Cecam tutorial - Wavelets in electronic structure

LYON - FRANCE

Introduction to the BigDFT library

Damien Caliste

L_Sim - CEA Grenoble

28 November 2007
First glimpse, “Do I need my umbrella?”

### The BigDFT library

<table>
<thead>
<tr>
<th>Wavelets</th>
<th>PW</th>
<th>Gaus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground state + forces</td>
<td>☀️</td>
<td>☀️</td>
</tr>
<tr>
<td>Isolated conditions</td>
<td>☀️</td>
<td>☁️</td>
</tr>
<tr>
<td>Arbitrary precision</td>
<td>☀️</td>
<td>☀️</td>
</tr>
<tr>
<td>Elec. field &amp; charged sys.</td>
<td>☀️</td>
<td>☁️</td>
</tr>
<tr>
<td>Collinear spin</td>
<td>☀️</td>
<td>☀️</td>
</tr>
<tr>
<td>Metallic systems</td>
<td>⌚️</td>
<td>☀️</td>
</tr>
<tr>
<td>Surfaces</td>
<td>⌚️</td>
<td>☁️</td>
</tr>
<tr>
<td>Periodic boundary conditions</td>
<td>⌚️</td>
<td>☀️</td>
</tr>
</tbody>
</table>

### Objectives of the lesson

- To know what are the main steps in a GS calculation
- To learn some elements on the manipulated objects

---

EU Nest Adventure - BigDFT project

www.abinit.org
Structure of the calculation in the wavelet basis set

1. The electronic minimisation loop
2. The input guess
3. Possible add-on calculations after fix point is reached
4. A fully parallelised code

Structure of the objects in the calculation

1. How to describing the electronic wavefunctions on a wavelet basis set?
2. Available operators in BigDFT
The DFT framework

The ground state is obtained by solving:

\[ H\psi = E\psi \]

- Expressed only for the electrons, using the Born-Oppenheimer approximation;
- Solved in the framework of Kohn-Sham, using one particule wavefunctions;
- Use of a self-consistency loop since the Hamiltonian \( H \) depends on \( \psi \).

What about wavelets?

Wavelets are the basis set for the representation of the wavefunctions. Thus things like \( \text{XC approximation} \) or Hellmann-Feynman theorem are still usable.
Elements of the Hamiltonian $H$

All classical parts present in plane-waves, can also be treated with wavelets.

**The kinetic operator $\nabla^2 \psi$**

Computed from the wavefunctions $\psi$, applying a **local filter**. Length for Daubechies is 28, see `Convolkinetic()` routine.

Finite differences in 1D:

$$\Delta f(x_n) = \frac{f(x_{n-1}) - 2f(x) + f(x_{n+1})}{h^2}$$

is a filter of $(1, -2, 1)$.

Precision is $O(h^{14})$ for arbitrary function. Even exact for linear combination of Daubechies functions:

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

$$\tilde{c}_\ell = \sum_j c_j a_{\ell-j}, \quad a_\ell \equiv \int \phi_0(x) \frac{\partial^2}{\partial x^2} \phi_\ell(x).$$
The non-local part of pseudo-potentials

Computed from the scalar product of wavefunctions $\Psi$ and the projectors, done in routine `applyprojectorsone()`. This is possible for GTH and HGH pseudo-potentials because of the spatial separability in their analytic expression.

Possible future developments may be done to include some other pseudo-potential families.
Elements of the Hamiltonian $H$

The local potentials, $V_H$ $V_{xc}$ $V_{psploc}$ & $V_{ionic}$

Hartree and ionic potentials are computed with a Poisson solver from the density $ρ(\vec{j})$.

$$V_H(\vec{j}) = \int d\vec{x} \frac{ρ(\vec{x})}{|\vec{x} - \vec{j}|}, \quad V_{xc}(\vec{j}) \text{ with ABINIT XC routines and}$$

$$V_{ionic}(\vec{j}) = \Sigma_κ \int d\vec{x} \frac{Z_κ}{|\vec{x} - \vec{j}|}$$

This Poisson solver is:

- **✓** very **fast and accurate**, with optimal parallelisation;
- **✓** can be used **independently** from the DFT code;
- **✓** integrated quantities (energies) are easy to extract;
- **✓** correct treatment of **isolated boundary conditions** and of charge offset;
- **✗** non-adaptive, needs data uncompression.

EU Nest Adventure - BigDFT project

www.abinit.org
Finding the fix point in Schrödinger equation

Two possible methods: the diagonalisation and the direct minimisation. The latter is fast and simple but restricted to systems with a non-zero HOMO-LUMO gap.

To improve convergency, preconditioning is done on the gradients $H\Psi$ using the routine `precondition()`.

$$\left(\frac{1}{2}\nabla^2 + C\right)\vec{p} = \vec{g}$$

$\vec{p}$ is computed in an iterative way (CG).

Classical gradient algorithms are applied for electronic minimisation: steepest descent or DIIS [Pulay, Chem. Phys. Lett. 73, 393 (1980)].

Future developments

Diagonalisation will be developed in later release. It will enable polarisation studies or fully metallic systems.
On the contrary to plane waves, wavelet representation requires a good input start for the minimisation loop.

Using atomic orbitals

The input guess is based on atomic orbitals, using the following scheme:

- project gaussian atomic orbitals in a Daubechies wavelet basis set (see the gauss_to_daub() routine);
- compute the associated hamiltonian;
- diagonalise it in this basis set;
- use the lowest states as input for the calculation.

This is done in input_wf_diag() routine.
The input guess, take a good start!

Tips! Get a look on the **evale** output

The eigenvalues in the input guess give an idea of the **electronic symmetry** of the system.

Computed eigenvalues of the atomic orbitals of a single carbon atom.

**BigDFT output of input guess**

```plaintext
evale(1)= -4.69845419157335E-01
evale(2)= -1.67943794696384E-01 <- Last eigenvalue for input wavefunctions
evale(3)= -1.67943794696384E-01 <- found degeneracy
evale(4)= -1.67941568022003E-01 <- found degeneracy
```

The first two eigenvalues are populated, but the two second are close enough to create a degeneracy around Fermi level.
## Additional computation after minimisation

### Getting the Kohn-Sham wavefunctions

A diagonalisation is done on output of the minimisation loop by the `KStrans()` routine.

### Compute the forces

The forces are sum of two parts:

- the local contribution, computed with the Poisson solver, see `local_forces()` routine;
- the non-local part, coming from the first derivatives of projectors. See `projectors_derivatives()` and `nonlocal_forces()` routines.

### Finite-size effects - an error estimation

After convergency, the basis set can easily be expanded to estimate these effects.

---

**EU Nest Adventure - BigDFT project**

[www.abinit.org](http://www.abinit.org)
Parallelism in BigDFT

What to do?
- Minimisation loop
- The input guess
- Post-minimisation

How to?
- Storing the WF
- Real space operators

BigDFT - an introduction

Wavefunctions
Orbitals are distributed over CPUs.

Density and potentials
Divided among CPUs: z-planes parallelisation.
Parallelism in BigDFT

**Wavefunctions**

Orbitals are distributed over CPUs.

Required communications:
- for orthogonality
  \[ \text{MPI\_ALLTOALL} \];
- for the density
  \[ \text{MPI\_REDUCE\_SCATTER} \].

**Density and potentials**

Divided among CPUs:
- \text{\textit{z}-planes parallelisation}.

Required communications:
- FFT communication in the Poisson solver.

**Full parallelisation**

There is no direct bottleneck with CPU numbers, making the code ready for both workstations and massively parallel computers.
- memory is divided;
- CPU time is also divided.
Overview

BigDFT library 1.0

Minimisation loop

Input guess in LCAO
- project atomic orb. in Daubechies;
- diagonalise + take lowest eigen-values.

Compute density.

Kinetic operator

Preconditioner

Local potentials

Non-local op.

Stp. desc. / DIIS

HΨ

ρ

V_{XC}

V_{H}

$V_I$

Ψ

Forces computation
- projector first derivatives;
- non local forces;
- local forces.

local_forces() & non_local_forces()

Kohn-Sham diag.

create Ionic Potential()

Overview

BigDFT - an introduction

What to do?
Minimisation loop
The input guess
Post-minimisation
Parallelism

How to?
Storing the WF
Real space operators

EU Nest Adventure - BigDFT project

www.abinit.org
Wavefunctions are expressed in a Daubechies wavelet basis set, for the following reasons:

**Orthogonality**

Daubechies are an orthogonal wavelet family. Scalar products are thus immediate.

**Multi-resolution**

*Locality:* density of degrees of freedom can be located near atoms. Void areas can be low density or even ignored zones.

*Adaptivity:* with pseudo-potential approximation, a 2-resolution level is enough for electronic computation.
The wavelet basis set

On a discretised mesh \([i,j,k]\) for one resolution:

\[
f(i,j,k) = \sum_{l,m,n} c_{l,m,n} \phi_{N_x}^l(i) \phi_{N_y}^m(j) \phi_{N_z}^n(k)
\]

Schematic view for a multi-resolution grid in 2D
The two-resolution grid around a water molecule

Number of degrees of freedom can vary with the position in space of each grid point.

The grid is divided in:
- **Low** resolution pts (SF, 1 DoF)
- **High** resolution pts (SF + W, 8 DoF)

Points of different resolution belong to the same grid.

**Advantage**

Number of data are more optimum. Data can thus be stored in compressed form.
Wavefunction storage in BigDFT

The compression is a compact storage of non-null wavelet coefficients in \( \psi(nvctr_c+7*nvctr_f,norb) \) array. Access keys are stored in the `wavefunctions_descriptors` public type.

The structure is symmetric for coarse and fine grid points.

- \( nseg \) is the number of segments \((nseg_c=2)\);
- \( nvctr \) is the number of non-null grid points \((nvctr_f=9)\);
- \( \text{keyg}(2,nseg) \) gives the start and stop of each grid segment \((\text{keyg_f}(:,1)=(2, 4))\);
- \( \text{keyv(nseg)} \) gives the segment positions in \( \psi \) array \((\text{keyv_c}(2)=6)\).
Overview of wavelet families used in BigDFT code

**Daubechies** \( f(x) = \sum_\ell c_\ell \phi_\ell(x) \)

- **Orthogonal set**
  \[ c_\ell = \int dx \phi_\ell(x)f(x) \]

No need to calculate overlap matrix of basis functions
Used for wavefunctions, scalar products

**Interpolating** \( f(x) = \sum_j f_j \phi_j(x) \)

- **Dual to dirac deltas**
  \[ f_j = f(j) \]

The expansion coefficients are the point values on a grid
Used for charge density, function products

**Magic Filter method (A.Neelov, S. Goedecker)**

The passage between the two basis sets can be performed without losing accuracy
Overview

BigDFT - an introduction

What to do?
Minimisation loop
- The input guess
Post-minimisation
Parallelism

How to?
- storing the WF
- Real space operators

BigDFT library 1.0

Overview:
- input_wf_diag()
- compute density
- eXch-Corr. pot
- Hartree potential
- Kinetic operator
- Local potentials
- Non-local op.
- Preconditioner
- Stp. desc. / DIIS

Forces computation:
- projector first derivatives
- non local forces
- local forces

Kohn-Sham diag.

Legend:
- public routine
- Daubechies wvl.
- Interpolating wvl.