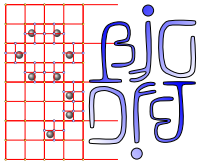


CECAM Tutorial on Wavelets in DFT

CECAM - LYON

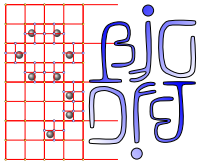
Poisson Solver, Pseudopotentials, Atomic Forces in the BigDFT code

Luigi Genovese

L_Sim - CEA Grenoble

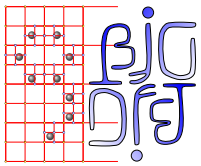
28 November 2007

Outline



1. The Poisson Solver with Interpolating Scaling Functions
   - Hartree Potential in DFT
   - Poisson Solver for Free BC
   - Interpolating scaling functions for electrostatic problems
   - Caclulation of the Poisson Kernel
   - Benchmarks and performances
   - A Poisson solver for surfaces BC
   - Performances
2. GTH-HGH pseudopoentials in BigDFT
   - Convenience of wavelet basis representation
3. Calculation of Atomic Forces
   - Treatment of the different terms

The Hartree potential



In the DFT calculation in the Kohn-Sham formalism we have:

Find a set of orthonormal orbitals $\Psi_i(\mathbf{r})$ that minimizes:

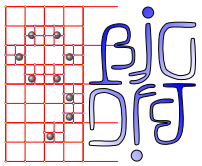
$$E = -\frac{1}{2}\sum_{i=1}^{N/2}\int \Psi_i^*(\mathbf{r})\nabla^2\Psi_i(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int \rho(\mathbf{r})V_H(\mathbf{r})d\mathbf{r} + E_{xc}[\rho(\mathbf{r})] + \int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$$

where

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

$$\nabla^2 V_H(\mathbf{r}) = -4\pi\rho(\mathbf{r})$$

The Poisson Solver in electronic structure calculation



During the minimization procedure we need to perform

Poisson's equation

Calculation of the self-consistent potential:

$$\nabla^2 V_H(\mathbf{r}) = -4\pi\rho(\mathbf{r})$$

Such equation should be solved at each minimisation iteration. Need of having an efficient and accurate formalism.

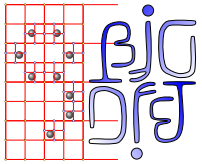
Plane waves approach

The most common approach. Uses the fourier components

$$f(x,y,z) = \sum_{p_x,p_y,p_z} e^{-2\pi i\left(\frac{p_x}{L_x}x+\frac{p_y}{L_y}y+\frac{p_z}{L_z}z\right)} f_{p_x,p_y,p_z}$$

The Poisson equation is algebraic in the Fourier coefficients

Poisson Solver with plane waves treatment



The Laplacian is diagonal in Plane waves representation

Immediate solution

$$V_{p_x,p_y,p_z} = \frac{1}{\pi} \frac{1}{\left(\frac{p_x}{L_x}\right)^2 + \left(\frac{p_y}{L_y}\right)^2 + \left(\frac{p_z}{L_z}\right)^2} \rho_{p_x,p_y,p_z} \ ,$$

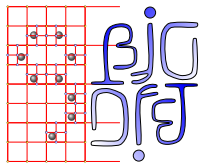
Characteristics

- Simple and fast, easy to parallelize (FFT)
- Automatically implement Periodic BC on a finite volume
- Do not fix the value of V_{p_x,p_y,p_z}

$\Rightarrow$ May result in problems for systems with other BC

How to solve this equation for other BC?

Isolated BC: the Green function treatment



Consider the Poisson equation for isolated BC. In this case the solution is given by

Green's function for Free BC (kernel)

$$\nabla^2 \frac{1}{r} = -4\pi\delta(r) \Longrightarrow V(\mathbf{r}) = \int \mathrm{d}\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}$$

An unambiguous solution

Such prescription is unique and is compatible with Free BC.

How to implement it?

Plane-wave based approaches:

- Truncated kernel, numerically or analytically
- Screening Functions

Approximate treatment, large box required.

Need of an accurate and efficient algorithm

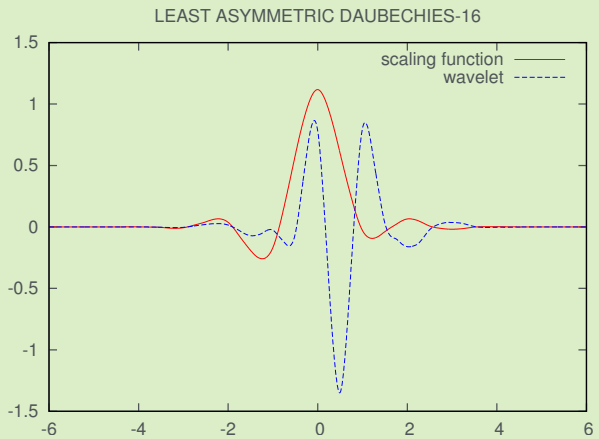
Poisson Solver with Interpolating Scaling Functions

A convenient basis for an electrostatic problem

Interpolating Scaling Functions

A set of localised functions centered on the nodes of a uniform mesh

$$\varphi_j(x) = \varphi_0(x-j)$$

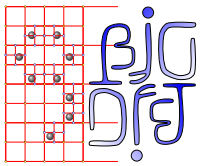


Undergo multiscale relation

$$\varphi(x) = \sum_{j=-m}^{m} \underbrace{h_j}_{\text{filters}} \varphi(2x-j)$$

- ✔ The expansion coefficients are the real space values $\rho_{j_x,j_y,j_z} = \rho(h_x j_x, h_y j_y, h_z j_z)$
- ✔ Represents exactly an order m polynomial
- ✔ The first m discrete and continuous moments coincide

A finite three-dimensional convolution



The expression of the potential in this basis is thus intuitive:

$$V(\mathbf{i}) = \sum_{\mathbf{j}} K_{\mathbf{ij}}\, \rho_{\mathbf{j}}$$

Where the central object is the

Poisson Kernel in the ISF basis

$$K_{\mathbf{ij}} = K_{\mathbf{i}-\mathbf{j}}\,, \qquad K_{\mathbf{i}} = \int K(|\mathbf{r}|)\varphi_{\mathbf{i}}(\mathbf{r})d\mathbf{r}\,, \quad K(r) = \frac{1}{r}$$

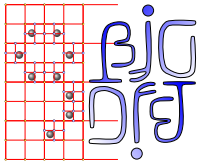
Values of the potential are obtained via a convolution

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

It can be treated via a zero-padded FFT algorithm

- ✔ Exact, easy to parallelize (FFT)
- ✔ For a box of N^3 points, it reduces the scaling from $O(N^6)$ to $O(N \log N)$

Characteristics of the approach



This approach is:

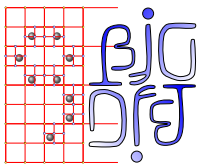
- Explicit, guarantees the good BC
- Real space-based, immediate interpretation of the expansion coefficients
- Can be combined with other real-space treatments of the density (e.g. XC) combined with ABINIT XC routines
- Can be used independently from the DFT code
- Preserves the first m (multipole) moments of the electrostatic potential

Requires only the evaluation of the kernel

We need to evaluate N^3 integrals

$$K_{\mathbf{i}} = \int K(|\mathbf{r}|)\varphi_{\mathbf{i}}(\mathbf{r})d\mathbf{r}$$

Gaussian tensor product decomposition



It can be shown that (Beylkin et *al.*)

Approximation with gaussians

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

with $k = 1, \cdots, 89$, p_k , ω_k suitably chosen

Accuracy of 10^{-8} for $r \in (10^{-9}, 1)$

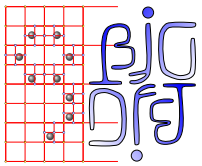
We can rescale for $R \in (0, L)$.

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

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

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

Other properties of the scaling functions



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

Moreover

The scaling property of the interpolets

$$\varphi_0(x/2) = \sum_j h_j \varphi_j(x) \ ,$$

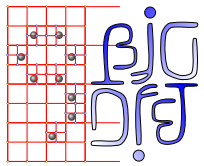
Implies similar condition of the one-dimensional function

$$K_j(4p) = \frac{1}{2} \sum_j \mathrm{h}_j K_{2i-j}(p) \ .$$

Thus we can evaluate the integrals for low p (not too sharp gaussians), then rescaling.

ISF properties allows us to gain in accuracy

Characteristics

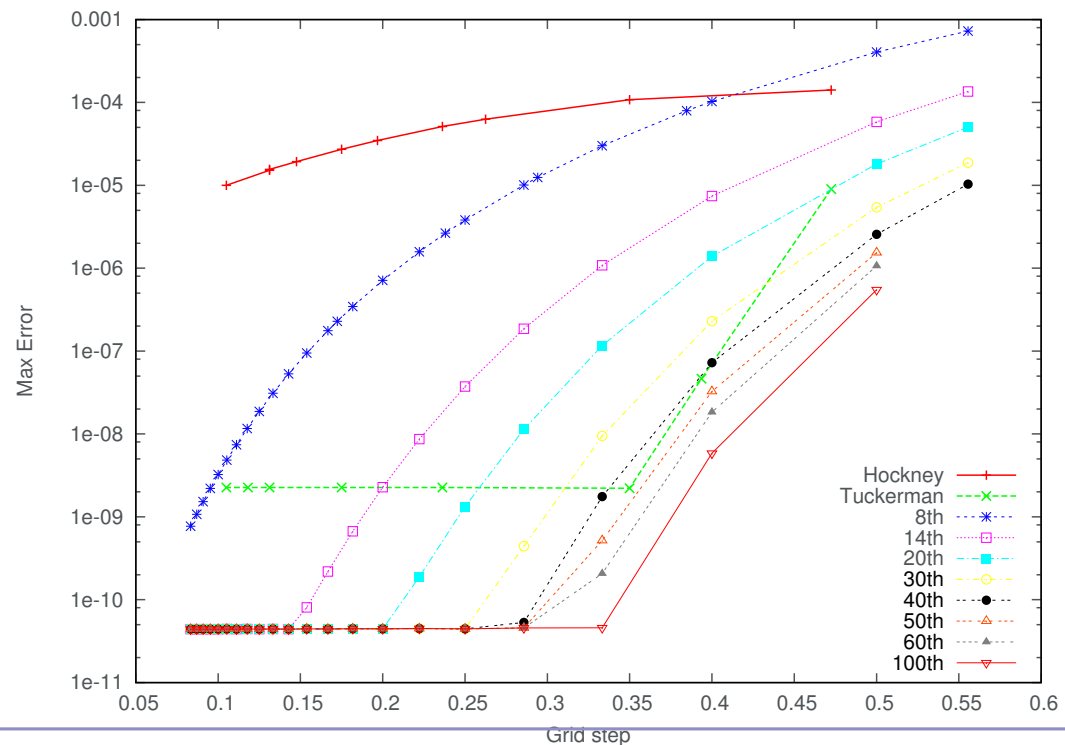


Very fast with moderate memory occupation:

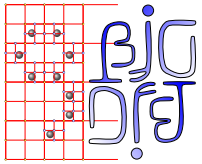
Elapsed Time for a 128^3 grid on a Cray XT3

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

More precise than other existing Free BC Poisson Solvers



Characteristics of the Solver



In summary, we have developed a technique

- Free boundary conditions
- Very high accuracy
- Good computational performance, easy to parallelize
- Can be used also in other contexts and/or combined with other treatment (e.g. XC)
- Coupled with ABINIT XC routines

In BigDFT code, for big systems it represents a small amount (typical values are less than 2-3% for big systems) of the overall computation

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 Surfaces boundary conditions

The same formalism can be applied to other BC

Surfaces BC

A domain isolated in one direction (say y) and periodic in x and z, with periods L_x and L_z. A function f which lives in such a domain can be expanded as

$$f(x,y,z) = \sum_{p_x,p_z} e^{-2\pi i\left(\frac{p_x}{L_x}x+\frac{p_z}{L_z}z\right)} f_{p_x,p_z}(y)$$

without any loss of generality.

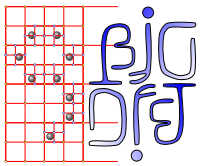
Mixed representation

For such functions the Poisson's equation become

$$\left(\partial_y^2 - \mu^2_{p_x,p_z}\right) V_{p_x,p_z}(y) = \rho_{p_x,p_z}(y)\,,$$

where $\mu^2_{p_x,p_z} = 4\pi^2(p_x/L_x)^2 + (p_z/L_z)^2$.

A Green's function formalism



The Green's function for the one-dimensional Helmoltz equation can be used

$$\left(\partial_y^2 - \mu^2\right) G(\mu; y) = \delta(y) \; ;$$

$$G(\mu; y) = \begin{cases} -\frac{1}{2\mu} e^{-\mu|y|} & \mu > 0 \\ \frac{1}{2}|y| & \mu = 0 \end{cases} ,$$

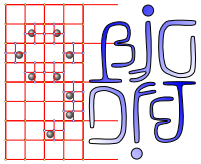
the components of the potential can be carried out:

A Green's function for each Fourier component

$$V_{p_x,p_z}(y) = \int \mathrm{d}y' \, G(\mu_{p_x p_z}; y - y') \rho_{p_x,p_z}(y') \; .$$

We can use Interpolating Scaling Functions for the Isolated direction.

Surfaces BC with Mixed representation



By defining the one-dimensional kernels

$$K(\mu_{p_x,p_z};j) = \int G(\mu_{p_x p_z};y)\varphi_{\mathbf{j}}(y)\mathrm{d}y$$

We can have a treatment similar to the Free BC case

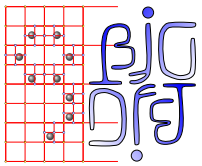
1-dim convolution for each reciprocal space component

$$V_{p_x,p_z}(i) = \sum_j K(\mu_{p_x,p_z};i-j)\rho_{p_x,p_z}(j)$$

From mixed representation to full real space

The calculation can be performed with a (semi-)zero-padded FFT algorithm. The I/O are function of the real domain.

Speed-up the calculation



The calculation of the kernel can be improved

Speed

The analytic form of the Green functions allows for recursion relations

$\longrightarrow$ improvement in speed of a factor of $(N_z + 1)/3$

Accuracy

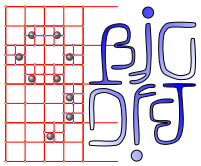
The scaling relation will again help in the kernel

$$K(2\mu; j) = \frac{1}{2} \sum_j \mathrm{h}_j K(2\mu, 2i - j)$$

A "Free-Lunch" case

We have the same advantages of the Free BC treatment with an explicit formalism conceived for surfaces BC

Poisson Solver for surface boundary conditions

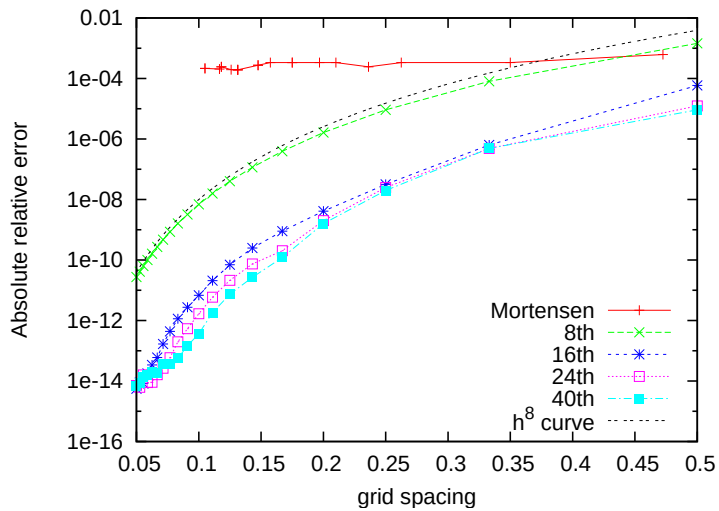


Poisson Solver, Pseudo, Atomic Forces

Elapsed Time on a Cray XT3, 128^3 grid

#proc	1	2	4	8	16	32	64
sec	.43	.26	.16	.10	.07	.05	.04

More precise than other treatments



L. Genovese, T. Deutsch, S. Goedecker

Efficient and accurate three dimensinal Poisson solver for surface problems, [arXiv: cond-mat/0703677], J. Chem. Phys. **127**, 054704 (2007)

A Poisson solver for surface problems

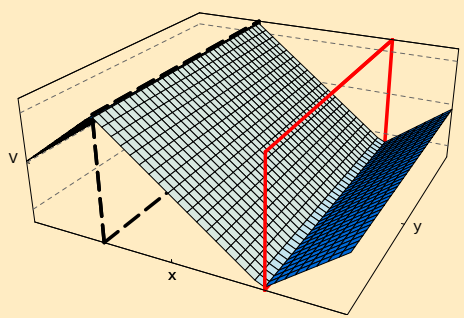
Like the Free BC case

We developed a technique

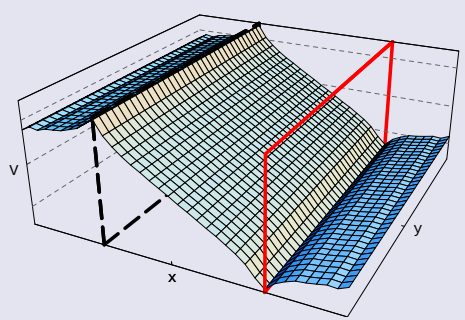
- ✔ Accurate and fast, easy to parallelize
- ✔ Can be applied both in real or reciprocal space codes
- ✔ Explicit treatment No supercell or screening functions
- ✔ More precise than other existing approaches
- ✔ Allows comparisons betweeen different backgrounds (charged systems)

Example of the plane capacitor:

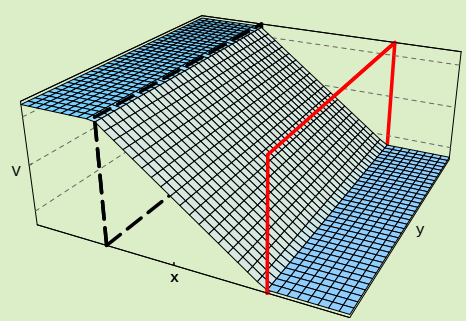
Periodic



Hockney



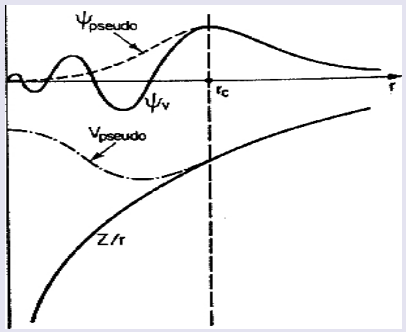
Our approach



Pseudopotential treatment

Smooth the wavefunction

Eliminate the rapid variations of the valence wavefunction in the core region. Thanks to them, only two levels of resolutions are enough for good accuracy



Pseudopotentials in BigDFT

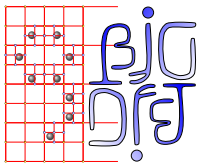
The pseudopotential used are of GTH-HGH type

$$V_{\text{loc}}(\mathbf{r}) = -\frac{Z_{\text{eff}}}{r}\frac{\text{erf}(\text{r})}{r} + e^{-\frac{1}{2}\left(\frac{r}{r_\ell}\right)} P\left[\left(\frac{r}{r_\ell}\right)^2\right]$$

$$V_{\text{nonloc}} = \sum_\ell \sum_{ij} \sum_{m=-\ell}^{\ell} h_{ij}^{(\ell)} |p_i^{\ell,m}\rangle\langle p_j^{\ell,m}|$$

$$\langle \mathbf{r}|p_i^{\ell,m}\rangle = Y_{\ell,m}(\theta,\varphi) f_i^{(\ell)}(r_\ell) e^{-\frac{1}{2}\left(\frac{r}{r_\ell}\right)}$$

Representation of the pseudopotential operators



These terms act differently on the energy calculation

Different treatment

- $E_{\text{loc}} = \int V_{\text{loc}}(\mathbf{r})\rho(\mathbf{r})\mathrm{d}\mathbf{r}$
  $V_{\text{loc}}(\mathbf{r})$ can be expressed in interpolating scaling functions basis (real space values)
- $E_{\text{nonloc}} = \sum_\ell \sum_{ij} \sum_{m=-\ell}^{\ell} h_{ij}^{(\ell)} \langle \psi | p_i^{\ell,m} \rangle \langle p_j^{\ell,m} | \psi \rangle$
  The important quantities are scalar products

The general term which appear in the application of the nonlocal operator is of the form

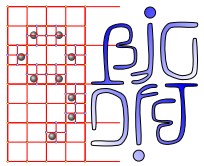
$$\int \mathrm{d}\mathbf{r}\psi(\mathbf{r})P(\mathbf{r})$$

where $P(\mathbf{r})$ is a gaussian times a separable polynomial

The projector is a tensor product

$$P(\mathbf{r}) = P_x(x)P_y(y)P_z(z)$$

Agrees with the wavefunction representation



For a wavefunction expressed in Daubechies wavelets

$$\psi(\mathbf{r}) = \sum_{i_x,i_y,i_z} c_{i_x,i_y,i_z} \phi_{i_x}(x)\phi_{i_y}(y)\phi_{i_z}(z)$$

the calculation of the scalar product can be decomposed

From N^3 to $3N$ calculations

$$\int \mathrm{d}\mathbf{r}\,\psi(\mathbf{r})P(\mathbf{r}) =$$

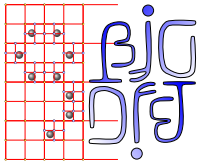
$$\sum_{i_x,i_y,i_z} c_{i_x,i_y,i_z} \int \phi_{i_x}(x)P_x(x)\mathrm{d}x \int \phi_{i_y}(y)P_y(y)\mathrm{d}y \int \phi_{i_z}(z)P_z(z)\mathrm{d}z$$

The numbers $\int \phi_{i_x}(x)P_x(x)$ are the expansion coefficients in Daubechies basis of the projectors.

Treatment simple and accurate

The application of the nonlocal projectors is easy and exact in the basis representation (no intrinsic approx.)

Atomic Forces



Suppose we find the ground state energy of an electronic system. The forces on an atom in the position **R** are

Gradient of the energy

$$F_i = -\frac{\mathrm{d}E}{\mathrm{d}R_i}$$

The DFT Hamiltonian is given in the Bohr-Oppenheimer approximation. For this approach it is valid the

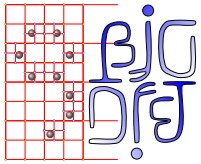
Feynman-Hellmann theorem

$$\frac{\mathrm{d}E}{\mathrm{d}R_i} = \left\langle \psi \left| \frac{\partial H}{\partial R_i} \right| \psi \right\rangle$$

for the normalised ground state eigenfunction

Equivalently, for this case we can write $F_i = -\frac{\partial E_{\mathrm{ext}}}{\partial R_i}$

Different terms



Which terms depend *explicitly* of the atomic positions?

- The external ionic energy:

$$E_{\text{ion}} = \frac{1}{2} \sum_{a \neq b} \frac{Z_a Z_b}{|\mathbf{R}_a - \mathbf{R}_b|}$$

- The local pseudopotential energy:

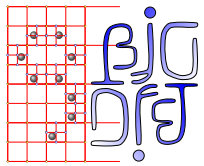
$$E_{\text{loc}} = \int \mathrm{d}\mathbf{r}\, V_{\text{loc}}\left(\mathbf{r}, \{\mathbf{R}_a\}\right) \rho(\mathbf{r})$$

- The nonlocal (separable) one:

$$E_{\text{nonloc}} = \sum_{i,j} h_{i,j} \left\langle \psi \middle| p_i(\mathbf{R}) \right\rangle \left\langle p_j(\mathbf{R}) \middle| \psi \right\rangle$$

Let us see the procedure used for each term separately

The calculation of each term



Immediate ionic force calculation

The contribution coming from the ionic energy is analytical

Local pseudopotentials

The local pseudopotential term can be separated in two parts:

$$V_{\mathrm{loc}}(\mathbf{r}) = -\frac{Z_{\mathrm{eff}}}{r}\frac{\mathrm{erf}(\mathrm{r})}{r} + e^{-\frac{1}{2}\left(\frac{r}{r_\ell}\right)} P\left[\left(\frac{r}{r_\ell}\right)^2\right] = V_e + V_g$$

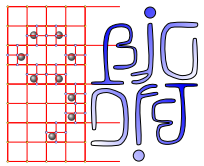
where

- V_g is a gaussian times a polynomial
- V_e is the solution of the isolated Poisson's equation

$$\nabla^2 V_e = -4\pi\rho_e$$

where $\rho_e \propto e^{-\frac{1}{2}\left(\frac{r}{r_\ell}\right)}$ is a gaussian

Local pseudopotential energy contribution



The contribution from the local term is thus

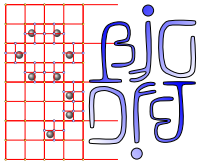
Local pseudopotentials

$$\begin{aligned}\mathbf{F}_a &= \int \vec{\nabla}_a V_g \rho + \vec{\nabla}_a \int \mathrm{d}\mathbf{r}\, V_e(\mathbf{r})\,\rho(\mathbf{r}) \\ &= \int \vec{\nabla}_a V_g \rho + \vec{\nabla}_a \int \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'\, \frac{\rho_e(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}\rho(\mathbf{r}) \\ &= \int \vec{\nabla}_a V_g \rho + \int \mathrm{d}\mathbf{r}\vec{\nabla}_a \rho_e(\mathbf{r})\, V_H(\mathbf{r})\end{aligned}$$

This approach

- Preserves locality
- Easy to parallelize
- Is of $O(n)$ scaling

Contribution from the Nonlocal pseudopotentials



Same treatment of the energy calculation

The projectors of the separable part are gaussian times polynomials (spherical harmonics). Their derivatives are still gaussian times polynomials. For each projector

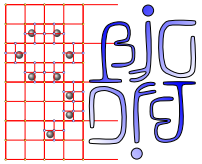
$$|p(\mathbf{R})\rangle$$

its derivative

$$\left|\frac{\partial p}{\partial \mathbf{R}}\right\rangle$$

is a function of the same kind and can thus be expressed with the same technique. We calculate the forces contribution by applying the Leibnitz rule to the energy expression

In Summary



We implemented the calculation of

$$F_i = -\frac{\partial E_{\text{ext}}}{\partial R_i}$$

where $E_{\text{ext}} = E_I + E_L + E_S$

Ionic

$$E_I = \frac{1}{2}\sum_{i \neq j} \frac{Z_i Z_j}{|\mathbf{R_{ij}}|}$$

Local

$$E_L = \int d\mathbf{r}\, V_{loc}(\mathbf{r}) \rho_{el}(\mathbf{r})$$

Separable

$$E_S = \sum_o n_o \langle \psi_o | H_S | \psi_o \rangle$$

Each term is calculated with a different method.
The complete implementation

- Preserves the locality (easy to parallelize)
- Compatible with the adaptivity
- The calculation can be performed with arbitrary accuracy (no intrinsic approximations)