CASTEP solvation forces: eCSE01-9

Ben Durham[1], Dr Jacek Dziedzic[2,3], Dr Phil Hasnip[1], Prof Matt Probert[1]

October 2022

[1] Department of Physics, University of York, Heslington, York YO10 5DD, U.K.
[2] School of Chemistry, University of Southampton, Highfiled, Southampton SO17 1BJ, U.K.
[3] Faculty of Applied Physics and Mathematics, Gdansk University of Technology, Gdansk, 80-233 Poland

Abstract

The aim of this eCSE was to add new functionality to the Density Functional Theory (DFT) program CASTEP to enable it to perform structural relaxation, transition state and/or molecular dynamics simulations in Open Boundary Condtions (OBCs) and in the presence of a solvent. Most CASTEP calculations use periodic boundary conditions and so are best suited to studying extended systems (i.e. solids), but sometimes it is helpful to have open boundary conditions instead, such as when studying a molecular system, or a system in the presence of a solvent. Whilst the solvent could be represented explicitly, this can be very computationally expensive, and so in a previous eCSE project (eCSE07-6), the "minimal-parameter solvent model" was added to CASTEP. In this model the solvent is treated implicitly via solving the generalised Poisson equation, by treating the dielectric permittivity of the system as a continuum and a function of the electronic density.

This original implementation was limited to calculating the free energy of solvation only, and could not relax the atomic positions for system under study in OBCs or in the presence of a solvent. Furthermore, the dielectric permittivity, and therefore the implicit model of the solvent, was only computed once at the beginning of each CASTEP execution, and was held fixed for the duration of the calculation. This meant that the model of the solvent was not updated to respond to changes in the electronic density; thus, even if the force terms had been implemented, once the atoms had moved significantly over the course of a Geometry Optimisation, or Molecular Dynamics simulation, then the implicit solvent model would have become obsolete and the results non-physical. An additional limitation was the most computationally intensive step in this model, the solving of the generalised Poisson equation, was not optimally parallelised. Hence, in this project, we added the calculation of implicit solvation forces, as well as a model for implicit solvent that allows the dielectric permittivity to be updated in a self-consistent manner with the electronic density and improved the parallelisation of these calculations. This significantly increases the usefulness of the OBCs/implicit solvation functionality and enables qualitatively new calculations.

1 Introduction

Density Functional Theory (DFT) is a quantum mechanical simulation method that has become one of the most widely used research tools for studying materials and systems of interest where the electrons dictate the behaviour. Modelling systems of interest exactly by solving the many interacting electron Schrödinger equation directly is not possible for systems of more than a few electrons, as the computer memory required for such problems scales exponentially with the number of electrons. Instead, the fundamental DFT theorems state that all properties of the system (observables) are functionals of the electron density $\rho(r)$, which does not have a problematic memory scaling with the number of electrons. The energy functional is of particular interest, as by knowing the energy and its derivatives one can predict the lowest energy (ground state) structure of a material/molecule. The ground state of a system is the most stable state and therefore the state most likely to be found in nature. Hence, determining the ground state structure must usually be done before attempting to calculate more complicated properties of the material/molecule. The exact energy functional is unknown, but with some relatively simple approximations one can yield highly accurate results, with approximately 30,000 papers using DFT published each year.

The CASTEP DFT code is a UK flagship code, specialised for solid materials, and is heavily used on ARCHER (300-400 active users). CASTEP's support for finding the ground state electronic and atomic configurations is very good for solid materials with ideal, periodic (repeated) crystal structures. However, computing these properties for isolated molecules, or molecules in solvent (for which there is a wide range of chemistry and materials science applications), was poorly supported in the CASTEP code before this eCSE was completed. Many molecules of interest exist dissolved in a solvent, an environment which will likely have a significant impact on the properties of the molecule. Modelling the effects of these environments on the solvated molecule by explicitly including the solvent material in the calculation is expensive using DFT, since the DFT computational cost scales as $O(N_e^3)$ with the methods used by CASTEP. The other issue with the standard CASTEP methodology for modelling molecules in such environments is the use of Periodic Boundary Conditions (PBCs), as the density of solvated molecules is rarely so high that an identical molecule in the same orientation and atomic configuration is found in close proximity. In a previous eCSE project, eCSE07-6, the periodic DFT approach of CASTEP was extended to enable simulations in Open Boundary Conditions (OBCs) and in the presence of a solvent, by adopting the Minimal-Parameter Solvent Model (MPSM) already developed in ONETEP which handles the effects of the solvent implicitly [1, 2, 3, 4]. This required adapting many of the operators (terms in the total energy expression) that are applied to the electron density, ρ_{elec}, such that they correctly represent OBC systems.

The modification of these operators was done in the expression for the total energy, E, such that the electronic ground state could be determined. But the algorithms that determine the ground state of the atomic configuration require derivatives of E w.r.t. ionic positions in the system, R_{Ion}, which is proportional to the force on the ions:

$$\bar{F}_{ion} = -\frac{dE}{d\bar{R}_{Ion}} \tag{1}$$

The force expressions in CASTEP were not modified to give the correct derivatives for OBC/implicit solvation calculations, which severely limited the usefulness of this functionality within CASTEP. Furthermore, the implementation of the MPSM was also limited in that the model of the solvent was fixed after its initial calculation and not updated if the atoms moved. This meant that unless the atomic configuration remained extremely close to its initial configuration, any results from CASTEP would be completely nonphysical and would not agree with ONETEP results. Hence, both these issues need to be addressed. This is explained in more detail below.

1.1 Minimal-Parameter Solvent Model in CASTEP

In vacuum, the Poisson equation (PE) must be solved to find the electrostatic potential, $\phi_{es}(\mathbf{r})$, due to the electron charge density, $\rho_{elec}(\mathbf{r})$:

$$\nabla^2 \phi_{es}(\mathbf{r}) = -\frac{4\pi}{\epsilon_0}\rho_{elec}(\mathbf{r}) \tag{2}$$

Here, ϵ_0 is the dielectric permittivity of free space. For a system with periodic boundary conditions, the solution can be found very straightforwardly via Fourier transforms. In OBCs, this method is not applicable as the resultant ϕ_{es} is the periodic potential of a periodic ρ_{elec}, and so another method must be used to find the OBC ϕ_{es}. This is done in ONETEP and CASTEP using the multigrid method via DL_MG, a library which uses the multigrid method to solve the PE. For more details on the multigrid method see [5, 6]. DL_MG is capable of handling fully periodic, fully open, and any mixture of boundary conditions when solving the PE. However, within CASTEP the other terms in the total energy expression have only been formulated in the fully open boundary condition. Although the multigrid method scales linearly with the number of points in the grid, it is an iterative procedure and often takes up a large proportion of the calculation time.

DL_MG is also capable of solving the Generalised Poisson Equation, GPE, (also known as the non-homogeneous Poisson equation), which allows the inclusion of the effects of a solvent on the electrostatics of a model system. The GPE is as follows:

$$\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi_{es}(\mathbf{r})) = -4\pi\rho_{sol}(\mathbf{r}) \tag{3}$$

where $\epsilon(\mathbf{r})$ is the non-uniform dielectric permittivity, $\phi_{es}(\mathbf{r})$ is the electrostatic potential due to $\rho_{sol}(\mathbf{r})$, and the total solvated charge density is $\rho_{sol} = \rho_{electrons} + \rho_{ions}$, where ρ_{ions} is the charge density of the ions of the molecule. It is via the non-uniform dielectric permittivity that the solvent can enter our model implicitly. The dielectric permittivity defines how a material polarises in the presence of an electric field, so by setting $\epsilon(\mathbf{r})$ to the value of the dielectric permittivity of water, the electrostatics of our model would act as if water was at point $\mathbf{r}$. There are other effects of the solvent on a solvated molecule that the MPSM handles, such as the entropic contribution from cavitation, and solvent-solute dispersion and repulsion. For more details on the CASTEP implementation of the MPSM, see the [1, 2, 3, 4] papers as well as the CASTEP documentation prepared as part of this eCSE (http://www.castep.org/CASTEP/Documentation).

The dielectric function $\epsilon(\cdot)$, defined in Eq. 7 of ref [3], and in Eq. 1 of ref [1], makes the dielectric permittivity a function of ρ_{elec}, such that where ρ_{elec} is large the dielectric permittivity is that of the vacuum and where

ρ_{elec} is small the dielectric permittivity is that of the solvent, with the function varying smoothly between these two values. The function used for dielectric permittivity is as follows:

$$\epsilon(\mathbf{r}) = 1 + \frac{\epsilon_\infty - 1}{2}\left(1 + \frac{1 - \left(\frac{\rho(\mathbf{r})}{\rho_0}\right)^{2\beta}}{1 + \left(\frac{\rho(\mathbf{r})}{\rho_0}\right)^{2\beta}}\right) \tag{4}$$

and so depends on just two parameters: ρ_0, an electronic density threshold where the transition to the solvent takes place and β, which controls the steepness of the change in ϵ. The third parameter in the above expression is ϵ_∞, the bulk permittivity of the solvent, which is a physical property of the system we are modelling. Thus, the model of the solvent will change every time the electronic density changes. From a physical point of view this is good, since it means the density can respond self-consistently to the polarisation of the solvent and vice versa. From a computational point of view this is rather inconvenient, because it requires extra terms in the energy gradients when determining the ground-state energy (see e.g. Eqs. 5, 6 and 14 in ref [3]). These terms tend to vary rapidly over very localised regions of space, and so accurate calculation usually requires unreasonably fine grids and thus it becomes prohibitively difficult for larger molecules. These additional terms in the energy gradients were not originally implemented in CASTEP due to time constraints in the previous eCSE.

One workaround for the missing gradients problem, which is straightforward, but introduces an approximation, consists in *fixing* the dielectric permittivity, i.e. the solvent, and not allowing it to change shape, for example by converging the system in-vacuum and then using the electron density of this system to construct the solvent model. Prior to the present eCSE, CASTEP worked exclusively using a fixed solvent which was kept fixed for the duration of a calculation and never updated, e.g. due to ionic motion. This meant that although the fixed solvent may be a reasonable approximation for the initial ionic configuration, if the ions were moved 'too far' from this configuration, the physical relevance of the model would become highly doubtful.

One further limitation of the original implementation of the OBCs/implicit solvation model prior to the completion of this eCSE, was the lack of parallelisation of the multigrid calculation. This was due to the incompatible nature of the CASTEP data decomposition of grids in real-space and the data decomposition required for DL_MG to operate in parallel using MPI. DL_MG requires each MPI process to have 1 block of real space data which is stored contiguously, where as the data decomposition in CASTEP has only columns of real-space points along the 'z-direction' being contiguous data, and these columns are divided among the MPI processes in such a way that leads to an effectively random decomposition of these columns among the MPI processes. This problem was overcome by passing all of the real space data onto the lead MPI process in the group which could then perform the DL_MG calculation across threads to obtain ϕ_{es}. This solution worked but drastically reduced the parallelism available on the most expensive part of the calculation and hence the original CASTEP implementation was significantly out-performed by the ONETEP implementation. Addressing this limitation was not one of the aims of the eCSE but it became necessary in order to make the calculation time practical, see section 3.4.

2 eCSE Aims

2.1 Enable New Class Of Calculations By Extending OBCs/Implicit Solvation Implementation in CASTEP

- **OB1a**: Implement force calculation for OBCs/implicit solvation systems.
- **OB1b**: Extend the functionality to include the self-consistently-updated cavity.
- **OB1c**: Validate the implementation against the existing ONETEP implementation.
- **OB1d**: Create QA, unit and regression test cases for inclusion in the CASTEP test suite.

2.2 Enable the CASTEP user community to use the Implicit Solvation model

The implementation of the above functionality will only benefit the CASTEP user community if they are made aware of these additions to the code base. Otherwise, these features will not get used, become under-supported and ultimately contribute to code-rot. In order to avoid this eventuality, work packages were proposed to advertise the functionality to the CASTEP user community and explain how to use the functionality effectively. A range of materials will be developed to support this; specifically:

- **OB2a**: Create a new CASTEP tutorial, focusing on the OBCs/implicit solvation functionality.
- **OB2b**: Create descriptions and examples for use on the CASTEP website.
- **OB2c**: Prepare a webinar to engage with the CASTEP user community more directly, both to promote the project and its outcomes and to introduce users to its capabilities and usage.

3 Work Done

3.1 OBCs/Implicit Solvation Forces

The additional analytical terms required to calculate the ionic force in OBCs/implict solvation calculations were implemented. These were tested in two ways, first via numerical calculations of the forces, calculated via numerical finite differences of the total energy, and second via comparison with a benchmark ONETEP calculation, with the same parameters for the MPSM and same grid size. In order for the comparison to go beyond comparing numerical noise in the forces, one atom in the system was displaced by 0.05 nm from the ground state atomic configuration calculated by ONETEP, which made the forces larger and made it easier to spot discrepancies. The systems chosen for the tests were a small group of similar organic molecules.

The finite difference method was applied to individual analytic force terms as well as the total derivatives of the energy. Each force term and the total energy derivative were found to be in agreement with the finite difference derivatives, to better than the expected numerical accuracy in both cases. When the calculated forces of CASTEP were compared with those calculated by ONETEP, they were found to be within good agreement, differing by less than 0.01 eV/Å. Investigating this small difference further, we found that in most cases the equivalent terms in both codes are in even better agreement than 0.01 eV/Å, and the disagreement appears in the terms due to the ϕ_{es}. It should be noted that there is a difference in how CASTEP and ONETEP pick an ideal grid size for the multigrid calculation, one that is divisible multiple times. ONETEP truncates the grid that is the natural grid for the rest of the ONETEP calculation down to the nearest ideal grid that is smaller than the ONETEP grid. CASTEP however, pads its natural grid up to the nearest ideal grid which is larger than the CASTEP natural grid. It is possible that this is causing the residual discrepancy between the calculated forces.

A regression test has been added to the CASTEP test suite in order detect any changes to the code base which might cause this functionality to break. This is tested via a benchmark implicit solvation calculation to which the calculated forces are compared.

3.2 Self-Consistent Model of Solvent

A new mode of operation for the solvation functionality was implemented. When the self-consistent mode for the dielectric permittivity is used, each time a new density is calculated, the model of the solvent is also updated. This means that the model of the solvent is always up to date with the electron density and when the density changes due to changes in the atomic positions, the model of the solvent responds to this in a physically relevant way. As mentioned in section 2.1, this mode means that there are additional terms in the energy gradient when determining the ground state electron density. These terms vary rapidly across small regions of space, and so very fine real space grids are required for these calculations to converge when calculating the electronic ground state. This is not a limitation of the implementation but rather a result of the choices made when constructing the model.

In order to test the implementation has been done correctly, benchmark calculations were again performed using ONETEP, with the same parameters for the MPSM and same grid size. The output value compared between the two implementations was the free energy of solvation, ΔG_{sol}, which is the difference in energy of the molecule in solvent and the isolated molecule in vacuum.

$$\Delta G_{sol} = E_{solvent} - E_{vacuum} \tag{5}$$

The CASTEP calculation of ΔG_{sol} should be in very good agreement with the ONETEP value for ΔG_{sol} for the same system. In tests across a small group of similar organic molecules the differences in the free energy of solvation of CASTEP and ONETEP was found to be less than 5×10^{-4} eV.

A regression test has been added to the CASTEP test suite in order detect any changes to the code base which might cause this functionality to break. This is tested via a benchmark calculation to which the calculated free energy of solvation is compared.

3.3 Documentation, Tutorial and Webinar

The documentation written for this functionality is distributed with the CASTEP source code, and can be found under `${CASTEP_SOURCE}/Doc/Implicit_Solvation_docs.pdf`. This PDF can also be found on the CASTEP website. The documentation gives a brief description of the model with references to papers that can provide more detailed information. The documentation gives details on how the model is implemented, what limitations there are, and details each of the relevant parameters. Each of the parameters is explained in detail with its importance, its uses, its default value(s) and any conflicting parameters.

A tutorial has also been written in similar style to those already on the CASTEP website. This provides a user with some example input files, takes them through some of the common pitfalls one can run into when running OBCs/implicit solvation calculations and what the output looks like in this scenario and what to do. It also guides the user through the steps to completing a successful calculation, and what physical results can be gleaned from the output. The tutorial will be uploaded to the CASTEP website after it has been quality checked. In addition a webinar will be delivered, recorded and uploaded alongside the tutorial to the CASTEP website.

3.4 Parallel Multigrid Calculations

One of the limitations of the original implementation of the solvation model in CASTEP prior to this eCSE, was its inability to perform the calculation of the electrostatic potential $\phi_{es}(\mathbf{r})$ via equation 2 or 3 in parallel across MPI processes. As noted above, this was due to the difference in the real space domain decomposition required by DL_MG and domain decomposition of CASTEP. CASTEP could still use DL_MG by following these steps:

1. Passing all the data for $\rho(\mathbf{r})$ and $\epsilon(\mathbf{r})$ to the lead MPI process.
2. Lead MPI process arranges the data to be contiguous in real space for DL_MG.
3. Lead MPI process performs the DL_MG calculation across the full real space grid to get $\phi_{es}(\mathbf{r})$.
4. The lead MPI process rearranges the $\phi_{es}(\mathbf{r})$ data into the CASTEP decomposition.
5. $\phi_{es}(\mathbf{r})$ data is passed to the other MPI processes in the calculation.

This was not an ideal implementation and meant that the large calculations required to test the force calculation took a long time to complete. However, by modifying the process described above, a fully parallel implementation of the DL_MG interface could be achieved. In this case the steps become:

1. Passing all the data for $\rho(\mathbf{r})$ and $\epsilon(\mathbf{r})$ to the lead MPI process.

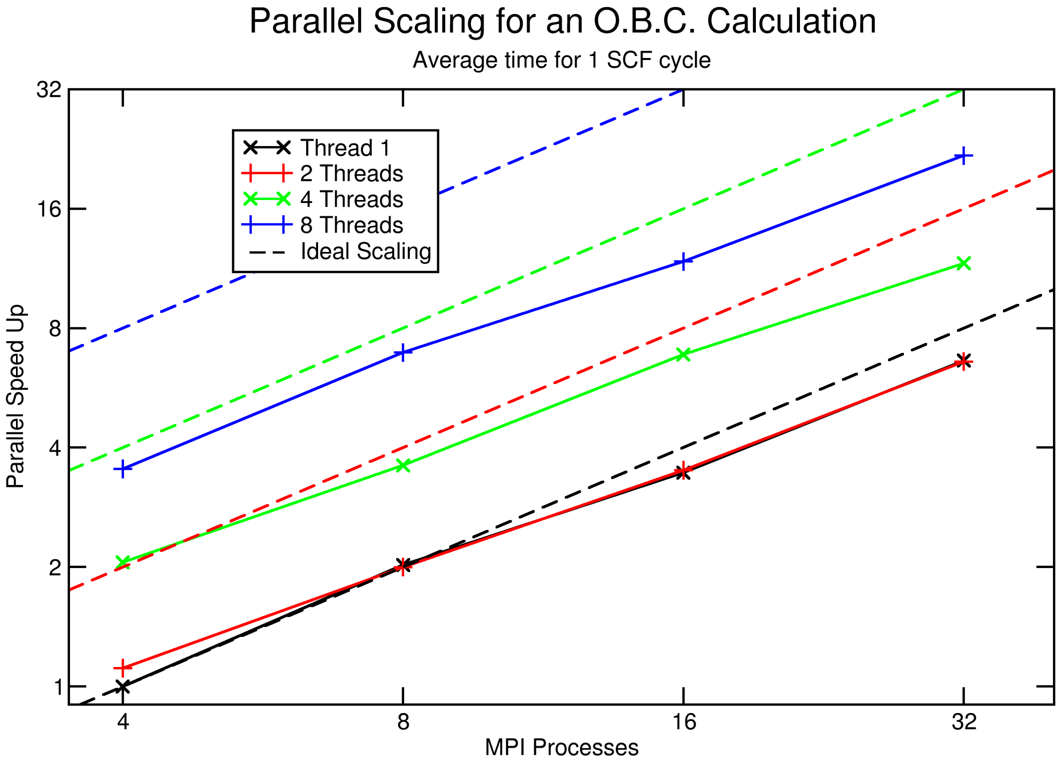

Figure 1: Scaling of Open Boundary Conditions calculation for small organic molecule. The time compared across is the average time taken to perform one Self-Consistent Field (SCF) cycle (each cycle includes one calculation of ϕ_{es} via DL_MG). The average time was determined by taking the average time per SCF cyle between the end of the second and sixth SCF cycle, which was averaged over the 3 different runs of the code with that number of MPI processes and threads. This should remove any noise due to initial set up of the calculation. Each of the calculations was run with the same parameter input and random seed. The parallel speed up is given in relation to the calculation with 4 MPI processes each using 1 thread. The solid lines show the actual scaling of the different number of threads, whilst the dashed line of the same colour shows the ideal scaling with that number of threads.

2. Lead MPI process arranges the data to be contiguous in real space for DL_MG.
3. Lead MPI process passes the contiguous real space data back to the other MPI processes via an `MPI_scatter`.
4. All MPI processes perform the DL_MG calculation to get a distributed $\phi_{es}(\mathbf{r})$.
5. All the $\phi_{es}(\mathbf{r})$ data is passed back to the lead MPI process via an `MPI_gather`.
6. The lead MPI process rearranges the $\phi_{es}(\mathbf{r})$ data into the CASTEP decomposition.
7. $\phi_{es}(\mathbf{r})$ data is passed to the other MPI processes in the calculation.

By this modification of the data decomposition process, a fully MPI parallel implementation of the DL_MG interface to CASTEP was achieved.

The method described above clearly has some communication overhead, which is not performed asynchronously, and so it is unlikely that this method will achieve ideal scaling. To test the scaling of CASTEP with this method, a simple organic molecule was simulated using OBCs. The number of MPI processes and OpenMP threads was varied from 4 → 32 and 1 → 8 respectively in steps of factors of 2. The parallel scaling achieved is shown in Figure 1. This shows that the MPI scaling performs very well across the range of MPI processes chosen and does this at all levels of threading (perhaps surprisingly well given the additional communication overhead described above). It can also be seen that the scaling across threads does not scale in the same way. Going from 1 thread to 2 threads shows little to no speed up, however, going from 2 → 4 approximately doubles the speed, and ditto going from 4 → 8 threads. Clearly there is some time overhead to threading this particular calculation that completely loses any calculation speed up achieved from going from 1 thread to 2 threads, but calculations with higher numbers of threads do overcome this cost and generate a very useful speed up. Clearly parallelism via MPI processes yields superior scaling to parallelism via OpenMP threads, but some speed up can also be achieved via threading. The combination allows one CASTEP calculation to fully use 2 nodes of ARCHER2, with 32 MPI processes with 8 threads each, which achieves a 21× speed up compared to a calculation on 1 node that uses 4 MPI processes with 1 thread.

4 Conclusion

In a previous eCSE project, eCSE07-6, the periodic DFT approach of CASTEP was extended to enable simulations in Open Boundary Conditions (OBCs) and in the presence of a solvent, by adopting the minimal-parameter solvent model already developed in ONETEP (http://www.onetep.org). However, the original implementation had some limitations: only the calculation of the electronic ground state was possible, the solvent could not be updated self-consistently with electronic state of the system, and the solving of the OBCs Poisson equation was performed only by the lead MPI process, significantly reducing the parallelism available on the most expensive part of the calculation.

This eCSE project aimed to address the first two of the above issues, but in the end managed to address all of them. Implementing the calculation of the forces means that it is now possible to optimise the structure in the presence of a solvent, and to perform molecular dynamics simulations to explore the properties and dynamics of the material at finite temperature. This brings the solvation functionality onto much more equal footing with other CASTEP functionality. The solvation functionality now has the option to be run such that the solvent is updated self-consistently with the electronic ground-state, although this requires finer real space sampling and therefore higher memory requirements. The interface to DL_MG (the library used as a Poisson equation solver) was extended to use more of the MPI processes available (in most cases all involved processes), significantly improving the parallel performance and scaling of the CASTEP OBCs/solvation functionality.

The OBCs/solvation functionality was relatively unknown and under-supported by the CASTEP userbase and CASTEP developer group. Up to date documentation has been written to address this, as well as a tutorial of the functionality and a webinar prepared (to be delivered) which will be uploaded to the CASTEP website (http://www.castep.org/CASTEP/Documentation). The increased performance allowed by the extension of the MPI parallelism will also make OBCs/solvation calculations more viable for usage in future.

The new code has been merged into the CASTEP development repository and will be included in the forthcoming 2023 release of the code, after which it will be widely available to ARCHER2 users. Key beneficiaries of the project include the CCP-NC, CCP9 and UKCP communities, where the new tools will aid in reconciling experimental observations with atomic-scale behaviour, helping to guide and interpret future experiments.

This work was funded under the embedded CSE programme of the ARCHER2 UK National Supercomputing Service (http://www.archer2.ac.uk).

References

[1] J. Dziedzic, H. H. Helal, C.-K. Skylaris, A. A. Mostofi, and M. C Payne, *Minimal parameter implicit solvent model for ab initio electronic-structure calculations*, EPL **95** (2011).

[2] J. Dziedzic, S. J. Fox, T. Fox, C. S. Tautermann, and C.-K. Skylaris, *Large-Scale DFT Calculations in Implicit Solvent – A Case Study on the T4 Lysozyme L99A/M102Q Protein*, International Journal of Quantum Chemistry **113** issue 6 (2013).

[3] D. A. Scherlis, J.-L. Fattebert, F. Gygi, M. Cococcioni, and N. Marzari, *A unified electrostatic and cavitation model for first-principles molecular dynamics in solution*, J. Chem. Phys. **124** (2006).

[4] O. Andreussi, I. Dabo and N. Marzari, *Revised self-consistent continuum solvation in electronic-structure calculations*, J. Chem. Phys. **136** (2012).

[5] J. C. Womack, L. Anton, J. Dziedzic, P. J. Hasnip, M. I. J. Probert, and C.-K. Skylaris, J. Chem. Theory Comput. **14**, 1412 (2018).

[6] L. Anton, J. Womack, and J. Dziedzic, *DL_MG multigrid solver* (2022) http://www.dlmg.org