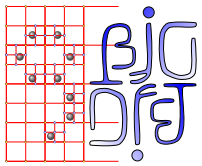


BigDFT

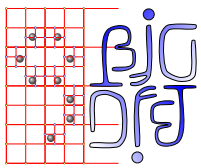
Linear Scaling DFT based on Daubechies Wavelets for Massively Parallel Archiecture

L Ratcliff[1], Stephan Mohr[1,2], Paul Boulanger[1,3], Luigi Genovese[1], Damien Caliste[1], Stefan Goedecker[2], Thierry Deutsch

[1]Laboratoire de Simulation Atomistique (L_Sim), CEA Grenoble
[2]Institut für Physik, Universität Basel, Switzerland
[3]Institut Néel, CNRS, Grenoble

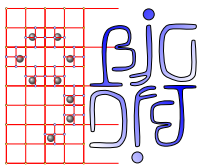
Centre Blaise Pascal, Lyon

Outline



BigDFT

A basis for nanosciences: the BigDFT project

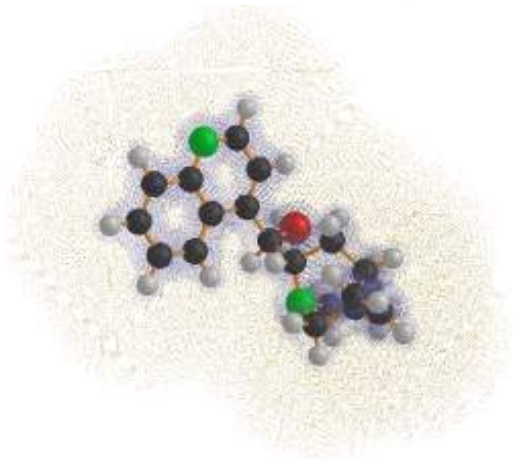


BigDFT

STREP European project: BigDFT(2005-2008)

Four partners, 15 contributors:
CEA-INAC Grenoble (T.Deutsch), U. Basel (S.Goedecker),
U. Louvain-la-Neuve (X.Gonze), U. Kiel (R.Schneider)

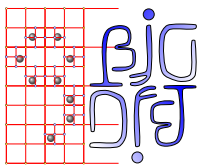
Aim: To develop an ab-initio DFT code based on Daubechies Wavelets, to be *integrated in ABINIT*, distributed under GNU-GPL license



After six years, BigDFT project still alive and well

- BigDFT formalism from HPC perspective
- Wavelets for $O(N)$ DFT
- Resonant states of open quantum systems

Why do we use wavelets in BigDFT?

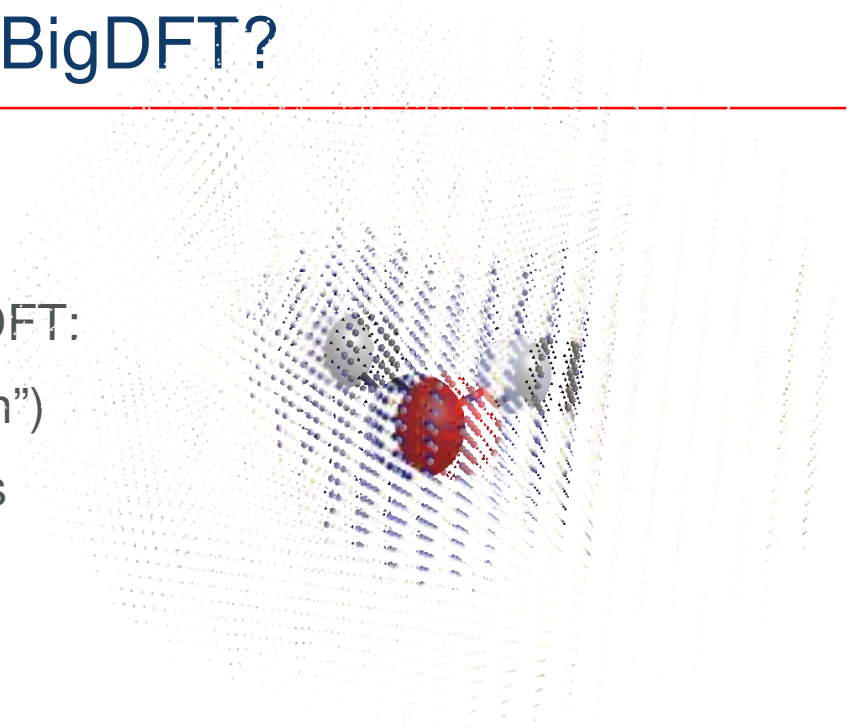


BigDFT

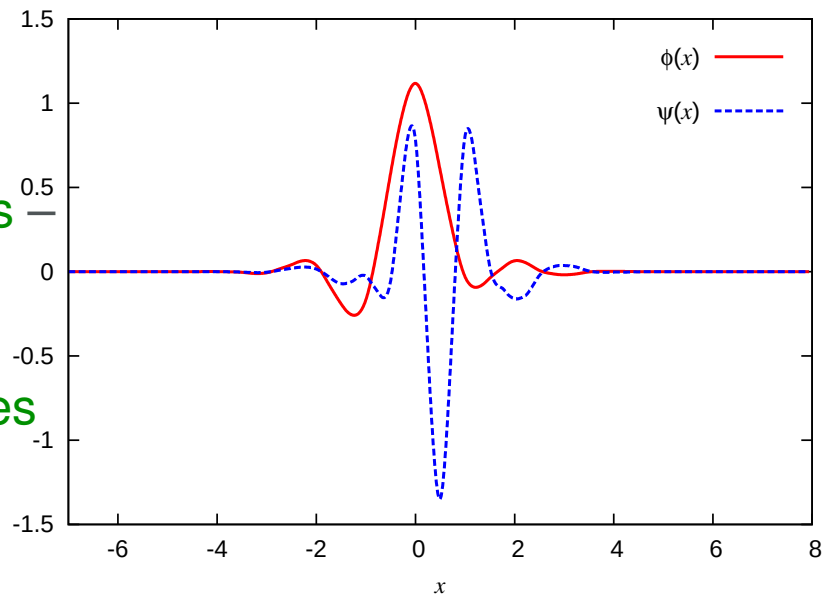
Adaptivity

One grid, two resolution levels in BigDFT:

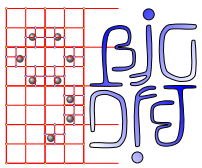
- 1 scaling function ("coarse region")
- 1 scaling function and 7 wavelets ("fine region")



Ideal for big inhomogeneous systems
Efficient Poisson solver, capable of handling different boundary conditions
free, wire, surface, periodic
Explicit treatment of charged systems
Established code with many capabilites



A brief description of wavelet theory



BigDFT

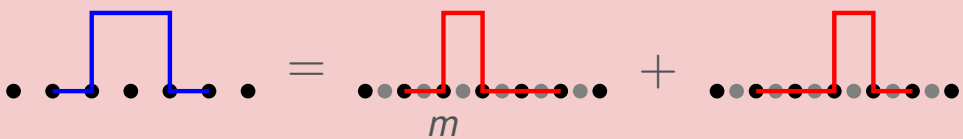
Two kind of basis functions

A Multi-Resolution real space basis

The functions can be classified following the resolution level they span.

Scaling Functions

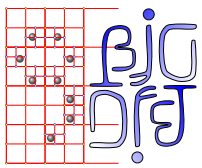
The functions of low resolution level are a linear combination of high-resolution functions



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

Centered on a resolution-dependent grid: $\phi_j = \phi_0(x - j)$.

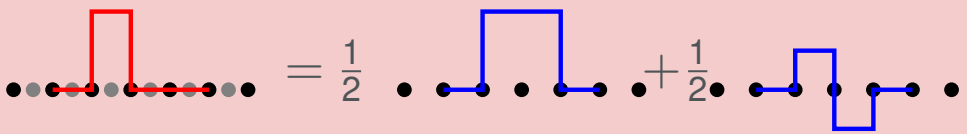
A brief description of wavelet theory



BigDFT

Wavelets

They contain the DoF needed to complete the information which is lacking due to the coarseness of the resolution.



$$\phi(2x) = \sum_{j=-m}^{m} \tilde{h}_j \phi(x-j) + \sum_{j=-m}^{m} \tilde{g}_j \psi(x-j)$$

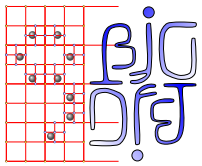
Increase the resolution without modifying grid space

SF + W = same DoF of SF of higher resolution

$$\psi(x) = \sum_{j=-m}^{m} g_j \phi(2x-j)$$

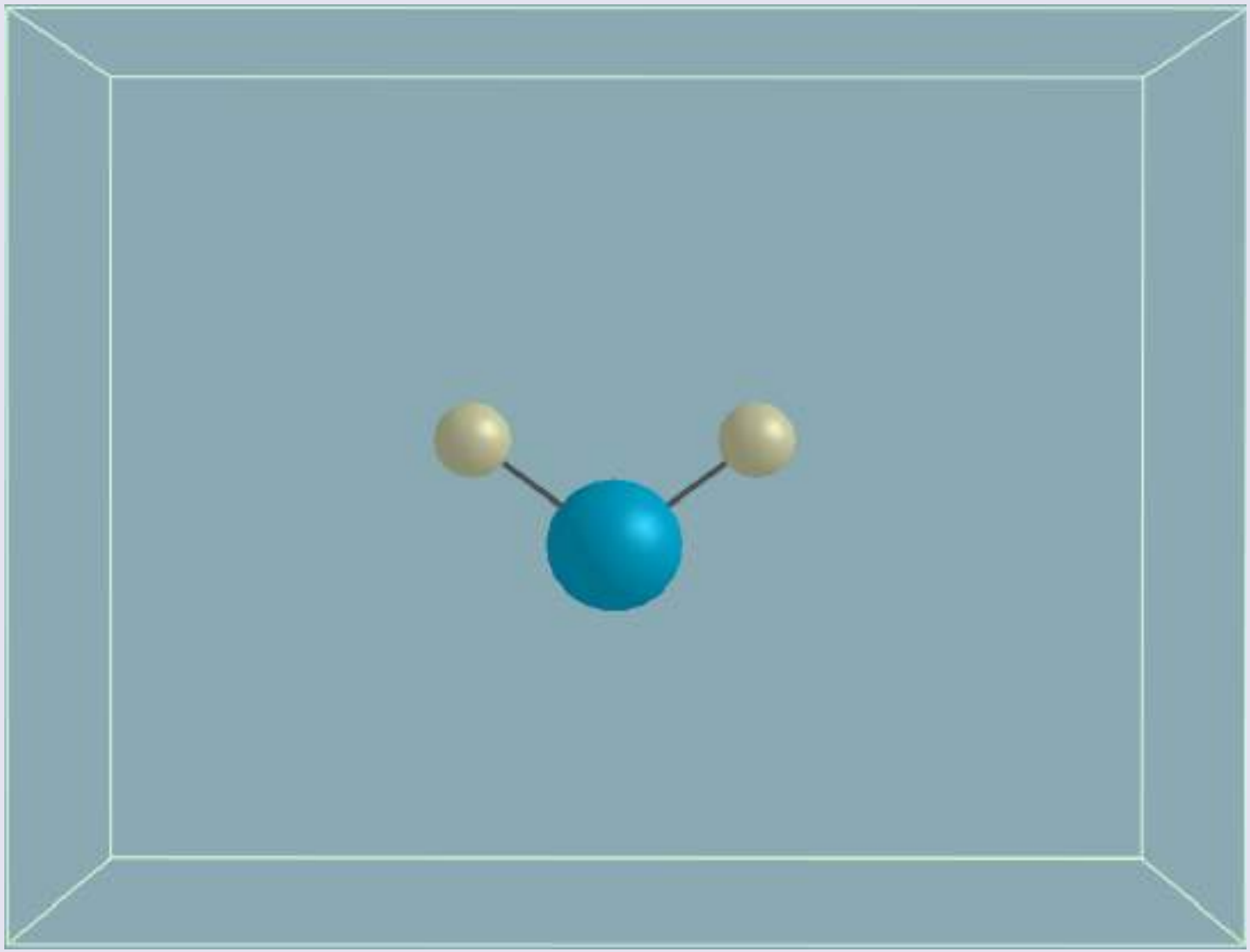
All functions have compact support, centered on grid points.

Adaptivity of the mesh

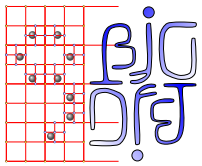


BigDFT

Atomic positions (H_2O)

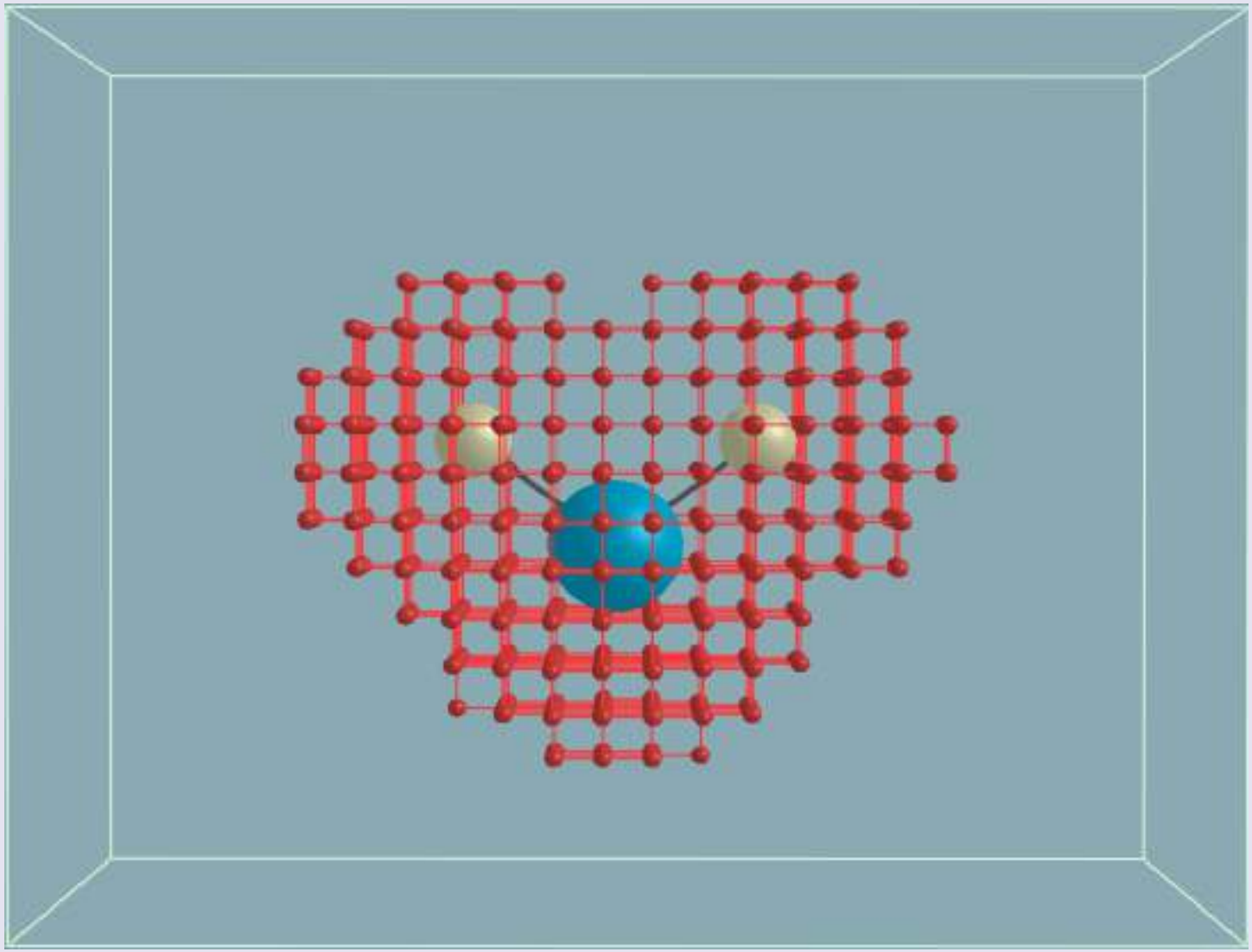


Adaptivity of the mesh

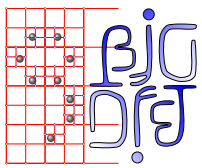


BigDFT

Fine grid (high resolution)

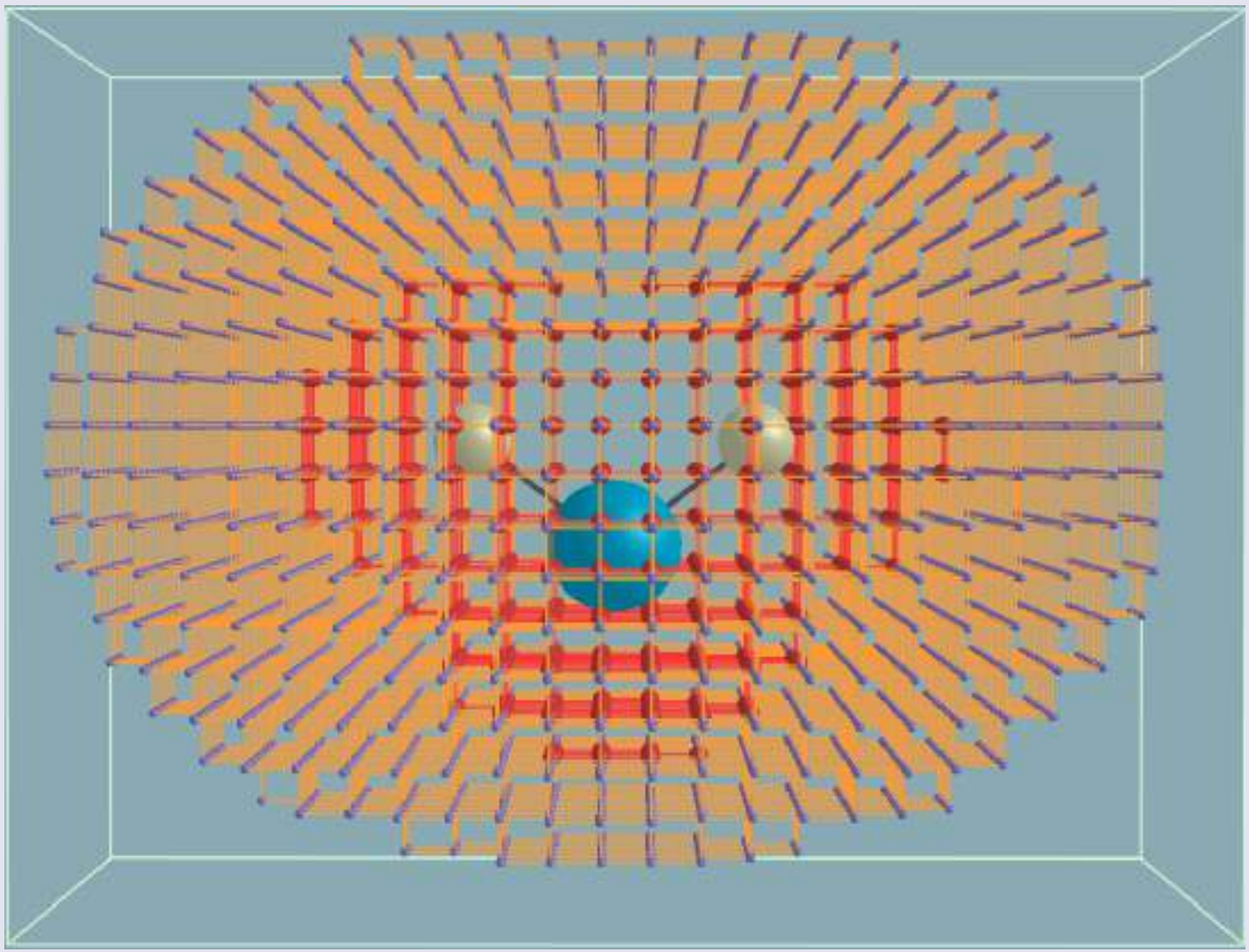


Adaptivity of the mesh

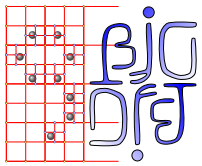


BigDFT

Coarse grid (low resolution)



Basis set features



BigDFT

Orthogonality, scaling relation

$$\int \mathrm{d}x\, \phi_k(x)\phi_j(x) = \delta_{kj} \qquad \phi(x) = \frac{1}{\sqrt{2}} \sum_{j=-m}^{m} h_j \phi(2x-j)$$

The hamiltonian-related quantities can be calculated *up to machine precision* in the given basis.

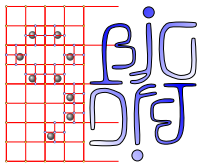
The accuracy is only limited by the basis set ($O\left(h_{\mathrm{grid}}^{14}\right)$)

Exact evaluation of kinetic energy

Obtained by convolution with filters:

$$f(x) = \sum_{\ell} c_\ell \phi_\ell(x)\,, \qquad \nabla^2 f(x) = \sum_{\ell} \tilde{c}_\ell\, \phi_\ell(x)\,,$$

$$\tilde{c}_\ell = \sum_{j} c_j\, a_{\ell-j}\,, \qquad a_\ell \equiv \int \phi_0(x)\partial_x^2 \phi_\ell(x)\,,$$

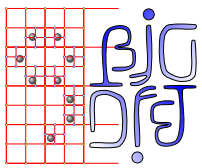


BigDFT

- Isolated, surfaces and 3D-periodic boundary conditions (k-points, symmetries)
- All XC functionals of the ABINIT package (libXC library)
- Hybrid functionals, Fock exchange operator
- Direct Minimisation and Mixing routines (metals)
- Local geometry optimizations (with constraints)
- External electric fields (surfaces BC)
- Born-Oppenheimer MD
- Vibrations
- Unoccupied states
- Empirical van der Waals interactions
- Saddle point searches (NEB, Granot & Bear)

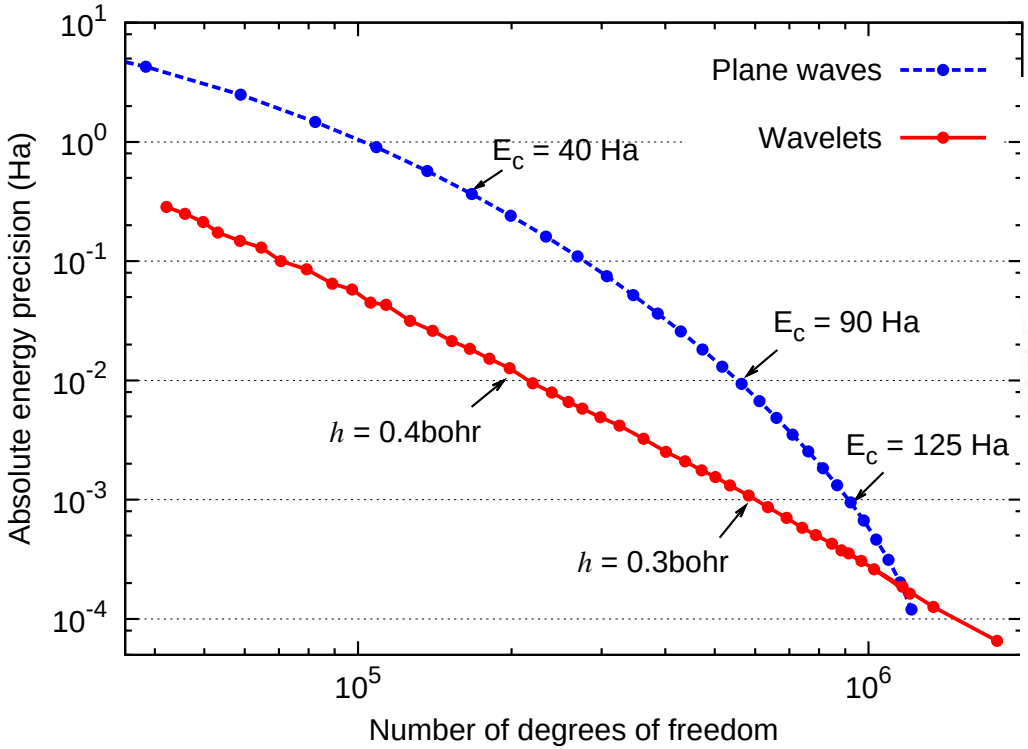
- All these functionalities are GPU-compatible

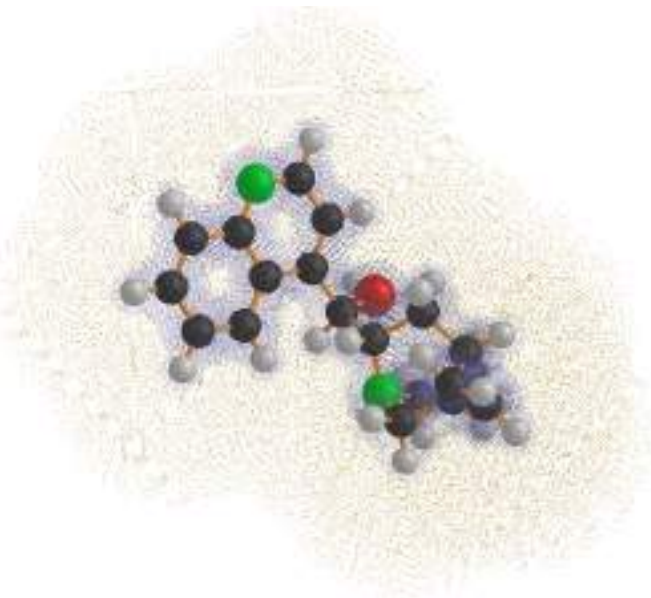
Optimal for isolated systems



BigDFT

Test case: cinchonidine molecule (44 atoms)





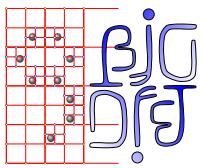
Allows a systematic approach for molecules

Considerably faster than Plane Waves codes.

10 (5) times faster than ABINIT (CPMD)

Charged systems can be treated *explicitly* with the same time

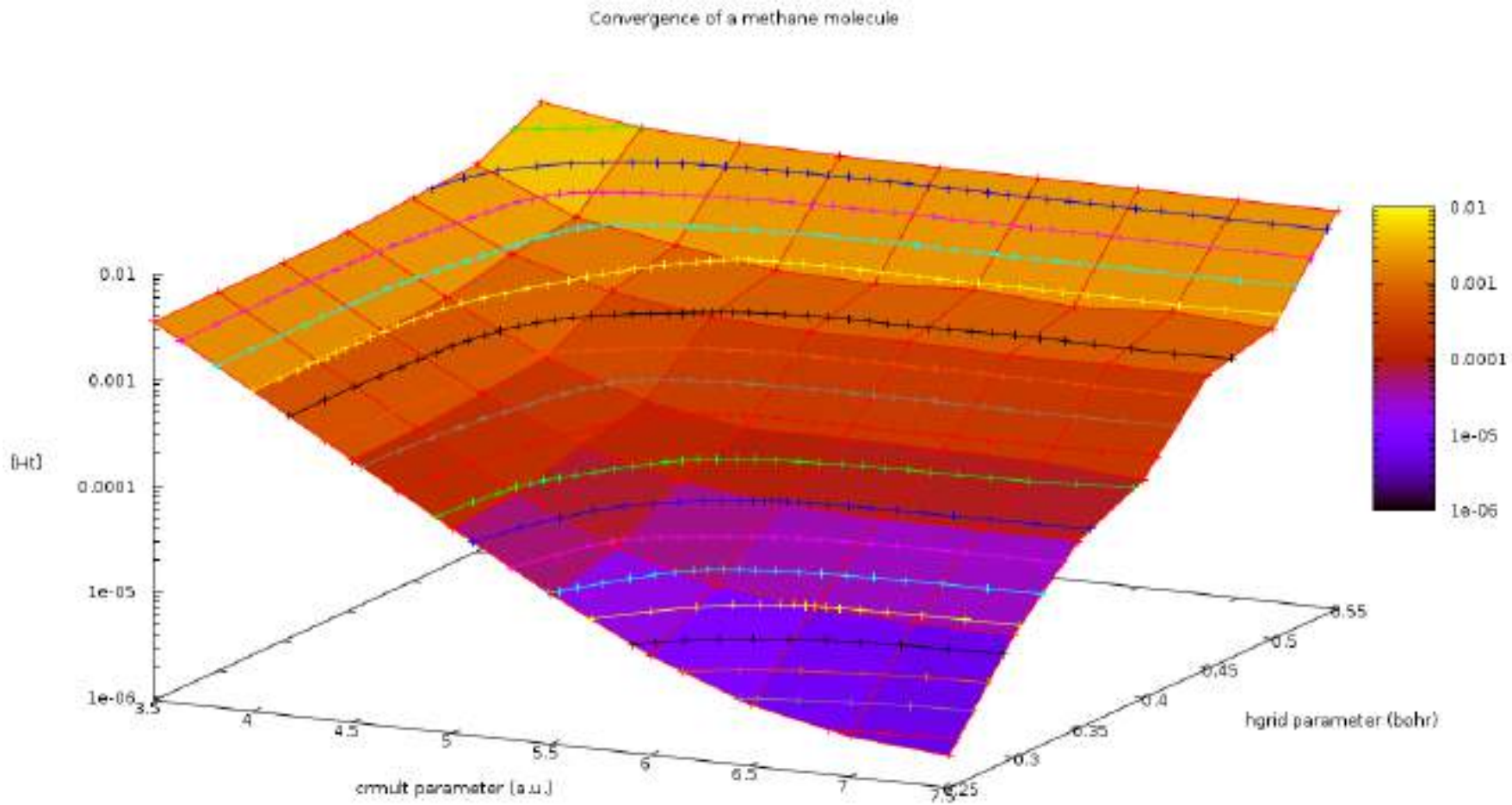
Systematic basis set



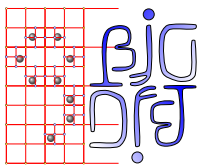
BigDFT

Two parameters for tuning the basis

- The grid spacing `hgrid`
- The extension of the low resolution points `crmult`



Massively parallel



BigDFT

Two kinds of parallelisation

- By orbitals (Hamiltonian application, preconditioning)
- By components (overlap matrices, orthogonalisation)

A few (but big) packets of data

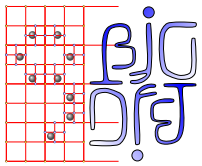
More demanding in bandwidth than in latency

- No need of fast network
- Optimal speedup (eff. $\sim$ 85%), also for big systems

Cubic scaling code

For systems bigger than 500 atomes (1500 orbitals) : orthonormalisation operation is predominant (N^3)

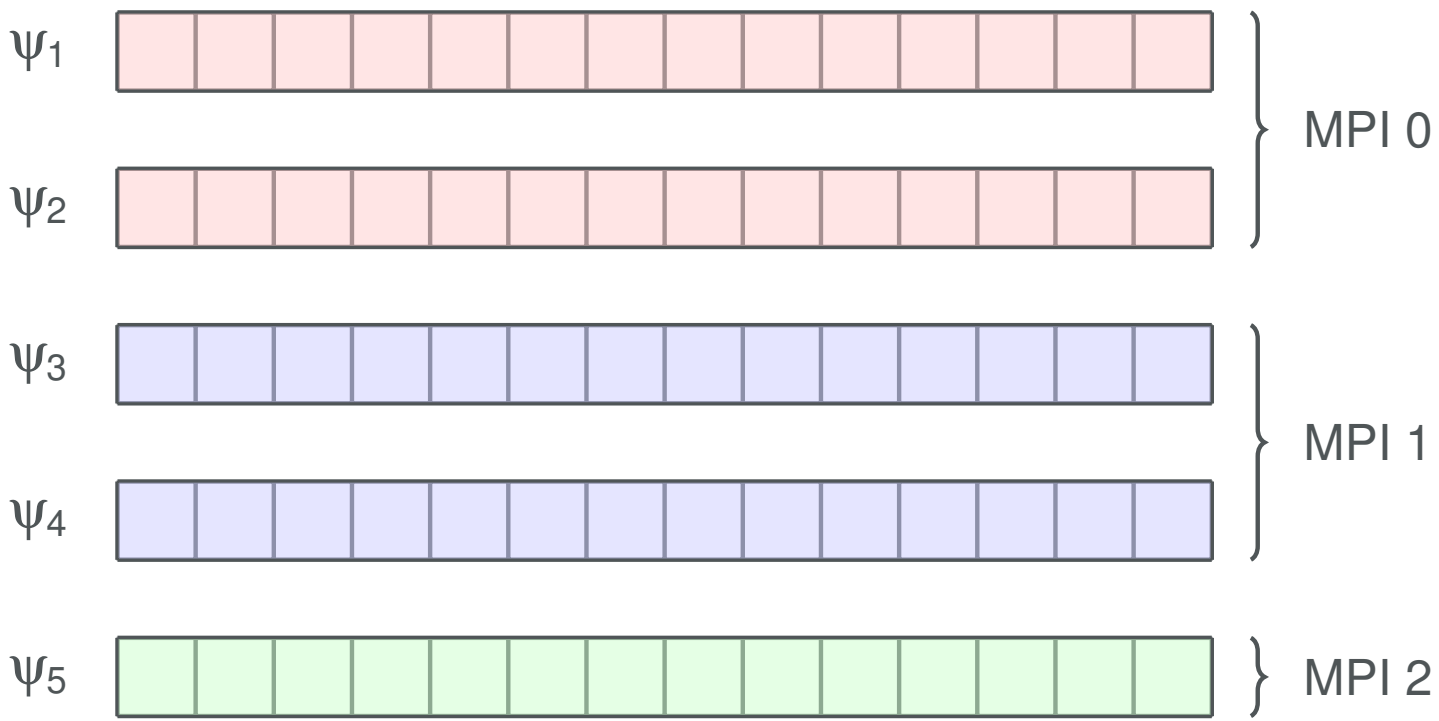
Orbital distribution scheme



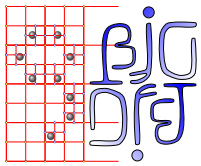
BigDFT

Used for the application of the hamiltonian

The hamiltonian (convolutions) is applied separately onto each wavefunction



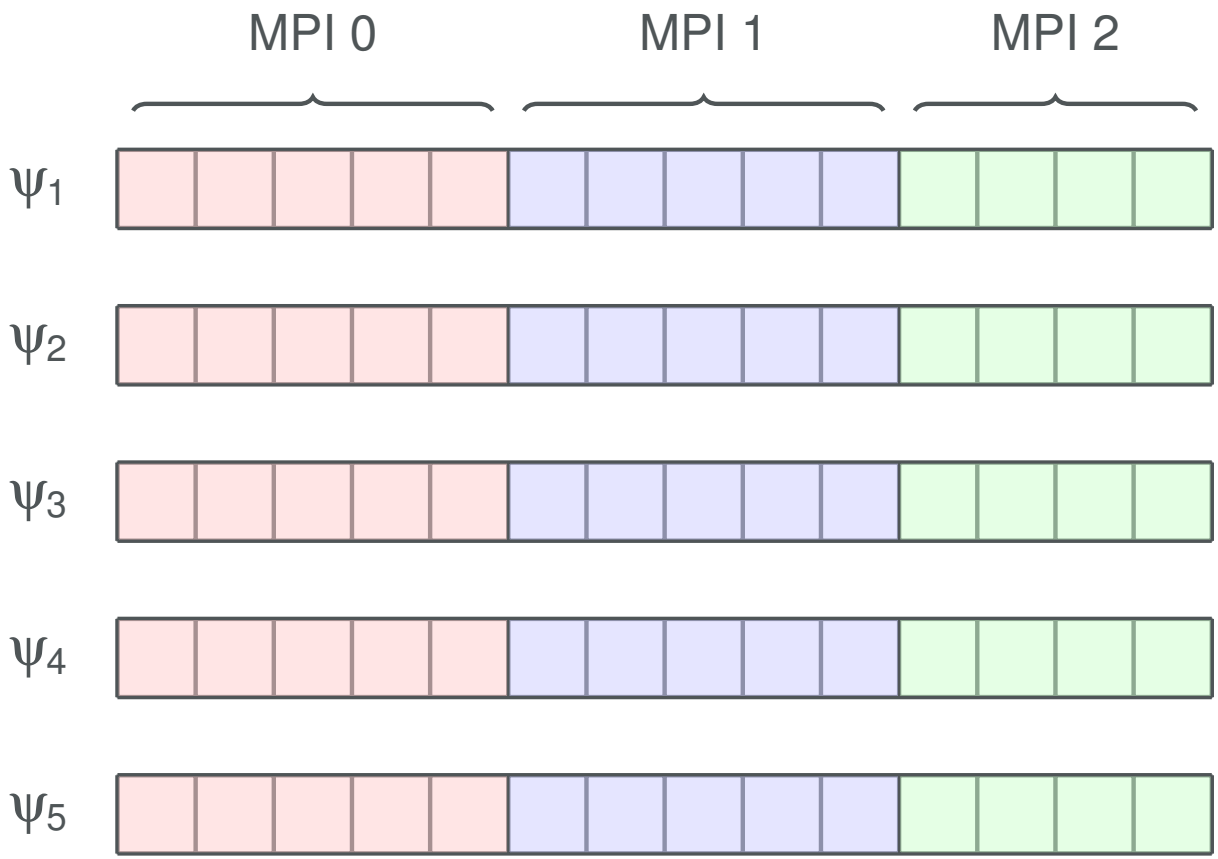
Coefficient distribution scheme



BigDFT

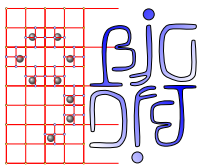
Used for scalar product & orthonormalisation

BLAS routines (level 3) are called, then result is reduced



Communications are performed via MPI_ALLTOALLV

OpenMP parallelisation



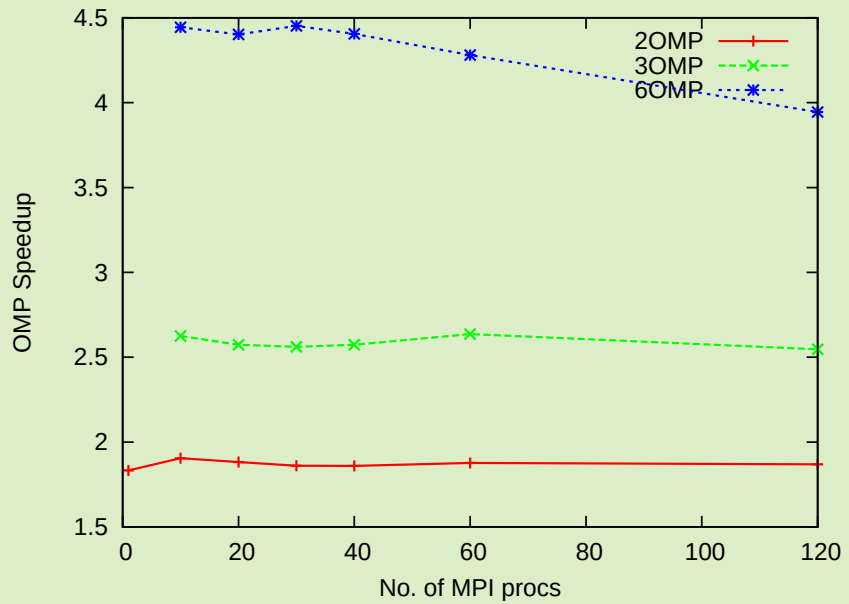
BigDFT

Innermost parallelisation level

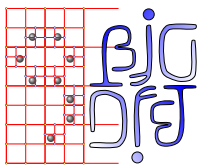
(Almost) Any BigDFT operation is parallelised via OpenMP

- ✔ Useful for memory demanding calculations
- ✔ Allows further increase of speedups
- ✔ Saves MPI processes and intra-node Message Passing

- ✖ Less efficient than MPI
- ✖ Compiler and system dependent
- ✖ OpenMP sections should be regularly maintained



High Performance Computing



BigDFT

Localisation & Orthogonality $\rightarrow$ Data locality

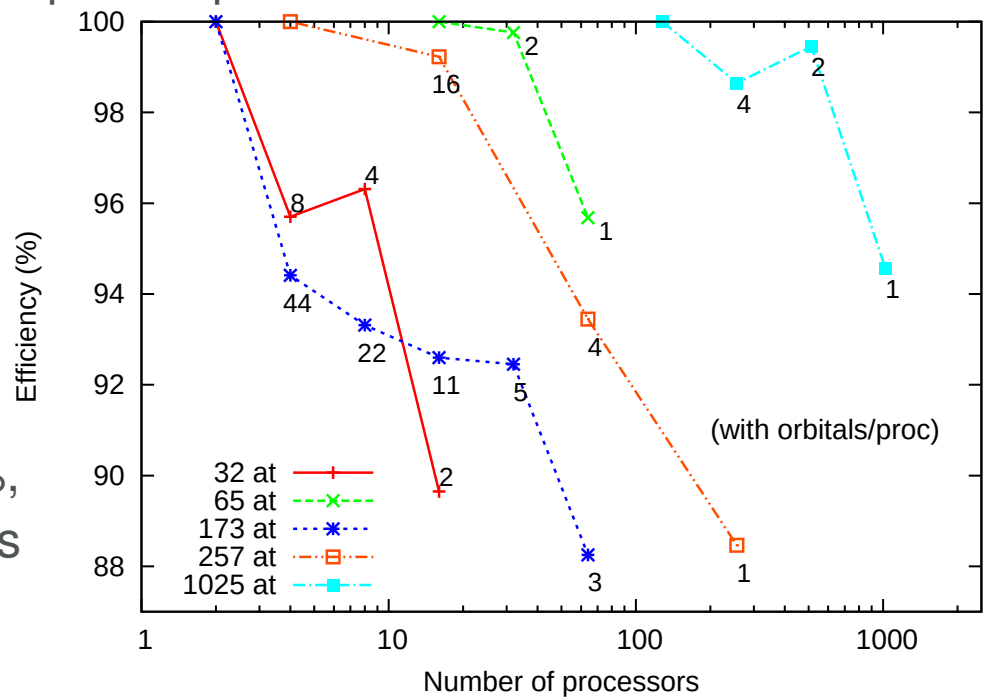
Principal code operations can be intensively optimised

Optimal for application on supercomputers

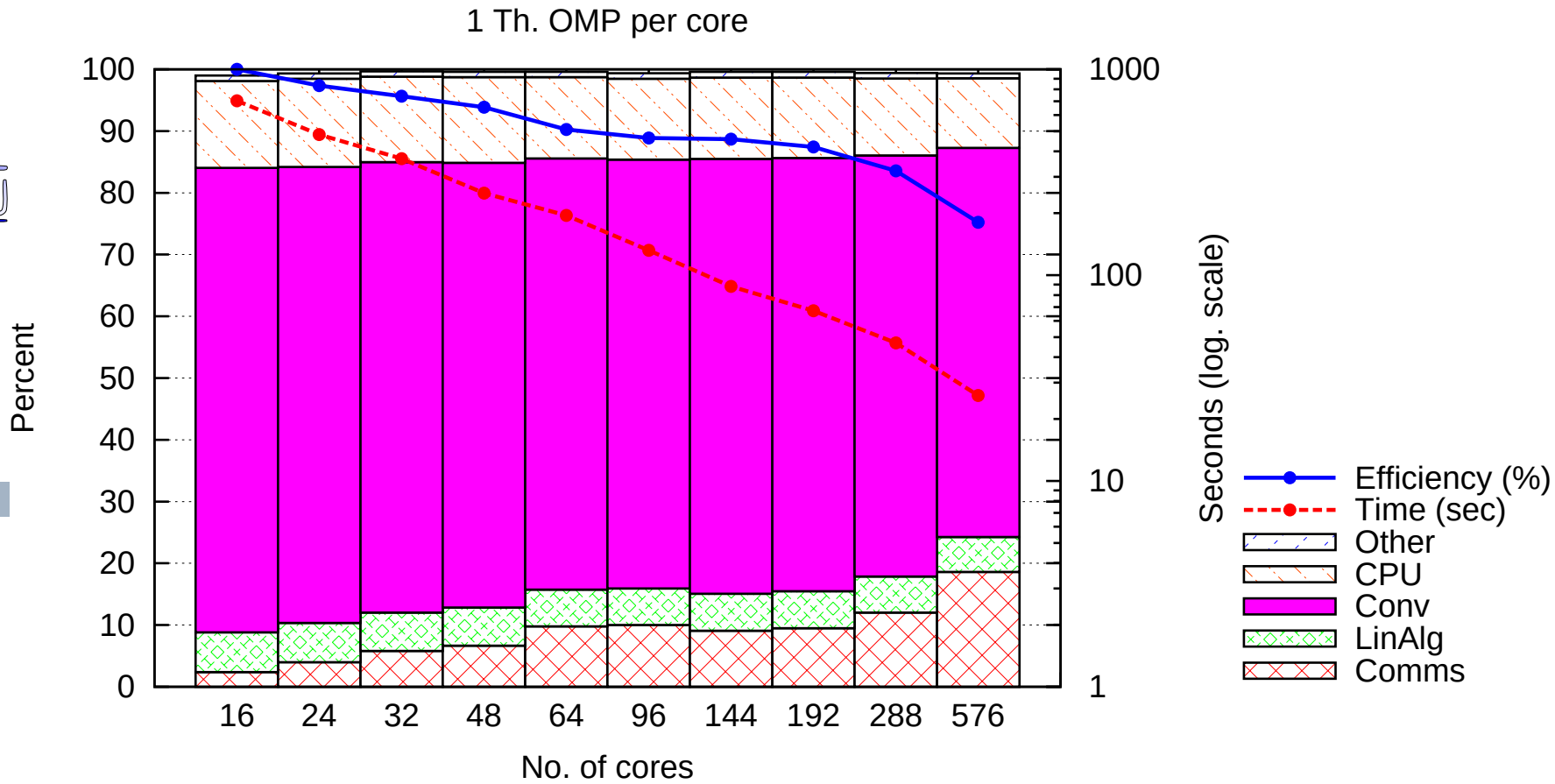
Little communication, big packets of data

- No need of fast network
- Optimal speedup

Efficiency of the order of 90%, up to thousands of processors



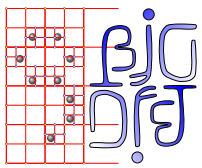
Task repartition for a small system (ZnO, 128 atoms)



Data repartition optimal for material accelerators (GPU)

Graphic Processing Units can be used to speed up the computation

Using GPUs in a Big systematic DFT code



BigDFT

Nature of the operations

- Operators approach via convolutions
- Linear Algebra due to orthogonality of the basis
- Communications and calculations do not interfere

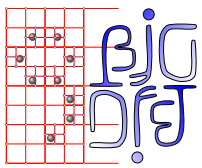
☛ A number of operations which can be accelerated

Evaluating GPU convenience

Three levels of evaluation

1. Bare speedups: GPU kernels *vs.* CPU routines
   Does the operations are suitable for GPU?
2. Full code speedup on one process
   Amdahl's law: are there hot-spot operations?
3. Speedup in a (massively?) parallel environment
   The MPI layer adds an extra level of complexity

Hybrid and Heterogeneous runs with OpenCL



BigDFT

NVidia S2070

Connected each to a Nehalem Workstation

BigDFT may run on both

ATI HD 6970

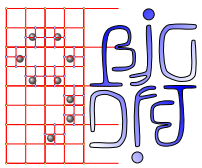
Sample BigDFT run: Graphene, 4 C atoms, 52 kpts

No. of Flop: $8.053 \cdot 10^{12}$

MPI	1	1	4	1	4	8
GPU	NO	NV	NV	ATI	ATI	NV + ATI
Time (s)	6020	300	160	347	197	109
Speedup	1	20.07	37.62	17.35	30.55	55.23
GFlop/s	1.34	26.84	50.33	23.2	40.87	73.87

Next Step: handling of Load (un)balancing

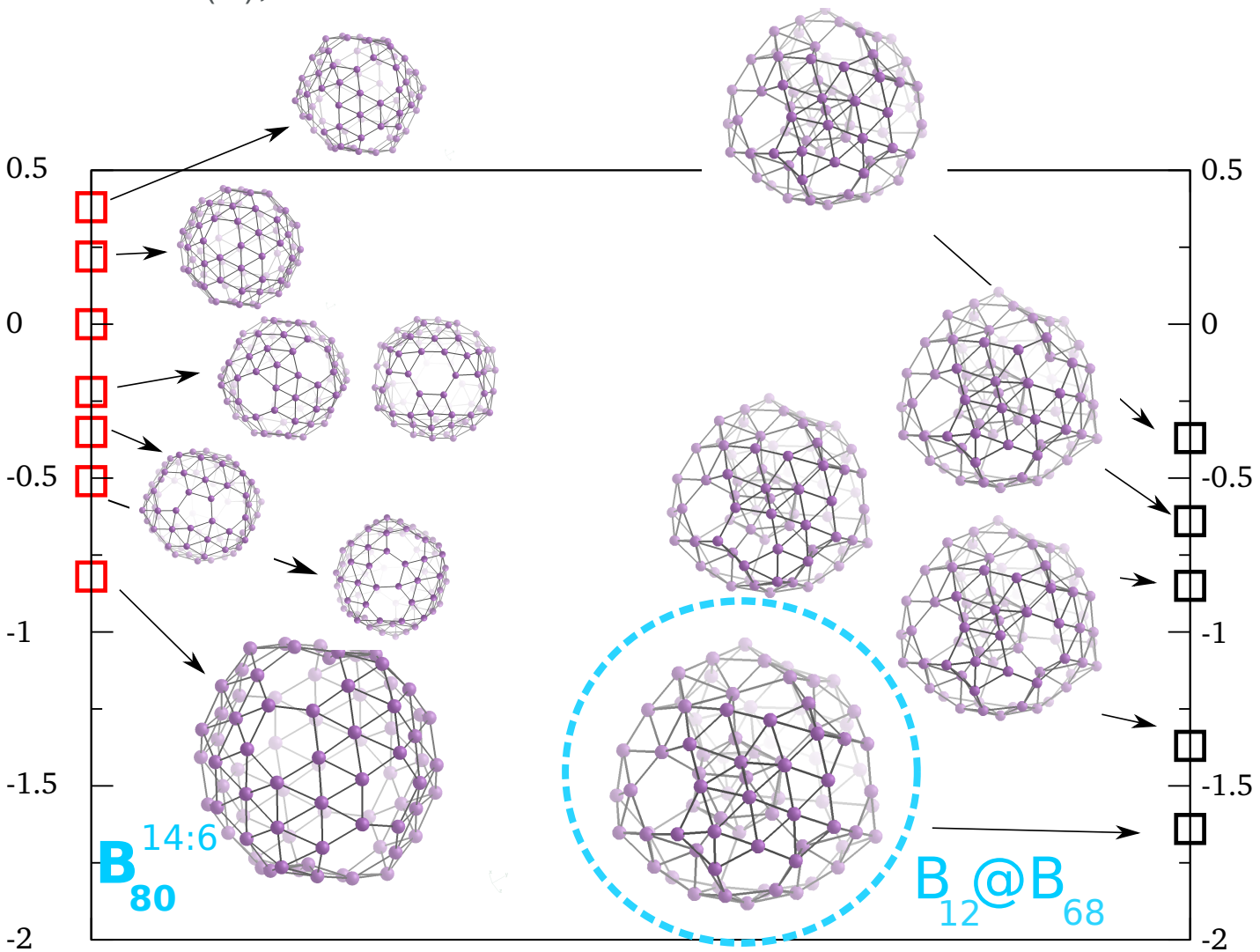
Configuration space of the cage-like boron clusters



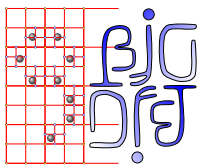
BigDFT

Stabilize the buckyball configuration of B_{80} systems

PRB 83 081403(R), 2011



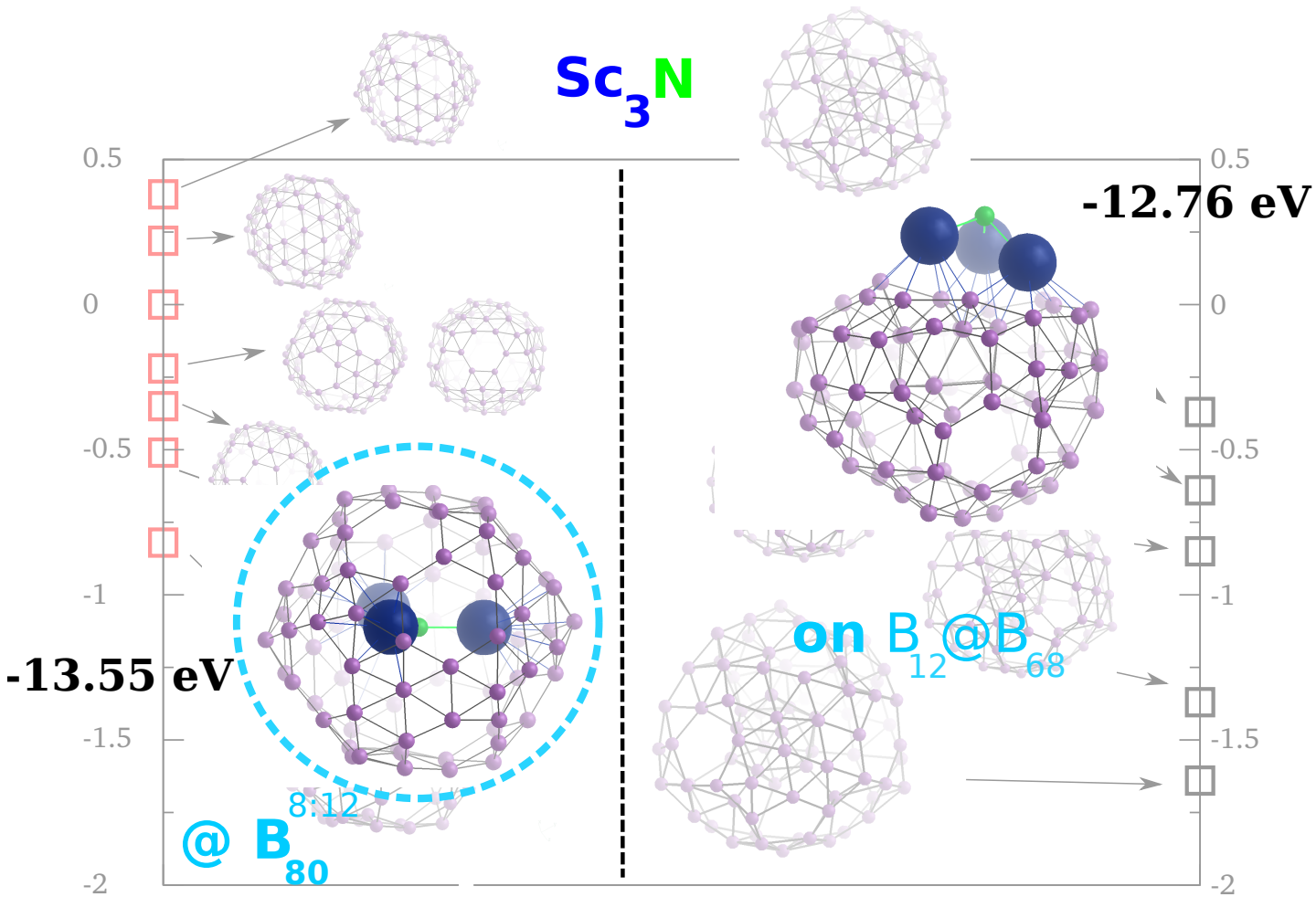
Configuration space of the cage-like boron clusters



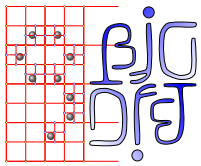
BigDFT

Stabilize the buckyball configuration of B_{80} systems

PRB 83 081403(R), 2011



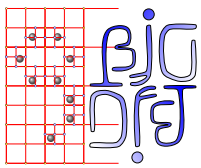
Outline



BigDFT

1. The machinery of BigDFT (cubic scaling version)
   - Mathematics of the wavelets
   - Main Operations
   - MPI/OpenMP and GPU performances
2. $\mathcal{O}(\mathcal{N})$ (linear scaling) BigDFT approach
   - Minimal basis set approach
   - Performance and Accuracy
   - Fragment approach
   - Applications
   - Perspectives
3. Conclusions

Scaling of BigDFT (I)



BigDFT

We can reach systems containing up to a few hundred electrons thanks to wavelet properties and efficient parallelization:

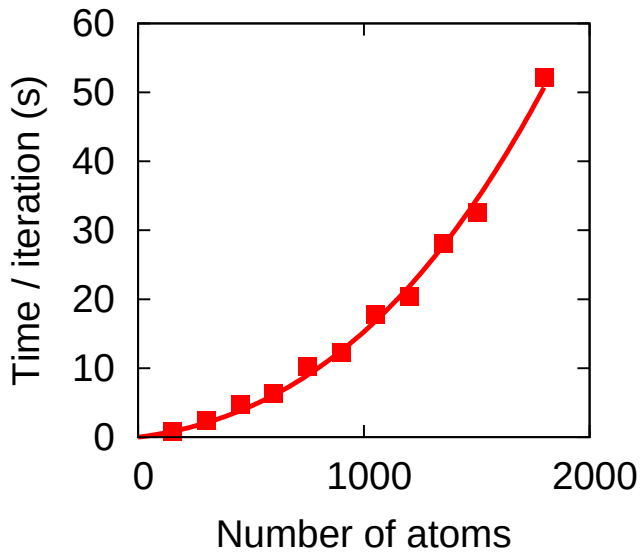
- MPI
- OpenMP
- GPU ported

But the various operations scale differently:

- $O(\mathcal{N}\log\mathcal{N})$: Poisson solver
- $O(\mathcal{N}^2)$: convolutions
- $O(\mathcal{N}^3)$: linear algebra

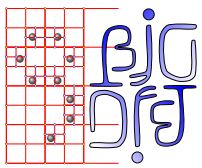
and have different prefactors:

- $c_{O(\mathcal{N}^3)} \ll c_{O(\mathcal{N}^2)} \ll c_{O(\mathcal{N}\log\mathcal{N})}$



$\longrightarrow$ for bigger systems the $O(\mathcal{N}^3)$ will dominate and so we need a new approach

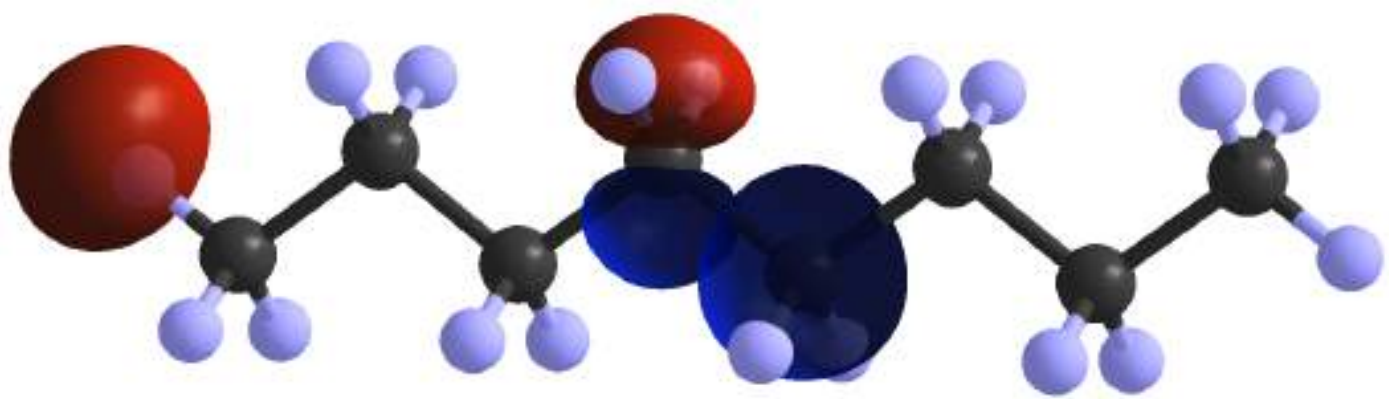
Scaling of BigDFT (II)



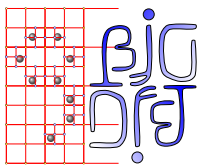
BigDFT

To improve the scaling and thus the scope of BigDFT, we must take advantage of the nearsightedness principle:

- the behaviour of large systems is short-ranged, or near-sighted
- the density matrix decays exponentially in insulating systems
- can take advantage of this locality to create linear-scaling $\mathcal{O}(\mathcal{N})$ DFT formalisms by having localized orbitals e.g. as in ONETEP, Conquest, SIESTA...
- we want to adapt this formalism for BigDFT



Minimal basis and density kernel



BigDFT

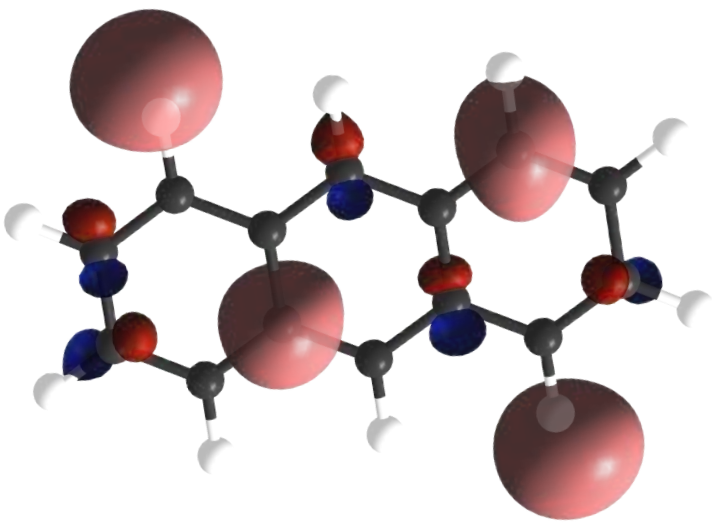
Write the KS orbitals as linear combinations of minimal basis set $\phi_\alpha(\mathbf{r})$:

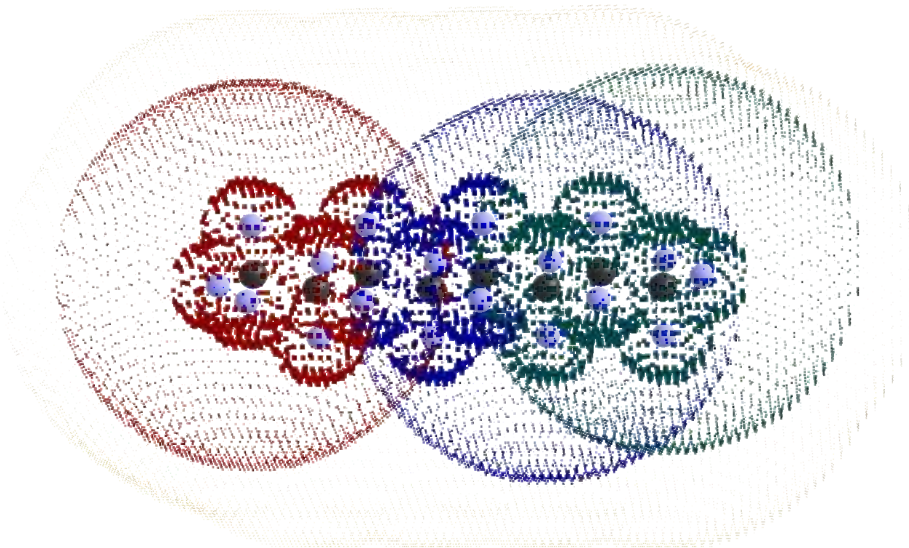
$$\Psi_i(\mathbf{r}) = \sum_\alpha c_i^\alpha \phi_\alpha(\mathbf{r})$$

- localized
- atom-centred
- expanded in wavelets

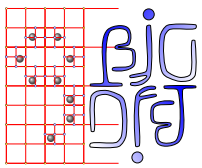
Define the density matrix $\rho(\mathbf{r},\mathbf{r}')$ and kernel $K^{\alpha\beta}$:

$$\begin{aligned}\rho(\mathbf{r},\mathbf{r}') &= \sum_i \left|\Psi_i(\mathbf{r})\right\rangle\left\langle\Psi_i(\mathbf{r}')\right| \\ &= \sum_{\alpha,\beta} \left|\phi_\alpha(\mathbf{r})\right\rangle K^{\alpha\beta}\left\langle\phi_\beta(\mathbf{r}')\right|\end{aligned}$$

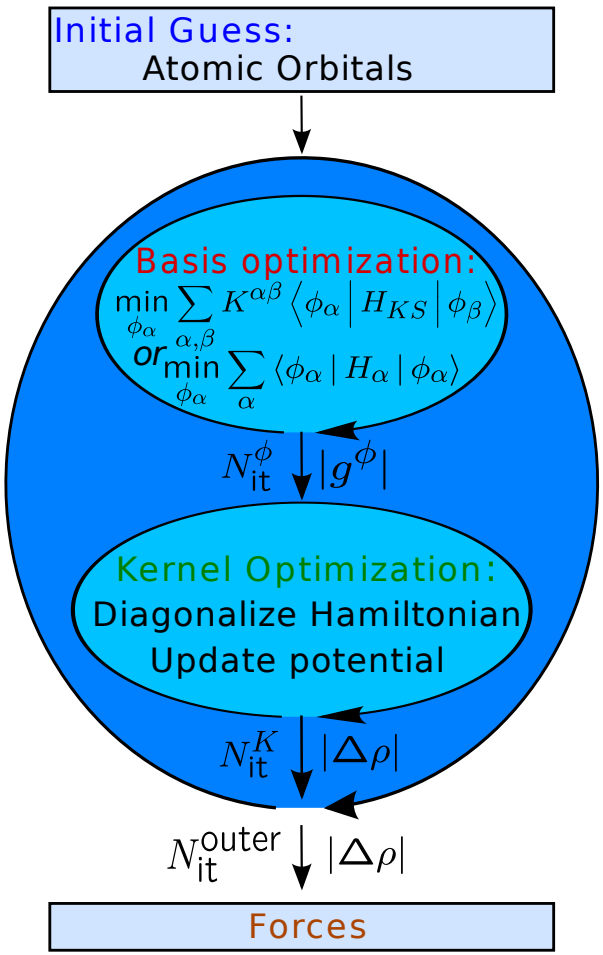




The algorithm



BigDFT



Minimal basis set

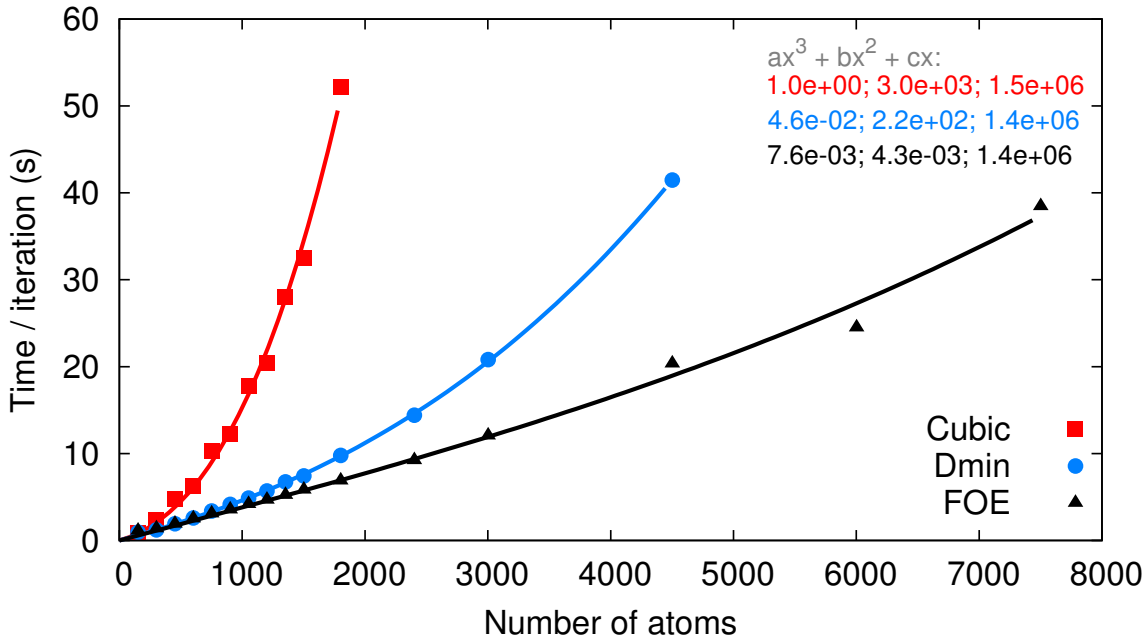
- Initialized to atomic orbitals, optimized using a quartic confining potential subject to an orthogonality constraint
- Minimize the 'trace', band structure energy or a combination at fixed potential
- SD or DIIS with k.e. preconditioning

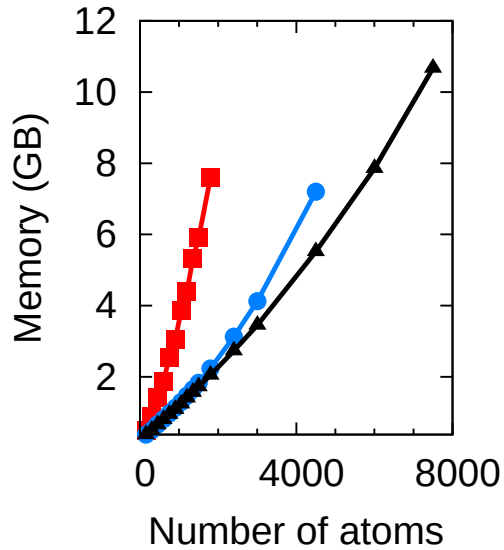
Density kernel

Three methods for self-consistent optimization:

- Diagonalization
- Direct minimization
- Fermi operator expansion (linear-scaling)

Scaling

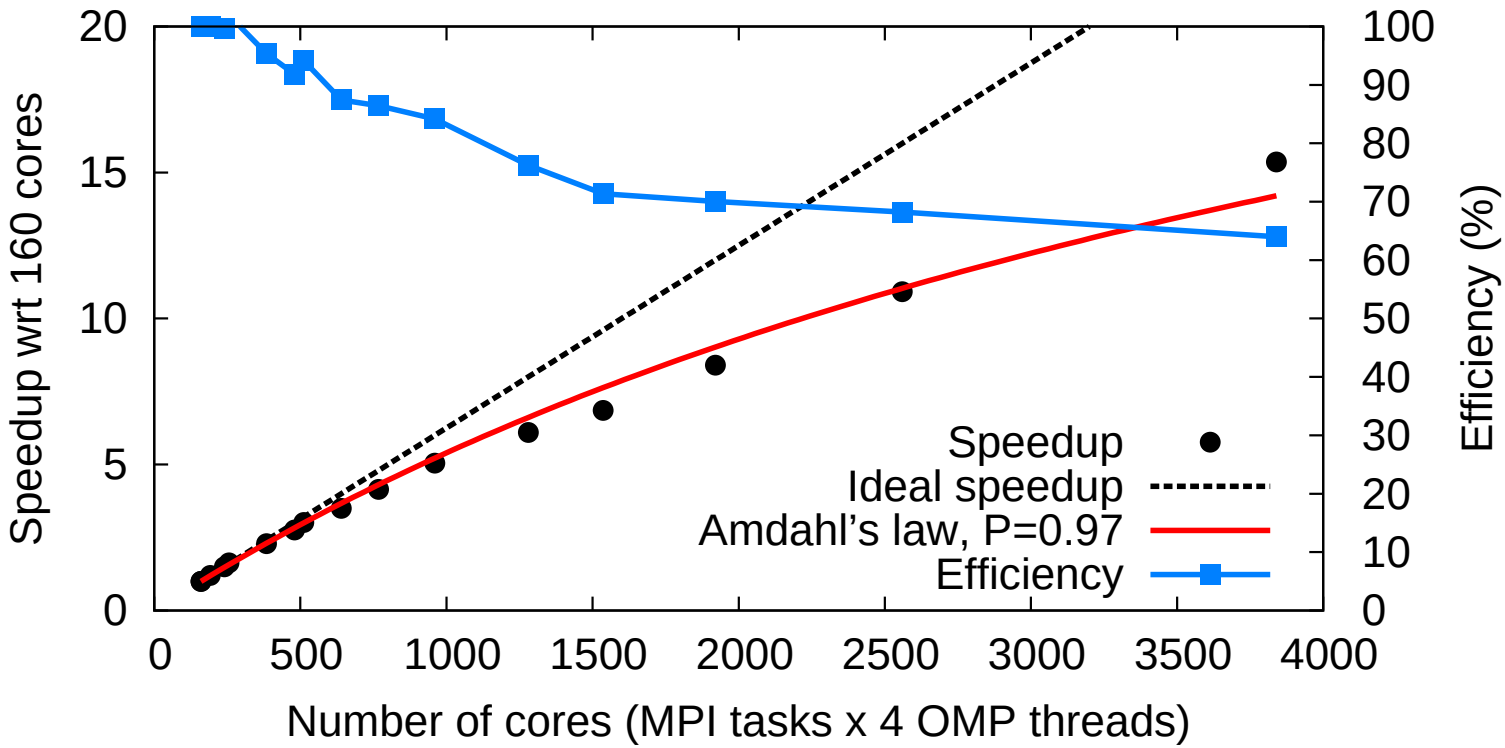




Improved time and memory scaling

- 301 MPI, 8 OpenMP for above results
- $\sim \mathcal{O}(\mathcal{N})$ for FOE
- need sparse matrix algebra for direct minimisation $\rightarrow \mathcal{O}(\mathcal{N})$

Parallelization – MPI

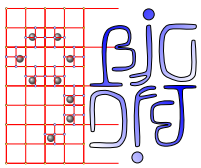


Efficient parallelization for 1000s of CPUs

- $\sim 97\%$ of code parallelized with MPI
- $\sim 93\%$ of code parallelized with OpenMP

960 atoms

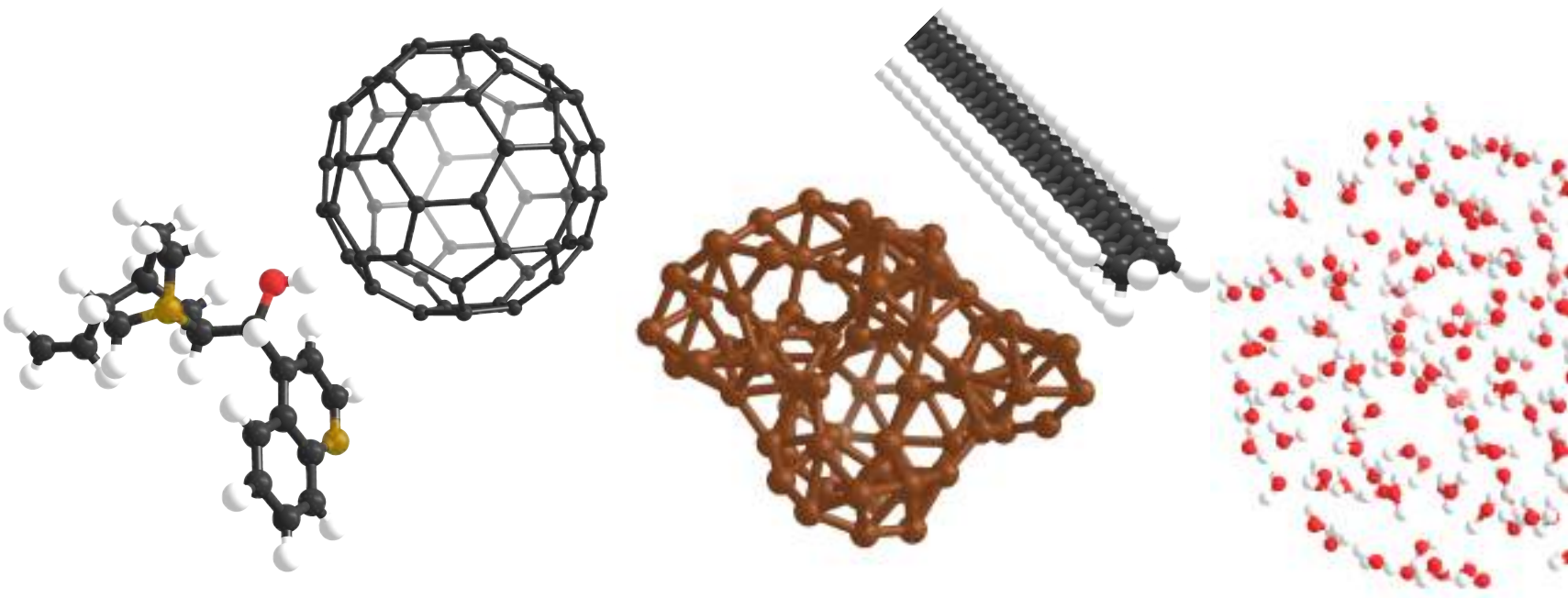
Accuracy: Total energies and forces



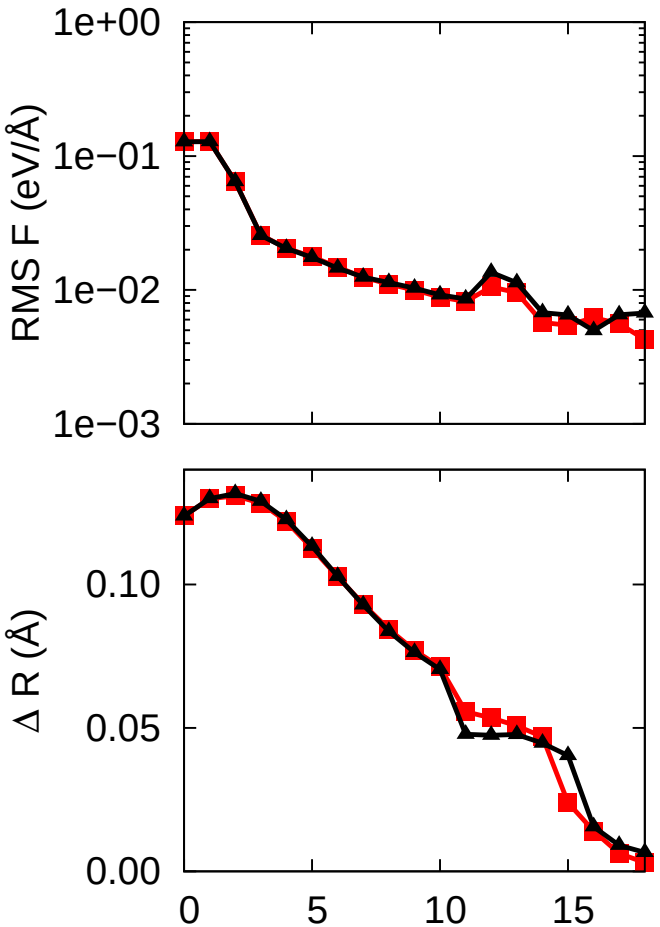
BigDFT

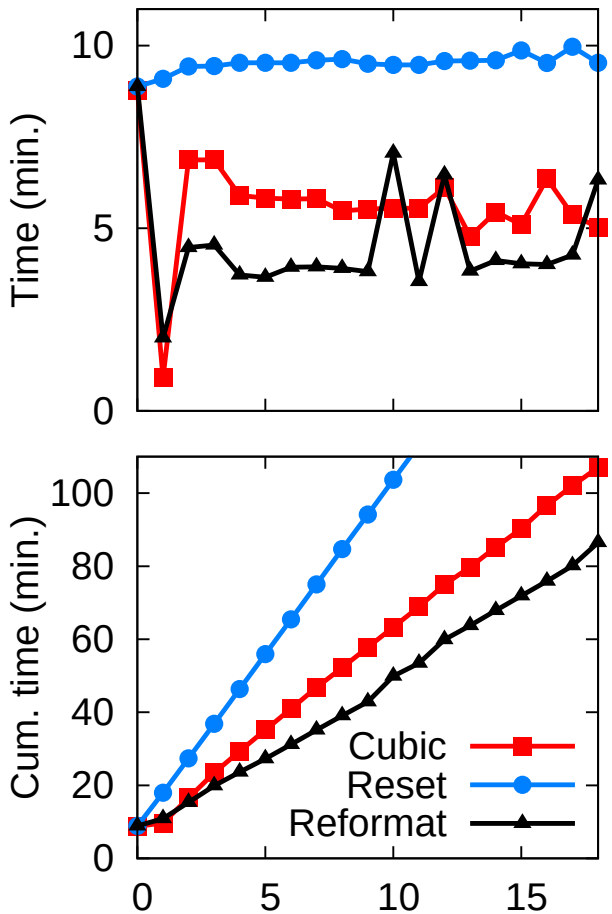
Several systems 44 – 450 atoms

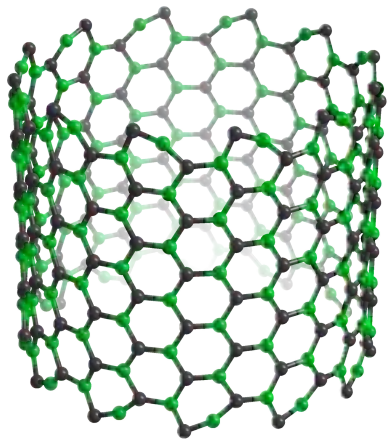
- compared with standard BigDFT
- total energies accuracy: $\sim$ 1 meV/atom
- forces accuracy: $\sim$ a few meV/bohr



Geometry optimization





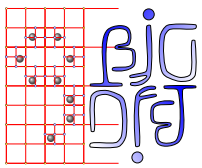


288 atoms

Further savings through support function reuse

- Pulay forces not needed
- lower prefactor than single point

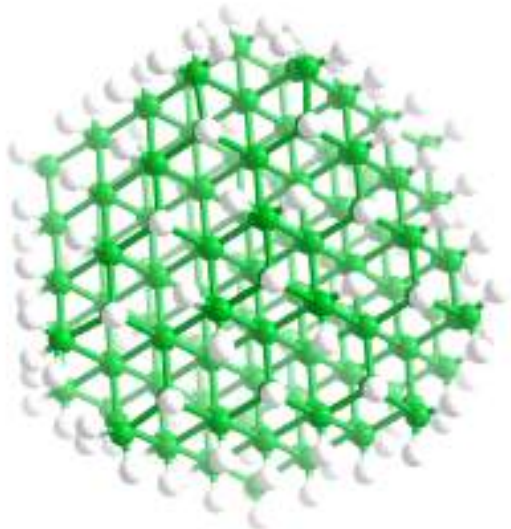
Energy differences



BigDFT

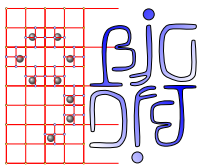
Silicon cluster with a vacancy defect

- 291 atoms
- need 9 support functions per Si
- accuracy in defect energy of 12 meV

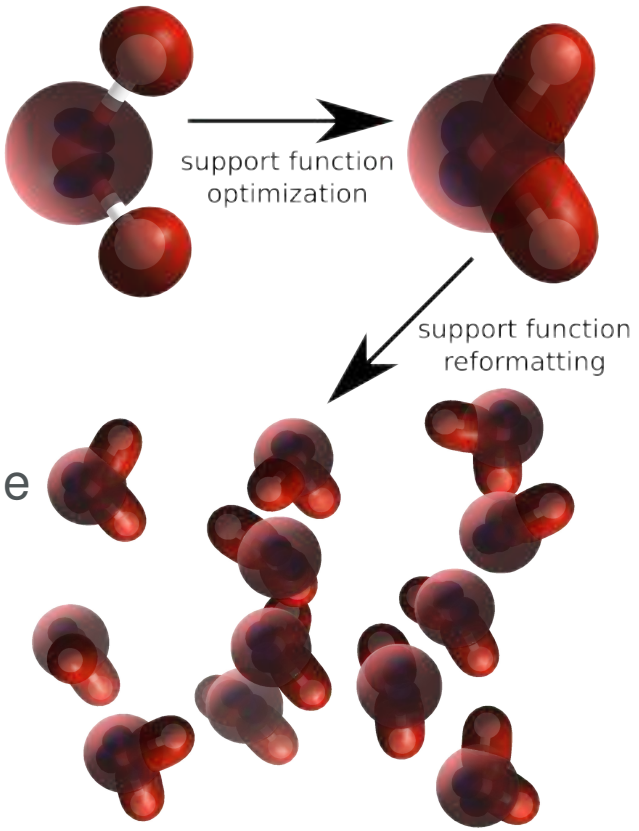


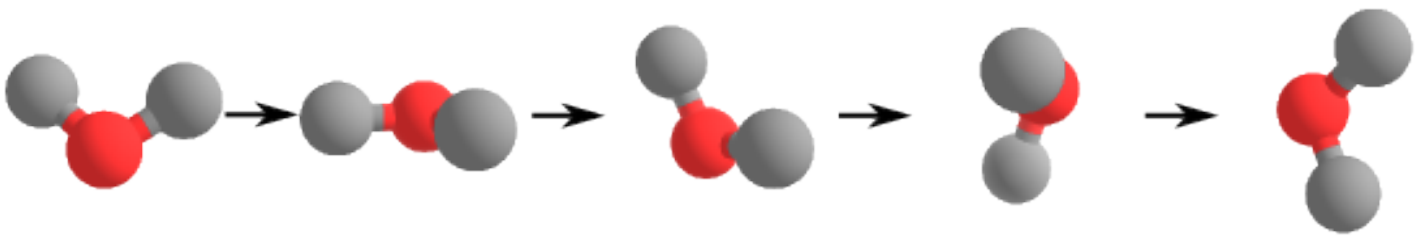
	pristine eV	vacancy eV	Δ eV	Δ-Δ_{cubic} meV
cubic	−20674.2	−20563.1	111.167	–
4/1 (*sp*-like)	−20667.6	−20556.5	111.038	129
9/1 (*spd*-like)	−20672.9	−20561.7	111.155	12

Fragment approach



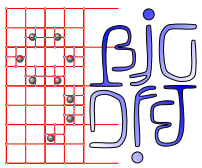
- By dividing a system into fragments, we can avoid optimizing the support functions entirely for large systems
- Substantially reduces the cost (need an efficient reformatting)
- Many useful applications, including the explicit treatment of solvents
- A necessary first step towards a tight-binding like approach, with each atom as a fragment





Reformatting the minimal basis set in the same grid

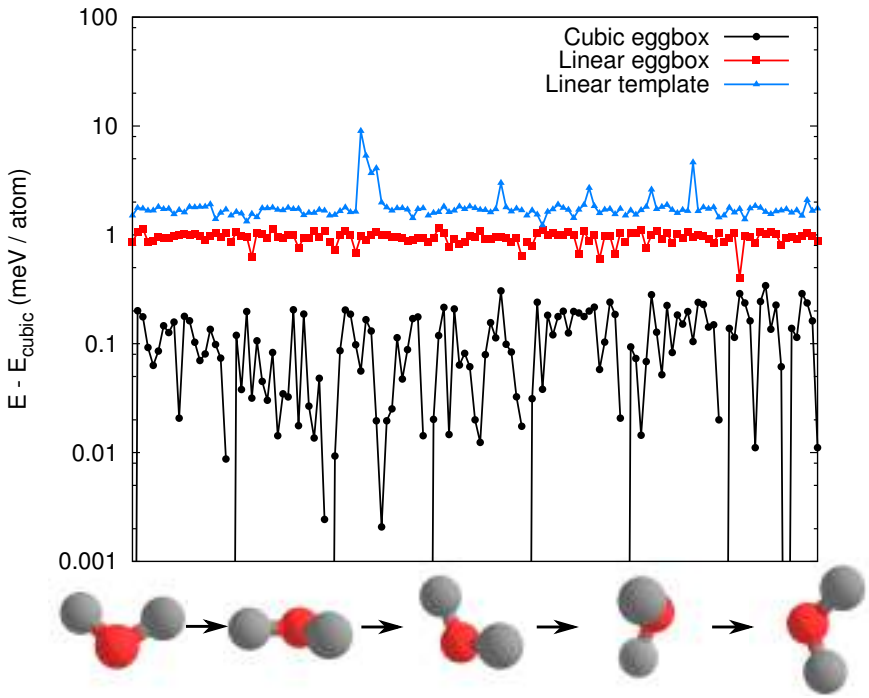
Support function reuse

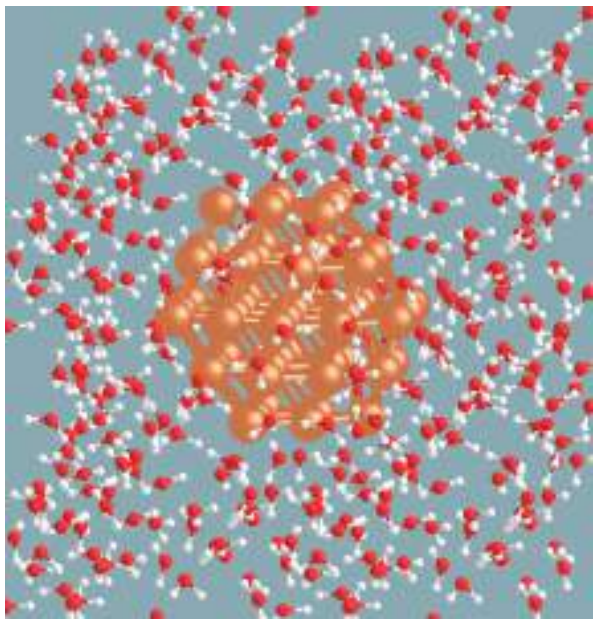


BigDFT

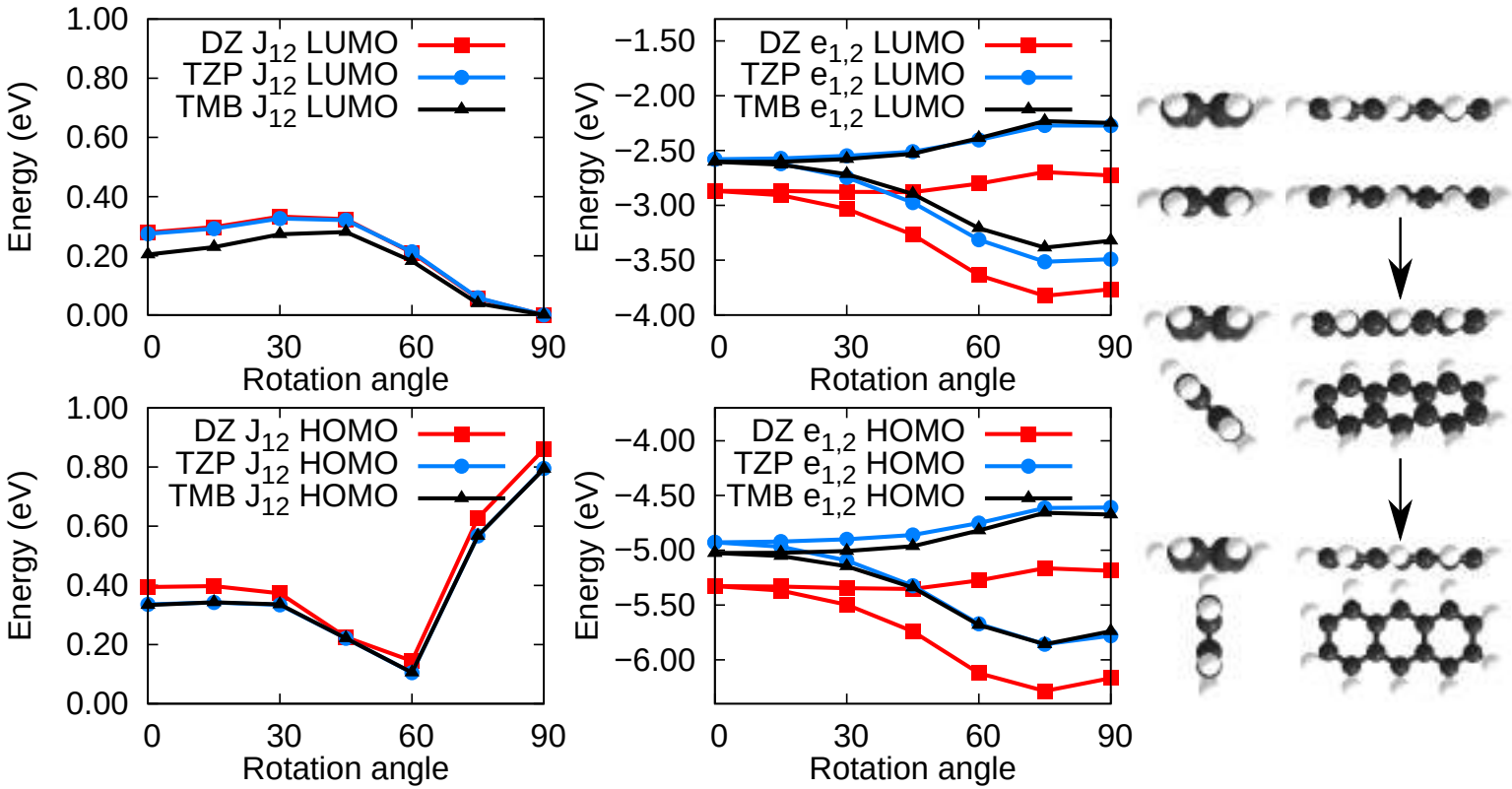
As we have seen, the reuse of support functions lends itself to some interesting applications, e.g. geometry optimizations, charged systems

For this, we need an accurate and efficient reformatting scheme





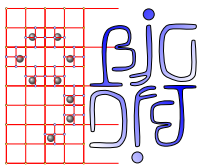
Transfer integrals and site energies



BigDFT compared to ADF fragment approach

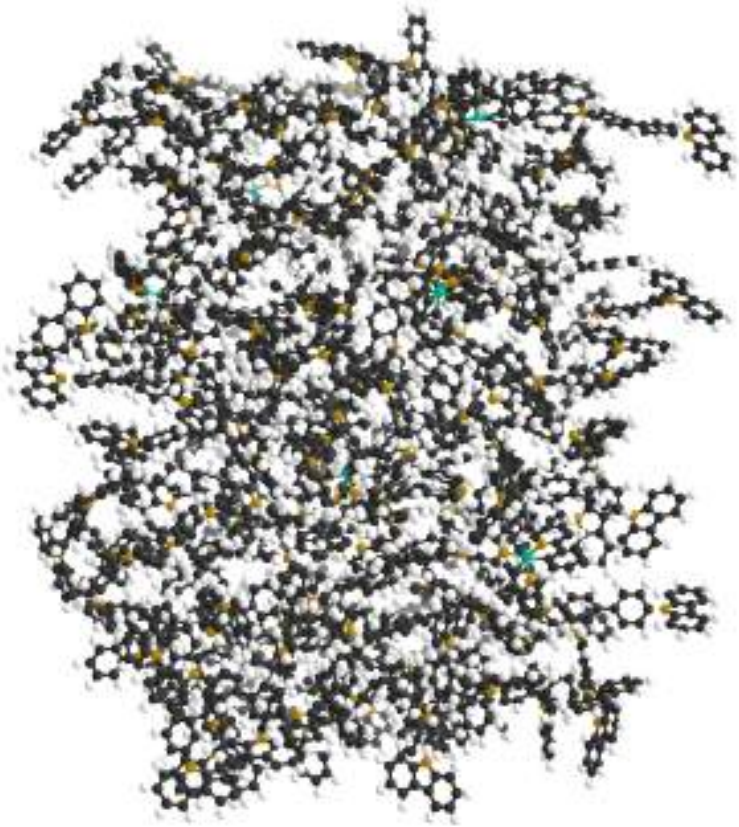
- support functions from molecules reused in dimers
- Application to OLED (Organic Light-Emitting Diode)

Transfer integrals for OLEDs (I)



BigDFT

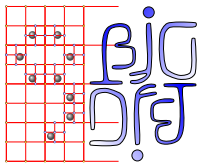
We want to consider environment effects in realistic 'host-guest' morphologies: 6192 atoms, 100 molecules



Host molecule

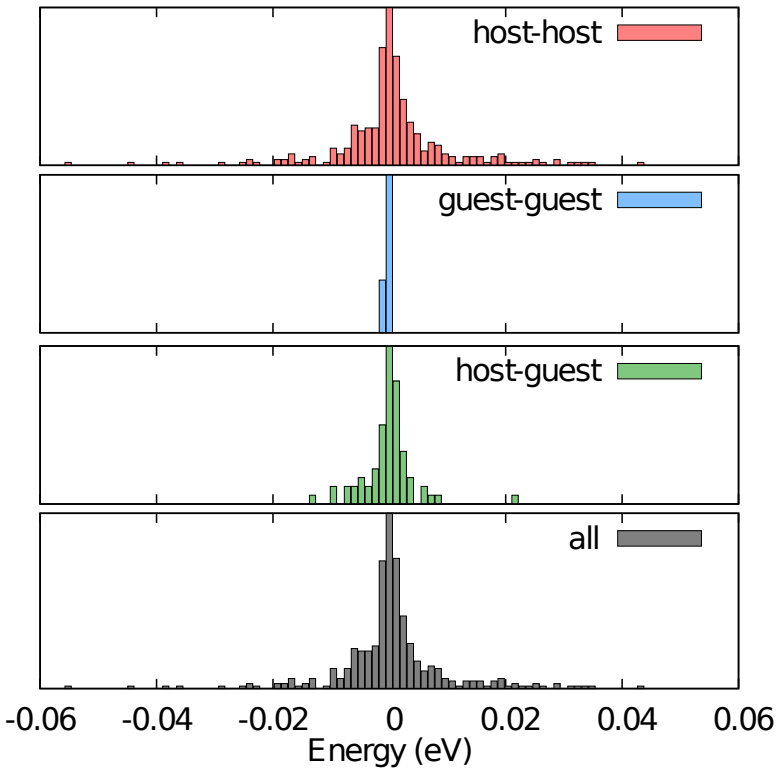
Guest molecule

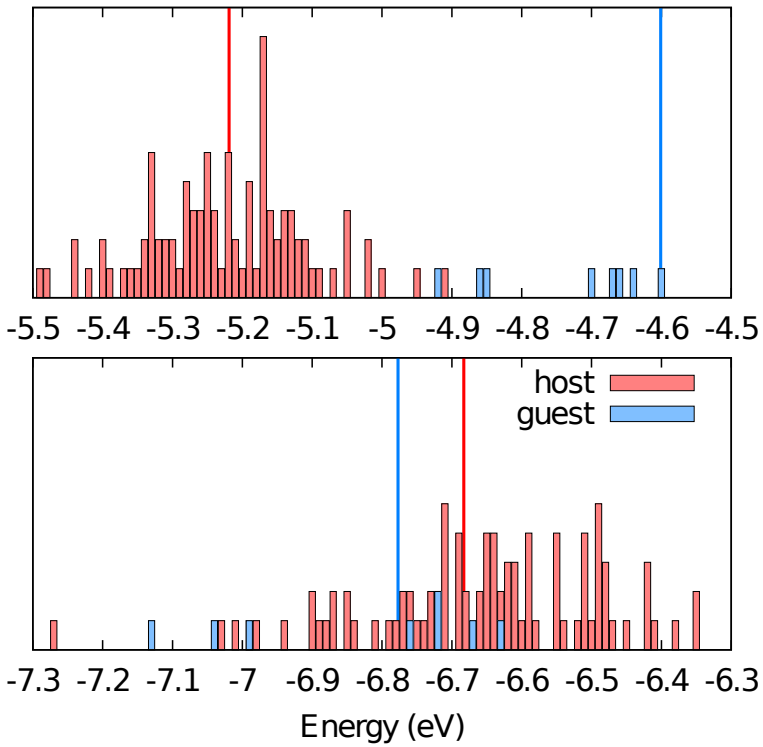
Transfer integrals for OLEDs (II)



BigDFT

Transfer integrals (left) and site energies (right) calculated with (bottom) and without (top) constrained DFT

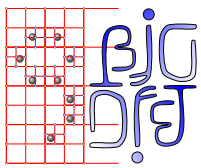




Generate good statistics.

Then use of Markus model to calculate the efficiency of OLEDS.

Perspectives of linear scaling



BigDFT

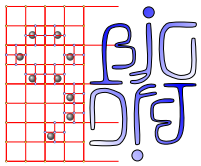
Combination of wavelets and localized basis functions

- Highly accurate (energies and forces) and efficient
- Linear scaling – Thousands of atoms
- Ideal for massively parallel calculations
- Flexible e.g. reuse of support functions to accelerate geometry optimizations and for charged systems

Fragment approach

- Accurate reformatting scheme – can accelerate calculations on large systems e.g. explicit treatment of solvents
- Wide range of applications: constrained DFT, transfer integrals
- Crucial for future developments e.g. tight-binding, embedded systems (QM/QM)

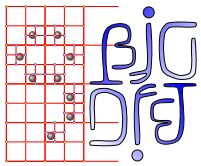
Outline



BigDFT

1. The machinery of BigDFT (cubic scaling version)
   - Mathematics of the wavelets
   - Main Operations
   - MPI/OpenMP and GPU performances
2. $\mathcal{O}(\mathcal{N})$ (linear scaling) BigDFT approach
   - Minimal basis set approach
   - Performance and Accuracy
   - Fragment approach
   - Applications
   - Perspectives
3. Conclusions

Conclusions



BigDFT

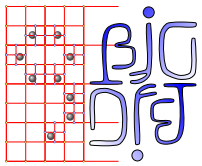
Wavelets: Powerful formalism

- Linear scaling – Thousands of atoms
- Accurate reformatting scheme – can accelerate calculations on large systems e.g. explicit treatment of solvents
- Wide range of applications: constrained DFT, transfer integrals
- Towards multi-scale approach (embedded systems)

Beyond DFT

- Resonant states are the way to describe fully a system
- Give a compact description (few states) of the systems
- Linear-response, TD-DFT, ...
- Many-body perturbation theory (GW, ...)

Acknowledgments



BigDFT

Main maintainer Luigi Genovese

Group of Stefan Goedecker A. Ghazemi, A. Willand, S. De, A. Sadeghi, N. Dugan (Basel University)

Link with ABINIT and bindings Damien Caliste (CEA-Grenoble)

Order N methods S. Mohr, L. Ratcliff, P. Boulanger (Néel Institute, CNRS)

Exploring the PES N. Mousseau (Montreal University)

Resonant states A. Cerioni, A. Mirone (ESRF, Grenoble), I. Duchemin (CEA) M. Morinière (CEA)

Optimized convolutions B. Videau (LIG, computer scientist, Grenoble)

Applications P. Pochet, E. Machado, S. Krishnan, D. Timerkaeva (CEA-Grenoble)