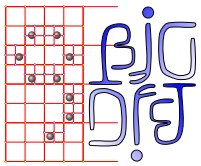


3rd International ABINIT Developer Workshop

LIÈGE - BELGIUM

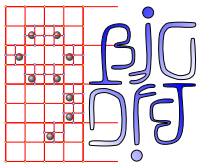
Introducing wavelet basis sets inside ABINIT via the BigDFT project

Damien Caliste & Luigi Genovese

PCPM - UCL / L_Sim - CEA Grenoble

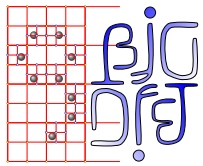
30 January 2007

Outline



1. Poisson Solver for free BC
   - Interpolating scaling functions
   - Separability
   - Performances
2. Computing Hartree's potential in ABINIT for isolated systems
   - The `02poisson` interface
   - Computing potentials in real space
   - Effects on sample potentials
3. Inclusion of a wavelet basis set into ABINIT, the BigDFT project
   - A quick view into BigDFT machinery
   - Interface between BigDFT and ABINIT
   - Convergence and results
4. Summary and outlook

Poisson Solver with interpolating scaling functions



Calculation of the self-consistent potential

Poisson Equation for isolated systems

$$\nabla^2 V_H(\mathbf{r}) = -4\pi\rho(\mathbf{r}) \quad \Longrightarrow \quad V_H(\mathbf{r}) = \int_\Omega K(|\mathbf{r}-\mathbf{r}'|)\rho(\mathbf{r}')d\mathbf{r}'$$

The kernel $K(\mathbf{r}) = 1/r$ with $\Omega = \mathbb{R}^3$

This equation is usually solved using plane waves. Their delocalization results in problems for isolated systems.
How to remove long-distance interactions?

Most commonly used methods (Hockney, Martina-Tuckerman)

Modify the kernel operator $K = K_{\text{short}} + K_{\text{long}}$

- Does not implement well short-distance behaviour
- We must consider a size that is larger than the size of the original system

Using a localized function set

Interpolating Scaling Functions (from wavelet theory)

- Localized in real and Fourier space
- Undergo multi-scale relations
- The expansion coefficients coincide with the values on a (uniform) grid
- Separable basis functions $\phi_{\mathbf{j}}(\mathbf{r}) = \phi_{j_x}(x)\phi_{j_y}(y)\phi_{j_z}(z)$

Optimal for isolated systems

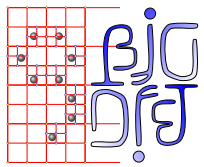
The potential is obtained via a *discrete convolution*

$$V(\mathbf{i}) = \sum_{\mathbf{j}} K_{\mathbf{i}-\mathbf{j}}\, \rho(\mathbf{j})$$

$$K_{\mathbf{i}} = \int K(|\mathbf{r}|)\Phi_{\mathbf{i}}(\mathbf{r})d\mathbf{r} = \int \frac{1}{r}\phi_{i_x}(x)\phi_{i_y}(y)\phi_{i_z}(z)d\mathbf{r}$$

Given $K_{\mathbf{i}}$, we can calculate convolution using *zero-padded FFT* ($O(N\log(N)$ scaling)

Gaussian tensor product decomposition



The calculation of the kernel $K_{\mathbf{i}}$ can be improved thanks to separability of the basis

Approximation with gaussians(Beylkin et *al.*)

$$\frac{1}{r} \simeq \sum_k \omega_k e^{-p_k r^2}$$

with very high accuracy

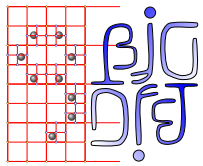
89 terms, p_k , ω_k suitably chosen.
The sum in gaussians allows us to write the tensor decomposition of the kernel:

The computational cost is reduced $N^3 \longrightarrow 89 \times N$

$$K_{\mathbf{j}} = \sum_{k=1}^{89} \omega_k K_{j_x}(p_k) K_{j_y}(p_k) K_{j_z}(p_k)$$

$$K_j(p) = \int \phi_0(x) e^{-p(x-j)^2} dx$$

Features

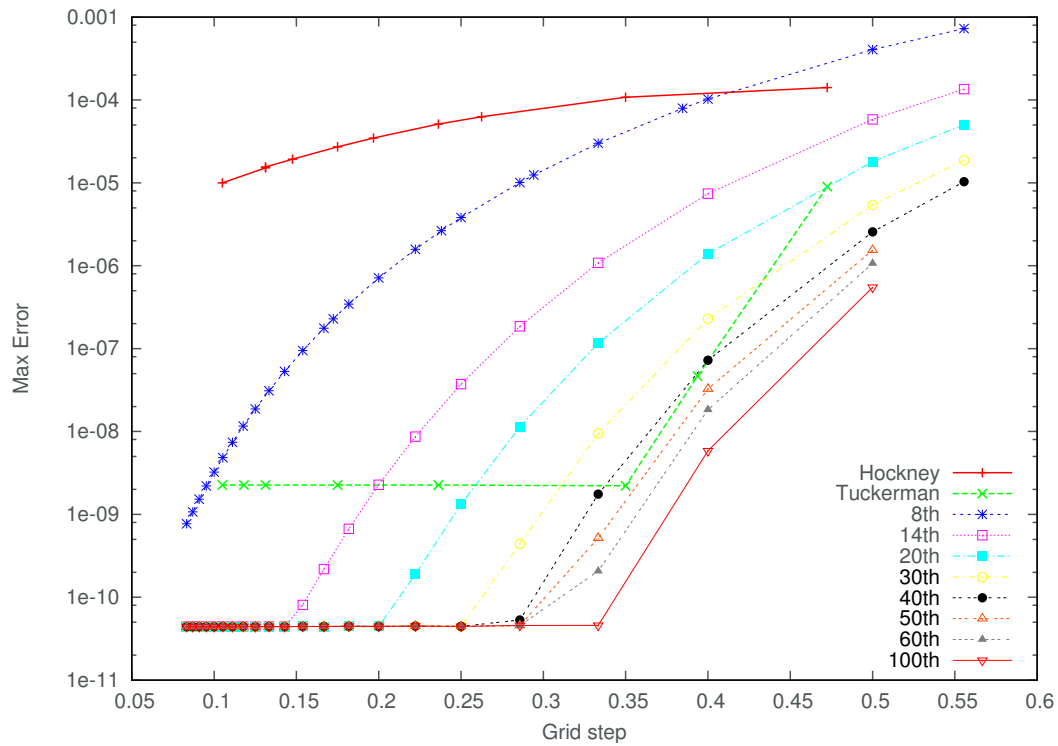


Very fast with moderate memory occupation:

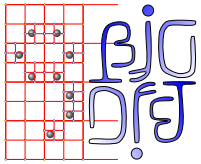
Elapsed Time on a Cray XT3,128^3 grid

np	1	2	4	8	16	32	64
s	.92	.55	.27	.16	.11	.08	.09

More precise than other existing Poisson Solvers:



Overview



In summary, we have developed a technique

- ✔ Free boundary conditions
- ✔ Very high accuracy
- ✔ Good computational performance, easy to parallelize
- ✔ Based on a real space grid, can be used independently
- ✔ Allows for MultiResolution Analysis (adaptivity)

Paper published

L. Genovese, T. Deutsch, A. Neelov,S. Goedecker, G. Beylkin *Efficient solution of Poisson's equation with free boundary conditions*, [arXiv: cond-mat/0605371], J. Chem. Phys. **125**, 074105 (2006)

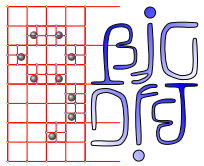
The 02poisson interface

- Branching into rhohxc_coll() between hartre() and PSolver_hartree()

```
if (dtset%icoultrtmt == 1) then
call PSolver_hartree(dtset, 1, nfft, ngfft, rhor, rprimd, vhartr)
else
call hartre(cplex,gmet,gsqcut,izero,mpi_enreg,nfft,ngfft,qphon,rhog,vhartr)
end if
```

- Two new input variables
  - icoultrtmt that controls the computation of Coulomb's potential ;
  - nscforder which commands the degree of interpolating scaling functions used in the kernel.

Realspace potentials within E_{total}



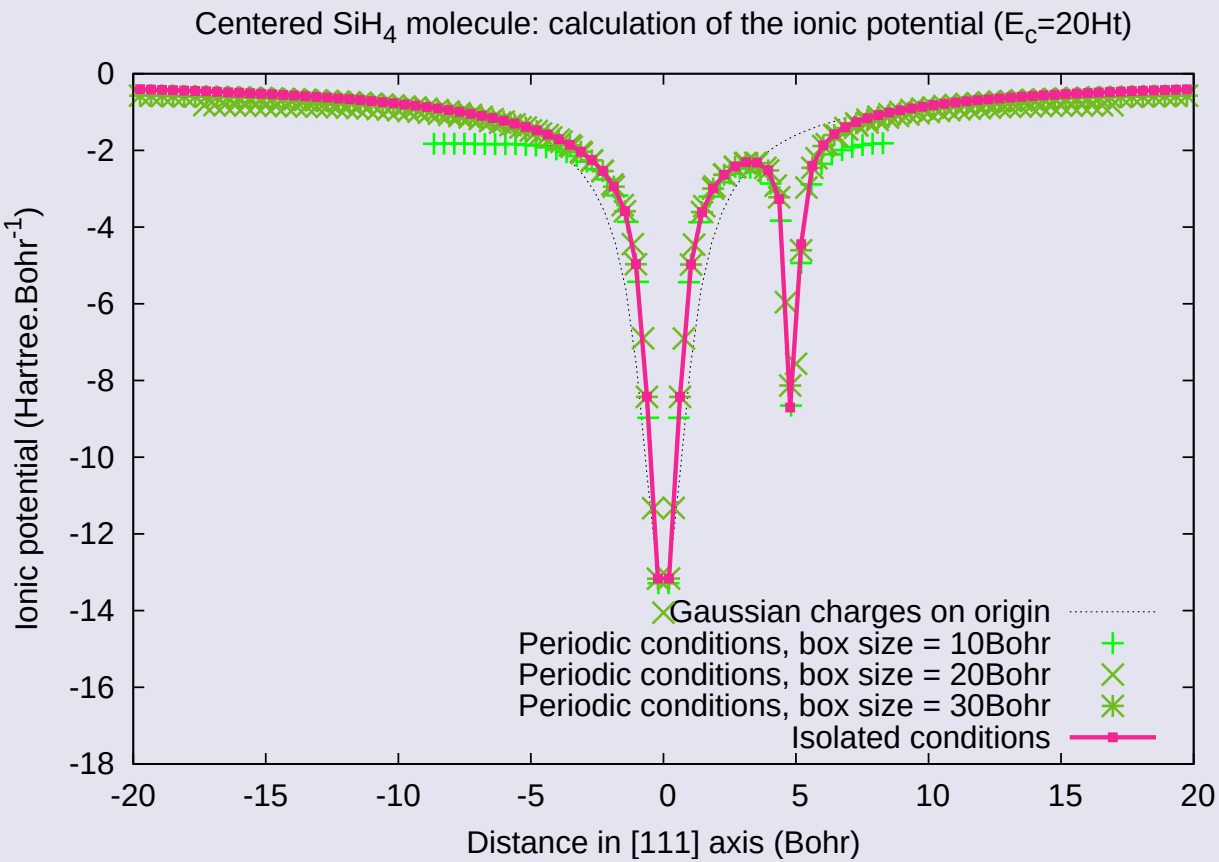
With `icoultrtmt == 1`, all local potentials are computed in realspace.

	Real space	Reciprocal space
E_{kin}	✘	✔
E_{hart}	`PSolver_hartree()`	`hartre()`
$E_{\text{psp\_nl}}$	✘	✔
$E_{\text{psp\_loc}}$	branched inside `mklocl()`	
	`mklocl_realspace()`†	`mklocl_recipspace()`
E_{ewald}	simple ion/ion interaction	
	$\sum_{\lambda,\kappa;\lambda\neq\kappa} \frac{Z^{\lambda}_{\text{ion}} Z^{\kappa}_{\text{ion}}}{\|\tau_\lambda - \tau_\kappa\|}$	
	`ionion_realspace()`	`ewald()`

† Computation of $E_{\text{psp\_loc}}$ in real space requires a description of pseudo-potentials in RS, currently implemented for GTH and HGH only.

Better estimated potentials for isolated systems

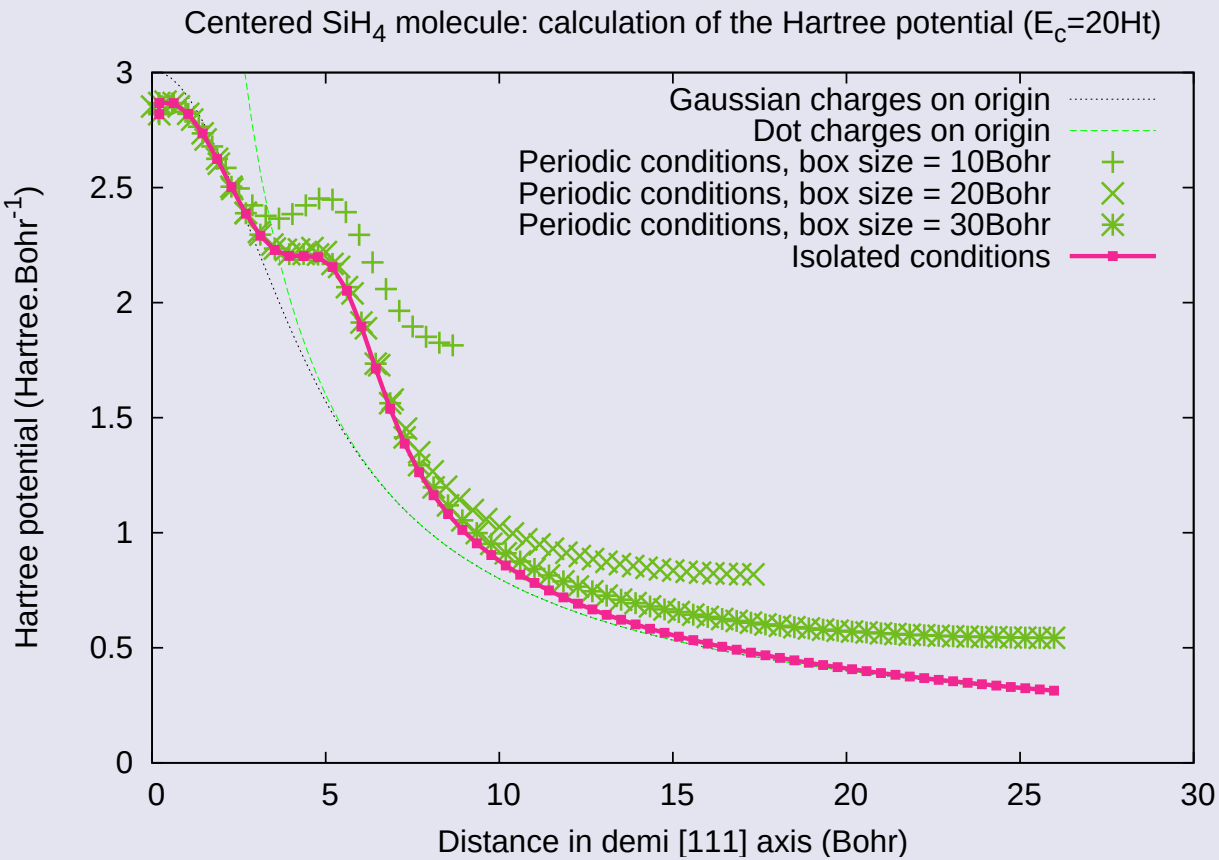
The local part from pseudo-potentials



- the potentials are correct in the boundary regions ;
- the $\frac{1}{\mathbf{r}}$ behavior is valid whatever the supercell size.

Better estimated potentials for isolated systems

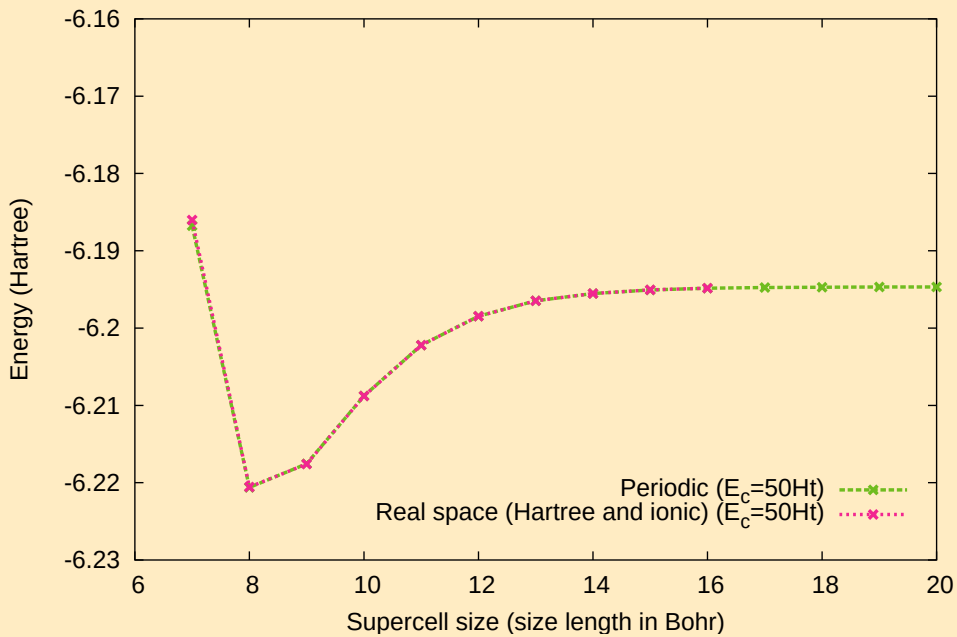
The Hartree's potential



- the potentials are correct in the boundary regions ;
- the $\frac{1}{\mathbf{r}}$ behavior is valid whatever the supercell size.

E_{total} is still affected by box size

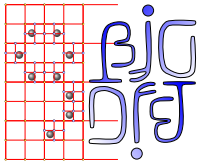
Total energy of a silane molecule



Total energy is not significantly modified by accurate potential treatment.

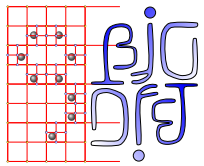
- the kinetic part of the energy is strongly dependent on the periodicity of the super-cell ;
- a full real-space is mandatory for isolated and/or inhomogeneous systems.

A quick view into BigDFT machinery



- Every steps of electronic calculations are treated in real space on a systematic basis set.
  - Pseudo-potentials are described by separable functions (GTH and HGH) ;
  - Wave-functions are expanded on a Daubechies' wavelet basis set (*i.e.* a set of coefficients on a real space adaptative grid).
- New tools have been developped to handle the Schrödinger's equation.
  - A real space laplacian, using filters on wavelets ;
  - Storage and matricial computation of wave-functions on a two-resolution grid ;
  - A real space preconditionner for wavelets.
- The ground state is computed in a direct minimisation loop, using either the steepest descent algorithm or a DIIS scheme.

Interface between BigDFT and ABINIT



All informations have been gathered into Fortran types.

Data storage

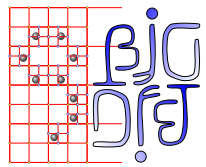
- `wvl_wf_type` - stores the decomposition of wave-functions ;
- `wvl_keyArrays_type` - description of the two-resolution grid ;
- `wvl_projectors_type` - stores the projector parts of pseudo-potentials.

Most of the main routines have been branched and call the BigDFT library (using a module interface).

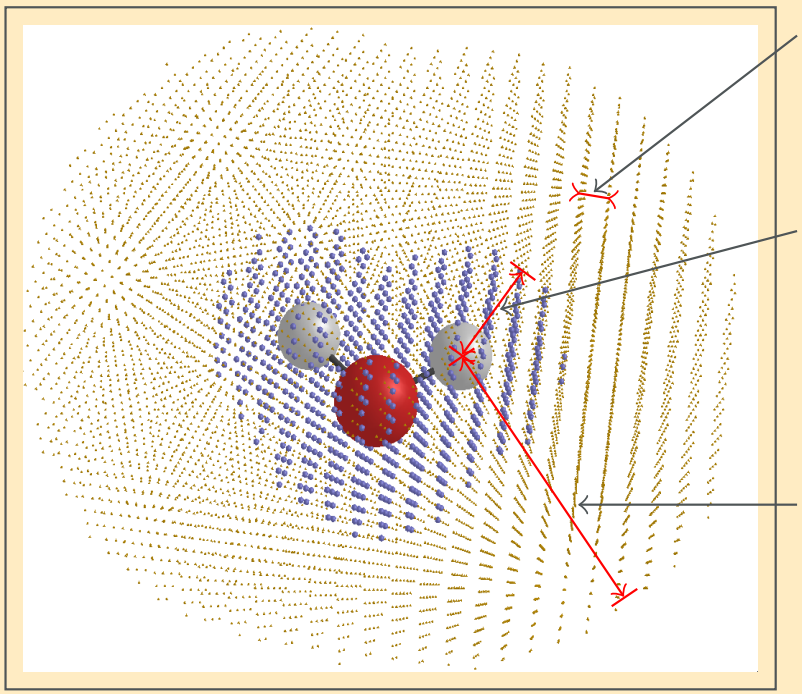
New interface routines

- `src/08seqpar/wvl_vtorho()` - the main part within the minimisation loop (build the Hamiltonian, compute the gradient and create new wave-functions using steepest descent or DIIS) ;
- `src/05common/wvl_mkrho()` - compute density from wave-functions ;
- `src/04wvl_wfs` - contains basic routines to create, free, manipulate wave-functions on a two-grid wavelet basis set ;

New input variables



Main specific variables to control the basis set



`wvl_hgrid` the grid size

`wvl_frmult` the expansion around atoms for fine grid

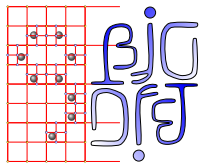
`wvl_crmult` the expansion around atoms for coarse grid

All wavelet specific input variable are prefixed by wvl_.

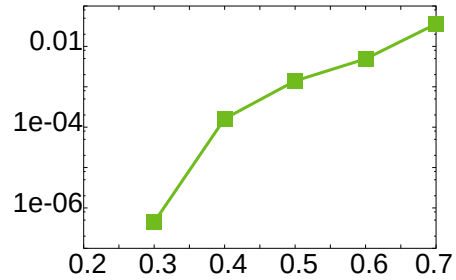
Other relevant variables

- `usewvl` to enable a wavelet computation ;
- `nwfshist` for the DIIS history.

An accurate total energy

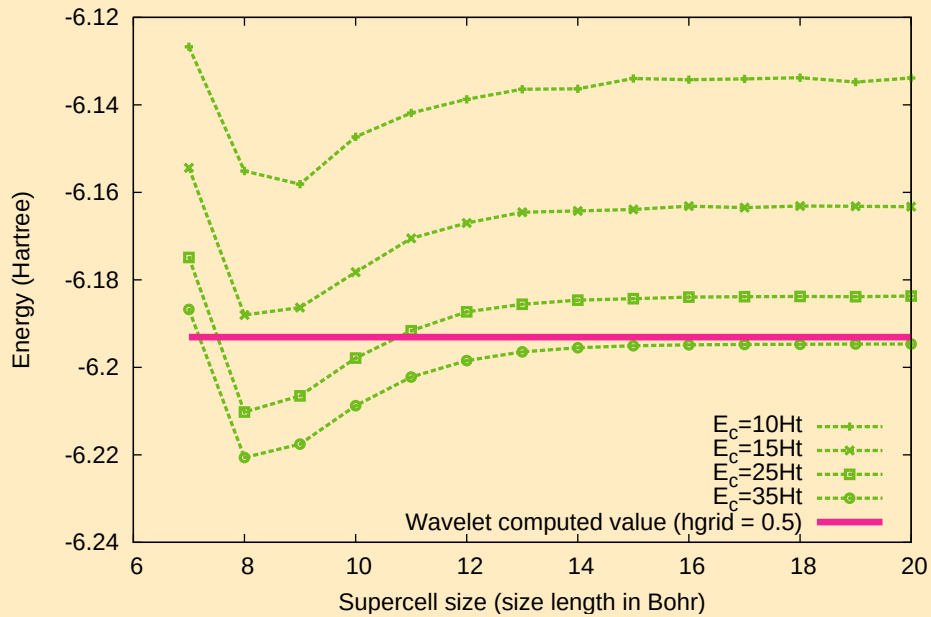


The main convergence parameter is `wvl_hgrid` and an usual value for organic molecule is 0.35Bohr.



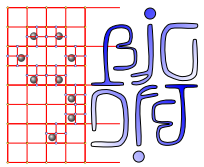
[Convergence of E_{total} versus `wvl_hgrid` for a silane]

Total energy for a silane molecule



$E_{\text{total}}^{\text{wvl}} = E_{\text{total}}^{\text{pw}}$ with a converged cut-off.

Summary and Outlook



The standalone code created in the frame of the European project BigDFT, has been included in ABINIT for developper usage. It is characterised by:

- a systematic real space basis set ;
- an accurate ground state computation.

The next development will import the capability to

- run on parallel with a split on bands (mature in BigDFT) ;
- compute forces for geometry optimisation (mature in BigDFT) ;
- use a scheme to improve accurancy without increasing the basis-set (recently introduced in BigDFT).

A simple linear scaling approach is under development in the BigDFT project and schedule for inclusion in ABINIT in 5.4 series.

Wavelet computations for end-users will be presented during next autumn in a CECAM tutorial.