D3.1

Report on identified algorithmic advances, and their software development plan

Luigi Genovese and Stefano Baroni, Stefano de Gironcoli, Pietro Delugas, Andrea Ferretti, Alberto Garcia, Paolo Giannozzi, Anton Kozhevnikov, Andrea Marini, Ivan Marri, Pablo Ordejon, Davide Sangalli, Daniel Wortmann.

Due date of deliverable 31/05/2019 (month 6)
Actual submission date 31/05/2019
Lead beneficiary CEA (participant number 5)
Dissemination level PU - Public
### Document information

<table>
<thead>
<tr>
<th>Project acronym</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project full title</td>
<td>Materials Design at the Exascale</td>
</tr>
<tr>
<td>Research Action Project type</td>
<td>European Centre of Excellence in materials modelling, simulations and design</td>
</tr>
<tr>
<td>EC Grant agreement no.</td>
<td>824143</td>
</tr>
<tr>
<td>Project starting/end date</td>
<td>01/12/2018 (month 1) / 30/11/2021 (month 36)</td>
</tr>
<tr>
<td>Website</td>
<td><a href="http://www.max-centre.eu">http://www.max-centre.eu</a></td>
</tr>
<tr>
<td>Deliverable no.</td>
<td>D3.1</td>
</tr>
</tbody>
</table>

### Authors

Luigi Genovese and Stefano Baroni, Stefano de Gironcoli, Pietro Delugas, Andrea Ferretti, Alberto Garcia, Paolo Giannozzi, Anton Kozhevnikov, Andrea Marini, Ivan Marri, Pablo Ordejon, Davide Sangalli, Daniel Wortmann.

### To be cited as


### Disclaimer

This document’s contents are not intended to replace consultation of any applicable legal sources or the necessary advice of a legal expert, where appropriate. All information in this document is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user, therefore, uses the information at its sole risk and liability. For the avoidance of all doubts, the European Commission has no liability in respect of this document, which is merely representing the authors’ view.
Contents

1 Executive Summary 4

2 Introduction 4

3 Organization of the workpackage 4
   3.1 Restructuring Stages 5
   3.2 Tasks of the workpackage 6

4 Report on identified algorithmic advances, and their software development plan 6
   4.1 BigDFT code 7
   4.2 SIESTA code 8
   4.3 QUANTUM ESPRESSO code 9
   4.4 YAMBO code 10
   4.5 FLEUR code 13
   4.6 CP2K code 14
   4.7 SIRIUS library 15

Acronyms 15

References 16
1 Executive Summary

Work package 3 is entitled “Code Evolutions, exploiting algorithmic advantages enabled by the exascale transition”. The idea here is to identify which advantages the exascale and pre-exascale machines might enable for the electronic structure calculation community. This work package is a sort of “hub” for other code-oriented activities, as there are several outcomes that might be put in relation with other WPs (in particular WP1, WP4, and WP6).

In this document, we will present the identified “algorithmic advances” (also called “actions” in the rest of the document) which are planned on the codes of MAX consortium. The implementation details of such actions have been discussed and planned, during the first six months of the MAX project, internally in each of the code owner’s groups. We here present the planned initiatives in the context of the WP, by classifying them in terms of their complexity and by proposing a tentative timeline for each of them.

2 Introduction

The increasing of the computational power of modern architectures represents at the same time an opportunity and a challenge, for code users and developers: on the one hand, the accessible CPU-hours are continuously increasing and researchers make routinely usage of computational research project demanding several millions of CPU hours. This enables new investigation directions with present-day research paradigms. On the other hand, the code and algorithmic designs that are used within such projects have been, for the great majority of the cases, conceived in an era – remarkably not long ago – when routine production calculations were only measured in thousands or even hundreds of CPU-hours. It appears therefore not surprising that the usage of large supercomputing facility is undergoing deep changes.

In this context, for electronic structure calculation codes, the question that arises is “which kind” of usage of the HPC platforms is able to bring new scientific results to the community. We insist here on this point: the focus is on the science that a novel, potentially disruptive set of paradigms might enable to address. But to meet such objective, essential actions about software restructuring and end-user code interfaces need to be identified and implemented in the software developers community: the objective of this workpackage is to find such actions and try to implement them for some use-cases of the codes of the consortium. We deliberately excluded in this context capacity-computing tasks like high-throughput.

To better clarify the role of WP3 in the evolution of the codes towards exascale, the workpackage deals with the concept of “intra-code” modifications, namely the actions for which the scope is within a code of the MAX consortium. Clearly, such actions may result from the integrations of libraries that are provided in other WP, and/or may be associated to new functionalities/algorithms.

3 Organization of the workpackage
3.1 Restructuring Stages

WP3 will therefore address the *progressive adaptation* of either the research community and the MAX software codes to the advent of the exascale era. Such adaptation might be declined in various stages:

**Stage1: Brute Force** exploitation of the computing power. At this stage the *existing* software codes and algorithms should address some of the research directions that might be faced thanks to the increasingly available computing power. Such stage might include developments at short and medium term, that might be realised even without altering the present structure of the related codes. The deliverables on such stage will provide interesting functionalities in the codes whilst at the same time introducing the awareness of the *design limitations* that have to be overcome in the codes to go to Stage 3.

**Stage Objectives**: hitting the “wall” of the limitations of present-day paradigms, showing what can be effectively done with the codes “as they are” today. As a deliverable, we will have a set of new functionalities that are added on top of existing codes, with pre-exascale awareness, yet still without altering the underlying infrastructure of the codes.

**Stage2: Alternative Research Paradigms** based on the *unconventional* usage of the existing codes. In this stage, we will show that *new* research directions are nowadays possible, that exploit HPC facilities and resources thanks to a paradigm shift in the approach: new physics can be accessible thanks to a *novel* approach to the calculations. This will include the set up and the preparation of complex simulation setup, as well as the integration of their results into post-processing and analysis tools. In this stage also a throughout discussion will be performed on which are the good practice to implement the I/O of a code in view of an algorithmic redesign.

**Stage Objectives**: showing that the consortium is able to identify and promote new user experiences, that connect HPC and novel approaches for the end-user. Ideally, we would like to have a set of scientific use-cases of the MAX codes, possibly written as a set of tutorials for the end-user, showing how to extract challenging results and prepare complex input files approaches via a workflow-based HPC approach.

**Stage3: Algorithmic design framework**. In this stage, the outcomes and the bottlenecks identified in the previous two stages will be critically considered in order to address *unbiased* discussions on how a given code should be designed to fully exploit its capabilities in the HPC platforms and at the same time to guarantee the flexibility that is required for the workflow approach described in Stage 2. Here the discussion should be conducted either on the *components* of the codes that have emerged from WP1, but also on which might be the software infrastructure (i.e. the high-level programming paradigm) that can “connect” together such components.

**Stage Objectives**: here the idea is to bring awareness of how to conceive the software codes in order to simplify its usage and its maintenance. The objectives are prototype examples that should suggest how the codes have to deal with the unavoidable process of fine-tuning to make different libraries/modules working together. Such stage of actions is certainly enabled by WP1 outcomes, in the sense
that some actions may explicitly employ the libraries released in WP1. We should not, in this context rewrite the MAX codes from scratch, but all the codes of the consortium should eventually identify how they should have been written to meet the ideal objectives.

3.2 Tasks of the workpackage

Such stages will therefore appear, declined in various flavours, in the identified actions of the MAX codes. Another classification of the workpackage actions will ease their positioning in the pre-exascale scenario. The following tasks have been identified:

T3.1 Software-related fault resilience algorithms and solutions: the items of this task are meant to prevent the users a waste of computing time due to incorrect behavior of the code or incorrect setup of the input parameters. This is not associated to traditional fault resilience intended as a feedback to a hardware problem, but rather to a software fault, in the sense explained above. Such “fault resilience” might be implemented in two ways:

- By creating new approaches for fault tolerance at the level of the algorithm itself, that can detect a runtime misbehaviour and provide a fallback solution.
- By correcting possible weaknesses of the code components in certain regimes; for example, by preventing the code to run in a regime for which a selected functionality has not been conceived.

T3.2 Enabling new code functionalities with an exascale mindset: the goal is to open research avenues that have up to now been considered closed or impractical within the mindset of current HPC tools, but that could be explored thanks to the increasing computing power. These actions will be by no means simple development initiatives aimed at the traditional expansion of code functionalities. The emphasis here is on the unlocking of computations whose time-to-solution would be too long nowadays.

T3.3 Exploitation of new algorithms at the pre-exascale. The code modifications which belong to this task will be associated to algorithmic advances which are enabled by the pre-exascale computing power. New workflows and different I/O will be implemented, in a context with high thread concurrency and increasing computational workload. Actions in this tasks will not only be associated to new functionalities but rather to new computational approaches for exploitation of supercomputers.

4 Report on identified algorithmic advances, and their software development plan

We here present the actions foreseen by each code development group in the context of WP3. To each of the actions we assign a label which will be used to identify its expected timeline in the Gantt chart shown in Fig. 2.
4.1 BigDFT code

B1 Input-file manipulation and handling: wildcard approaches.

The variety of the input parameters that is present in a DFT code is often a source of mistakes by the non-expert users, especially in the context of non-conventional simulation on massively parallel architectures. In this situation it is important for the developers to provide internal cross-checks of the code input parameters. We plan to insert in the input file system of BigDFT a system of automatic cross-check between input variables to avoid inconsistencies in the specification of the input variables. This would limit - if not avoid - the potential wasting of computational resources which would result from a bad specification of input files in a massively parallel calculations. In BigDFT, like in other DFT codes, there are many parameters which can be varied, however it is possible to achieve robust and reliable results using a pre-defined set of values for the majority of these variables. We can exploit this by importing a profile, which already defines a set of common values. All these techniques are going to be achieved with the help of the FUTILE library, delivered in the context of WP1. For this reason, they might be also imported in other DFT codes which use a representation of the input file as key/value pairs.


The BigDFT code has been proved to handle calculation of very large systems - up to many thousands electrons [1] - with Hybrid Functionals thanks to an accurate and unconstrained application of the Fock operator, that may also benefit from acceleration by Graphic Processing Units. We plan to further improve this functionality by introducing mixed precision approach for the self-consistency cycle of hybrid functionals calculations, by calculating with simple precision the first part of the SCF cycle and having double precision Fock operator for highly precise calculations. In addition, the same techniques might be used for the wavelets convolutions - important for the representation of the Kohn-Sham Hamiltonian operator - thanks to the extensibility of the libconv library, which will be shipped with BigDFT suite.

B3 Usage of Pseudo-Fragment approaches for extended systems in the Support Functions formalism.

Thanks to the specific properties of BigDFT code, we plan to present a computational approach which is tailored for reducing the complexity of the description of extended systems in density functional theory. The idea is to define a recipe for generating a set of localized basis functions which are optimized either for the accurate description of pristine, bulk-like Wannier functions, or for the in situ treatment of deformations induced by defective constituents such as boundaries or impurities. With this approach, we would like to identify the regions of an extended system which require dedicated optimization of the Kohn–Sham degrees of freedom, and provides the user with a reliable estimation of the errors – if any – induced by the locality of the approach. Such a method facilitates on the one hand an effective reduction of the computational degrees of freedom needed to simulate systems at the nanoscale, while in turn providing a description that can be
straightforwardly put in relation to effective models, like tight binding Hamiltonians. The wavelet-based method employed in this paper makes possible calculation of systems with different dimensionalities, including slabs and fully periodic systems. Such an approach will be made possible by the usage of PyBigDFT library released in the context of WP1.

**B4 Exact exchange for $O(N)$ implementation.**

We plan to extend the calculation of hybrid functionals to the formalism of the Support Functions which is used in the Linear-Scaling algorithm of the BigDFT code. Such an extension will be done by further specialising the communication method of round-robin scheme that is now done in FUTILE library. Such calculations will benefit from the sparsity of the density kernel matrix that would preserve the $O(N)$ scaling for this very challenging calculations, albeit with a considerably larger prefactor than the present-day calculations with semilocal functionals. The scheduling of the communications which will be needed to do exact exchange operator contracted with a sparse density kernel is of a different kind of what exists nowadays, as it has to consider a different workflow than the round-robin scheme implemented for the cubic-scaling approach presented in [1]. Extensive tests are planned in order to find the crossover points between the implemented approach in the cubic-scaling code and this implementation.

**4.2 SIESTA code**

**S1 Basis-set contraction.**

We plan to generalize the internal interfaces in SIESTA that deal with the information about basis orbitals to provide two levels of operation: one using the “primitive” basis set employed now, and the other using a “contracted” basis set with lower cardinality. The latter will be constructed “on the fly” [see [2] and refs. therein; also similar work in BigDFT] by an extra optimization step. These ‘minimal basis sets’ will have a very large impact on performance in general, and are essential in particular for the efficiency of the CheSS O(N) solver, whose operation count depends on the size of the spectrum of the Hamiltonian.

**S2 Exact-exchange.**

A “classic” algorithm based on intermediate conversion of the PAOs to Gaussians is currently being (re)-implemented and is almost ready. But we have in our plans also the potentially better performing ACE (Adaptively Compressed Exact-Exchange) [3] algorithm.

**S3 Improved algorithms for scf convergence.**

Here we have started by extending the basic toolbox, and adding mechanisms for changes in the mixing recipe as the calculation progresses, under various conditions (see the new blocks description in the manual). However, more work needs to be done to advance the robustness and efficiency, both by refinement of existing techniques and by the implementation of new algorithms. The exploration of the proper heuristics can be made more efficient in terms of programmer’s time by the use of Lua scripts, which can access the data structures of the program and
modify the convergence parameters on-the-fly. New ideas can be tried without re-compilation. This Lua-embedding is a major feature in SIESTA that now sets it apart from other codes, but that is going to be offered as a module in WP1. This action might be considered to span two task realms: that of the fault resilience (non-convergence should be avoided), and that of the implementation of novel algorithms (the scf convergence loop might be substituted by other techniques (e.g., direct minimization)).

S4 Break-even points for various solvers and automatic dispatch based on learned heuristics.

The solvers currently available in SIESTA have different complexity dependencies on the size of the system, and different use-case features: Diagonalisation is a cubic scaling direct method, and it comes in several flavors (ScaLAPACK: standard, MRRR, etc; ELPA as a more efficient re-implementation of the basic algorithms); The Orbital Minimisation Method (OMM) is a cubic scaling solver algorithm that allows reuse of previous information. The Pole Expansion and Selected Inversion (PEXSI) solver affords a multi-level parallelisation with reduced scaling (at most quadratic with system size), and CheSS, the density-matrix purification method within ELSI-NTPoly, and the legacy O(N) method in SIESTA based on OMM-plus-localization, are linear-scaling algorithms.

Each solver has an effective prefactor which depends on the details of the system and the use case (e.g. single-point vs. MD runs). We will investigate the break-even points as a function of system size and characteristics, and collect the information into a form that can be dynamically re-used in actual runs for the most efficient selection of the solver. Steps in this direction are also being taken in the ELSI project. This action can also be seen as dealing with resilience (the use of the wrong solver should be avoided) and with novel algorithmic ideas (including the dynamical combination of algorithms).

S5 Re-design of the legacy $O(N)$ solver.

The original linear-scaling functionality in SIESTA was based on the combination of the OMM method with orbital localization. We plan to re-design it by exploiting the libDBCSR library for sparse-matrix multiplication produced in T.1.1 (WP1). In fact, whole new families of O(N) methods can be implemented on that foundation (connection to the use of the Algorithm Development Platform of T1.2 (WP1)).

4.3 QUANTUM ESPRESSO code

Q1 Improved diagonalization algorithms.

A possible point of failure for density-mixing QUANTUM ESPRESSO pw.x runs is the appearance of non positive-definite overlap matrices during the iterative diagonalisation. For this we will try alternative diagonalisation algorithms with reduced subspace diagonalisation, that are both more robust and more scalable, these new developments will be incrementally added to the KS_library released by WP1.

Q2 Direct energy minimization schemes.
For non periodic systems a global minimization approach will fix both diagonalisation failures as well provide a more robust self consistent method.

**Q3 RPA based advanced exchange and correlation functionals.**

Advanced exchange and correlation functionals based on the Random Phase Approximation (RPA) [4] and its extensions including exchange-kernel corrections (RPAx) [5] have recently emerged as promising, yet extremely costly, accurate functionals naturally incorporating static and dynamical correlation effects. In this action we will modify existing specialised codes in QUANTUM ESPRESSO performing total energy RPA and RPAx calculations to fully exploit high-level parallelisation of the required integration in the imaginary frequency domain, enabling scalability to pre-exascale machines. We will apply this approach to several benchmark systems probing different correlation regimes.

**Q4 Extension of the localised inner-projection method to EXX.**

We will extend the localised inner-projection approach [3, 6] to exact exchange, successfully demonstrated for the ground state to excited-state properties within time-dependent DFT and MBPT, as well as to real-time simulations of electronic excitations. We expect that this action will finally close the gap between accurate but expensive excited-state simulations using plane waves, with less expensive, but less accurate, ones using localised basis sets.

**Q5 Adaptive parallelisation schemes.**

On the hardware architectural side, we plan to implement and adopt advanced adaptive parallelisation schemes. Currently the parallelisation strategy is defined via command-line options at startup. We will implement a dynamical definition of communicators and grids in real and reciprocal space, to be activated at run time, switching between different data distributions according to the task being performed, in order to optimise communication balance. Actions of this side are also planned in WP2. The developments of this side will be eventually added to the UtilXlib library distributed by WP1.

**4.4 YAMBO code**

**Y1 Restart structure, parallel IO and database re-organization: Application to BSE and QP calculations**

With the introduction of parallel I/O based on NetCDF and HDF5, YAMBO databases can now be created independently on the parallel structure used during the simulation. This allows the user to restart an interrupted run with using a different parallel structure, which is useful in general to address software resilience, and particularly important in specific cases, as listed below.

An example is the calculation and solution of the excitonic Hamiltonian. The construction of the excitonic Hamiltonian is very time consuming, and may not be finalised within a single run of the code (consider eg all issues threatening resilience). The restart technique that we plan to code allows for a proper restart which is independent from the parallel structure used and from the way the simulation ended. This will be obtained via the use of character matrix, shadow to the
complex BSE matrix. The extra disk space required is almost negligible. We also plan to apply the same strategy to restart the calculation of quasi-particles (QP). As above, this computational step is typically very time-consuming and it would be definitely important to store the data (QP corrections) already computed for later use or restart. A flexibly and parallel I/O database handling is foreseen to be a very effective solution also for this case.

Y2 Exploitation of mixed precision algorithms.

YAMBO supports both single and double precision (DP) arithmetic. The single precision (SP), in particular, allows the user to reduce the overall computational and storage requirements and often the time-to-solution. In some cases, however, the use of SP floating point during the whole computational workload could produce a lack of precision in the final results. For this reason, some specific parts of YAMBO (e.g., extensive accumulations as in the calculation of the GW self-energy) are executed in DP. We plan to extend the implementation of the mixed precision paradigm to other parts of YAMBO (these parts will be identified by performing specific simulations in both SP and DP) in order to significantly improve memory, disk and bandwidth usage without affecting the results precision. Moreover, we aim at improving numerical accuracy and numerical robustness in highly parallelised applications, especially when the Open-MP paradigm is extensively exploited. Finally, we will also consider and evaluate the use of the mixed-precision FFT library to be developed within the QUANTUM ESPRESSO consortium.

Y3 Real-time propagation with atomic motion and interface with QE.

The real-time module of the YAMBO code allows the user to describe electronic propagation using non-local self-energies in an efficient way. This is achieved by projecting the equation of motion (EOM) in the basis-set of the Kohn–Sham (KS) wave–functions. The propagation of the EOM has, then, the same cost with local and with non-local potentials. Moreover, the choice of the number of states in the KS basis-set naturally imposes a cutoff on the maximum frequency oscillation permitted. This makes possible the use of larger time-steps compared to plane-wave or real-space implementations. The drawback is that the dynamics of electrons away from the equilibrium position of the nuclei (where the KS wave–functions are centered) can be hardly described. This is a limitation if atomic motion, for example within Ehrenfest dynamics, is considered. Indeed the atomic displacement carries the electronic density away from the initial position. The solution is to update the basis-set during the dynamics. We will implement a scheme which has been already used for a pure TD-DFT code [7]. This will require an on the fly interface between YAMBO and the pw.x code from QUANTUM ESPRESSO, where YAMBO will feed pw.x with the atomic positions (and eventually the electronic density) and receive the KS wave–functions.

Y4 Advanced approaches for full-frequency GW.

Simple approximations (such as the plasmon-pole models) to deal with dynamical nature of the screened Coulomb interaction $W$, and to ease the frequency convolution needed to compute the GW self-energies are still very common in today’s GW calculations, but may break down and severely limit the overall accuracy of the...
implementation. In the exascale perspective, removing this limitation not only becomes feasible for a large class of systems, but is also an interesting way to exploit the available computing power. A number of methods have already been proposed in the literature, such as real axis integration [8], analytic continuation and Padé approximants [9, 10, 11], contour deformation [12] to name a few. Here we plan to improve the full frequency implementation of YAMBO (real axis method) by implementing more advanced techniques (either among the existing ones or newly developed) to address this point.

Y5 Advanced self-energies from MBPT. With the boost in computational capabilities expected with pre-exascale machines, the calculation of advanced self-energies in many-body perturbation theory methods may become possible and usable for realistic systems. In the MAX work-plan we consider two types of such developments:

- GW self-energies plus environment: with the aim of describing the quasiparticle levels of systems (molecules, nanostructures, layers) in the proximity of a dielectric/metallic environment (such as a surface or a solvent), we plan to extend the calculation of the GW self-energy by including environment effects (e.g. following the polarisable continuum model, PCM, approach) [13, 14]. We plan to interface YAMBO with the ENVIRON module, already available also from QUANTUM ESPRESSO.
- SOSEX self-energies: beyond GW methods are emerging as higher accuracy MBPT methods for excited state properties. Upon a critical evaluation of the accuracy of this class of methods (ongoing work by some of the YAMBO developers on model systems), we consider to implement the SOSEX approximation [15, 16] also for 3D realistic system in YAMBO. This approach is numerically very challenging and would fit naturally the capabilities of pre- and exascale machines.

Y6 YAMBO without empty states.

Excited state calculations are severely limited by the large number of (empty) states involved in sum-over-states operations used to converge the physical quantities (e.g., the self–energy or the screened Coulomb interaction). Lanczos-based algorithms have been applied to GW, BSE, and TD-DFT methods [17, 18, 19, 20, 21, 22, 23] and demonstrated as a feasible option to address the above problems. In fact, when applied to solvers for the Dyson equation, for the electronic Green’s function and for the polarisation function, these algorithms do not require the use of virtual orbitals nor the explicit inversion of dielectric matrices. In this context we plan to interface YAMBO with the linear response module of QUANTUM ESPRESSO to take advantage of Lanczos-based solvers in the calculation of the screened Coulomb interaction W and possibly of the GW self-energy.

Y7 Real time parallelisation.

The time propagation of the equation of motion for a time dependent function is intrinsically serial. This is a serious drawback for simulations using a large number of CPUs, since frequent all-to-all communications are required. However YAMBO

\[ \text{http://www.quantum-environment.org/} \]

\[ \text{http://www.max-centre.eu} \]
does not propagate a simple function but a time dependent matrix. This offers the opportunity to distribute the matrix over different groups of CPUs, thus parallelising the propagation and reducing the communication process inside the CPU subgroups. In this action such approach will be developed to perform NEQ–MBPT simulations on (pre-)exascale machines.

4.5 FLEUR code

Most development effort on the FLEUR code will focus on the implementation of ideas that enable more efficient utilization of current and future exascale technology. In particular, we will spend some efforts in the redesign of existing functionality in order to achieve performance, scalability and robustness by modifications and improvements of the fundamental algorithms used.

F1 Finishing the restructuring of the Hamiltonian setup into high-level operations

In a standard DFT self-consistency cycle the bulk of the computational effort goes into the matrix-diagonalization and the Hamiltonian-setup. While this Hamiltonian setup can be written as straight-forward matrix operations, such an implementation does not take advantage of the specific structure of the problem and hence leads to significant inefficiencies. The full-exploitation of the symmetries and the specific structure on the other hand does not allow the use of optimized libraries available for the standard matrix operations and thus also does not provide satisfactory performance. Hence these two approaches have to be combined to keep the possibility to harvest the performance and performance-portability proved by the use of standard operations and libraries with the exploitation of a certain set of symmetries present in the problem. We already made significant advances in this process and will finish the implementation within this project. Besides providing a general performance boost, our effort will also lead to a better utilization of computing resources in different parallelization strategies and hence lower the burden on the user to identify an optimal parallel setup. The utilization of standard libraries also enable us to push the code to the limits of machine performance. The functionality implemented here will constitute a key component of the LAPWlib developed in WP1.

F2 Evaluation of the Coulomb kernel in LAPW

The evaluation of the coulomb kernel in LAPW is a key requirement in the efficient implementation of hybrid functionals. This operation is significantly more computational challenging than the tasks of a standard DFT calculation and hence must be optimized carefully. We plan to utilize our knowledge from the construction of the Hamiltonian to adopt a similar strategy for this code and thus proceed by analysing the existing implementation, identify the possibilities to exploit standard HPC libraries and to keep optimizations exploiting the specific structure of the problem. Similar to the previous action item the work done here will be strongly linked to the library on LAPW operations (LAPWlib) as developed in WP1.

F3 Improved charge-density mixing schemes

A key task in any DFT calculation is to achieve a self-consistent solution of the electronic density. This is usually done by relatively simple iteration schemes in which convergence is not guaranteed.
Convergence of this iterative approach usually requires the use of quasi-Newton methods for mixing the density. The use of preconditioners can also increase the convergence rate and the stability of the procedure. Within the linearized augmented plane-wave approach implemented in FLEUR many of these established preconditioners are difficult to implement due to the complexity of the representation of the charge. Hence, we plan to explore some ideas that are more adopted to our basis set. These developments are crucial for HTC and needed to enable the efficient use of the code on (pre-)exascale machines.

4.6 CP2K code

Implementation of forces and stress tensor for RPA and double-hybrid functionals in CP2K requires a distributed multiplication of tall and skinny matrices (matrices with very different dimension sizes). At the moment this functionality is provided by the ScaLAPACK library which is CPU-only and which has a poor performance for this kind of matrices. The work is in progress to create a communication-optimal library COSMA for the efficient multiplication of matrices of arbitrary shapes. The technical report on the COSMA algorithm is available here: http://spcl.inf.ethz.ch/Publications/.pdf/mmm-tr.pdf.

The following developments have been completed:

• a single-node tiled multiplication of large matrices which outperforms the cublasXt; source code is available here: https://github.com/kabicm/Tiled-MM

• a grid-to-grid converter to change matrix distribution from one layout to another, for example, from COSMA to ScaLAPACK or vice versa; source code is available here: https://github.com/kabicm/grid2grid

The following actions are planned next:

C1 Integrate COSMA library into CP2K using an intermediate ScaLAPACK matrix layout.

In the current implementation of CP2K ScaLAPACK is used to perform distributed matrix-matrix multiplication. As a first step of incorporating COSMA library into CP2K the ScaLAPACK data layout will be preserved and the following chain of operations will be executed: transformation of wave-functions from internal CP2K format to ScaLAPACK format → transformation from ScaLAPACK to COSMA layout using grid2grid library → matrix multiplication with COSMA → transformation of the resulting matrix from COSMA layout back to ScaLAPACK layout using grid2grid library.

C2 Transform CP2K matrices directly to COSMA layout without a need of ScaLAPACK.

In this stage the further optimization will be implemented and the wave-functions stored in the internal CP2K format will be converted to COSMA layout directly without an intermediate ScaLAPACK matrix representation.
C3 Switch to COSMA in RPA calculations.

In the final step the COSMA library will be fully integrated into CP2K which will accelerate the RPA and related calculations (such as forces and stress tensor) where multiplication of tall and skinny matrices is a bottleneck.

4.7 SIRIUS library

The domain-specific SIRIUS software development platform aims at providing efficient algorithms for the DFT total energy minimization. The work has been started to implement the direct solvers for the wave-function optimization. The following optimizers for the plane-wave pseudopotential method have been implemented:

- orbital transformation method for insulators
- direct minimization for ensemble density-functional theory

The optimizers have been successfully tested on few structures where the classical density mixing scheme doesn’t converge. We plan the following next steps:

U1 Prototype and implement conjugate gradient method.

This is an implementation of direct minimization for ensemble electronic structure calculations by Baarman et al. [24] in which the update operator for the electronic orbitals takes the structure of the Stiefel manifold into account. In this method the optimization scheme for the occupation numbers ensures that the constraints remain satisfied.

U2 Prototype and implement proximal gradient method.

This is an implementation of proximal gradient method for ensemble density functional theory by Ulbrich et al. [25] which is suitable for metallic systems.

U3 Interface advanced density optimizers with QUANTUM ESPRESSO and CP2K.

Once the wave-function optimizers are prototyped and proved to be working they will be implemented in a highly-efficient way directly in the SIRIUS library. This will allow QUANTUM ESPRESSO and CP2K codes to call the optimizers from SIRIUS (we consider direct or reverse communication implementations) and find the ground states of systems which can’t be converged using standard mixing techniques.

Acronyms

DFT Density Functional Theory. 10

MBPT Many Body Perturbation Theory. 10

RPA Random Phase Approximation. 10
Figure 1: Table of the proposed WP3 actions classified according to the Stage and Tasks. Actions of increasing difficulty are associated to higher values of Stages, whereas increasing complexity correspond to tasks of higher number.

References


Figure 2: (cont.) Gantt diagram.


