

# eCSE018 Technical report

## Tuning FHI-Aims for complex simulations on CRAY HPC platforms

M.R. Farrow<sup>a</sup>, A.J. Logsdail<sup>a</sup>, A.A. Sokol<sup>a</sup>, V. Blum<sup>†</sup>, C.R.A. Catlow<sup>a</sup>, and S.M. Woodley<sup>a</sup>

<sup>a</sup>*Department of Chemistry, Kathleen Lonsdale Materials Chemistry, University College London, 20 Gordon Street, London, WC1H 0AJ, United Kingdom*

<sup>†</sup>*Duke University, MEMS Department, 144 Hudson Hall, Box 90300, Durham, NC 27708, USA*

---

### Abstract

The quantum mechanical software code FHI-aims has been ported to the ARCHER platform, with recommended compiler settings with the intel compiler using the -O2 compiler option, and with the MKL maths libraries. Profiling using CRAY’s multi-processor performance tools have been used to target areas for code development. The FHI-aims software is available as a module on ARCHER, saving researcher’s time in compiling and also to allow for the latest version (071914.7) of the code to be available to all users.

---

### Introduction

The quantum mechanical (QM), all-electron electronic structure code FHI-aims [1] has been developed specifically for highly accurate atomistic modelling of a many different system types, including, periodic and clusters at the level of Density Functional Theory (DFT) and higher. The methodology adopted in the FHI-aims code uses numerical atomic orbitals, that are, in theory, extremely efficient for parallelisation and allows the study of systems of different dimensionality, including aperiodic boundary conditions, at a similar computational cost. This is a major advantage that the FHI-aims code has over other QM codes, in particular, those that use plane-wave basis sets. The development of the FHI-aims code has been specifically focussed towards High-Performance Computing (HPC) platforms and has been shown to achieve scalability to over 64,000 processors on a Blue-Gene/P architecture while maintaining chemical accuracy [2]. We report the work that has been performed in optimising the FHI-aims software for the ARCHER hardware and the features and enhancements that have been developed for the code. We have chosen four “exemplar” systems to study, and these are 1. A zinc metal periodic system consisting of two atoms in the unit cell, 2. Al<sub>2</sub>O<sub>3</sub> (corundum) periodic system with thirty atoms in its unit cell, 3. A spinel cubic system ZnAl<sub>2</sub>O<sub>4</sub> periodic system with fourteen atoms in its unit cell, and finally, 4. Nanocluster systems that are known as “double bubble” (AB)<sub>12</sub>@(CD)<sub>48</sub> that consists of one hundred and twenty atoms, where AB = ZnO or GaN and CD = GaN or ZnO. For each of these systems we investigate a GGA functional, PBEsol [3, 4], and a hybrid functional PBEsol0 which includes 25% Hartree-Fock-like electron exchange.

### Compiling, scaling and profiling

The FHI-aims code was extracted from the repository and compiled with the intel ifort compiler (version 14.0.1) and used both the intel MKL maths libraries with scalapack, and CRAY’s LibSci maths libraries. The MPICH2 environment was used to create an executable capable of running over multiple processors. The level of compiler-driven optimisation was investigated, and the results are summarised in table 1, where the average (over two runs) of a singlepoint energy calculation for each of the test systems using the PBEsol functional, performed on ARCHER, is reported. It is clear that the compiler can make significant performance improvements over using no optimisation (compiler option

-O0). However, the results do appear to be showing discrepancies in expected behaviour.  $\text{ZnAl}_2\text{O}_4$  is the only system that behaves as expected with approximately a 1.8x increase in computational time as the optimisation level is increased. The  $\text{Al}_2\text{O}_3$  system seems to prefer -O2 over the more optimisation intensive -O3 option, whereas the Zn metal system has the best performance for -O1. However, for Zn metal using -O3, one of the runs took ten self-consistency cycles whereas the other took twelve, indicating some numerical instabilities are occurring during the calculation. Overall, the MKL maths libraries provided by the intel compiler seemed to give the faster compute time. Moreover, it was noticed that using option -O2 with the MKL libraries resulted in more consistency of the timings over the runs performed as each run was within a few percent of each other. Therefore, it was recommended that the FHI-aims software be compiled with -O2 compiler option and the MKL maths libraries for a consistent executable. The release version ((071914.7) of FHI-aims was then compiled using the above options and added to the ARCHER module database. This is a key benefit for current and future users as they will all be able to work from the same base code and not need to compile the code for themselves.

Table 1: Computation time for a singlepoint energy calculation using the PBEsol functional.

System	cores	wall clock (cpu1) at optimisation level (seconds)						
		-O0	-O1	-O1 (MKL)	-O2	-O2 (MKL)	-O3	-O3 (MKL)
Zn Metal	24	106.70	85.186	69.89	71.38	71.12	69.84	73.12
Zn Metal	12	322.71	142.46	136.90	145.75	128.82	135.10	135.08
Zn metal	2	1496.94	553.64	547.55	580.16	587.40	509.50	586.96
$\text{Al}_2\text{O}_3$	24	4736.82	2951.28	2821.46	2831.71	2760.86	2845.26	2741.42
$\text{Al}_2\text{O}_3$	12	8677.31	5081.07	4934.25	5026.10	4923.55	5095.75	4891.90
$\text{ZnAl}_2\text{O}_4$	24	1683.10	972.59	994.151	959.51	947.89	957.31	963.31
$\text{ZnAl}_2\text{O}_4$	12	3452.06	1947.38	1946.59	1946.32	1936.15	1951.86	1927.59

Scaling runs were also performed for both PBEsol and PBEsol0 shown graphically in figures 1 and 2, respectively. The PBEsol scaling shows excellent scaling up to 72 processors for all the systems. It is also worth noting that all the systems exhibit the same trend, indicating that the GGA functional routines are well optimised. When comparing these results to those obtained from PBEsol0, we can immediately see that the hybrid functional routines are much less optimal.

We also explored how increasing the number of basis sets from the minimum up to a complete “standard” basis set (known as Tier 1) changed the calculation time. Figure 3 shows the results for the zinc test system and we can see that significant time saving can be achieved by reducing the basis sets by one or two. It is unadvisable to go beyond this however, as accuracy will be affected. Whether the system under consideration would maintain accuracy after the basis set reduction is very much system dependant and it is recommended that users perform some test calculations for their systems first before removing basis sets.

Figure 4 shows a snapshot from the Apprentice2 visualisation package for the Cray profiling tool, perftools, for the  $\text{ZnAl}_2\text{O}_4$  test system. It was noticed that the function `prune_density_matrix_sparse` was taking a lot longer than expected and so this was targeted for optimisation.

### Optimisation of compute time

At the start of a DFT calculation, the wavefunction has to be converged and this is done through a Self-Consistent Field (SCF) process. During this process, the charge density (`rho`) and the derivative of the charge density (`rho_gradient`) have to be computed. The latter is computationally intensive and so there is would be a large performance improvement in optimising this process. In order to speed up the computation, the charge density (`rho`) is splined to create a multipole expansion (`rho_multipole`), which in turn, is used to calculate the gradient (`rho_multipole_grad`) much more efficiently as the

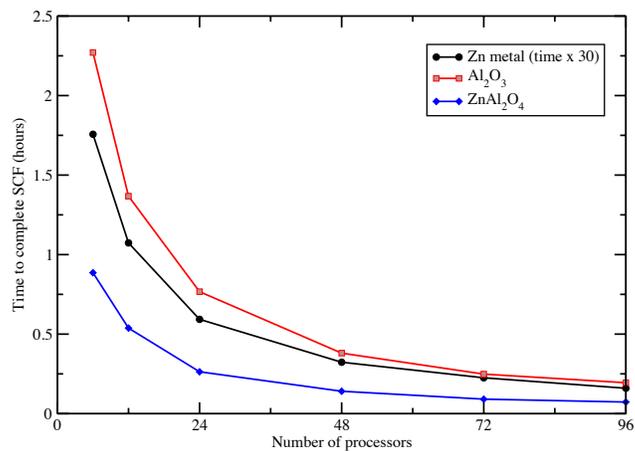


Figure 1: PBEsol singlepoint energy calculations scaling for three of the test systems.

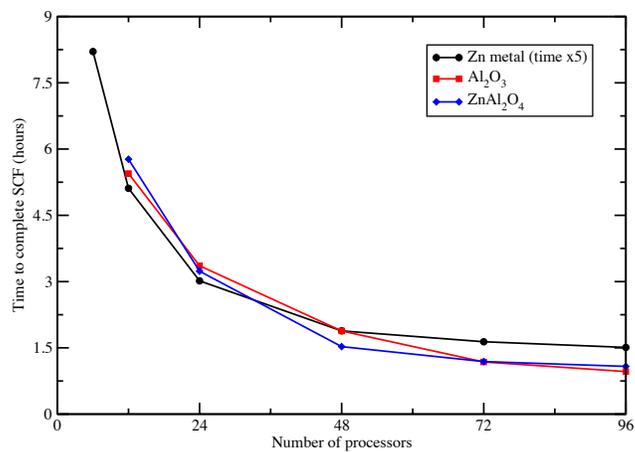


Figure 2: PBEsol0 singlepoint energy calculations scaling for three of the test systems.

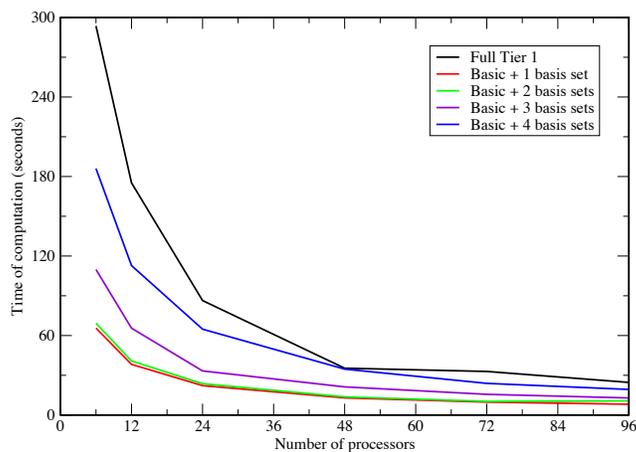


Figure 3: Example plot of how the computational expense of a calculation increases with basis sets, shown for zinc test system.

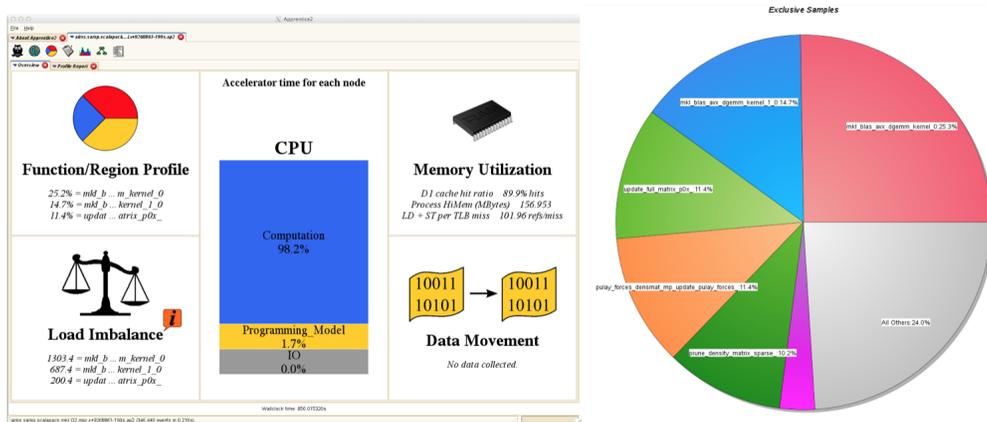


Figure 4: Visualisation (via Apprentice2 software package) of singlepoint energy calculation on  $\text{ZnAl}_2\text{O}_4$  system.

computationally expensive `rho_gradient` does not have to be computed at all. A new subroutine was created to perform this function although it is still under development.

## Implementation of new functionality

State-of-the-art, meta-GGA density functional methods offer improved chemical accuracy when compared to generalised gradient and local density approximations (GGA and LDA, respectively), as they depend not only on the density and the gradient of the density, but also on the kinetic energy density. The implementation of meta-GGAs within FHI-aims was non-self-consistent, *i.e.* a single point energy calculation was performed that depended on a pre-converged solution to the Kohn-Sham equations. In this work, we implemented meta-GGA density functionals that are self-consistent. In order to achieve self-consistency, it was necessary to establish several steps in our work. The first step was a re-structuring of the software to allow the calculation of the kinetic energy density during the self-consistent cycle, the next step interfaced the pre-existing meta-GGA density functionals into the FHI-aims so that the energy and potential can be calculated, as required for the self-consistent cycle, without dependence on re-implementing the calculations. Finally, we derived and implemented the calculation of the derivative of the kinetic energy density with respect to density, as needed for the calculation of the potential during the self-consistent cycle.

During the first step, appropriate data objects and routine calls were introduced in the software, as well as the propagation of data objects. Namely, the calculation of the kinetic density was introduced inside the routine `scf_solver`, which is a general routine that governs the overall SCF cycle. Due to the parallel nature of FHI-aims it was necessary to ensure synchronisation across the MPI environment, which was done using internal `MPI_sync` routines. When comparing run time for each process, it was clear that the both the calculation and synchronisation of the kinetic density was not a time consuming process. With the calculated kinetic density, it was necessary to propagate the data object for the kinetic density to the appropriate part of the software for calculation the exchange-correlation (xc) energy and potential. This is handled via the subroutine `evaluate_xc`, and routines therein, and so we updated these to accept the necessary objects, along with all hierarchical calls to this subroutine between `scf_solver` and `evaluate_xc`. Within `evaluate_xc`, we implemented functionality for the local meta-GGAs of Zhao and Truhlar [5] and Peverati and Truhler [6], commonly known as M06-L and M11-L, respectively. Furthermore, pre-existing implementations for the energy calculation using the density functionals TPSS [7], revTPSS[8, 9], and TPSSloc[10] were updated to allow for their use in full self-consistent calculations. It was also necessary to derive the appropriate physical expression and implement a method for calculating the exchange-correlation potential. This term includes an additional derivative when compared to the previous software, namely the derivative of the kinetic energy density with respect to the density, for which we followed a method as previously outlined by Neumann, Nobes and Handy[11]. Once derived, the necessary parts of the calculation were built inside the subroutine `integrate_hamiltonian_solver_p2`, specifically, the left-hand and right-hand sides of a dot product that depends on the gradient of the basis functions, grid point weightings and also the derivative of the exchange-correlation energy with respect to the kinetic energy density. This leads to the calculation of a dot product between two matrices of size  $(3*n\_points\_in\_batch, n\_basis\_functions)$ , resulting in an output square matrix of size  $n\_basis\_functions$ . This output matrix is the contribution to the Hamiltonian from the kinetic-energy density for the self-consistent cycle, and thus allows an iterative process of solving the Kohn-Sham equations to be solved self-consistently for the meta-GGA functionals. Moreover, our work here is the basis for a step-wise development of hybrid meta-GGAs, where a part of the exchange energy from the density functional is replaced by the non-local exact exchange from Hartree-Fock.

## Conclusions and discussion

The quantum mechanical all-electron code FHI-aims has been ported and optimised to run on the ARCHER platform. It has been incorporated into the ARCHER module database so that current

and future users can get access without having to compile the code themselves. This has an added benefit of enabling all users to use the latest version of the code which contains many performance enhancements and bug fixes. During the testing and optimisation of the FHI-aims code, we investigated the optimisation and properties of so-called “double bubble” structures [12, 13] and the work was presented at conferences as a showcase of the capabilities of the FHI-aims software. A new feature, self-consistent meta-GGA functionals have been implemented into the FHI-aims software package. These functionals are enable users to perform calculations to a higher accuracy than standard GGA / LDA functionals alone. Finally, work is progressing on an optimisation to the SCF cycle that avoids the calculation of the computationally expensive derivative of the gradient of the charge density. This new subroutine takes a spline fit of the charge density and uses that for the calculation of the derivatives. This has the potential to significantly speed up the time to SCF convergence. The PBEsol0 hybrid functional was found to be sub-optimal and areas of the code were identified for optimisation, that will be done in future work.

## Acknowledgements

This work was funded under the embedded CSE programme of the ARCHER UK National Supercomputing Service (<http://www.archer.ac.uk>)

## References

- [1] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, Ab initio molecular simulations with numeric atom-centered orbitals, *Computer Physics Communications* 180 (11) (2009) 2175–2196. doi:10.1016/j.cpc.2009.06.022.  
URL <http://linkinghub.elsevier.com/retrieve/pii/S0010465509002033>
- [2] V. Blum, The Nuts and Bolts of Electronic Structure Theory Basis sets, Real-Space Grids, Relativity, Scalability, [http://th.fhi-berlin.mpg.de/th/Meetings/DFT-workshop-Berlin2011/presentations/2011-07-13\\_Blum\\_Volker.pdf](http://th.fhi-berlin.mpg.de/th/Meetings/DFT-workshop-Berlin2011/presentations/2011-07-13_Blum_Volker.pdf), [Online; accessed 15 July 2015] (2013).
- [3] J. Perdew, K. Burke, M. Ernzerhof, Generalized Gradient Approximation Made Simple., *Physical Review Letters* 77 (18) (1996) 3865–3868. doi:10.1103/PhysRevLett.77.3865.  
URL <http://www.ncbi.nlm.nih.gov/pubmed/10062328>
- [4] J. Perdew, A. Ruzsinszky, G. Csonka, O. Vydrov, G. Scuseria, L. Constantin, X. Zhou, K. Burke, Restoring the Density-Gradient Expansion for Exchange in Solids and Surfaces, *Physical Review Letters* 100 (13) (2008) 136406. doi:10.1103/PhysRevLett.100.136406.  
URL <http://link.aps.org/doi/10.1103/PhysRevLett.100.136406>
- [5] Y. Zhao, D. G. Truhlar, A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions., *The Journal of chemical physics* 125 (19) (2006) 194101. doi:10.1063/1.2370993.  
URL <http://scitation.aip.org/content/aip/journal/jcp/125/19/10.1063/1.2370993>
- [6] R. Peverati, D. G. Truhlar, M11-L: A Local Density Functional That Provides Improved Accuracy for Electronic Structure Calculations in Chemistry and Physics, *The Journal of Physical Chemistry Letters* 3 (1) (2012) 117–124. doi:10.1021/jz201525m.  
URL <http://dx.doi.org/10.1021/jz201525m>
- [7] J. Tao, J. Perdew, V. Staroverov, G. Scuseria, Climbing the Density Functional Ladder: Nonempirical Meta-Generalized Gradient Approximation Designed for Molecules and Solids, *Physical Review Letters* 91 (14) (2003) 146401. doi:10.1103/PhysRevLett.91.146401.  
URL <http://link.aps.org/doi/10.1103/PhysRevLett.91.146401>

- [8] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, J. Sun, Workhorse Semilocal Density Functional for Condensed Matter Physics and Quantum Chemistry, *Physical Review Letters* 103 (2) (2009) 026403. doi:10.1103/PhysRevLett.103.026403.  
URL <http://link.aps.org/doi/10.1103/PhysRevLett.103.026403>
- [9] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, J. Sun, Erratum: Workhorse Semilocal Density Functional for Condensed Matter Physics and Quantum Chemistry [*Phys. Rev. Lett.* 103, 026403 (2009)], *Physical Review Letters* 106 (17) (2011) 179902. doi:10.1103/PhysRevLett.106.179902.  
URL <http://link.aps.org/doi/10.1103/PhysRevLett.106.179902>
- [10] L. A. Constantin, E. Fabiano, F. D. Sala, Semilocal dynamical correlation with increased localization, *Physical Review B* 86 (3) (2012) 035130. doi:10.1103/PhysRevB.86.035130.  
URL <http://link.aps.org/doi/10.1103/PhysRevB.86.035130>
- [11] R. Neumann, R. H. Nobes, N. C. Handy, Exchange functionals and potentials, *Molecular Physics* 87 (1) (1996) 1–36. doi:10.1080/00268979600100011.  
URL <http://www.tandfonline.com/doi/abs/10.1080/00268979600100011>
- [12] A. A. Sokol, M. R. Farrow, J. Buckeridge, A. J. Logsdail, C. R. A. Catlow, D. O. Scanlon, S. M. Woodley, Double bubbles: a new structural motif for enhanced electron-hole separation in solids., *Physical chemistry chemical physics : PCCP* 16 (39) (2014) 21098–105. doi:10.1039/c4cp01900h.  
URL <http://pubs.rsc.org/en/content/articlehtml/2014/cp/c4cp01900h>
- [13] M. Farrow, J. Buckeridge, C. Catlow, A. Logsdail, D. Scanlon, A. Sokol, S. Woodley, From Stable ZnO and GaN Clusters to Novel Double Bubbles and Frameworks (May 2014). doi:10.3390/inorganics2020248.  
URL <http://www.mdpi.com/2304-6740/2/2/248>