Extended software development workshop: electronic structure library coding - solvers

Location: CECAM-ES
Webpage: https://www.cecam.org/workshop-0-1274.html
Dates: June 6, 2016 to June 17, 2016

Organizers

Artacho, Emilio
CIC NanoGUNE Consolider, Donostia-San Sebastián, Spain, Spain
Blum, Volker
Duke University, Durham, NC, USA, USA
Corsetti, Fabiano
Imperial College London, United Kingdom
Pouillon, Yann
University of Cantabria, Spain

1 State of the art

Electronic structure software and methods development still work predominantly within the historical paradigm of separate, complete and self-contained packages, typically depending only on a few compilers and basic libraries. This approach has undoubtedly been successful in producing the rich variety of electronic structure codes available today. However, it has led to a lot of replicated development, and makes it hard to introduce common data standards. It has also made it increasingly difficult for researchers to contribute new ideas without becoming deeply involved with the development of one of the pre-existing packages.

At the moment, it is widely recognized in the community that these are important, yet unsolved, problems. There are however a number of interesting new European initiatives aiming to help communication between codes, in conjunction with big data research (e.g., NOMAD, AiiDA, MARVEL) and close integration with HPC (e.g., E-CAM, MaX). It is also important to note that there are already a few examples of communal software libraries specific to electronic structure which have found some success in bridging between different codes, e.g., Wannier90 for maximally-localized Wannier functions, and Libxc for exchange and correlation functionals.

The aim of the Electronic Structure Library (ESL, esl.cecam.org) initiative is to develop the research infrastructure underpinning a huge amount of scientific research, spanning materials physics, physical chemistry and biology, nanotechnology and nanomedicine, Earth science, and more. Indeed, the importance of atomistic modelling from first principles electronic structure is widely recognized, and its applications outside of academia are many.

In particular, the opportunity to collaborate with researchers in industry involved in such modelling projects would allow us to develop software that could drive innovation and growth.

Our goal is to create a common repository of high-quality software and data standards in the field of electronic structure simulation, which will facilitate reuse of code, interoperability between different code bases, and development of new methodologies. This ESL initiative was born two years ago from a CECAM extended software development workshop; the current workshop is the third annual one, plus a January 2016 workshop organized in collaboration with the NOMAD project.

2 Training provided

The theme of this year's workshop was "solvers", as our objective was to develop three libraries focusing on eigensolvers, Poisson solvers, and atomic solvers.

The first day of the workshop was dedicated to talks and discussion. These revolved around three topics: the integration of ESL within the larger E-CAM project, technical issues relating to the quality and usability of software libraries produced with the ESL, and the details of the specific libraries which we planned to work on during the workshop. We had representatives present both from E-CAM and commercial atomistic simulation ventures (QuantumWise, SIMUNE).

The rest of the workshop was dedicated to coding and related work (software documentation, ESL wiki maintenance, code repositories) in small teams of 2-4 people.

3 Software development projects

The major outcomes are summarized below:

Integration of ESL and E-CAM: creation of the E-CAM GitLab code repository; migration of ESL projects to GitLab; creation of online collaboration tools for ESL projects on the E-CAM server; initial work on adopting the EasyBuild installation framework for ESL modules.

Eigensolver library: development work on the ELSI project (Electronic Structure Infrastructure), including the ELPA, libOMM and PEXSI eigensolvers; initial work on including also the CheSS solver from BigDFT; complete restructuring of the ELSI interface to increase flexibility and portability by making use of the MatrixSwitch and FUTILE libraries; creation of a new subproject for automatic generation of realistic test Hamiltonians.

Poisson library: creation of the PoKE project (Poisson Kernel for Electrons), based partly on a previous effort to create a Poisson solver library within BigDFT; discussions on the interface; implementation of periodic and free boundary conditions; initial testing.

Atomic library: creation of the SQARE project (Solvers for Quantum Atomic Radial Equations); setup of coding framework; discussions on the interface and code layout; implementation of ODE solvers.

ESL wiki: restructuring and clean up, addition of data standard documentation (UPF).

There has already been some noticeable improvement in the computational infrastructure for the ESL project, partly as a result of our activities during this workshop and our ongoing collaboration with E-CAM. Currently, CECAM is hosting the ESL wiki on one of its servers, while the software has been partly migrated to a new GitLab repository hosted by E-CAM. Some of thesesoftware modules are part of D2.1 of WP2.

Planned future developments, which we hope to be able to implement in coming years, include the regulated mirroring of externally-developed software (with the permission and collaboration of the original authors) and integration of the ESL libraries into package management systems for both personal workstations and HPC centres.

It would be important for the future strategy of the ESL to provide dedicated machines not just for storing code but also performing nightly builds and automatic testing. This would help the ESL maintainers to ensure consistent high quality amongst all contributions. Similarly, quick access to HPC facilities would allow for in-house development and testing of massively parallel code.

Finally, providing training in collaboration with the ESL could be useful for attracting new researchers and giving them the tools to contribute to the projects being developed. Training areas of interest would be software engineering, modern coding practices, data analysis tools, and development for HPC. On top of this, ESL maintainers could benefit from meeting HPC companies and learning about developments in hardware.