Extension of the computation of density to non-diagonal band occupations with the KGB parallelization

T. Cavignac

CEA, DAM, DIF, 91297 Arpajon Cedex, France

École Centrale de Lyon

Abinit developer Workshop, Mai 2019

Table of Contents

1 paral_kgb and the repartition of data

2 Non local density, DMFT and DFT

3 Solving this incompatibility

4 Validation process and results

PARAL_KGB AND THE REPARTITION OF DATA

Reminder on Abinit parallelization

The `paral_kgb` mode imply:

- Parallelization over *k* vectors, which is trivial because they are always orthogonal.

And depending on the context one of:

- Parallelization over *plane waves* (*g*)
- Parallelization over *bands* (*b*)

Also in some cases there is locally parallelization over atoms, PAW projectors...

Computation of the density in DFT

$$n(\vec{r}) = \sum_{i \in \text{bands}} d_i \, \phi_i(\vec{r})^* \phi_i(\vec{r}) \tag{1}$$

The density have is easily done in real space with a few bands on each CPU.

Parallelization for the computation of the density (plane waves part)

1. Before `mkrho`, diagonalization of the hamiltonian ⇒representation in reciprocal space and plane waves components distributed (for linear algebra, named *linalg* layout).
2. Transposition of coefficients (gather plane waves, distribute bands, layout called *fft*)
3. Inverse Fourier transform (change the representation from reciprocal space to real space)
4. Density efficiently computed in real space with a natural parallelization over bands

Parallelization for the computation of the density (PAW part)

In PAW part:

- no plane waves by definition
- few components in the PAW base

Then band parallelization is the default. Bands are distributed among CPUs.

Non local density, DMFT and DFT

The DMFT, a non local density theory

- Dynamic Mean Field Theory (DMFT): one of the theory developed to address the problem of correlated electrons in transition metals and lanthanides
- deals with non-local density and is totally different from the DFT

How to integrate it with Abinit ?

- Define non-diagonal occupations of the Kohn-Sham vectors from the DFT
- Compute the density from these occupations and the Kohn-Sham vectors

From DFT density to DMFT+DFT density

Density expressed in terms of Kohn-Sham vectors and occupations:

$$n(\vec{r}) = \sum_{i \in \text{bands}} d_i \; \phi_i(\vec{r})^* \phi_i(\vec{r}) \tag{2}$$

With non-diagonals occupations

$$n(\vec{r}) = \sum_{i,i' \in \text{bands}} d_{i,i'} \; \phi_{i'}(\vec{r})^* \phi_i(\vec{r}) \tag{3}$$

We have now to compute products of vectors from different bands

Conclusion

Computation of the density from DFT+DMFT is not possible as is because a given CPU would need to access arbitrary pairs of bands at a time where bands are distributed over CPUs.

Solving this incompatibility

Solution for the plane wave part of the density

Take advantage of the initial state of data in `mkrho`

Goal

Temporarily modify the data to make it look like normal DFT data.

Let

$$\Psi(\vec{r}) = \begin{pmatrix} \phi_1(\vec{r}) \\ \vdots \\ \phi_N(\vec{r}) \end{pmatrix} \text{ and } F = (d_{i,i'})_{i,i' \in \text{bands}} \tag{4}$$

Rewrite (3)

$$n(\vec{r}) = \sum_{i,i' \in \text{bands}} d_{i,i'}\, \phi_{i'}(\vec{r})^* \phi_i(\vec{r}) = \Psi^*(\vec{r}) F \Psi(\vec{r}) \tag{5}$$

Diagonalising F (D diagonal and R unitary such that $F = R^* D R$) gives us

$$n(\vec{r}) = \Psi^*(\vec{r}) R^* D R \Psi(\vec{r}) = (R\Psi(\vec{r}))^* D (R\Psi(\vec{r})) \tag{6}$$

$\widehat{d}_i$ the coefficients of D (eigen values of F)
$\widehat{\phi}_i$ the rotated components of $R\Psi$ (rotated Kohn-Sham vectors)

$$n(\vec{r}) = (R\Psi(\vec{r}))^* D(R\Psi(\vec{r})) = \sum_{i \in \text{bands}} \widehat{d}_i \widehat{\phi}_i^{\,*}(\vec{r}) \widehat{\phi}_i(\vec{r}) \qquad (7)$$

The $\widehat{d}_i$ are our new occupations and the components $\widehat{\phi}_i$ are our new Kohn-Sham vectors.

Solution for the PAW part of the density

- Band distributed everywere ⇒no tricks this time

But:

- PAW components are really few compared to planewaves
- PAW density computation is rather light

A carefully crafted set of point-to-point MPI communications will do the job just well.

The algorithm is the following:

```
if the current CPU uses correlated bands then
    for each correlated band do
        if the band is available then
            Extract the data and put it in the buffer;
            for each remote CPU do
                if it uses correlated bands then
                    if it needs the current band then
                        send the band;
        else
            if this CPU need this band then
                receive the band from the CPU that own it and
                  put it in the buffer;
```

It prevents deadlocks and grant that data are available when they are used.

Some precision about the actual implementations:

- MPI communications are implemented as asynchrone communications they are initialized and then the computation can start with already available bands
- Since correlated bands form a block arround the Fermi level, not all CPUs are concerned
- This part could probably be optimized further but as we will see it does not worth it

Validation process and results

Validation process

- Comparison of the results with the new method and the old one at the 11th decimal of total energy in a few test cases with up to 100 steps
- Comparison of various intermediates quantities on the first iterations
- Comparison of the results with differents diagonalization algorithm for the Hamiltonian (LOBPCG, Chebychev, Conjugate gradient)
- Comparison of the results with differents CPU configurations
- Add of `paral[84]` and `paral[86]` to the testsuite

Results

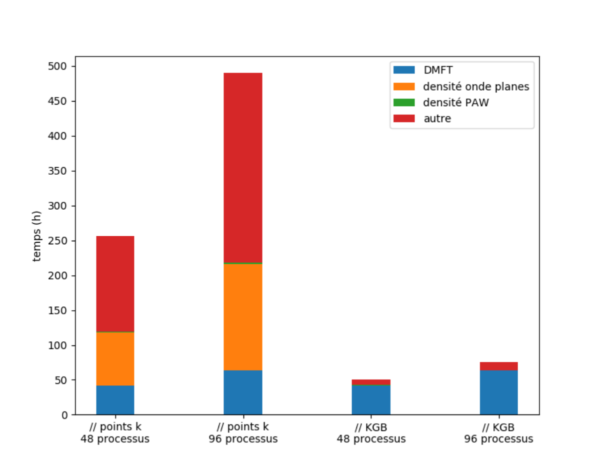

Figure: Drastic effect of the use of paral_kgb on a DMFT computation

Finally I want to express my gratitude to Marc Torrent, Jordan Bieder, for the time they gave me to answer my questions and facilitate my discovery of ABINIT, its code and its ecosystem. Of course, I also want to sincerely thank Bernard Amadon for his great supervision all along this work.

Thank you for your attention !

Questions ?