# Matrix Structure Exploitation in Generalized Eigenproblems Arising in Density Functional Theory

E. Di Napoli and P. Bientinesi

RWTHAACHEN
UNIVERSITY

AICES

List of AICES technical reports: http://www.aices.rwth-aachen.de/preprints

# Matrix Structure Exploitation in Generalized Eigenproblems Arising in Density Functional Theory

Edoardo Di Napoli and Paolo Bientinesi

*RWTH Aachen, AICES, Schinkelstr. 2, 52062 Aachen, Germany*

**Abstract.** In this short paper, the authors report a new computational approach in the context of Density Functional Theory (DFT). It is shown how it is possible to speed up the self-consistent cycle (iteration) characterizing one of the most well-known DFT implementations: FLAPW. Generating the Hamiltonian and overlap matrices and solving the associated generalized eigenproblems $Ax = \lambda Bx$ constitute the two most time-consuming fractions of each iteration. Two promising directions, implementing the new methodology, are presented that will ultimately improve the performance of the generalized eigensolver and save computational time.

**Keywords:** Generalized Eigenproblems, Density Functional Theory, Matrix Analysis, Iterative Solvers, Self-consistent cycle
**PACS:** 02.60.Dc, 02.60.Pn, 07.05.Kf, 31.15.E-, 71.15.Mb

## INTRODUCTION TO DENSITY FUNCTIONAL THEORY

Density Functional Theory (DFT) [1] is a powerful method of investigation that has become the "standard model" of material science. DFT is one of the most effective frameworks for studying complex quantum mechanical systems. DFT-based methods are growing as the standard tools for simulating new materials. The core of the method relies on the simultaneous solution of a set of Schrödinger-like equations. These equations are determined by a Hamiltonian operator $\hat{H}$ containing an effective potential $v_0[n]$ that depends functionally on the one-particle electron density $n(\mathbf{r})$. In turn, the wave functions $\phi_i(\mathbf{r})$, which solve for the equations, compute the one-particle electron density used in determining the effective potential. The latter depends explicitly on the external atomic Coulomb potential $v_I$ and the Hartree term $w(\mathbf{r},\mathbf{r}')$ describing interactions between electrons.

$$
\begin{aligned}
&v_0([n],\mathbf{r}) = v_I(\mathbf{r}) + \int w(\mathbf{r},\mathbf{r}')n(\mathbf{r}) + v_{xc}([n],\mathbf{r}) \\
&\hat{H}\phi_i(\mathbf{r}) = \left(-\frac{\hbar^2}{2m}\nabla^2 + v_0(\mathbf{r})\right)\phi_i(\mathbf{r}) = \varepsilon_i\phi_i(\mathbf{r}) \quad ; \quad \varepsilon_1 \leq \varepsilon_2 \leq \dots \\
&n(\mathbf{r}) = \sum_i^N |\phi_i(\mathbf{r})|^2
\end{aligned}
\tag{1}
$$

In practice, this set of equations, also known as Kohn-Sham (KS) [2], is solved self-consistently; an educated guess for $n(\mathbf{r})$ is used to calculate the effective potential $v_0$ that, in turn, is inserted in the Schrödinger-like equations whose solutions, $\phi_i(\mathbf{r})$, are used to compute a new charge density $n'(\mathbf{r})$. Convergence is checked by comparing the new density to the starting one. If convergence is not reached, an opportune mixing of the two densities is selected as a new guess, and the process is repeated. This is properly called an outer-iteration of the DFT self-consistent cycle.

In principle, the theory only requires as input the quantum numbers and the positions of the atoms constituting the investigated system. In practice, DFT implementations depend on the particular modeling of the atomic structure and the orbital basis used to parametrize it. The Full-potential Linearized Augmented Plane Wave (FLAPW) method [3, 4] is one of the many techniques implementing DFT that is based on plane wave expansion of $\phi_{\mathbf{k},v}(\mathbf{r})$, where the Bloch vector $\mathbf{k}$ and the band index $v$ replace the generic index $i$. The Bloch wave function $\phi_{\mathbf{k},v}(\mathbf{r}) = \sum_{|\mathbf{G}+\mathbf{k}|\leq \mathbf{K}_{max}} c_{\mathbf{k},v}^{\mathbf{G}}\phi_{\mathbf{G}}(\mathbf{k},\mathbf{r})$ is expanded in terms of a basis set $\phi_{\mathbf{G}}(\mathbf{k},\mathbf{r})$ indexed by vectors $\mathbf{G}$ lying in the lattice reciprocal to configuration space. 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. The basis set $\phi_{\mathbf{G}}(\mathbf{k},\mathbf{r})$ takes a different expression depending on the region

$$
\begin{cases}
e^{i(\mathbf{k}+\mathbf{G})\mathbf{r}} & \text{Interstitial} \\
\sum_{l,m}\left[a_{lm}^{\alpha,\mathbf{G}}(\mathbf{k})u_l^{\alpha}(r) + b_{lm}^{\alpha,\mathbf{G}}(\mathbf{k})\dot{u}_l^{\alpha}(r)\right]Y_{lm}(\hat{\mathbf{r}}_{\alpha}) & \text{Muffin Tin}
\end{cases}
\tag{2}
$$

where the coefficents $a_{lm}^{\alpha,\mathbf{G}}(\mathbf{k})$ and $b_{lm}^{\alpha,\mathbf{G}}(\mathbf{k})$ are determined by imposing continuity of $\phi_{\mathbf{G}}(\mathbf{k},\mathbf{r})$ and its derivative at the boundary of MT. Due to this expansion the KS equations naturally translate to a set of generalized eigenvalue problems $\sum_{\mathbf{G}'}[A_{\mathbf{G}\mathbf{G}'}(\mathbf{k}) - \lambda_{\mathbf{k}\nu}B_{\mathbf{G}\mathbf{G}'}(\mathbf{k})]c_{\mathbf{k}\nu}^{\mathbf{G}'} = 0$ for the coefficients of the expansion $c_{\mathbf{k}\nu}^{\mathbf{G}'}$ where the Hamiltonian and overlap matrices $A$ and $B$ are given by multiple integrals and sums

$$\{A(\mathbf{k}),B(\mathbf{k})\} = \sum \iint \phi_{\mathbf{G}}^*(\mathbf{k},\mathbf{r})\{\hat{H},\hat{\mathbb{1}}\}\phi_{\mathbf{G}'}(\mathbf{k},\mathbf{r}). \tag{3}$$

In conclusion, the core of the FLAPW self-consistent scheme is formed by a sequence of outer iterations, each one containing multiple large dense generalized eigenpencils. In order to numerically compute the charge density $n(\mathbf{r})$ at each iteration, the matrices $A$ and $B$ need to be initialized for each $\mathbf{k}$-point and the generalized eigenproblem $Ax = \lambda Bx$ solved. These two computing instances are the most machine-time consuming part of each iteration. It is our intention in this short paper to present some preliminary results and future ideas to reduce the computational time and improve the performance of the self-consistent cycle.

## A NEW COMPUTATIONAL PHILOSOPHY

It is important to look at the entire DFT iterative process as a series of correlated problems $\{P_i\}$: it starts from an initial charge density $n_0(\mathbf{r})$, executes a series of iterations $P_1 \dots P_i P_{i+1} \dots P_N$ and converges to a final density $n_f(\mathbf{r})$. The problems are correlated since the output of a certain iteration $P_i$ is used as input for the next one $P_{i+1}$. Current codes that implement FLAPW (i.e. FLEUR, WIEN, FLAIR, Exciting, ELK) follow a simple approach: 1) compute every mathematical quantity involved in the solution of each $P_i$, 2) use libraries to solve the generalized eigenproblems as black boxes. This approach allows for standard and accurate calculations but is quite time-consuming and often sub-optimal. Our philosophy follows a counter-intuitive path: increase the number of iterations $\tilde{P}_i$, $i = 1 \dots M$, $M \gg N$ but make them fast so that the overall process $\{\tilde{P}_i\}$ is faster than $\{P_i\}$.

We plan to achieve the target by acting on the two most expensive sections of the self-consistent cycle, namely the matrix initialization and the eigenproblem solutions. Based on the observation that a considerable amount of matrix entries do not change between iterations, we want to save on computations that update the Hamiltonian $A$ and the overlap matrix $B$. At the same time we will show that eigenvectors from one iteration can be used to compute the eigenpairs of the next. Clearly, using information from previous iterations will speed up execution time of the single $\tilde{P}_i$. It will also make each $\tilde{P}_i$ less accurate, requiring a higher number of iterations in order to reach convergence. In the end we believe that the advantage gained by speeding up each single $\tilde{P}_i$ will overcome the disadvantage of having a larger number of iterations leading to a net saving for the set of correlated problems $\{\tilde{P}_i\}$.

We have conducted a preliminary study running a DFT process using the FLEUR code [5] to simulate two simple metallic systems: bulk copper (Cu_b) and an iron multilayer (Fe_l). The simplicity of the systems allows for a rather small matrix size M, where M$\simeq$50 for Cu_b and M$\simeq$400 for Fe_l. For the copper system, the simulation starts converging rapidly after the 4-5th iteration while the iron system starts converging after the 8-9th iteration. For each step of the two convergence processes we have collected the $A$ and $B$ matrices for each $\mathbf{k}$-point and analyzed them using either a specifically designed MatLab routine or existing libraries.
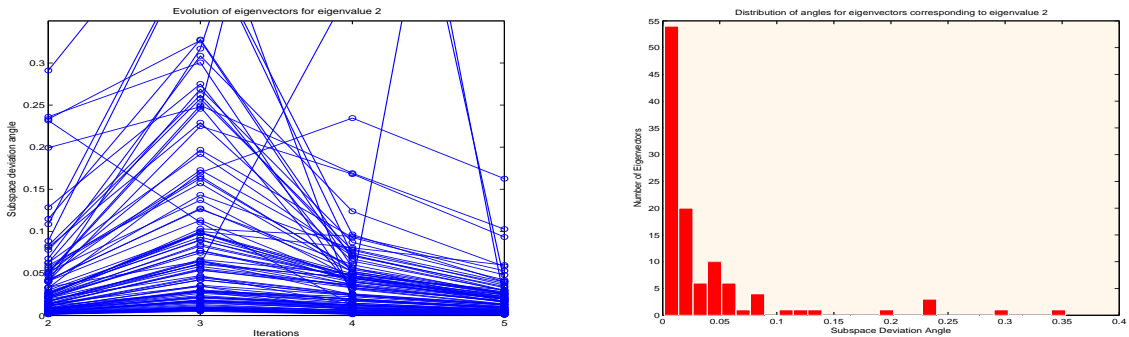


**FIGURE 1.** Cu_b $N = 5$ case. On the left: the evolution of the subspace deviation angle for each of the $\mathbf{k}$-point eigenvectors corresponding to the 2nd lowest eigenvalue. Angles exceeding graph borders correspond to degenerate eigenspaces. On the right: distribution of deviation angles for the same set of eigenvectors limited to the second iteration. All the angles are normalized to one ($1.00 = \frac{\pi}{2}$).

**Eigenvectors evolution** — In order to study how eigenvectors evolve from one iteration to the next, we needed to establish a criterion that allows comparison of each eigenvector of $A$ at iteration $i$ with its corresponding eigenvector at the following iteration $i+1$. This is not a trivial task since the indexing of a set of eigenvalues, ordered by magnitude, can change substantially from one iteration to the next. The indexing of the eigenvectors meets the same fate, interfering with the ability to find a one-to-one correspondence between vectors of neighboring iterations. On the contrary, subspace deviation angles between corresponding eigenvectors of adjacent iterations should take comparable values and be uniformly distributed. Taking advantage of the latter observation, we designed an angle minimization procedure that made it possible to re-index the eigenvectors of the new iteration. The deviation angles were then computed from simple scalar products between re-ordered eigenvectors of adjacent iterations and were used as indicators of the evolution of the vectors.

We systematically looked at the deviation angles of a generic eigenvector $x(\mathbf{k}, \lambda_i, n)$, corresponding to eigenvalue $\lambda_i$, as it evolves from iteration $n = 1$ to iteration $n = N$. In all of the cases analyzed, the angles start decreasing monotonically after the first 2-3 iterations until they become almost negligible when the process $\{P_i\}$ is about to converge. Even at $n = 2, 3$ the average of the deviation angle among all the eigenvectors for all $\mathbf{k}$-points is quite small $\langle \theta_{dev} \rangle = 0.04$ ($1.00 = \frac{\pi}{2}$) and its distribution quite peaked ($\sigma_{\theta_{dev}} = 0.135$) (Fig. 1).

Based on this study we have implemented a process based on our new computational philosophy. We start with few initial standard iterations $P_1 \ldots P_i \quad i \ll N$ computed with the full direct eigensolver so as to allow the simulation to bypass the phase with relatively large deviation angles. Then a series of fast iterations $\tilde{P}_i \ldots \tilde{P}_{j-1}$ are carried out using the eigenvectors from prior iterations to initialize an iterative eigensolver. This procedure continues until progress towards convergence flattens out. At this point we re-start with a full data iteration $P_i$ in order to refresh the process, followed by another series of fast iterations $\tilde{P}_{j+1} \ldots \tilde{P}_{k-1}$, and so on.

We have tested this procedure on both systems Cu_b and Fe_l using the LAPACK [6] QR algorithm for the eigenproblems of the $P$ iterations and the ARPACK [7] Implicit Restarted Arnoldi Method (IRAM) for eigenproblems of the fast iterations $\tilde{P}$. The results are somewhat encouraging but also show that the blind use of a Krylov process to implement the fast iterations does not work in all cases. In fact, while in the Fe_l case the $\tilde{P}$s were more than five times faster than $P$s, the opposite was true for the Cu_b case. These results suggest a more careful approach should be used in implementing the fast iterations.

**Matrix generation** — In a second set of experiments, we compared Hamiltonian matrices $A$ at the same $\mathbf{k}$-point between adjacent iterations. The objective was to understand how much variation the matrix entries would undergo from one cycle to the next and how much this variation depends on the progress towards convergence of the entire DFT process. A favorable answer to this question would allow us to apply the new computational philosophy in the same form expressed for eigenvectors evolution. The difference would be that now the series of fast iterations $\tilde{P}_i \ldots \tilde{P}_{j-1}$ would be represented by generalized eigenproblems involving matrices where only a fraction of the entries have been updated.
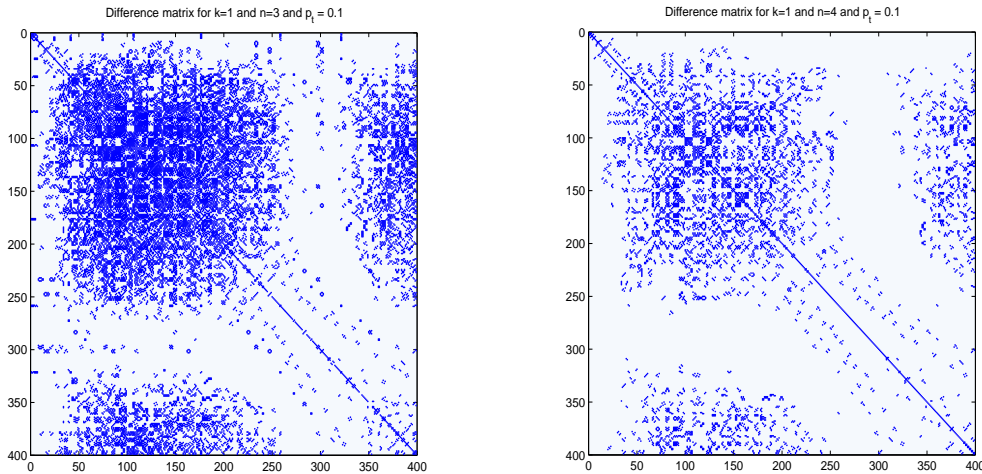


**FIGURE 2.** Fe_l $N = 11$ case. On the left: plot of $\tilde{A}(\mathbf{k} = 1, n = 3)$ for $p_t = 0.1$ On the right: plot of $\tilde{A}(\mathbf{k} = 1, n = 4)$ for $p_t = 0.1$. Cross-like unchanging patterns are visible in both plots. At fixed $p_t$ the amount of entries above threshold has clearly decreased from $n = 3$ to $n = 4$.

First we had to establish the most appropriate "metric" to gauge when a variation is substantial and when it is not. In the end we chose as a metric the maximal entry variation for each matrix difference $\delta_{n+1} = \max(A(\mathbf{k}, n+1) - A(\mathbf{k}, n))$ and normalized the entries of the difference with respect to it. The resulting matrices $\tilde{A}(\mathbf{k}, n+1) = \frac{A(\mathbf{k},n+1) - A(\mathbf{k},n)}{\delta_{n+1}}$ were analyzed for different values of the variation threshold and for different values of the iteration level $n$. For the sake of conciseness, we present here some sample results taken only from the Fe_l system. Similar results have been obtained for the Cu_b and other simple crystals.

Our qualitative study shows that no matter how low the threshold value $p_t$ (measured in percentage of $\delta_{n+1}$) is, the set of difference matrices $\{\tilde{A}(\mathbf{k}, n)\}$ shows extended areas where entries do not vary considerably. Moreover, as we progress to higher iteration levels $n \to N$, the number of entries of $\tilde{A}(\mathbf{k}, n)$ that undergo a variation above threshold decreases monotonically (Fig. 2). This behavior is expected since, as we advance towards convergence, the set of basis functions $\phi_{\mathbf{G}}(\mathbf{k}, \mathbf{r})$ needs lesser refinements in order to minimize the ground energy of the quantum mechanical system. In conclusion, it seems that the Hamiltonians (we expect a similar behavior from the overlap matrices) follow a universal behavior independent from the physical system under consideration. This fact implies that we can save a great deal of computational time if we avoid updating the corresponding matrices. For instance if, in the Fe_l case, we refrained from updating the entries of $A(\mathbf{k}, 4)$ for a threshold value $p_t = 0.1$, we would save computing about 40% of the entries - quite a substantial portion of the total time spent generating the entire matrix.

## SUMMARY AND CONCLUSIONS

In this short paper we illustrated a new computational approach to DFT-based methods. Two promising research directions emerge from our preliminary study: exploit limited eigenvectors evolution and save on matrix entry generation. Several steps still require investigation in this line of research. Current work on eigenvectors evolution is focusing on building a set of tailored algorithms that would efficiently use the information provided by eigenvectors of prior iterations. The target is to achieve consistency even for physical systems of small dimensions or special geometry. Major candidates, in this line of research, are a new eigensolver based on the Sakurai-Sugiura [8] method and an improved version of the subspace iteration method. In matrix generation we are working to clarify the relation among threshold value $p_t$, speed up of the single iteration $\tilde{P}$, and the accuracy of solutions of the generalized eigenvalue problems. It is crucial that the fast iterations not loose too much accuracy in their advance towards convergence while gaining in speed. A clear ratio between speed up and progress of the entire process needs to be set and correlated with the choice of threshold value for each iteration. In a following phase we want to develop an automated self-tuning slider that adjusts the value of $p_t$ in order to optimize the trade-off between speed and accuracy.

Last but not least, it is evident that there are clear patterns in the unchanging portions of the matrices. This fact strongly points to a more profound explanation that is likely connected to the physics embedded in the self-consistent cycle. If we can uncover the physical rationale behind the unchanging patterns, we can directly influence the choice of orbital functions and improve the DFT method before the numerical computation is initiated. This would constitute a very important contribution to any DFT-based method reversing the usual flow of information that goes from theory to computation. It would establish a unique example where computer simulations provide insight into theoretical models by simple analysis of the numerical inputs and outputs.

## ACKNOWLEDGMENTS

## REFERENCES

1. R. M. Dreizler, and E. K. U. Gross, *Density Functional Theory*, Springer-Verlag, 1990.
2. W. Kohn, and L. J. Sham, *Phys. Rev.* **A 140** 1133 (1965).
3. A. J. Freeman, H. Krakauer, M. Weinert, and E. Wimmer, *Phys. Rev.* **B 24** 864 (1981).
4. A. J. Freeman, and H. J. F. Jansen, *Phys. Rev.* **B 30** 561 (1984).
5. The Jülich FLEUR project, *http://www.flapw.de*.
6. LAPACK – Linear Algebra PACKage, *http://www.netlib.org/lapack/*.
7. ARPACK – ARnoldi PACKage, *http://www.caam.rice.edu/software/ARPACK/*.
8. T. Sakurai, and H. Sugiura, *J. Comp. Appl. Math.* **159** 119 (2003).