parallelized with OpenMP, CUDA

  - OpenMP speedup: 3 to 4.5 on 6 cores
  - CUDA speedup: 20-30

- Dense linear algebra: Orthogonalization, cubic scaling
- Sparse linear algebra: Nonlocal part of pseudopotential

Advantages of GPUs

- High degree of parallelism: 240-512 cores
- High hardware memory bandwidth by the use of GDDR memory
- Good use of the bandwidth by latency hiding through thread switching
- Reduced instruction data traffic by the use of kernels

Questions concerning GPUs

- Is part of the performance gain only due to the fact that CUDA gives more control at the hardware level?
- Is programming in CUDA a good long term investment?
- Which features of GPUs will soon go into standard CPUs.

Main goals of the project

- Improve the strong scaling of the BigDFT code: More multi-threaded routines
- Improve the weak scaling of the BigDFT code in the context of a linear scaling algorithm
- Add an additional parallelization level by calling BigDFT from a parallel global optimization algorithm (Minima Hopping)

Expected obstacles

- Bandwidth of the network connecting different nodes
- Bisectional bandwidth does not scale with the number of nodes on most machines
- Efficiency of MPI algorithms/libraries
- Which programming language will prevail (Cuda C, Cuda Fortran, Coarray Fortran, OpenCL)
- Programming effort
- Reliability of compilers/software, fault tolerance of hardware

The team

- Stefan Goedecker (Computational physics, university of Basel)
- 1 PhD student (probably Stephan Mohr, master thesis student, physics, Basel)
- 2 Postdocs to be hired

Hardware

- 12 node cluster (each node has 2 Tesla cards)
- Various workstations equipped with GPU’s