Correlations in sequences of generalized eigenproblems arising in Density Functional Theory\textsuperscript{*,**}

Edoardo Di Napoli\textsuperscript{a,c,*}, Stefan Blügel\textsuperscript{b,**}, Paolo Bientinesi\textsuperscript{b,***}

\textsuperscript{a}RWTH Aachen University, AICES, Schinkelstr. 2, 52062 Aachen, Germany
\textsuperscript{b}Peter Grünberg Institut and Institute for Advanced Simulation, Forschungszentrum Jülich and JARA, 52425 Jülich, Germany
\textsuperscript{c}Jülich Supercomputing Centre, Institute for Advanced Simulation, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

Abstract

Density Functional Theory (DFT) is one of the most used \textit{ab initio} theoretical frameworks in materials science. It derives the ground state properties of multi-atomic ensembles directly from the computation of their one-particle density \( n(\mathbf{r}) \). In DFT-based simulations the solution is calculated through a chain of successive self-consistent cycles; in each cycle a series of coupled equations (Kohn-Sham) translates to a large number of generalized eigenvalue problems whose eigenpairs are the principal means for expressing \( n(\mathbf{r}) \). A simulation ends when \( n(\mathbf{r}) \) has converged to the solution within the required numerical accuracy. This usually happens after several cycles, resulting in a process calling for the solution of many sequences of eigenproblems. In this paper, the authors report evidence showing unexpected correlations between adjacent eigenproblems within each sequence and suggest the investigation of an alternative computational approach: information extracted from the simulation at one step of the sequence is used to compute the solution at the next step.

The implications are multiple: from increasing the performance of material simulations, to the development of a mixed direct-iterative solver, to modifying the mathematical foundations of the DFT computational paradigm in use, thus opening the way to the investigation of new materials.

Keywords: Density Functional Theory, sequence of generalized eigenproblems, FLAPW, eigenvector evolution

1. Introduction

Density Functional Theory \cite{1,2} is a very effective theoretical framework for studying complex quantum mechanical problems in solid and liquid systems. DFT-based methods are growing as the standard tools for simulating new materials. Simulations aim at recovering and predicting physical properties (electronic structure, total energy differences, magnetic properties, etc.) of large molecules as well as systems made of many hundreds of atoms. DFT reaches this result by solving self-consistently a rather complex set of quantum mechanical equations leading to the computation of the \textit{one-particle density} \( n(\mathbf{r}) \), from which physical properties are derived.

In order to preserve self-consistency, numerical implementations of DFT methods consist of a series of iterative cycles: at the end of each cycle a new density is computed and compared to the one calculated in the previous cycle. The end result is a series of successive densities converging to a \( n(\mathbf{r}) \) approximating the exact density within the desired level of accuracy. Each cycle consists of a complex series of operations: calculation of full Coulomb and exchange-correlation potentials, basis functions generation, numerical integrations, generalized eigenproblems solution, and ground state energy computation. In one particular DFT implementation, namely the Full-potential Linearized Augmented Plane Wave (FLAPW) method \cite{4,5}, matrix entry initialization and generalized eigenvalue problem solution are the most time consuming stages in each iterative cycle (Fig. 1).

The cost for solving the mentioned stages is directly related to the number of generalized eigenproblems involved and to their size. Above a certain threshold, the eigenproblem size is proportional to the third power of the number of atoms of the physical system, while the number of eigenproblems ranges from a few to several hundred per cycle. Typically, each of the problems is dense and a significant fraction of the spectrum is required in order to compute \( n(\mathbf{r}) \). The nature of these eigenproblems forces the use of direct methods, such as those included in LAPACK or ScaLAPACK \cite{10,15}. All of the most common simulation codes implementing the FLAPW method (WIEN2k, FLEUR, FLAIR, Exciting, ELK \cite{11,12}), despite successfully simulating complex materials \cite{12,15}, treat each eigenproblem of the series of iterative cycles...
in isolation. This implies that no information embedded in the solution of eigenproblems at one cycle is used to speed up the computation of problems at the next. While these routines provide users with accurate algorithms to be used as black-boxes, they do not offer a mechanism for exploiting extra information relative to the application.

The line of research pursued here takes inspiration from the necessity of exploring a different computational approach in the attempt of developing a high performance algorithm specifically studied for the FLAPW method. The entire succession of iterative cycles making up a simulation can be seen as made up of a few dozen sequences of generalized eigenproblems. By mathematical construction, each problem in a sequence is, at most, weekly connected to the previous one. At odds with this observation, we present evidence showing that there is an unexpectedly strong correlation between eigenproblems of adjacent cycles in each sequence. We discuss how this extra information should be used to improve the performance of the current state-of-the-art routines, eventually leading to a computational scheme in which direct and iterative methods are used in different phases of the simulation.

In Sec. 2 we introduce the reader to the DFT framework and more specifically to the FLAPW method. We explain in more detail the series of self-consistent cycles, the computational bottle-necks inside each cycle, and how, from this picture, the importance of sequences of eigenproblems emerges. In Sec. 3 we illustrate the investigative tools we employ in studying the eigensequences, namely eigenvector evolution and unchanging matrix patterns. We then present our experimental results, and extensively discuss their interpretation from the numerical and physical point of view. Finally in Sec. 4 we draw our conclusions and explain how it is possible to exploit the experimental results to improve the performance of a DFT simulation.

2. Physical framework

DFT methods are based on the simultaneous solution of a set of Schrödinger-like equations [eq. (1)]. These equations are determined by a Hamiltonian operator $\hat{H}$ that, in addition to a kinetic energy operator, contains an effective potential $v_0[n]$, which functionally depends only on the one-electron particle density $n(r)$. In turn, the wave functions $\phi_i(r)$, which solve the Schrödinger-like equations for $N$ electrons, compute the one-electron particle density [eq. (2)] used in determining the effective potential. The latter is explicitly written in terms of the nuclei atomic Coulomb potential $v_C(r)$, a Hartree term $v_H(r, r')$ describing repulsions between pairs of electrons, and the exchange correlation potential $v_{xc}[n](r)$ summarizing all other collective contributions [eq. (3)]. This set of equations, also known as Kohn-Sham (KS) [2], is solved self-consistently. In other words the equations must be solved subject to the condition that the effective potential $v_0[n]$ and the electron density $n(r)$ mutually agree.

$$\hat{H}\phi_i(r) = \left( -\frac{\hbar^2}{2m} \nabla^2 + v_0(r) \right) \phi_i(r) = \epsilon_i \phi_i(r) \quad (1)$$

$$n(r) = \sum_i |\phi_i(r)|^2 \quad (2)$$

$$v_0(r) = v_C(r) + \int w(r, r')n(r')dr' + v_{xc}[n](r) \quad (3)$$

Computational implementations of DFT depend on the particular modeling of the effective potential and on the orbital basis used to parametrize the eigenfunctions $\phi_i(r)$. In the context of periodic solids, the vector $k$ and band $v$ indices replace the generic index $i$; the Bloch vector $k$ is an element of a three-dimensional Brillouin zone discretized over a finite set of values, called the set of $k$-points. In the FLAPW method [4, 5], the orbital function $\phi_{k,v}(r)$ are expanded in terms of a basis function set $\psi_G(k, r)$ indexed by vectors $G$ lying in the lattice reciprocal to the configuration space

$$\phi_{k,v}(r) = \sum_{|G+k| \leq K_{max}} c^G_{k,v} \psi_G(k, r). \quad (4)$$

In FLAPW, the configuration (physical) space of the quantum sample is divided into spherical regions – called Muffin-Tin (MT) spheres – centered around atomic nuclei, and interstitial areas between the MT spheres. Within the volume of the solid’s unit cell $\Omega$, the basis set $\psi_G(k, r)$ takes a different expression depending on the region

$$\psi_G(k, r) = \begin{cases} \frac{1}{\sqrt{A}} e^{(k+G)r} & \text{– Interstitial} \\ \sum_{l,m} a^{G}_{lm}(k) u^G_l(r) + b^{G}_{lm}(k) \dot{u}^G_l(r) \end{cases} Y_{lm}(\hat{r}_\alpha) \quad \text{– MT.}$$

For each atom $\alpha$, the coefficients $a^{G}_{lm}(k)$ and $b^{G}_{lm}(k)$ are determined by imposing continuity of the wavefunctions $\phi_{k,v}(r)$ and their derivatives at the boundary of the MT sphere. The $Y_{lm}(\hat{r}_\alpha)$ are spherical harmonics of the $\alpha$-atom, $\hat{r}_\alpha \equiv \frac{\hat{r}}{r_\alpha}$ is a unit vector, and $r_\alpha$ is the distance from the MT center. The radial functions $u^G_l(r)$ and their time derivatives $\dot{u}^G_l(r)$ are obtained by a simplified Schrödinger equation, written for a given energy level $E_i$, containing only the spherical part of the effective potential $v_{0ph}(r)$

$$\left\{ \frac{\hbar^2}{2m} \frac{\partial^2}{\partial r^2} + \frac{\hbar^2 (l+1)}{2m r^2} + v_{0ph}(r) - E_i \right\} \dot{u}^G_l(r) = 0. \quad (5)$$

Thanks to this expansion, the KS equations naturally translate to a set of generalized eigenvalue problems

$$\sum_{G'} \{ A_{GG'}(k) - \lambda_{kk'} B_{GG'}(k) \} c^G_{kk'} = 0 \quad (6)$$

where the coefficients of the expansion $c^G_{kk'}$ are the eigenvectors, while the Hamiltonian and overlap matrices $A$ and
Initialization:
non-interacting spherical $v_{0}\phi(r)$

Calculation of full-potential $v_{0}[n](r)$

Charge density for next cycle $n(r)$

Calculation of ground state energy $E_{n} \rightarrow n(r)$

Basis functions generation $\psi_{G}(k, r)$

Matrices generation $\{A(k), B(k)\} = \sum \int \psi_{G}^{*}(k, r)\{\hat{H}, \hat{1}\}\psi_{G}(k, r)$. (7)

Generalized eigenproblems $A(k)x = \lambda B(k)x$

Convergence check. Iteration setup

Eigenpairs selection

Figure 1: A schematic rendering of the principal stages of the self-consistent cycle. Colors indicate the computational cost of each stage of the cycle.

$B$ are given by volume integrals and a sum over all MT spheres

\[
\{A(k), B(k)\} = \sum \int \psi_{G}^{*}(k, r)\{\hat{H}, \hat{1}\}\psi_{G}(k, r).
\] (7)

In practical numerical computations, a solution is reached by setting up a multi-stage cycle (Fig. 1). An initial educated guess for $n(r)$ is used to calculate the effective full-potential $v_{0}[n]$ using the Pseudo-Charge [6] in combination with Fast Fourier Transform (FFT) methods. The potential, in turn, is inserted into the simplified Schrödinger equation (5) whose solutions, together with the coefficients $a^{G}_{\alpha m}(k)$ and $b^{G}_{\alpha m}(k)$, lead to the basis functions $\psi_{G}(k, r)$. The latter are used to calculate the entries of the matrices $A(k)$ and $B(k)$, an operation that requires the computation of three-dimensional spherical integrals [eq. (7)] for the non-symmetric part of the potential appearing in $\hat{H}$.

In the next stage the matrices just computed are the input in dozens to hundreds of generalized eigenvalue problems [eq. (6)] that are solved simultaneously. Each eigenproblem is of the form $Ax = \lambda Bx$, where both $A$ and $B$ are dense Hermitian matrices, $B$ is additionally positive definite, and $x$ and $\lambda$ form a sought-after eigenpair. In FLAPW-related applications, usually only a fraction of the lower part of the spectrum is computed and retained based on the Fermi energy value. The stored eigenpairs are then used to evaluate the ground state energy of the physical system, followed by the computation of a new charge density $n'(r)$.

At the end of the cycle, convergence is checked by comparing $n'(r)$ with $n(r)$. If $|n'(r) - n(r)| > \eta$, where $\eta$ is the required accuracy, a suitable mixing of the two densities is selected as a new guess, and the cycle is repeated. This process is properly referred to as an outer-iteration of the DFT self-consistent cycle. Convergence is guaranteed by the Hohenberg-Kohn theorem [3] stating that there exists...
a unique electron density \( n_0(r) \) locally minimizing an energy functional \( E[n] \) closely related with the Hamiltonian operator \( \hat{H} \).

In conclusion, the FLAPW self-consistent scheme is formed by a series of outer-iterations, each one containing multiple large generalized eigenproblems. In order to numerically compute the charge density \( n(r) \) at each iteration, the matrices \( A \) and \( B \) need to be initialized for each \( k \)-point and the generalized eigenproblem \( Ax = \lambda Bx \) solved. These two stages are the most machine-time consuming part of the cycle, each accounting between 40\% and 48\% of the total computational time (Fig. 1). Moreover, the more complex the material, the larger the matrices, and the slower the convergence, resulting in an increase in the number of outer-iterations.

### 3. An alternative viewpoint

The results presented in this paper originate from the deliberate choice of studying the DFT self-consistent cycle from a different perspective. The entire outer-iterative process is regarded as a set of sequences of eigenproblems \( P_i \). This interpretation is based on the observation that, for each \( k \)-point, the solution of a problem at a certain iteration \( P_i(k) \) is a prerequisite for setting up the next one \( P_{i+1}(k) \).

Considering the single eigenproblems \( P_i \) (the \( k \) index is suppressed for sake of simplicity) as part of a sequence \( \{P_i\} \) can have far reaching consequences: it might help unravel correlations among them, and ultimately suggest an entirely different computational approach to solve them as the simulation progresses. Since DFT is one of the most important \emph{ab initio} electronic structure frameworks, the study of a computational procedure that would lead to high-performance solutions to \( \{P_i\} \) is of crucial importance.

In order to study the evolution of the generalized eigenproblems as part of the sequence \( \{P_i\} \), we focus our attention on the transformation of eigenvectors and the variation of the matrix entries of the Hamiltonian matrix \( A \) (the same could be done for the overlap matrix \( B \)). For a fixed \( k \), each eigenvector at iteration \( i \) is compared with its corresponding eigenvector at iteration \( i + 1 \). A similar comparison is performed between the values of the entries of adjacent Hamiltonian matrices. Despite the simplicity of the strategy, the realization of a comparative tool that quantitatively describes the evolution of \( \{P_i\} \) is not a trivial matter.

#### 3.1. Eigenvector evolution

In this section we focus on a single sequence and describe a procedure to study the evolution of the eigenvectors solving for \( P_i \). The results obtained are independent of \( k \) and can be applied to any sequence in the simulation. In order to carry out our plan, we need an associative criterion that allows comparison between eigenvectors of successive iterations. This is not a simple task since the ordering of a set of eigenpairs can change substantially from one iteration to the next.

For instance, one can arrange the eigenvectors by the increasing magnitude of their respective eigenvalues and compare two eigenvectors, say \( x^{(i)}_\ell \) and \( x^{(i+1)}_\ell \), with the same eigenvalue index \( \ell \). This naive comparison is bound to fail due to the fact that eigenvalues close in magnitude often swap positions across iterations. Consequently, identifying eigenvectors becomes rather difficult as the sequence advances. In other words the iterative process interferes with the ability to find a one-to-one correspondence between vectors of neighboring iterations.

#### 3.1.1. Computational scheme

For a correct comparison between adjacent eigenvectors we developed an algorithm that establishes a univocal correspondence based on two observations: 1) a DFT simulation is basically a minimization procedure, and as such favors small eigenpair variations in its progress towards convergence, and 2) all eigenpairs contribute more or less “democratically” to the progression of the sequence. We noticed that these statements translate directly into two specific behaviors of the eigensolutions. First, scalar products between an eigenvector \( x^{(i)}_j \) at iteration \( i \) and any of the eigenvectors \( x^{(i+1)}_\ell \) at iteration \( i + 1 \) have a gaussian distribution narrowly peaked at around one value \( \sigma^{(i)}_j \).

Second, the set of largest scalar products, \( \{\sigma^{(i)}_j\} \), has a flat and almost constant distribution. In mathematical terms, they can be written as

\[
\forall j \not\equiv \ell : \quad \sigma^{(i)}_j = \langle x^{(i)}_j, x^{(i+1)}_\ell \rangle \gg \langle x^{(i)}_j, x^{(i+1)}_\ell \rangle_{\ell \neq \ell} \quad (8)
\]

\[
\forall (j_1, j_2) : \quad \frac{\sigma^{(i)}_{j_1} - \sigma^{(i)}_{j_2}}{\sigma^{(i)}_{j_1} + \sigma^{(i)}_{j_2}} \ll 1. \quad (9)
\]

These observations motivated the design of a routine that, far from being unique or optimized, succeed to correctly relate two successive eigenvectors. Specifically, it identifies, for each eigenvector \( x^{(i)}_j \), the largest scalar product \( \sigma^{(i)}_j \) subject to the condition that the \((i+1)\)-iteration index \( \ell \), associated with \( j \), is not associated with any other \( j' \neq j \). By construction, this procedure establishes a one-to-one correspondence between the eigenvectors of successive iterations, \( x^{(i)}_j \Leftrightarrow x^{(i+1)}_\ell \), whose information is stored in a permutation operator \( \Pi \)

\[
\forall j, \exists! \ell : \quad j = \Pi(\ell) \quad \text{and} \quad \forall \ell, \exists! j : \quad \ell = \Pi^{-1}(j). \quad (10)
\]

Using \( \Pi \), the positions of columns of the matrix of scalar products \( \langle x^{(i)}_j, x^{(i+1)}_\ell \rangle \) can be re-arranged so as to easily obtain the largest scalar products \( \{\sigma^{(i)}_j\} \) from its main
diagonal. From this matrix we can easily extract the subspace deviation angles, automatically normalized to one, between corresponding eigenvectors of adjacent iterations

$$\theta_{ij}^{(s)} = \text{diag}\left(1 - \langle x_j^{(s)}, x_{\Pi(i+1)}^{(s)} \rangle\right). \tag{11}$$

These angles provide the means for studying the evolution of the eigenvectors of the sequence of generalized eigenproblems \(\{P_i\}\).

Collecting all the angles computed in one simulation results in a large set of data (there are \(N = \dim(A)\) angles for each iteration and each \(k\)). For our statistical analysis, we manipulate the angles so as to plot them in three different ways depending on which parameter characterizing the data is kept fixed. First, fixing the iteration index and a specific eigenvalue, we look at how the angles are distributed among the \(k\)s. Then, we choose a random \(k\) and look at how all the deviation angles vary as the sequence progresses. Finally we select an eigenvalue and examine the evolution of the angles for all \(k\) as the iteration index increases.

In order to perform the entire computational process, from eigenvector pairing to deviation angle plots, we built a Matlab analysis toolkit. The input is the set of matrices \(A\) and \(B\) of all the eigenproblems appearing in the sequences of a simulation. Simulations of the physical systems analyzed were performed using the FLEUR code \(\text{[8]}\) running on JUROPA, a powerful cluster-based computer operating in the Supercomputing Center of the Forschungszentrum Jülich. For each physical system studied we produced outputs for a consistent range of parameters.

### 3.1.2. Experimental evidence

We present here a numerical study for two typical physical systems. The first one, bulk copper, is a noble metal with atoms positioned on a face-centered-cubic lattice. The second example, zinc oxide, is an ionic bonded material arranged on a wurtzite lattice – a multilayered hexagonal lattice with 2 Zn and 2 O atoms per unit cell. For each material we ran a simulation whose specifics are described in Table 1.

<table>
<thead>
<tr>
<th>Material</th>
<th># of k-points</th>
<th># of Iterations</th>
<th>Avg size of matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cu</td>
<td>110</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>ZnO</td>
<td>40</td>
<td>9</td>
<td>490</td>
</tr>
<tr>
<td>Fe50A</td>
<td>15</td>
<td>11</td>
<td>400</td>
</tr>
</tbody>
</table>

For each case study we plot a set of histograms showing, for all k-points, the distribution of angle deviations for the four smallest eigenvalues at a specifically chosen iteration (Fig. 2). In each histogram the distribution is sharply peaked on the lowest end of the interval and has null or negligible tails. This result supports our analysis on the “democratic” contribution of all angles to the progression of the sequence [eq. (11)].

Figure 2: Histogram showing a qualitative distribution of the deviation angles of each k-point eigenvector corresponding to the lowest four eigenvalues at a fixed iteration. All angles are normalized to one (1.00 = π/2). Above: the bulk copper case plotted at the 3rd iteration. Below: the zinc oxide case plotted at the 4th iteration.

For both physical systems we also show two diagrams, one that plots angles for all eigenvectors at one specific k-value and the other that select the angles of the \(j\)th eigenvector at each k-point (Fig. 3). Both graphs are plotted against the iteration index in semilog scale to better display the evolution of the deviation angles. We can immediately notice the almost monotonic decrease of the deviation angles as the sequences progress towards convergence. Small upward oscillations are probably due to an excess of localized charge that may cause a partial restart of the sequence. We have also observed that the angles corresponding to the lowest 20% of the spectrum are, on average, higher than the rest. Moreover, as can be seen from the bottom plot of Fig. 3, some k-points, for each se-
lected eigenvalue, have a larger angle evolution suggesting a slightly larger weight in influencing the simulation.

Figure 3: Eigenvector angles of successive iterations for copper: evolution of angles for all eigenvectors of the sequence corresponding to k-point 33 (above) and the evolution of angles for all 110 k-points of eigenvectors corresponding to the 7th lowest eigenvalue (below). All the angles are normalized to one ($1.00 = \pi/2$).

In all other multi-atomic systems studied, besides the ones shown here, the great majority of angles after the 3rd or 4th iteration are very small. Contrary to intuition the simulation is far from converged at this stage, implying again a sort of “democracy” of contribution, where all eigenvectors positively influence the process of minimizing the energy functional that depends on $n(r)$. This behavior has a universal character since we observed it in bulk, layer, metallic, or ionic material analyzed.

Table 2: Deviation Angles Means

<table>
<thead>
<tr>
<th>Material</th>
<th># or relevant eigs</th>
<th>$\bar{\theta}^{(3)}$</th>
<th>$\bar{\theta}^{(9)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cu</td>
<td>9 (18.8 %)</td>
<td>$1.13 \times 10^{-2}$</td>
<td>$2.70 \times 10^{-3}$</td>
</tr>
<tr>
<td>ZnO</td>
<td>27 (5.6 %)</td>
<td>$7.29 \times 10^{-2}$</td>
<td>$0.6 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

In order to give a more quantitative flavor of the eigenvector evolution we have tabulated the mean angle value $\bar{\theta}^{(i)}$ for an iteration at the beginning and one at the end of the simulation (Table 2). Due to their physical relevance, we used only deviation angles of those eigenvectors whose eigenvalues represent energies below Fermi level. As can readily be seen, the mean values at the end of the simulation are sensibly smaller than those at the beginning, confirming the described qualitative picture.

Figure 4: Eigenvectors angles of successive iterations for zinc oxide: evolution of angles for all eigenvectors of the sequence corresponding to k-point 21 (above) and the evolution of angles for all 40 k-points of eigenvectors corresponding to the 7th lowest eigenvalue (below). All the angles are normalized to one ($1.00 = \pi/2$).

3.2. Matrix entry variation

We systematically look at the variations in the entries in adjacent $A$ matrices for two main reasons. First, we plan to identify those portions of entries of $A$ that undergo little or no change at all in order to avoid re-calculating those entries at each iteration, and in doing so saving computing time. Second, we believe that the connection between successive eigenvectors should somehow surface in how much the matrices defining the eigenproblems vary across iter-
tions: both are the indirect consequence of changes in the set of basis wavefunctions \( \psi_G(k, r) \).

### 3.2.1. Computational scheme

Despite its manifest simplicity, comparison between matrix entries across adjacent iterations can be tricky. In fact, variations of the single entries of \( A_k^{(i)} \) with \( A_k^{(i+1)} \) span a range of several orders of magnitude and need to be rescaled opportunistically. Our strategy is to normalize all variations so as to map them onto a \([0, 1]\) interval. Subsequently we introduce a threshold parameter that cuts off all variations below a certain value. This strategy helps in identifying those areas of the matrices where the entries undergo relatively large variations; it also allows to study the percentage of entries that varies as a function of the cut-off value. Through this procedure, we can determine the value of the threshold that might be chosen for saving computing time without compromising the accuracy of the eigensolutions too much (i.e. speed vs accuracy).

First we had to establish the most appropriate “metric” to gauge the relative size of entry variation. The choice of the metric influences the mapping of the variations onto the specified interval. In this study we chose, for the metric, the maximal entry variation for each matrix difference \( \delta^{(i)} = \max(|A_k^{(i+1)} - A_k^{(i)}|) \) and normalized each entry of the difference with respect to it. The entries of the resulting matrix \( \tilde{A}_k^{(i)} = \frac{A_k^{(i+1)} - A_k^{(i)}}{\delta^{(i)}} \) are clearly mapped onto the \([0, 1]\) interval. Then the threshold is measured as a fraction of \( \delta_k \), and the cut-off value, \( p_t \), is represented by a number \( \in [0, 1] \). It has to be noted that, contrary to common intuition, the lower the cut-off value \( p_t \), the greater the number of non-zero entries of \( \tilde{A} \) is.

All \( \tilde{A}_k^{(i)} \), extracted from the simulation of a physical system, were analyzed, at a fixed \( k \), for different values of the cut-off and for different iteration levels \( i \). As for the eigenvector evolution, the input for our analysis is the set of matrices \( A \) that defines the eigenproblems appearing in sequences of a simulation. All the simulations of the physical systems were performed using the FLEUR code running on JUROPA.

### 3.2.2. Experimental evidence

We analyzed the simulation of a 5 layer film of iron, denoted by Fe$_5$, with a \((100)\) surface orientation modeled by a simple tetragonal lattice containing 5 atoms in the unit cell embedded in two semi-infinite vacui. The specific characteristics of this simulation are listed in Table \ref{table:simulation}. In Fig.\ref{fig:fig5} we give a qualitative picture of the portion of \( A \) that changes, for a specific k-point and iteration index. Observe that the empty portions of \( A \) preserve their shape and localization as the cut-off decreases. In other words those parts of \( A \) that do not vary seem to be almost independent from \( p_t \) and so have a sort of “universal” character. Despite the fact that the basis functions set \( \psi_G(k, r) \) changes substantially between successive iterations, it would seem that certain subsets of basis functions contribute very little to some of the volume integrals in eq.\ref{eq:7}.

![Figure 5: Visualization of \( \tilde{A}_k^{(i)} \) for \( k = 1 \) and \( i = 4 \), excerpted from a Fe$_5$ simulation. Above: plot of all the entries above the cut-off value \( p_t = 0.10 \). Below: plot of all the entries above the cut-off value \( p_t = 0.05 \).](image)

In Fig.\ref{fig:fig5} we give a more quantitative description of the number of matrix entries that change as the cut-off value changes. These plots lead us to two important conclusions: first, they show that only for \( p_t \leq 0.1 \) does the percentage of varied entries become significant. This behavior indicates that overall there are very few entries undergoing major changes; most of the variations are concentrated in the low end of the metric. This last observation suggests that different metrics can have a different impact on our analysis. Secondly, contrary to what is suggested by the eigenvector evolution, matrix entry variation doesn’t seem to decrease as the sequence of eigenproblems progresses.
We analyzed plots for $2 \leq i \leq 11$ (here we show only $i = 4$ and $i = 10$) and concluded they do not present substantial differences, indicating that patterns of variation do not change during the whole simulation. This is quite a surprising result signaling that unchanging patterns in $\{P_i\}$ and small eigenvector deviation angles may have different origins.

**Figure 6:** Percentage of varying matrix entries plotted versus cut-off values in semilogarithmic scale for the Fe$_8$ system. Above: $k = 1$ and $i = 4$. Below: $k = 1$ and $i = 10$.

Can the large number of unchanging entries be used to speed up computations at every DFT iteration? In order to answer this question it needs to be understood how the trade off between speed and accuracy depends on the choice of cut-off value and, even more importantly, on the choice of metric for the threshold. A conclusive answer is not possible at this stage and we refer the reader to future publications of our current research.

There is an observation to be made here. The percentages plotted in Fig. 6 take, as natural point of reference, the total number of non-zero entries of $A_k^{(i)}$. Due to numerical artifacts coming from subtractions of very small numbers, the latter is often doubly more dense than $A_k^{(i)}$. On the other hand, if we want to exploit the unchanging portion of $A_k^{(i)}$, it is its total number of non-zero entries that should be the point of reference. Thus, numerical artifacts should be taken into account and carefully eliminated when computing $A_k^{(i)}$.

### 4. Correlation and exploitation

In the previous section, we have deliberately analyzed DFT-based simulations from a non-conventional perspective. Stemming from the assumption that such simulations are formed by a set of sequences of eigenproblems $\{P_i(k)\}$, we provided experimental evidence suggesting a connection between problems that are adjacent. In particular, we uncovered a strong correlation between eigenvectors of successive problems. We illustrated how this correlation is strongly linked to the convergence process of the simulation: as the iteration index increases, the eigenvector deviation angles become, on the average, smaller.

Despite the fact that the problem $P_{i+1}$ at iteration $i + 1$ is determined by the orbital wave functions obtained by the solution of the problem $P_i$ at iteration $i$, such a correlation is unexpected. Because each $P_i$ is influenced by the basis functions $\psi_G(k, r)$ computed at each new iteration, there are two reasons to be startled: 1) matrix entries defining the eigenproblems are given by volume integrals involving basis functions [eq. (7)], and 2) the eigenvectors are the n-tuple of coefficients expressing orbital wave functions in terms of a linear combination of basis functions [eq. (8)]. As a consequence of these two observations, eigenvectors are, in principle, very loosely connected; this compels us to give great relevance to the evidence of a strong correlation.

While having found a strong eigenvector correlation is in itself an important result, it is even the more so because it opens the way to the exploration of new computational strategies. In particular, the performance of the entire DFT simulation can be improved by boosting the performance of the sequences of eigenproblems. The idea is to take advantage of repetitive patterns in the eigenvector (Hamiltonian and Overlap) and in the eigenvector evolution. In practice, the key element is to reuse, in the eigenproblems at iteration $i + 1$, numerical quantities calculated at previous iterations.

To this end we propose a 2-step process that combines direct and iterative methods at different phases of the sequence: the result is a sequence-solver that after a few initial iterations using a direct method switches to an iterative one until termination. In a first phase, we envision an automatic matrix entry reuse between adjacent eigenproblems to enhance the direct eigensolver. As described in eq. (10), the matrices $A$ and $B$ are the result of costly multiple numerical integrations, to the point that their generation is as expensive as the solution of the eigenproblem itself (Fig. 1). One could save on the generation cost by setting up an automatic process to identify patterns in the matrix structure that do not vary between successive iterations.

In the following stage, due to the quasi-collinearity of
eigenvectors after a few iterations, an iterative eigensolver is the natural method of choice, where the eigenvectors computed at one iteration would be used as a starting guess for the next. It is a well known fact that iterative methods are mostly used for sparse eigenproblems when the fraction of eigensolutions required is very small. In our case, in spite of the fact that we deal with dense problems, the choice of an iterative solver is dictated by its ability to solve simultaneously for a substantial portion of eigenpairs as required by the calculation of $n(r)$. This is part of a study that is underway and will be presented in a future publication.

5. Conclusions

The overall results described in this paper are the first example of a study that, starting from simulations, provides a method to analyze the possible improvements of the algorithmic realization of a mathematical model on which the simulations are based. This approach reverses the usual direction that goes from theoretical model to material simulations; it is an example of how to look at DFT as an inverse problem. As such we would like to refer to our approach as a “reverse simulation” method. This methodological viewpoint would have by far more important consequences than just improving the computational approach of the simulations: it would allow us to go beyond the conventional FLAPW method and create a more efficient mathematical paradigm.

6. Acknowledgements

We thank Dr. Daniel Wortmann, Dr. Gustav Bihlmayer and Gregor Michalicek for discussions and their help in dealing with the FLEUR code. The computations were performed under the auspices of the Jülich Supercomputing Centre at the Forschungszentrum Jülich, whose generous support of CPU time is hereby acknowledged. We would also like to thank the AICES graduate school for hosting some of the authors and contributing to the success of the project.

Financial support from the following institutions is gratefully acknowledged: the JARA-HPC through the Midterm Seed Funds 2009 grant, the Deutsche Forschungs gemeinschaft (German Research Association) through grant GSC 111, and the Volkswagen Foundation through the fellowship “Computational Sciences”.

References

WIEN2k - http://www.wien2k.at/
The Jülich FLEUR project - http://www.flapw.de
The Exciting Code - http://exciting-code.org/
The Elk Code Manual (Ver. 1.2.20) - http://elk.sourceforge.net/
Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.