



Article Performance Enhancement of APW+lo Calculations by Simplest Separation of Concerns

Long Zhang ^{1,2,3}, Anton Kozhevnikov ⁴, Thomas Schulthess ⁴, Hai-Ping Cheng ^{1,2,3} and Samuel B. Trickey ^{1,2,3,*}

- ¹ Department of Physics, University of Florida, Gainesville, FL 32611, USA; longzhang2008@gmail.com (L.Z.); hping@ufl.edu (H.-P.C.)
- ² Quantum Theory Project, University of Florida, Gainesville, FL 32611, USA
- ³ Center for Molecular Magnetic Quantum Materials, University of Florida, Gainesville, FL 32611, USA
- ⁴ Swiss National Supercomputing Centre, 6900 Lugano, Switzerland; kozhevnikov@cscs.ch (A.K.); schulthess@cscs.ch (T.S.)
- * Correspondence: trickey@qtp.ufl.edu

Abstract: Full-potential linearized augmented plane wave (LAPW) and APW plus local orbital (APW+*lo*) codes differ widely in both their user interfaces and in capabilities for calculations and analysis beyond their common central task of all-electron solution of the Kohn–Sham equations. However, that common central task opens a possible route to performance enhancement, namely to offload the basic LAPW/APW+*lo* algorithms to a library optimized purely for that purpose. To explore that opportunity, we have interfaced the Exciting-Plus ("EP") LAPW/APW+*lo* DFT code with the highly optimized SIRIUS multi-functional DFT package. This simplest realization of the separation of concerns approach yields substantial performance over the base EP code via additional task parallelism *without* significant change in the EP source code or user interface. We provide benchmarks of the interfaced code against the original EP using small bulk systems, and demonstrate performance on a spin-crossover molecule and magnetic molecule that are of size and complexity at the margins of the capability of the EP code itself.

Keywords: LAPW method; APW+lo method; all-electron DFT

1. Dedication

Much credit for the widespread use of full-potential linearized augmented plane wave (LAPW) methodology to solve the Kohn–Sham (KS) [1] equation for solids goes to Karlheinz Schwarz. The history of that contribution is related in Section 3 of Ref. [2]. It suffices to say here that Heinz started using the original APW in his thesis work, then came to Gainesville and Quantum Theory Project (QTP) in 1969 to work with Prof. J.C. Slater, the inventor of the APW method. That is how the last author of this paper became a collaborator with Heinz and a friend.

Years later, when linearized methods removed the explicit energy dependence difficulty in the APW basis, Heinz undertook development of the code that became WIEN [3]. Again there was a collaboration involving QTP, University of Florida, and the last author. Apparently that was the first publicly available FLAPW code. By now, it has evolved to WIEN2k [2,4]. During that evolutionary period, methodological developments led to revival of the APW scheme via the APW plus local orbitals (APW+*lo*) combination (summarized below). That has been instantiated in several other codes as well as WIEN2k, e.g., ELK [5], FIEUR [6], *exciting* [7], and Exciting-plus [8]. Here, we are pleased to contribute to the further advance of this important methodology and particularly pleased to be able to do so in honor of Heinz' birthday.



Citation: Zhang, L.; Kozhevnikov, A.; Schulthess, T.; Cheng, H.-P.; Trickey, S.B. Performance Enhancement of APW+*lo* Calculations by Simplest Separation of Concerns. *Computation* **2022**, *10*, 43. https://doi.org/ 10.3390/computation10030043

Academic Editor: Henry Chermette

Received: 7 February 2022 Accepted: 4 March 2022 Published: 16 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

2. Motivating Physical Systems

The materials physics problem class driving our effort is molecular magnetism, in particular, the contriving of condensed aggregates of molecular magnets into materials of relevance to quantum information systems, notably, quantum computing [9,10]. The computational challenge is to predict promising molecular magnets [11,12], and promising aggregates of them as well as parametrizing spin Hamiltonians and aiding interpretation of experimental data. The molecules themselves are large and complicated.

A pertinent example is the molecular magnet $NiCl_2-[SC(NH_2)_2]_4$, dichloro-tetrakisthourea-nickel, commonly called "DTN" and its Co analogue, DTC [13]. DTN is important in this context because of its multi-ferroicity, the coexistence of ferromagnetism and ferroelectricity [14], in contrast with the absence of multi-ferroicity in DTC. The DTN cubic molecular crystal structure has two Ni atoms as magnetic centers in the unit cell. Each Ni has four S atoms and two Cl atoms as nearest neighbors, forming an octahedral structure (like the BO₆ octahedra in ABO₃ perovskites). The unit cell has 70 atoms and 444 electrons,

Spin-crossover systems are a closely related, highly relevant class, as they are candidate linkers for quantum information systems [15–17]. The electronic structure challenge is to calculate the low-spin, high-spin energy difference and provide the potential surface to calculate the vibrational entropic contributions. A particular significant example is the so-called [Mn(taa)] molecule ([$Mn^{3+}(pyrol)_3(tren)$]), a meridional pseudo-octahedral chelate complex of a single Mn as the magnetic center and the hexadentate tris[(E)-1-(2azolyl)-2-azabut-1-en-4-yl]amine ligand. It has 53 atoms and 224 electrons. Calculating its spin-crossover energy with low-computational-cost, commonly used density functional methodology without user intervention and tuning has proven to be a formidable task [18].

While these two examples are convenient for the demonstration of capacity focus of this paper, they actually are a bit on the small side for the investigation of molecular magnetic materials in general. An illustration of that challenge is a molecule of current interest, the $[Mn_{12}O_{12}(O_2CPh)_{16}(H_2O)_4]$ complex [19,20]. It has spin S = 10 from 176 atoms and 1210 electrons.

Essential computational issues are made evident by these examples. The individual molecules are structurally and electronically intricate. Typically they have complicated spin manifolds that are strongly structurally dependent. Their condensed aggregates are correspondingly complicated and demanding. Moreover, the presence of heavy nuclei and the importance of anisotropy both implicate the significance of relativistic effects, including spin-orbit coupling. In sum, predictive, materials-specific simulations of condensed magnetic molecule systems and spin-crossover systems are extremely challenging tasks.

In the remainder of this paper, we describe the context and need for all-electron computational methods with emphasis on LAPW and APW+*lo* methodology, then discuss impediments to use of existing codes on the physical systems of interest, introduce separation of concerns as identifying and off-loading algorithmic elements common to any LAPW/APW+*lo* code, and the SIRIUS package used as a library for that off-loading, show how interfacing between the Exciting-Plus code and SIRIUS can be achieved, and give numerical examples and timings for the combination.

3. Predictive Computational Approaches

Balance of computational cost-effectiveness and accuracy in treatment of electronic structure of challengingly complicated systems is the pragmatic reason for prevalent use of density functional theory (DFT) [21] in its KS variational form [21–24]. In the context motivating this work, accuracy is crucial for predicting both structural properties and characterizing spin manifolds. The primary choices regarding accuracy are the selection of an exchange-correlation (E_{xc}) approximation and selection of a method for solving the resulting KS equations. We address the second. The first is an arena of intense effort that has provided many options.

Most "electronic structure methods" come down to the choice of a basis set (and its truncation) by which to reduce the KS equation to a linear algebra problem. The obvious,

naïve basis for periodic systems is plane waves. It provides systematic enrichment and is unbiased with respect to ionic charge. The long-known limitation is that the basis becomes unmanageably large if the oscillations of near-nucleus orbitals caused by the bare Coulomb potential are included [25]. This burden is removed by use of a pseudo-potential instead [26] or, more recently, use of projector augmented waves (PAWs) [27]. Widely used examples of such "plane-wave pseudo-potential-PAW" (PW-PP-PAW) codes include VASP [28], QuantumEspresso [29], and ABINIT [30–32].

Such calculations intrinsically are not truly all-electron. Pseudo-potentials eliminate core states, while PAWs reconstruct them. There is a need therefore to test and cross-check plane-wave pseudo-potential calculations against truly all-electron calculations. For only three examples of many, see Refs. [33–35]. Another example is a comparatively early use of all-electron calculations for materials-by-design [36]. That was a study of Li-ion battery formulations with the WIEN2k code. Though nontrivial (especially at the time), at 14 atoms per cell with 170–178 electrons, those systems were smaller than the motivating examples discussed above. Cross-validation is particularly important in the context of molecular magnetic quantum materials because of core contributions to spin manifolds and spin-orbit interaction effects.

The all-electron methodology of choice is the LAPW method or its close kin, APW+*lo* [37]. Basis set construction is by use of the "muffin-tin" potential, the spherical average of the KS potential in non-overlapping, nuclear-centered spheres and a constant average elsewhere (the "interstitial" region). "Full-potential" denotes use of the whole KS crystalline potential, not just the so-called muffin-tin (MT) part. Historically that was an important distinction but today the MT potential is used only for basis set construction. Both LAPW and APW+*lo* are rooted in Slater's original APW scheme [38–43]. Within the MT spheres, all three sets have basis functions that are atomic-like solutions of spherical potentials. Those are matched with plane waves in the interstitial region.

Original LAPW literature is extensive, see Refs. [44–58]. Subsequently there was a particular kind of local orbital ("LO", not "*lo*") added [59], and then the closely related APW+*lo* scheme [60]. These are covered in at least two other books [37,61] as well as various review chapters (e.g., Refs. [2,62]). Therefore here we display only those equations directly relevant to our discussion of codes and algorithmic libraries.

The original APW basis function for Bloch wave-vector *k* and plane-wave vector *G* is

$$\varphi_{k}^{G}(\mathbf{r}) := \sum_{\ell,m} \sum_{\nu} A_{\ell m \nu}^{\alpha, \mathbf{k}}(G) u_{\ell \nu}^{\alpha}(\mathbf{r}) Y_{\ell m}(\hat{\mathbf{r}}) \quad \mathbf{r} \in \alpha \\
= (1/\sqrt{\Omega}) e^{i(G+k) \cdot \mathbf{r}} \quad \mathbf{r} \notin \alpha$$
(1)

Here $u_{\ell\nu}^{\alpha}(r)$ is the solution of the (energy-dependent, ϵ) radial Schrodinger equation in the MT sphere labeled α , that is regular at the origin with principle quantum number ν , $Y_{\ell m}(\hat{r})$ are spherical harmonics, $A_{\ell m\nu}^{\alpha,\mathbf{k}}(\mathbf{G})$ are the coefficients for matching with the interstitial plane wave, ℓ and m are the azimuthal and magnetic quantum numbers in a particular sphere. (The APW basis does not have continuous radial first derivatives at the sphere surfaces). Since the radial functions are ϵ -dependent, continuity at the sphere surface requires that energy to correspond to a KS eigenvalue. The APW secular equation thus is highly non-linear in the one-electron energies. That non-linearity induces both an important computational computational cost and difficult-to-manage singularities whenever a radial basis function has a node on a sphere surface.

The LAPW basis addresses those difficulties by using both the radial functions $u(r; \epsilon_{\ell})$ at reference energies ϵ_{ℓ} and their energy derivatives

$$\dot{u}_{\ell\nu} := \frac{\partial u_{\ell\nu}(r;\epsilon)}{\partial \epsilon}\Big|_{\epsilon_{\ell}} \,. \tag{2}$$

(The "dotted" notation for the energy derivative is conventional in LAPW literature). Thus the basis functions become

$$\varphi_{k}^{G}(r) := \sum_{\ell,m} \sum_{\nu} [A_{\ell m \nu}^{\alpha,k}(G) u_{\ell \nu}^{\alpha}(r; \epsilon_{\ell}) + B_{\ell m \nu}^{\alpha,k}(G) \dot{u}_{\ell \nu}^{\alpha}(r, \epsilon_{\ell})] Y_{\ell m}(\hat{r}) \quad r \in \alpha$$

$$:= \frac{1}{\sqrt{\Omega}} \exp^{i(G+k) \cdot r} \quad r \notin \alpha .$$
(3)

The coefficients follow from making each basis function continuous with continuous first radial derivative at each sphere boundary. Unlike the APW case, the KS secular equation in the LAPW basis is of the ordinary linear variational form. The only user-dependent choices are the reference energies ϵ_{ℓ} and muffin-tin radii.

As the LAPW linearization is not unique, exploration of options eventually led [60] to the recognition that a more efficient linearization combines the original APW basis functions inside the sphere *at fixed reference energies* with a set of linearized (in energy) radial functions inside the sphere, each of which vanishes at the sphere surface. This "APW+*lo*" basis consists of Equation (1) enhanced with different local orbitals ("*lo*"),

$$\varphi_{lo}(\mathbf{r}) := \sum_{\ell,m} \sum_{\nu} [A^{\alpha}_{\ell m \nu} u^{\alpha}_{\ell \nu}(\mathbf{r}; \boldsymbol{\epsilon}_{\ell}) + B^{\alpha}_{\ell m \nu} \dot{u}^{\alpha}_{\ell \nu}(\mathbf{r}, \boldsymbol{\epsilon}_{\ell})] Y_{\ell m}(\hat{\mathbf{r}}) , \quad \mathbf{r} \in \alpha .$$
(4)

These localized basis functions do not have continuous derivatives at the sphere boundary, so surface terms arise in the kinetic energy and in any gradient-dependent exchange correlation functional.

The LAPW and APW+*lo* basis sets can be used together with suitable reference energy choices and consideration of the atomic structure differences among spheres. Observe that both basis sets are adaptive in that the radial functions evolve as the KS potential evolves from SCF iteration to iteration.

The forms of the electron number density $n(\mathbf{r})$ and the KS effective potential $v_{KS}(\mathbf{r})$ matrix elements in these basis sets are given in Appendix A for completeness.

4. Codes and Libraries

4.1. Base Code

The present work focuses on the *Exciting-Plus* code, hereafter "EP" for brevity [8]. EP was developed from an early version of the ELK/*exciting* code, that was branched at the time when independent evolution of *exciting* and ELK had just begun. EP was developed with emphasis on post-ground-state calculations such as for the density response function [63] and RPA [8] and GW [64] calculations. Ground state KS calculations are done in EP with *k*-point task distribution and LAPACK [65] diagonalization support. EP also implemented a convenient *mpi-grid* task parallelization in several independent variable dimensions, e.g., *k*-points, *i-j* index pairs of KS states, and *q* points in the calculation of the KS density response function.

EP was constructed conscientiously in terms of coding practices. However, its design did not focus on high performance for multi-atom unit cells. Our context makes that important. Our goal is to retain the features and capabilities of EP while making it fast enough for routine all-electron DFT calculations to be feasible for large, complicated systems such as the magnetic molecules, spin-crossover molecules, and aggregates discussed at the outset.

4.2. Separation of Concerns and the SIRIUS Package

LAPW/APW+*lo* codes evidently share their central formalism. Because their basis sets start from plane waves, those codes also share significant procedural elements with PW-PP-PAW codes. Shared tasks include unit cell setup, atomic configurations, definition and generation of reciprocal lattice vectors *G*, combinations with Bloch vectors G + k, definition of basis functions on regular grids as Fourier expansion coefficients, construction

of the plane wave contributions to the KS Hamiltonian matrix, generation of the charge density, effective potential, and magnetization on a regular grid, iteration-to-iteration mixing schemes for density and potential, and diagonalization of the secular equation. Compared to PW-PP-PAW codes, LAPW/APW+*lo* codes additionally have spatially decomposed basis sets as outlined above.

These extensive commonalities constitute an opportunity for performance enhancement via separation of concerns. Computer scientists can bring their skills to bear on the shared algorithmic core of LAPW/APW+*lo* methodology while computational materials physicists can focus on implementation of analysis, post-processing, better exchangecorrelation functionals, etc.

With achievement of the benefits of this separation in mind, an optimized package, SIRIUS [66], was created by some of us. It has explicit, focused, highly refined implementation of LAPW/APW+*lo* commonalities (and PW-PP-PAW to the extent of the broader commonality) as the goal. That is, abstracting and encapsulating objects common to LAPW and APW+*lo* as the design objective for SIRIUS. By concept, it had both task parallelization and data parallelization. It has been optimized for multiple MPI levels as well as OpenMP parallelization and for GPU utilization.

SIRIUS can be used two ways, as a library or as a simple LAPW/APW+lo code. Elsewhere, we will report on its use in the latter way [67]. In that case the compromise involved is to accept the functionality limits of SIRIUS in return for being able to handle very large systems by both task and data parallelism. Here we report on exploitation of SIRIUS purely as a DFT library by construction of an EP-SIRIUS interface using the SIRIUS API. The expected gain is speed-up while retaining the familiar user-interface and post-processing functionalities of EP. Figure 1 illustrates the scheme. The intrinsic limitation of separation of concerns is that the resulting package has limitations that, in essence, are the union of the limitations of the host code and of the library. We discuss that briefly at the end.



Figure 1. General scheme for utilization of SIRIUS as a library to enhance performance of a host code.

4.3. SIRIUS Characteristics and Features

SIRIUS is written in C++ in combination with the CUDA [68] back-end to provide (1) low-level support (e.g., pointer arithmetic, type casting) as well as high-level abstractions (e.g., classes and template meta-programming); (2) easy interoperability between C++ and widely used Fortran90; (3) full support from the standard template library (STL) [69]; and (4) easy integration with the CUDA nvcc compiler [70]. The SIRIUS code provides dedicated API functions to interface to *exciting* and to QuantumEspresso [29,71,72].

Virtually all KS electronic structure calculations rely at minimum on two basic functionalities: distributed complex matrix-matrix multiplication (e.g., pzgemm in LAPACK) and a distributed generalized eigenvalue solver (e.g., pzhegvx also in LAPACK). SIRIUS handles these two major tasks with data distribution and multiple task distribution levels.

The eigenvalue solver deserves particular attention. Development of *exciting* led to significant code facilities to scale the calculation to larger numbers of distributed tasks than originally envisioned by making the code switchable from LAPACK to ScaLAPACK. This can be verified by comparing the task distribution and data distribution of the base ground state subroutine in recent versions of *exciting* (version Nitrogen for example) and ELK (version 5.2.14 or earlier for example). ELK development appears to have emphasized physics features and functionalities rather than adding ScaLAPACK support. The EP situation is similar. It has only LAPACK support and does not have data distribution of large arrays. Table 1 summarizes the diagonalization methods available in these codes.

nalization
gorithm
-

Table 1. Eigensolver options.

Eigenvalue solver performance depends strongly upon the algorithm type. Widely used linear algebra libraries (e.g., LAPACK, ScaLAPACK) implement robust full diagonalization. They can handle system size up to about 10⁶. Unfortunately for LAPW/APW+*lo* calculations on systems as large as 100+ atoms, the eigensystem often is several times larger. A Davidson-type iterative diagonalization algorithm is appropriate in that case because it typically suffices to solve for the lowest 10–20 percent of all occupied eigenvalues and associated eigenvectors up through and somewhat above the Fermi energy.

Davidson-type diagonalization algorithms are available in some APW+*lo* and PW-PP-PAW codes, e.g., WIEN2k [73] and PWscf [74] respectively. They are not offered in standard linear algebra libraries however. At least in part that is because such algorithms repeatedly apply the Hamiltonian to a sub-space of the system. Therefore the algorithm depends upon details of the Hamiltonian matrix, hence upon the specific basis-set formalism. By virtue of focus on tasks central to LAPW/APW+*lo* and PW-PP calculations, the SIRIUS package can provide an efficient implementation of Davidson-type diagonalization [75] for LAPW/APW+*lo* and PW-PP-PAW codes.

4.4. Interfacing Exciting-Plus with SIRIUS

Despite its many attractive features, especially for important post-ground state calculations, EP has some significant limitations in regard to ground state calculations on large systems such as magnetic and spin-crossover molecules. Those limits include: (1) provision of only the LAPACK eigensolver; and (2) *k*-point-only MPI parallelization. This second limit renders the code completely serial for single *k*-point calculations, e.g., on an isolated molecule in a big cell.

We frame the task therefore as straight-forward interfacing to SIRIUS as an unaltered library with comparatively minimal modification of EP. This black-box approach is pure separation of concerns, since it is the simplest route an experienced EP user could take to try to gain advantage from SIRIUS without investing effort in learning its inner workings. A benefit is that the user interface to EP+SIRIUS is essentially unaltered EP, yet the combined system provides (a) ScaLAPACK support, (b) Davidson iterative eigensolver, (c) band MPI parallelization for one *k*-point, and (d) thread-level OMP parallelization per *k*-point per band. It also exposes some oddities introduced by the black-box strategy.

Interface implementation benefits from the FORTRAN API functionalities provided by SIRIUS. Listing 1 displays the FORTRAN API function calls for parsing the atomic configuration, the APW and *lo* basis from EP, and passing them to SIRIUS.

Listing 1: Setting up atomic configuration.

```
call sirius_set_atom_type_coniguration(sctx, string(trim(label)),&
    & spn(ist,is), spl(ist,is), spk(ist,is), spocc(ist,is),&
    & logical(spcore(ist,is),kind=c_bool))
enddo
```

The code segment in Listing 1 loops over the number of states of a single atom type atom (spnst: species' number of states). For each state, the API call provides to SIRIUS the quantum numbers n, l and k for each state (spn, spl and spk), the occupation of that state (spocc), and whether that state is treated as a core state (spcore).

The first of the two double loops in the code chunk shown in Listing 2 goes over the APWs of one atom type and the ℓ -channels of each APW. For each ℓ -channel, the API call passes the following information to SIRIUS: principle quantum numbers n (apwpqn), value of ℓ (1), value of the initial linearization energy (apwe0), the order of energy derivative of that APW (apwdm), and whether the linearization energy is allowed to be adjusted automatically (autoenu). The second loop is over the total number of local orbitals (nlorb) of one atom type and the *orders* (lorbord) of each local orbital (i.e., number of u(r) or $\dot{u}(r)$ terms in that local orbital). The API call passes the following information to SIRIUS: quantum numbers n and l (lopqn and lorbl), initial linearization energy is allowed to be adjusted automatically (autoenu).

```
Listing 2: Fortran API for basis description.
```

```
! parsing APW descriptions from host code to SIRIUS
do 1 = 0, lmaxapw
do io = 1, apword(l, is)
autoenu = .false.
if (use_sirius_autoenu.and.apwve(io,1,is)) autoenu = .true.
call sirius_add_atom_type_aw_descriptor(sctx, string(trim(label)),&
&apwpqn(1,is), 1, apwe0(io, 1, is), apwdm(io, 1, is),&
&logical(autoenu,kind=c_bool))
enddo
enddo
! parsing LO/lo description from host code to SIRIUS
do ilo = 1, nlorb(is)
do io = 1, lorbord(ilo, is)
autoenu = .false.
if (use_sirius_autoenu.and.lorbve(io, ilo, is)) autoenu = .true.
call sirius_add_atom_type_lo_descriptor(sctx, string(trim(label)),&
&ilo, lopqn(ilo,is), lorbl(ilo, is),lorbe0(io, ilo, is),&
&lorbdm(io, ilo, is), logical(autoenu,kind=c_bool))
enddo
enddo
```

General input parameters such as the plane-wave cutoff, ℓ cutoff for the APWs and for density and potential expansion, *k*-points, lattice vectors and atom positions, etc., all are set as usual in the EP input file. Then they are passed to SIRIUS via its built-in import and set parameter functionalities. Other important parameters such as the fast Fourier transform grid, radial function grid inside each MT sphere, and number of first variational states [37] often are not set in EP input files but defaulted. For EP+SIRIUS, however, those also must be passed to SIRIUS in the initialization step to ensure that the Hamiltonian matrix and eigenvectors are precisely the same in EP and SIRIUS. Other information such as specification of core states, linearization energy values and MT radii defined in the so-called species files of EP is passed to SIRIUS at the beginning of the calculation to overwrite the corresponding SIRIUS default values. Consider Listing 3 therefore.

Listing 3: Fortran API for setting inputs for SIRIUS.

```
call sirius_set_parameters( sctx,&
&use_symmetry=bool(.true.),&
&valence_rel=string('zora'),&
&core_rel=string('none'),&
&auto_rmt=0,&
&fft_grid_size=ngrid(1),&
&fft_grid_size=ndmag,&
&num_mag_dims=ndmag,&
&num_fv_states=nstfv,&
&pw_cutoff=gmaxvr,&
&gk_cutoff=gkmax,&
&lmax_apw=lmaxapw,&
&lmax_rho=lmaxvr,&
&klmax_pot=lmaxvr )
```

The code chunk shown in Listing 3 is an example of basic inputs that are added to EP in the initialization step, in the piece of code named init0.f90. Most of the meanings are explicit in the name. zora means zero-order relativistic approximation. ngrid is the FFT grid set up in EP and passed to SIRIUS. Plane wave cutoff and |G + k| cutoff values are gmaxvr and gkmax in EP. The lmaxapw and lmaxvr are the angular momentum cutoff for APW and for charge density (and potential) inside the MT.

The inserted code shown in Listing 4 supplies SIRIUS with additional parameters for the Davidson method if it is used. After ensuring that the setup of input quantities is identical between the host code (EP) and SIRIUS, the ground state calculation is done solely by SIRIUS. The results, eigenvalues and eigenvectors, are passed back to EP for further calculation.

Listing 4: Eigen-solver selection and Davidson solver parameter setup.

```
if (sirius_davidson_eigen_solver) then
call sirius_import_parameters(sctx, string('{''iterative_solver'' : {''type'' :
     ''davidson'' }}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     {''energy_tolerance'' : 1e-13}}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     {''residual_tolerance'' : 1e-6}}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     {''num_steps'' : 32}}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     {''subspace_size'' : 8 }}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     {''converge_by_energy'' : 1 }}'))
call sirius_import_parameters(sctx, string('{''iterative_solver'' :
     { '`num_singular'' : 20 }}'))
else
! otherwise use full eigen solver from LAPACK
call sirius_set_parameters(sctx, iter_solver_type=string('exact'))
endif
```

Next we display, in Listing 5, a code segment with the typical API calls from EP to retrieve the resulting eigenvalues and eigenvectors. It is inserted in the ground state subroutine, the piece of code named gndstate.f90.

Listing 5: Fortran API for retrieving eigenvalues and eigenvectors from SIRIUS.

```
! get local fraction of eigen-vectors
do ikloc=1,nkptloc
ik=mpi_grid_map(nkpt,dim_k,loc=ikloc)
call sirius_get_fv_eigen_vectors(ks_handler, ik, evecfvloc(1, 1, 1, ikloc),
     nmatmax. nstfv)
call sirius_get_sv_eigen_vectors(ks_handler, ik, evecsvloc(1, 1, ikloc), nstsv)
enddo !ikloc
! get all eigen-values and band occupancies
do ik = 1, nkpt
if (ndmag.eq.0.or.ndmag.eq.3) then
call sirius_get_band_energies(ks_handler, ik, 0, evalsv(1, ik))
call sirius_get_band_occupancies(ks_handler, ik, 0, occsv(1, ik))
else
call sirius_get_band_energies(ks_handler, ik, 0, evalsv(1, ik))
call sirius_get_band_energies(ks_handler, ik, 1, evalsv(nstfv+1, ik))
call sirius_get_band_occupancies(ks_handler, ik, 0, occsv(1, ik))
call sirius_get_band_occupancies(ks_handler, ik, 1, occsv(nstfv+1, ik))
endif
enddo
```

Care is needed in dealing with the MPI task schedules when interfacing to SIRIUS as a library because typically the host code will have an MPI implementation that differs from that in SIRIUS. For EP as the host, the task is simplified because EP has only *k*-point parallelization in the ground-state calculation. In the initialization step, we set the SIRIUS MPI communicator to be derived from the global MPI communicator (MPI_COMM_WORLD) of the host code so that all MPI ranks will be used by SIRIUS. Then the user needs to specify how SIRIUS will carry out the *k*-point distribution, how to plan further band parallelization within a *k*-point, and thread-level parallelization. The schedules of *k*-point parallelization also has additional inputs which are specified in the run job script.

If band parallelization is used in SIRIUS, the eigenvalues and eigenvectors associated with a single *k*-point are distributed in multiple MPI tasks. It therefore is necessary to combine the band subset results before transmitting the eigenvalues and eigenvectors back to EP. Thus, after SIRIUS finishes the ground state calculation but before calling the API to return the eigenvalues and eigenvectors to EP, SIRIUS will do *mpi_reduce* in the MPI band dimension and prepare full eigenvalues and eigenvectors labeled by *k*-points and by the global band index at each *k*-point.

The last piece of the interface provides the additional inputs for the SIRIUS Davidson diagonalization algorithm. These are adjustable numerical parameters passed directly to SIRIUS by EP.

As anticipated, the MPI parallelization in the band degree of freedom is one major gain from interfacing EP to SIRIUS. We noted above that EP runs entirely in non-parallel mode for a single *k*-point calculation (often a "Gamma-point calculation" or "Balderschi-point calculation"), such as is typical for isolated molecule calculations. Hence the SIRIUS-enhanced-EP has the same scaling as SIRIUS alone in the case of single *k*-point calculations. This is an example of the antithesis of the union of limitations that is inherent in separation of concerns. Here, separation of concerns actually avoids a limitation of the host code.

To illustrate, Figure 2 displays the benchmark of band-parallelization on the DTN molecule (brief details about the molecule are below). It is placed in a $10 \times 10 \times 10$ Å cubic unit cell, with plane-wave cutoff $20 a_0^{-1}$ (inverse Bohr radius) and angular momentum cutoff = 7. All jobs were set to 16 multi-threads in one task in accord with the hardware configuration. The recorded time is for the first 100 SCF iterations using the Davidson diagonalization eigensolver. Note that the figure also shows that employment of EP as a front-end to SIRIUS

does not introduce any significant overhead. The timings for EP+SIRIUS are almost identical to those for SIRIUS alone. Timings compared to PW-PP-PAW codes are in the next section.



Figure 2. Benchmark of band parallelization in single *k*-point jobs. $n \times n$ ranks are used for one *k*-point.

5. EP+SIRIUS: Verification Tests on Small Solid State Systems

The EP+SIRIUS combination was bench-marked first against SIRIUS standalone on ground-state calculations of the total energy (and magnetization for magnetic systems) for the simple bulk materials Al, Ni, Fe, NiO, C, Si, Ge, and GaAs. For each system, identical input parameters were used for the SIRIUS and EP+SIRIUS runs. To be systematic, we adopted the experimental lattice parameters for all systems. The APW+lo and LAPW bases were used. Both local density approximation (LDA) and generalized gradient approximation (GGA; PBE [76]) exchange-correlation functionals were used. In the interstitial potential and charge density expansions, the maximum length of the reciprocal lattice vector |G| used as plane wave cut-off for the APW was set to $12 a_0^{-1}$ for all systems. The angular momentum truncation was taken as $\ell_{max} = 8$ for APW, with the same value used for the charge density, potential, and orbital inside the MT sphere. The linearization energy associated with each APW radial function was chosen at the center of the corresponding band with ℓ -like character for all systems. Sampling of the first Brillouin zone was by a dense 16 16 \times 16 k-mesh for all systems. All parameters were tested carefully to achieve total energy convergence (tolerance = 10^{-6} Hartree). For the EP+SIRIUS calculations, diagonalization always was done with the Davidson iterative eigensolver.

Table 2 summarizes the APW+*lo* basis configuration, settings other than those already stated, and the converged total energy and magnetization of these small systems. Table 3 summarizes the same calculation setup with the LAPW basis. The good agreements in total energy and magnetization in these tests validate the assumption that the identical basis setup was invoked and that the constructed interface linked the SIRIUS calculation properly with the EP host code.

	Al (Non-Mag.)	NiO (Non-Mag.)	Ni (FM)	Fe (FM)
crystal structure	fcc	rock-salt	fcc	bcc
latt. const. (Å)	4.05	4.17	3.52	2.87
$R_{mt}(a_0)$	1.8	1.8, 1.6	2.0	2.0
valence relativity	z.o.r.a.	z.o.r.a.	z.o.r.a.	z.o.r.a.
lo config.	Al: <i>s</i> , <i>p</i>	Ni: s, p, d O: s, p, d	Ni: <i>s</i> , <i>p</i> , <i>d</i>	Fe: <i>s</i> , <i>p</i> , <i>d</i>
LO for semi-core	$\epsilon'_{2p} = -2.55$	$\epsilon'_{Ni,3d} = -0.33$ $\epsilon'_{O,2s} = -0.87$	$\epsilon'_{Ni,3d} = -0.33$ $\epsilon'_{Ni,3p} = -2.59$	$\epsilon'_{Fe,3d} = -0.28$ $\epsilon'_{Fe,3p} = -2.18$ $\epsilon'_{Fe,3s} = -3.43$
treated as core state	1s, 2s	Ni: 1s, 2s, 2p, 3s O: 1s	1s, 2s, 2p, 3s	1s, 2s, 2p
LDA: (unit: Ha, μ_B) E_{tot} , SIRIUS E_{tot} , EP+SIRIUS μ_{tot} , SIRIUS μ_{tot} , EP+SIRIUS	-241.40085447 -241.40085447	-1593.13659104 -1593.13659102	-1518.09194282 -1518.09194282 0.564822 0.564825	-1270.11766996 -1270.11766997 2.308247 2.308245
GGA-PBE: (unit: Ha, μ_B) E_{tot} , SIRIUS E_{tot} , EP+SIRIUS μ_{tot} , SIRIUS μ_{tot} , EP+SIRIUS	-241.54245824 -241.54245824		-1518.15356943 -1518.15356942 0.563466 0.563467	-1270.18442575 -1270.18442575 2.327534 2.327530

Table 3. EP+SIRIUS vs. SIRIUS, using LAPW.

	Al (Non-Mag.)	NiO (Non-Mag.)	Ni (FM)	Fe (FM)	
$R_{mt}(a_0)$	1.8	1.8, 1.6	2.0	2.0	
	Nc	LAPW has same linearization energy as APW. No more <i>lo</i> configurations. <i>LO</i> configuration is same as Table 2. other parameters are also same as in Table 2			
LDA: (unit: Ha, μ_B) E_{tot} , SIRIUS E_{tot} , EP+SIRIUS μ_{tot} , SIRIUS μ_{tot} , EP+SIRIUS	-241.40085321 -241.40085422	-1593.13659761 -1593.13659902	-1518.09194596 -1518.09194752 0.564830 0.564827	-1270.11766882 -1270.11766822 2.308243 2.308240	
GGA-PBE: (unit: Ha, μ_B) E_{tot} , SIRIUS E_{tot} , EP+SIRIUS μ_{tot} , SIRIUS μ_{tot} , EP+SIRIUS	-241.54245124 -241.54245372	-1593.27058546 -1593.27058701	-1518.15356717 -1518.15356932 0.563460 0.563466	-1270.18442211 -1270.18442394 2.327533 2.327528	

6. EP+SIRIUS: Two Molecular Examples

6.1. [Mn(taa)] Molecule

As the first known example of a manganese(III) d^4 spin-crossover system [77], [Mn(taa)] is a system of long-standing interest. Experiment shows that the Mn^{3+} cation goes from a low-spin state (LS) to a high-spin state (HS) at a transition temperature of about 45 K.

The [Mn(taa)] structure (see Figure 3) is sufficiently large that it has non-negligible intramolecular dispersion interactions with significant HS-LS dependence. The HS ground state involves anti-bonding molecular orbital occupation, hence the octahedral HS complex tends to have weaker and therefore longer metal-ligand bonds than in the LS ground state.





Figure 3. [Mn(taa)] molecule.

This combination of spin- and structural dependence makes [Mn(taa)] a significant challenge to the computational determination of the two ground states. The purely molecular (non-thermal) $\Delta E_{HL} := E_{HS} - E_{LS}$ is small compared to the total energies. Estimates are about 50 ± 30 meV but as high as a few hundred meV. Extensive details of studies with various codes are in Ref. [18]. Several factors can affect a DFT calculation of the molecular ΔE_{HL} . For consistency with condensed phase calculations, it is appropriate to study the isolated molecule in a large, periodically bounded box. Appropriate accuracy necessitates a rather large plane wave cutoff, a need that is worsened by the amount of vacuum in the unit cell. (We remark that the self-interaction error of the usual GGA exchange-correlation functions (e.g., PBE) tends to cause the the LS state to be favored, hence cause overestimated ΔE_{HL} values. That is not of concern here since what we are testing is algorithmic efficiency. Similarly we did not use Hubbard *U*).

For the test of EP + SIRIUS, we used the experimentally determined HS and LS [Mn(taa)] structures and did PBE calculations for a single molecule in a $10 \times 10 \times 10$ Å³ box. Comparison data are from VASP calculations on optimized structures, also with PBE and without *U*. Notice, however, that the VASP calculations used a $20 \times 20 \times 20$ Å³ box. Table 4 gives the parameters and results for the LS state. Its total energy is determined to be about 412 meV below that of the HS state. In contrast, the VASP results are 458-497 mev (at the optimized geometry) with the variation arising from whether the Mn pseudo-potential has 7, 13, or 15 Mn valence electrons. This illustrates the kind of assessment that all-electron calculations facilitate. Regarding timing, observe that the EP + SIRIUS timing is for 16 (4 × 4) MPI tasks with 8 cores per task.

Table 5 compares timing for the EP only and EP+SIRIUS all-electron calculations and VASP PW-PP-PAW calculations. Evidently EP-only is not competitive but EP+SIRIUS is, at least on a per iteration basis.

There is a difficulty hidden in these results however. The lesser aspect is that we cannot run EP alone at all in a $20 \times 20 \times 20^3$ box. The appropriate cutoffs for such a large vacuum region cause out-of-memory problems with EP because of the way its arrays are structured. The more severe consequence is that we also cannot do a full EP+SIRIUS run, in the sense of returning solutions from SIRIUS to EP for post-processing, on that size box. In effect, EP+SIRIUS is limited in this situation to being an EP user interface for input and control of SIRIUS. The work goes to SIRIUS from EP but the results cannot be returned to EP. Examination of EP suggests that it would take some significant restructuring to remedy the problem, a task well outside the scope of this work or of the separation of concerns approach.

structure	[Mn(taa)], LS state structure
unit cell	$10 imes10 imes10$ Å 3 box
number of atoms in unit cell	55
$R_{mt}(a_0)$	Mn: 2.0; C/N: 1.2; H: 1.0;
$G_{max} (a_0^{-1})$ for APW	20
l_{max} for APW and ρ , $(l_{max}^{APW} \text{ and } l_{max}^{\rho})$	8
l_{\max} for V_{eff} , (l_{\max}^{pot})	8
k-points grid	$1 \times 1 \times 1$
(L)APW configuration for $l \le l_{\max}^{APW}$	$\epsilon_l = -0.15 \mathrm{eV}; \partial_E = 0;$
lo configuration	Mn: <i>s</i> , <i>p</i> , <i>d</i> ; O/C: <i>s</i> , <i>p</i> ; H: <i>s</i> ;
LO for semi-core	$\epsilon'_{Mn,3d} = -0.32; \epsilon'_{Mn,3p} = -2.45$
treated as core state	Mn: 1s, 2s, 2p, 3s; O/C: 1s
μ_{tot} (μ_B)	total: 2.00 Mn atom: 1.65
Minutes per SCF cycle EP+SIRIUS	4.5
Minutes per SCF cycle VASP	3.03–3.71

Table 4. Parameters and results for the APW+*lo* calculation of the [Mn(taa)] LS state, with comparison with VASP timings.

Table 5. For $|G + k|_{max} \cdot R_{MT} = 4$, the average time (seconds) consumed per SCF iteration of EP-SIRIUS for single [Mn(taa)] in 10 × 10 × 10 Å³ box, single *k*-point calculation over 60 min of iterations. Comparison is to VASP for three different pseudo-potentials (see text) in a 20 × 20 × 20 Å³ box.

	$ G^{\rho,v} _{max} (a_0^{-1}) = 12$	14	16	18	20
EP-only (1 MPI task)	1140 (s)	1180	1227	1275	1323
EP+SIRIUS (1 MPI task)	156	190	178	181	318
EP+SIRIUS (4 MPI task)	45	60	42	46	72
VASP (1 MPI task)	181–223				

6.2. EP+SIRIUS: DTN Molecule

The challenges and opportunities posed by the DTN molecule were summarized in Section 2. In essence one has two transition metals in a complicated structure reminiscent of the perovskites such that the system is both ferromagnetic and ferroelectric. See Figure 4. Recall that the molecule has 70 atoms and 444 electrons.



Figure 4. DTN molecule crystal.

We used EP+SIRIUS to calculate the AFM ground state of DTN. We make no attempt at a thorough study, but simply use DTN to show the speed of an all-electron APW+*lo* calculation done with EP+SIRIUS versus with the conventional implementation in EP. Table 6 shows the parameters used and the basic results.

	DTN
unit cell	$\approx 10 \times 10 \times 10 \text{ Å}^3 \text{ box}$
number of atoms in unit cell	70
$R_{mt}(a_0)$	Ni: 2.0; Cl/S: 1.2; C/N: 1.0; H: 0.75;
$G_{\max}(a_0^{-1})$ for APW	20
l_{max} for APW, (l_{max}^{APW})	8
l_{\max} for ρ , (l_{\max}^{ρ})	8
l_{max} for V_{eff} , (l_{max}^{pot})	8
k-points grid	$2 \times 2 \times 2$
(L)APW configuration for $l \le l_{\max}^{APW}$	$\epsilon_l = -0.15 \text{ eV}; \partial_E = 0;$
lo configuration	Na: <i>s</i> , <i>p</i> , <i>d</i> Cl/S/C/N: <i>s</i> , <i>p</i> ; H: <i>s</i>
LO for semi-core	$arepsilon_{Ni,3d}^{\prime}=-0.28 \ arepsilon_{Ni,3p}^{\prime}=-2.18$
treated as core state	Ni: 1s, 2s, 2p, 3s Cl/S/C/N: 1s
$\mu_{\rm tot} (\mu_B)$	total: 0.0 Ni atom: +/-0.72

Table 6. Input parameters and outputs of DTN.

Table 7 shows the average time per scf iteration as a function of the longest expansion vector $G^{\rho,v}|_{max}$ for the density and potential. Notice that the main gain from EP+SIRIUS over EP alone at the level of one MPI task per k-vector is that the iteration time is almost independent of that vector magnitude. The bigger gain comes from the multiple MPI tasks.

Table 7. For the DTN MOF structure, with $|G + k|_{max} \cdot R_{MT} = 4$, the average time (seconds) consumed per SCF iteration as a function of longest expansion vector for the potential and density. $2 \times 2 \times 2$ k-points, run of 60 min.

DTN	$G_{max}^{ ho,v}(a_0^{-1})=12$	14	16	18	20
EP-only (1 MPI task per k-point)	420 (s)	420	450	515	515
EP+SIRIUS (1 MPI task per k-point)	420	440	430	440	440
EP+SIRIUS (4 MPI task per k-point)	171	184	171	195	180

7. Summary and Conclusions

To summarize, we have implemented a performance enhancement strategy for the Exciting-Plus LAPW/APW+*lo* code by interfacing it with SIRIUS used as a library. We have explored the simplest possible approach to exploiting the separation-of-concerns design philosophy of SIRIUS, namely to interface to it as a black box. The interface outsources the central tasks of the ground-state KS problem from EP to SIRIUS. The objective is to embed a SIRIUS SCF loop inside EP. The implementation effort involved is moderate, benefiting from the similarity of the data structures between EP and the LAPW/LAPW+*lo* elements of SIRIUS.

The EP+SIRIUS combination provides performance gains through diagonalization and parallelization improvements while retaining the user interface and post-processing functionalities of EP. The result is a major advance in capability for treating large, complex molecular aggregates. From the user perspective, only small modifications to the original EP input files are needed. A few lines to select use of SIRIUS and to specify the additional parameters for the Davidson eigensolver are the only changes.

This simplest separation of concerns implementation resolves the eigenvalue solver bottleneck in EP that comes from use of LAPACK full diagonalization. (It cannot handle Hamiltonian matrices larger than $\approx 10^6$). The hand-off to SIRIUS provides the option to use diverse diagonalization algorithms (Davidson, ScaLAPACK, or LAPACK). Use of Davidsontype diagonalization of the Hamiltonian in the self-consistent loop thus benefits from multiple level parallelization within *k*-points and bands. The eigenvalues and eigenvectors resulting from the SIRIUS calculation have the same structure as those of EP. The design intent therefore is to transfer them back to EP. However, the array structure design of EP inhibits this, as we found with [Mn(taa)]. We return to that below.

For testing and validation, we showed results from small bulk systems calculated in both the APW+*lo* basis and the LAPW basis. The resulting total energy and magnetization show no meaningful deviation from the SIRIUS standalone runs. Two very much larger molecular systems were calculated using the APW+*lo* basis using both EP alone and EP+SIRIUS. Good scaling in band parallelization for a single *k*-point is observed, The parallelization of the interfaced code works well on high-performance computers, and the computational time is drastically reduced in comparison with the original EP.

The main advantage of the interfaced code is the ease of its construction and the support from advanced eigensolvers. We expect similar interface construction can be done with the ELK or *exciting* codes without unreasonable effort.

Looking ahead, we have found that the non-distributed large arrays defined in EP have become the new bottleneck. That is especially the case when dealing with molecular systems containing more than ≈ 100 atoms in a large unit cell. The primary cause of the bottleneck is the high plane wave *G* cutoff for large systems, and the fact that some fundamental multi-dimensional arrays are defined with one dimension containing all indices of the *G* vector or G + k vector. Examples of such fundamental quantities include the augmentation wave part $(u \cdot Y_{lm} \text{ part})$ of the APW basis and the so-called structure factor, the form $\exp[i(G + k) \cdot r]$. These basic quantities are used in many places in the host code. It is not an easy job to change them to be distributed data in all occurrences. Although that system-size limitation remains, calculations based on the current EP+SIRIUS can handle larger systems than the original Exciting-Plus and offer a significantly improved foundation for examining the validity of the results from calculations based on various pseudo-potentials.

Author Contributions: Conceptualization, T.S. and H.-P.C.; Formal analysis, A.K. and S.B.T.; Methodology, L.Z., A.K., T.S., H.-P.C. and S.B.T.; Resources, H.-P.C.; Software, L.Z., A.K. and T.S.; Supervision, H.-P.C. and S.B.T.; Validation, L.Z.; Writing—original draft, L.Z.; Writing—review & editing, S.B.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The work of L.Z., H.-P.C., and S.B.T. was supported as part of the Center for Molecular Magnetic Quantum Materials, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences under Award No. DE-SC0019330. Computations were performed at the U.S. National Energy Research Supercomputer Center and the University of Florida Research Computer Center. SBT thanks Angel M. Albavera Mata for providing some [Mn(taa)] calculation timing data from VASP and a helpful remark.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Expressions for Number Density and KS Potential

In the LAPW and APW+lo basis sets, the number density and KS potential obviously are adapted, through their matrix elements, to the MT subdivision of the unit cell. In the interstitial region they are expanded in plane waves and inside MT spheres in real spherical harmonics $R_{\ell m}(\mathbf{r})$:

$$n(\mathbf{r}) = \begin{cases} \sum_{\ell m} n^{\alpha}_{\ell m}(r) R_{\ell m}(\hat{\mathbf{r}}), & \mathbf{r} \in \alpha \\ \sum_{\mathbf{G}} \tilde{n}(\mathbf{G}) e^{i\mathbf{G} \cdot \mathbf{r}}, & \mathbf{r} \notin \alpha \end{cases}$$
(A1)

and

$$v_{KS}(\mathbf{r}) = \begin{cases} \sum_{\ell m} v_{\ell m}^{\alpha}(\mathbf{r}) R_{\ell m}(\hat{\mathbf{r}}), & \mathbf{r} \in \alpha \\ \sum_{\mathbf{G}} \tilde{v}(\mathbf{G}) e^{i\mathbf{G}\cdot\mathbf{r}}, & \mathbf{r} \notin \alpha \end{cases}$$
(A2)

Here $n_{\ell m}^{\alpha}(r)$, $\tilde{n}(G)$, $v_{\ell m}^{\alpha}(r)$, and $\tilde{v}(G)$ are expansion coefficients determined through the self-consistent solution of the KS equation.

References

- Kohn, W.; Sham, L.J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* 1965, 140, A1133–A1138. https://doi.org/10.1103/PhysRev.140.A1133.
- 2. Blaha, P.; Schwarz, K.; Trickey, S. Electronic structure of solids with WIEN2k. Mol. Phys. 2010, 108, 3147–3166.
- Blaha, P.; Schwarz, K.; Sorantin, P.; Trickey, S. Full-potential, Linearized Augmented Plane Wave Programs for Crystalline Systems. Comput. Phys. Commun. 1990, 59, 399–415.
- Blaha, P.; Schwarz, K.; Tran, F.; Laskowski, R.; Madsen, G.; Marks, L. WIEN2k: An APW+lo program for calculating the properties of solids. J. Chem. Phys. 2020, 152, 074101. https://doi.org/10.1063/1.5143061.
- 5. The Elk Code. Available online: http://elk.sourceforge.net (accessed on 16 October 2020).
- 6. The Fleur Code. Available online: http://www.flapw.de (accessed on 11 December 2020).
- Gulans, A.; Kontur, S.; Meisenbichler, C.; Nabok, D.; Pavone, P.; Rigamonti, S.; Sagmeister, S.; Werner, U.; Draxl, C. Exciting—A full-potential all-electron package implementing density-functional theory and many-body perturbation theory. *J. Phys. Condens. Matter* 2014, 26, 363202.
- Kozhevnikov, A.; Eguiluz, A.G.; Schulthess, T.C. Toward First Principles Electronic Structure Simulations of Excited States and Strong Correlations in Nano- and Materials Science. In Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, New Orleans, LA, USA, 13–19 November 2010u; pp. 1–10. https://doi.org/10.1109/SC.2010.55.
- Wasielewski, M.R.; Forbes, M.D.E.; Frank, N.L.; Kowalski, K.; Scholes, G.D.; Yuen-Zhou, J.; Baldo, M.A.; Freedman, D.E.; Goldsmith, R.H.; Goodson, T., III; et al. Exploiting chemistry and molecular systems for quantum information science. *Nat. Rev. Chem.* 2020, 4, 490–504. https://doi.org/10.1038/s41570-020-0200-5.
- 10. Gaita-Ari no, A.; Luis, F.; Hill, S.; Coronado, E. Molecular spins for quantum computation. *Nat. Chem.* **2019**, *11*, 301–309. https://doi.org/10.1038/s41557-019-0232-y.
- Castro, S.L.; Sun, Z.; Grant, C.M.; Bollinger, J.C.; Hendrickson, D.N.; Christou, G. Single-Molecule Magnets: Tetranuclear Vanadium(III) Complexes with a Butterfly Structure and an S = 3 Ground State. *J. Am. Chem. Soc.* 1998, 120, 2365–2375, https://doi.org/10.1021/ja9732439.
- 12. Bagai, R.; Christou, G. The Drosophila of single-molecule magnetism: [Mn₁₂O₁₂(O₂CR)₁₆(H₂O)₄]. *Chem. Soc. Rev.* 2009, *38*, 1011–1026. https://doi.org/10.1039/B811963E.
- 13. Mun, E.; Wilcox, J.; Manson, J.L.; Scott, B.; Tobash, P.; Zapf, V.S. The Origin and Coupling Mechanism of the Magnetoelectric Effect in TMCl₂-4SC(NH₂)₂ (TM = Ni and Co). *Adv. Condens. Matter Phys.* **2014**, 2014, 512621.
- 14. Jain, P.; Stroppa, A.; Nabok, D.; Marino, A.; Rubano, A.; Paparo, D.; Matsubara, M.; Nakotte, H.; Fiebig, M.; Picozzi, S.; et al. Switchable electric polarization and ferroelectric domains in a metal-organic-framework. *NPJ Quantum Mater.* **2016**, *1*, 16012.
- 15. Kepp, K.P. Consistent descriptions of metal–ligand bonds and spin-crossover in inorganic chemistry. *Coord. Chem. Rev.* 2013, 257, 196–209.
- Shatruk, M.; Phan, H.; Chrisostomo, B.A.; Suleimenova, A. Symmetry-breaking structural phase transitions in spin crossover complexes. *Coord. Chem. Rev.* 2015, 289–290, 62. https://doi.org/10.1016/j.ccr.2014.09.018.
- 17. Collet, E.; Guionneau, P. Structural analysis of spin-crossover materials: From molecules to materials-Etudes structurales des materiaux a conversion de spin : De la molecule aux materiaux. *Comptes Rendu Chim.* **2018**, *21*, 1133–1151. https://doi.org/10.1016/j.crci.2018.02.003.
- Mejía-Rodríguez, D.; Albavera-Mata, A.; Fonseca, E.; Chen, D.T.; Chen, H.P.; Hennig, R.G.; Trickey, S. Barriers to Predictive High-throughput Screening for Spin-crossover. *Comput. Mater. Phys.* 2022, 206, 111161.

- Caneschi, A.; Gatteschi, D.; Sessoli, R.; Barra, A.L.; Brunel, L.C.; Guillot, M. Alternating current susceptibility, high field magnetization, and millimeter band EPR evidence for a ground S = 10 state in [Mn₁₂O₁₂(Ch₃COO)₁₆(H₂O)₄]·2CH₃COOH·4H₂O. *J. Am. Chem. Soc.* **1991**, *113*, 5873–5874, https://doi.org/10.1021/ja00015a057.
- Sessoli, R.; Tsai, H.L.; Schake, A.R.; Wang, S.; Vincent, J.B.; Folting, K.; Gatteschi, D.; Christou, G.; Hendrickson, D.N. High-spin molecules: [Mn₁₂O₁₂(O₂CR)₁₆(H₂O)₄]. *J. Am. Chem. Soc.* **1993**, *115*, 1804–1816, https://doi.org/10.1021/ja00058a027.
- 21. Hohenberg, P.; Kohn, W. Inhomogeneous electron gas. Phys. Rev. 1964, 136, B864.
- 22. Parr, R.G.; Yang, R.G.P.W. Density-Functional Theory of Atoms and Molecules; Oxford University Press: Oxford UK, 1989.
- Dreizler, R.; Gross, E. Density Functional Theory: An Approach to the Quantum Many-Body Problem; Springer: Berlin/Heidelberg, Germany, 1990.
- 24. Engel, E.; Dreizler, R.M. Density Functional Theory; Springer: Berlin/Heidelberg, Germany, 2013.
- Schlosser, H.; Marcus, P.M. Composite Wave Variational Method for Solution of the Energy-Band Problem in Solids. *Phys. Rev.* 1963, 131, 2529–2546.
- 26. Vanderbilt, D. Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. Phys. Rev. B 1990, 41, 7892.
- Blöchl, P. Projector augmented-wave method. *Phys. Rev. B* 1994, *50*, 17953–17979. https://doi.org/10.1103/PhysRevB.50.17953.
 Kresse, G.; Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* 1996, *54*, 11169–11186. https://doi.org/10.1103/PhysRevB.54.11169.
- Giannozzi, P.; Baroni, S.; Bonini, N.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Chiarotti, G.L.; Cococcioni, M.; Dabo, I.; et al. QUANTUM ESPRESSO: A modular and open-source software project for quantum simulations of materials. *J. Phys. Condens. Matter* 2009, 21, 395502.
- Gonze, X.; Amadon, B.; Anglade, P.M.; Beuken, J.M.; Bottin, F.; Boulanger, P.; Bruneval, F.; Caliste, D.; Caracas, R.; Côté, M.; et al. ABINIT: First-principles approach to material and nanosystem properties. *Comput. Phys. Commun.* 2009, 180, 2582–2615. https://doi.org/10.1016/j.cpc.2009.07.007.
- Gonze, X.; Beuken, J.M.; Caracas, R.; Detraux, F.; Fuchs, M.; Rignanese, G.M.; Sindic, L.; Verstraete, M.; Zerah, G.; Jollet, F.; et al. First-principles computation of material properties: The ABINIT software project. *Comput. Mater. Sci.* 2002, 25, 478–492. https://doi.org/10.1016/S0927-0256(02)00325-7.
- Torrent, M.; Jollet, F.; Bottin, F.; Zèrah, G.; Gonze, X. Implementation of the projector augmented-wave method in the ABINIT code: Application to the study of iron under pressure. *Comput. Mater. Sci.* 2008, 42, 337–351. https://doi.org/10.1016/j.commatsci.2007.07.020.
- 33. Paier, J.; Marsman, M.; Hummer, K.; Kresse, G.; Gerber, I.C.; G.Ángyán, J. Screened hybrid density functionals applied to solids. J. Chem. Phys. **2006**, 124, 154709.
- 34. Haas, P.; Tran, F.; Blaha, P. Calculation of the lattice constant of solids with semilocal functionals. Phys. Rev. B 2009, 79, 085104.
- 35. Fu, Y.; Singh, D. Applicability of the Strongly Constrained and Appropriately Normed Density Functional to Transition-Metal Magnetism. *Phys. Rev. Lett.* **2018**, *121*, 207201.
- 36. Eglitis, R.I.; Borstel, G. Towards a practical rechargeable 5 V Li ion battery. Phys. Status Solidi A 2005, 202, R13–R15.
- 37. Singh, D.J.; Nordstrom, L. Plane Waves, Pseudopotentials, and the LAPW Method, 2nd ed.; Springer: New York, NY, USA, 2006.
- 38. Slater, J. Wave Functions in a Periodic Potential. Phys. Rev. 1937, 51, 846–851.
- 39. Slater, J. An Augmented Plane Wave Method for the Periodic Potential Problem. Phys. Rev. 1953, 92, 603–608.
- 40. Leigh, R.S. The Augmented Plane Wave and Related Methods for Crystal Eigenvalue Problems. *Proc. Phys. Soc. A* **1956**, *69*, 388–400.
- 41. Loucks, T. The Augmented Plane Wave Method; Benjamin: New York, NY, USA, 1967.
- 42. Slater, J. Energy Bands and the Theory of Solids. In *Methods of Computational Physics*; Alder, B., Fernbach, S., Rotenberg, M., Eds.; Academic: New York, NY, USA, 1968; Volume 8, pp. 1–20.
- Mattheiss, L.; Wood, J.; Switendick, A. A Procedure for Calculating Electronic Energy Bands Using Symmetrized Augmented Plane Waves. In *Methods of Computational Physics*; Alder, B., Fernbach, S., Rotenberg, M., Eds.; Academic: New York, NY, USA, 1968; Volume 8, pp. 63–147.
- 44. Marcus, P. Variational methods in the computation of energy bands. Int. J. Quantum Chem. 1967, 1, 567–588.
- 45. Koelling, D. Linearized form of the APW method. J. Phys. Chem. Solids 1972, 33, 1335–1338.
- 46. Andersen, O. Simple Approach to the Band-Structure Problem. Solid State Commun. 1973, 13, 133–136.
- 47. Andersen, O. Linear methods in band theory. Phys. Rev. B 1975, 12, 3060–3083.
- Koelling, D.; Arbman, G. Use of energy derivative of the radial solution in an augmented plane wave method: Application to copper. J. Phys. F 1975, 5, 2041–2054.
- 49. Koelling, D.; Harmon, B. A technique for relativistic spin-polarised calculations. J. Phys. C 1977, 10, 3107–3114.
- Wimmer, E.; Krakauer, H.; Weinert, M.; Freeman, A. Full-potential self-consistent linearized-augmented-plane-wave method for calculating the electronic structure of molecules and surfaces: O₂ molecule. *Phys. Rev. B* 1981, 24, 864–875.
- 51. Weinert, M. Solution of Poisson's equation: Beyond Ewald-type methods. J. Math. Phys. 1981, 22, 2433–2439.
- 52. Weinert, M.; Wimmer, E.; Freeman, A. Total-energy all-electron density functional method for bulk solids and surfaces. *Phys. Rev.* B **1982**, *26*, 4571–4578.
- 53. Blaha, P.; Schwarz, K. Electron densities and chemical bonding in TiC, TiN, and TiO derived from energy band calculations. *Int. J. Quantum Chem.* **1983**, *XXIII*, 1535–1552.

- 54. Jansen, H.; Freeman, A. Total-energy full-potential linearized augmented-plane-wave method for bulk solids: Electronic and structural properties of tungsten. *Phys. Rev. B* **1984**, *30*, 561–569.
- 55. Blaha, P.; Schwarz, K.; Herzig, P. First-Principles Calculation of the Electric Field Gradient of Li₃N. *Phys. Rev. Lett.* **1985**, *54*, 1192–1195.
- Wei, S.; Krakauer, H.; Weinert, M. Linearized augmented-plane-wave calculation of the electronic structure and total energy of tungsten. *Phys. Rev. B* 1985, 32, 7792–7797.
- 57. Mattheiss, L.; Hamann, D. Linear augmented-plane-wave calculation of the structural properties of bulk Cr, Mo, and W. *Phys. Rev. B* **1986**, *33*, 823–840.
- 58. Goedecker, S.; Maschke, K. Operator approach in the linearized augmented-plane-wave method: Efficient electronic-structure calculations including forces. *Phys. Rev. B* **1992**, *45*, 1597–1604.
- 59. Singh, D.J. Ground-state properties of lanthanum: Treatment of extended-core states. Phys. Rev. B 1991, 43, 6388–6392.
- 60. Sjöstedt, E.; Nordstrom, L.; Singh, D. An alternative way of linearizing the augmented plane-wave method. *Solid State Commun.* **2000**, *114*, 15–20.
- Cottenier, S. Density Functional Theory and the Family of (L)APW-Methods: A Step-by-Step Introduction; Self-Published: Zwijnaarde, Belgium, 2013; ISBN 978-90-807215-1-7.
- 62. Schwarz, K. Computation of Materials Properties at Atomic Scale. In *Selected Topics in Applications of Quantum Mechanics;* Pahlavani, M.R., Ed.; IntechOpen: London, UK, 2015; pp. 275–310. https://doi.org/10.5772/59108.
- 63. Petersilka, M.; Gossmann, U.J.; Gross, E.K.U. Excitation Energies from Time-Dependent Density-Functional Theory. *Phys. Rev. Lett.* **1996**, *76*, 1212–1215. https://doi.org/10.1103/PhysRevLett.76.1212.
- Chu, I.H.; Trinastic, J.P.; Wang, Y.P.; Eguiluz, A.G.; Kozhevnikov, A.; Schulthess, T.C.; Cheng, H.P. All-electron self-consistent *GW* in the Matsubara-time domain: Implementation and benchmarks of semiconductors and insulators. *Phys. Rev. B* 2016, *93*, 125210. https://doi.org/10.1103/PhysRevB.93.125210.
- Anderson, E.; Bai, Z.; Bischof, C.; Blackford, S.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum, A.; Hammarling, S.; McKenney, A.; et al. LAPACK Users' Guide, 2rd ed.; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1999.
- Solcá, R.; Kozhevnikov, A.; Haidar, A.; Tomov, S.; Dongarra, J.; Schulthess, T.C. Efficient Implementation of Quantum Materials Simulations on Distributed CPU-GPU Systems. In *Proceedings of the International Conference for High Performance Computing*, *Networking*, Storage and Analysis, ser. SC'15; ACM: New York, NY, USA, 2015; pp. 1–12. https://doi.org/10.1145/2807591.2807654.
- 67. Zhang, L.; Kozhevnikov, A.; Schulthess, T.C.; Trickey, S.B.; Cheng, H.P. Large-scale All Electron APW+*lo* calculations using SIRIUS. 2022, *unpublished*.
- CUDA ToolKit Documentation. Version: v11.2.1. Available online: https://docs.nvidia.com/cuda/index.html (accessed on 16 February 2021).
- 69. Josuttis, N.M. The C++ Standard Library: A Tutorial and Reference, 3rd ed.; Addison-Wesley: Philadelphia, PA, USA, 2000.
- 70. NVCC Compiler. Available online: https://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/ (accessed on 16 February 2021).
- Giannozzi, P.; Andreussi, O.; Brumme, T.; Bunau, O.; Nardelli, M.B.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Cococcioni, M.; et al. Advanced capabilities for materials modelling with QUANTUM ESPRESSO. J. Phys. Condens. Matter 2017, 29, 465901.
- 72. Giannozzi, P.; Baseggio, O.; Bonfà, P.; Brunato, D.; Car, R.; Carnimeo, I.; Cavazzoni, C.; de Gironcoli, S.; Delugas, P.; Ferrari Ruffino, F.; et al. Quantum ESPRESSO toward the exascale. *J. Chem. Phys.* **2020**, *152*, 154105. https://doi.org/10.1063/5.0005082.
- Blaha, P.; Hofstätter, H.; Koch, O.; Laskowski, R.; Schwarz, K. Iterative diagonalization in augmented plane wave based methods in electronic structure calculations. J. Comput. Phys. 2010, 229, 453–460. https://doi.org/10.1016/j.jcp.2009.09.036.
- Chandran, A.K. A Performance Study of Quantum ESPRESSO's Diagonalization Methods on Cutting Edge Computer Technology for High-Performance Computing. Master's Thesis, Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy, 2017.
 Gulans, A. Implementation of Davidson Iterative Eigen-solver for LAPW. *to be published*.
- 76. Perdew, J.P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. Phys. Rev. Lett. 1996, 77, 3865–3868.
- Sim, P.G.; Sinn, E. First manganese(III) spin crossover, first d4 crossover. Comment on cytochrome oxidase. J. Am. Chem. Soc. 1981, 103, 241–243, https://doi.org/10.1021/ja00391a067.