

ARCHER eCSE technical report

Implementation and optimisation of advanced solvent modelling functionality in CASTEP and ONETEP

James C. Womack^a, Lucian Anton^b, Jacek Dziedzic^{a,c}, Phil J. Hasnip^d,
Matt I. J. Probert^d, and Chris-Kriton Skylaris^a

^a*Department of Chemistry, University of Southampton, Highfield, Southampton, SO17 1BJ, United Kingdom*

^b*Cray U.K. Limited, Broad Quay House, Prince Street, Bristol, BS1 4DJ, United Kingdom*

^c*Faculty of Applied Physics and Mathematics, Gdańsk University of Technology, Gdańsk 80-233, Poland*

^d*Department of Physics, University of York, Heslington, York YO10 5DD, United Kingdom*

21st August 2017

Abstract

Implicit solvent models provide a simple, yet accurate means to incorporate solvent effects into electronic structure calculations. Such models avoid the computational expense of explicitly modelling solvent molecules by representing the solvent implicitly, for example as a polarizable dielectric medium. In this report, we describe the implementation and extension of an one such model, the minimal parameter solvation model (MPSM), in two electronic structure packages: CASTEP and ONETEP. In the MPSM, the electrostatic potential which describes the interaction of the implicit solvent and solute is determined by direct solution of the nonhomogeneous Poisson equation (NPE), with a dielectric permittivity derived directly from the quantum mechanical electron density.

The MPSM was previously implemented in ONETEP, a linear scaling density functional theory (DFT) package. In this implementation, an efficient second-order multigrid solver, DL_MG, was employed to solve the NPE with open (Dirichlet) boundary conditions (BCs). To allow the model to describe the solvation of systems with natural periodicity, such as surfaces and polymers, the model was extended to support periodic BCs. We describe the modifications to the model and software necessary to support periodic BCs and present the results of test calculations which demonstrate the validity of our implementation.

To improve the accuracy of the second-order solutions produced by DL_MG for use in the MPSM, the high-order defect correction method is employed. During this project we ported the original implementation of this method in ONETEP to DL_MG, allowing the solver to directly produce higher-order-corrected solutions to the NPE. In this report, we describe the changes made to DL_MG in order to incorporate and optimize the defect correction. We also present results which demonstrate the improved performance of the defect correction in DL_MG—a 1.2–1.5× speed-up was demonstrated over the original implementation in large-scale DFT calculations.

The full MPSM was implemented in CASTEP, a state-of-the-art plane-wave-pseudopotential DFT code, using DL_MG to solve the NPE. This involved some modifications to the behaviour of the model compared to the implementation in ONETEP, to account for the differences in the theoretical methods employed by CASTEP. These modifications are outlined in the report, alongside a general description of the major new functionality implemented in CASTEP. We also present test calculations on several small molecules which validate the implementation in CASTEP, producing free energies of solvation which differ from ONETEP by ~ 0.1 kcal mol⁻¹ or less.

While significant new capabilities were added to DL_MG, CASTEP and ONETEP during the course of this project, some components of the original project goals were not completed. We therefore end the report by defining the key tasks that remain to be completed and suggesting how these can be attempted in the context of the foundations laid during this project.

1 Project overview

1.1 Introduction

Modern electronic structure software packages enable the quantum behaviour of matter to be modelled with great accuracy. By efficiently utilizing massively parallel computer hardware, packages such as CASTEP¹ and ONETEP² have enabled complex and subtle quantum phenomena to be studied at scales that would have been inconceivable only decades ago. This capability has opened the door to electronic structure calculations for technologically interesting extended systems, such as biomolecules, nanoparticles and materials. For these large-scale electronic structure simulations to reliably inform the design and development of new technologies, the effect of the environment surrounding the quantum system must be accounted for. In many cases, this is some kind of liquid solvent.

One approach to account for solvent effects in electronic structure simulations is simply to add a number of solvent molecules to the quantum system. While conceptually simple, this “explicit solvent” approach is challenging from both theoretical and computational perspectives. Theoretically, it requires some method of dealing with very large numbers of degrees of freedom, while the vast increase in the number of atoms that must be treated is computationally problematic. Implicit solvent models address both of these issues by representing the solvent environment in an implicit manner, forgoing full atomic detail in favour of simplified theoretical and computational methods.

A successful and long-standing family of implicit solvent models represent the solvent as a polarizable dielectric continuum surrounding the solute, which exists within a cavity in the continuous medium. The interaction of the solute with the solvent is represented by a reaction potential which is incorporated into the quantum mechanical Hamiltonian. The mutual polarization of solvent and solute can then be captured by self-consistently solving for the reaction potential and solute charge. The polarizable continuum model (PCM)³ and conductor-like screening model (COSMO)⁴ are two well-known and widely adopted models of this type. For a detailed review describing the continuum dielectric self-consistent reaction field (SCRF) method and various formulations of this, see Ref. 5.

This project is focused upon a specific variant of the continuum dielectric SCRF method in which the dielectric cavity is defined in terms of the quantum mechanical charge density of the solute and the reaction potential is obtained by direct solution of the non-homogeneous Poisson equation (NPE):

$$\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) = -4\pi n(\mathbf{r}). \quad (1)$$

Given a solute charge density, $n(\mathbf{r})$, and non-uniform dielectric permittivity, $\epsilon(\mathbf{r})$, Eq. 1 can be solved to obtain the total electrostatic potential,

$$\phi(\mathbf{r}) = \phi_0(\mathbf{r}) + \phi_r(\mathbf{r}), \quad (2)$$

which is the sum of the electrostatic potential, $\phi_0(\mathbf{r})$, due to the solute charge (i.e. the total charge from nuclei and electrons) and the reaction potential, $\phi_r(\mathbf{r})$, arising from the polarization of the surrounding dielectric medium. The solvent model in question was developed by Dziedzic *et al.*^{6,7} for use in large-scale density functional theory (DFT) calculations, and is a refinement of earlier work by Fattebert and Gygi^{8,9} and Scherlis *et al.*¹⁰ To distinguish the revised model from the earlier Fattebert-Gygi-Scherlis (FGS) model and other FGS-derived models, such as the self-consistent continuum solvation (SCCS) model of Andreussi *et al.*,¹¹ the model used in this work will be referred to as the minimal parameter solvation model (MPSM). This name emphasizes one of the key advantages of the model—the use of a very small number of empirical parameters compared to other widely used continuum solvent models. The theoretical underpinnings of the FGS model and refinements offered by the MPSM are outlined in section 2.1.

A means of efficiently solving Eq. 1 is critical for the application of the MPSM to extended systems of technological interest. A key aspect of this in the context of large scale electronic structure calculations is parallel scaling—in order for a solver to be suitable for use with highly parallelized packages such as CASTEP and ONETEP, it must scale effectively to thousands of CPU cores. To satisfy this requirement,

DL_MG, a parallel Poisson solver for electronic structure calculations was developed, with initial funding under the HeCTOR dCSE programme.¹² The solver uses a multigrid approach to solve Eq. 1 with a second-order discretization of the operator, $\nabla \cdot \epsilon \nabla$. A later ARCHER eCSE project extended DL_MG, adding the capability to solve the Poisson-Boltzmann equation,

$$\nabla \cdot (\epsilon(\mathbf{r}) \nabla \phi(\mathbf{r})) = -4\pi \left(n(\mathbf{r}) + n_{\text{ionsol}}[\phi](\mathbf{r}) \right), \quad (3)$$

which allows solvation in saline solutions to be modelled by the addition of a density of ions in solution, $n_{\text{ionsol}}[\phi](\mathbf{r})$ which depends non-linearly on the potential, $\phi(\mathbf{r})$.¹³ The design and implementation of DL_MG, as relevant to this work, is outlined in section 2.2.

A full implementation of the MPSM in ONETEP, using DL_MG to solve the Poisson equation, has enabled ONETEP users to perform large scale DFT calculations in the presence of implicit solvent. A few notable examples of recent work supported by the solvation model in ONETEP include:

- the development of a quantum mechanics-based Poisson-Boltzmann surface area (QM-PBSA) method for computing free energies of binding for ligands and proteins using ONETEP;¹⁴
- the implementation of an energy decomposition analysis (EDA) scheme in ONETEP and its application to the study of biomolecular interactions;¹⁵ and
- the development of a computational approach for predicting the effect of solvent on optical transitions (solvatochromism) using time-dependent DFT (TDDFT) and a hybrid explicit/implicit solvent representation.¹⁶

A significant reason for the success of the MPSM in ONETEP is the use of the defect correction method to iteratively correct the second-order solutions returned by DL_MG to higher orders. This technique is essential for obtaining acceptable accuracy in solvation calculations—as demonstrated in Ref. 7, defect correction can dramatically improve the accuracy of solutions to Eq. 1 with respect to the original second-order solution. A key benefit of the defect correction method from a software engineering perspective is its relative simplicity—it is a simple iterative scheme which requires no modification of the multigrid solver (see section 2.3). This is how the defect correction was originally implemented for the MPSM in ONETEP, with the defect correction procedure occurring outside of DL_MG, which acts purely as a second-order solver.

The development of the MPSM in ONETEP was initially targeted at the study of isolated solutes and the model was thus implemented using fully open (Dirichlet) boundary conditions (BCs). To mitigate the potentially significant computational cost of evaluating the Dirichlet BCs on the cell boundaries, a coarse-graining scheme was developed (see section 2.1), which reduces the prefactor of this operation by 2-3 orders of magnitude.⁷ While open BCs are suitable for many types of solvated systems, there are significant classes of materials where periodic BCs along one or more dimension are more appropriate. Examples include polymers, nanotubes, solid-liquid interfaces and surface-supported nanoparticles, relevant to the development of a range of technologies, including batteries, solar cells and catalysts.

This project was designed to capitalize on the success of the ONETEP/DL_MG implementation of the MPSM by improving the existing implementation and extending its availability. In particular, the project was conceived with the following broad goals:

Enable the CASTEP user community to perform simulations using the MPSM

Implementing the full MPSM in CASTEP would allow CASTEP simulations to incorporate solvent effects in a simple and efficient manner, without the need to resort to explicit solvent representations.

Enable MPSM simulations to be performed with generalized periodic/open BCs

Support for generalized periodic/open BCs would allow the MPSM to be applied to a range of materials which exhibit natural periodicity along one or more dimensions.

Enable the defect correction method to be applied directly within DL_MG

The integration of the defect correction into DL_MG would make the library more versatile, allowing it to provide high-order-corrected solutions to the Poisson equation directly to electronic structure codes.

The remainder of this report describes work undertaken over the 12-month duration of this project to realize the goals just outlined. The key technical objectives of the project, as originally proposed, are described in section 1.2, while the theoretical underpinnings of this work are presented in section 2. A section is dedicated to each of the major technical work packages (as outlined in section 1.2), describing the work completed, any significant limitations, implementation highlights and results (sections 3 to 5). Finally, concluding remarks and reflections on the project as a whole are offered in section 6.

1.2 Objectives

This project consists of three major work packages (WP1-WP3) relevant to the broad goals outlined in section 1.1. A fourth work package (WP4) is concerned with publication, reporting and dissemination of code, and includes the creation and distribution of this report. This report will primarily deal with the work performed for WP1-WP3.

The major technical objectives for WP1, WP2 and WP3, as initially conceived, are as follows:

WP1: DL_MG defect correction

- Implement the defect correction method within DL_MG to enable higher-order solution of the Poisson equation within the solver.
- Implement alternative iterate change criteria for the multigrid solver to improve convergence control.

WP2: ONETEP solvent model extensions

- Add support for fully periodic and mixed periodic and open boundary conditions (BCs) to the solvent model in ONETEP to complement the existing support for fully open BCs.
- Implement support for all BC types in both ONETEP and CASTEP to allow systems with and without inherent periodicity to be studied in solution.

WP3: CASTEP solvation

- Port the complete implicit solvent model from ONETEP to CASTEP to enable CASTEP calculations to account for solvation effects.
- Ensure that the ported model is adapted to fit within the data representation and parallel strategy employed by CASTEP.

An overarching goal of this project, relevant to WP1-WP3, is that the developments outlined above should be completed while maintaining or improving upon the numerical accuracy and parallel performance of existing implementation of the model. In the case of WP3, where the objective is to fully implement the model in CASTEP, the goal is to obtain good numerical agreement with the existing model in ONETEP and parallel performance comparable to standard CASTEP calculations in vacuum.

2 Theoretical background

2.1 Implicit solvent model

In the continuum dielectric SCRF methods, the interaction of the polarizable dielectric, which represents the solvent, and solute occurs via a reaction potential, $\phi_r(\mathbf{r})$. The reaction potential forms part of the total

electrostatic potential, $\phi_{\text{NPE}}(\mathbf{r})$ (Eq. 2), which may be obtained by solving the non-homogeneous Poisson equation (NPE, Eq. 1) for a given charge density, $n(\mathbf{r})$, and non-uniform dielectric permittivity, $\varepsilon(\mathbf{r})$.

The dielectric medium is polarized by the total charge of the solute, so the NPE should be solved for the combined charge of the nuclei and electrons,

$$n_{\text{tot}}(\mathbf{r}) = n_{\text{elec}}(\mathbf{r}) + n_{\text{ionic}}(\mathbf{r}), \quad (4)$$

to obtain the total electrostatic potential, $\phi_{\text{NPE}}(\mathbf{r})$, representing the potential due to the total charge of the solute and the polarization of the dielectric continuum. The total electrostatic energy of the system is then given by

$$E_{\text{es}}[n_{\text{tot}}] = \frac{1}{2} \int d\mathbf{r} n_{\text{tot}}(\mathbf{r}) \phi_{\text{NPE}}[n_{\text{tot}}](\mathbf{r}), \quad (5)$$

where the factor of 1/2 ensures that interactions are not double-counted.

In an SCRF calculation, the reaction field and solute charge density are solved for self-consistently. In the context of DFT and Hartree-Fock theory, this can be achieved by modifying the one-particle Hamiltonian, replacing the usual vacuum electrostatic potential with the electrostatic potential in solvent. Self-consistently solving the Kohn-Sham equations subject to this potential naturally incorporates the electrostatic interactions of the solute with the dielectric continuum into the DFT calculation.

The defining feature of the Fattbert-Gygi electrostatic solvation model,⁸ upon which the MPSM is built, is the use of a charge-density-dependent non-uniform dielectric function which smoothly varies between $\varepsilon(\mathbf{r}) = 1$ (inside the cavity) and $\varepsilon(\mathbf{r}) = \varepsilon_{\infty}$ (in the bulk solvent). This function has the form

$$\varepsilon(\mathbf{r}) = 1 + \frac{\varepsilon_{\infty} - 1}{2} \left(1 + \frac{1 - (n_{\text{elec}}(\mathbf{r})/n_0)^{2\beta}}{1 + (n_{\text{elec}}(\mathbf{r})/n_0)^{2\beta}} \right), \quad (6)$$

and is defined in terms of the electron density at \mathbf{r} , $n_{\text{elec}}(\mathbf{r})$, the bulk permittivity of the solvent, ε_{∞} , and two parameters: β and n_0 . The two parameters modify the characteristics of solvent/cavity interface: n_0 is the electron density in the centre of the interface region (i.e. where $\varepsilon(\mathbf{r}) = 1 + (\varepsilon_{\infty} - 1)/2$), while β determines the width of the interface. These two parameters can be empirically determined based on experimental data (see Refs. 6, 8, 10).

The electrostatic interaction of the solvent and solute plays an important role in the solvation process, but is not the full story. To predict how favourable the solvation process for a given solute and solvent is, we can calculate the free energy of solvation, ΔG_{sol} , which we define as the change in free energy upon transferring the solute from vacuum to solution with a fixed nuclear configuration. This process involves non-electrostatic effects, in particular due the work of creating a cavity in the solvent and dispersion-repulsion interactions between the solvent and solute. In developing theoretical treatments of solvation, it is conceptually useful to partition the free energy of solvation into electrostatic and non-electrostatic components,

$$\Delta G_{\text{sol}} = \Delta G_{\text{es}} + \Delta G_{\text{non-es}}, \quad (7)$$

which can then be treated independently.

In the MPSM, the non-electrostatic component of the free energy of solvation is computed based on the surface area of the cavity, $S[n_{\text{elec}}]$, and macroscopic surface tension of the solvent, γ , as described in Ref. 10. The MPSM builds upon the model of Ref. 10 by introducing an effective surface tension, γ_{eff} , scaled by a constant to account for solute-solvent dispersion repulsion (see Ref. 6):

$$\Delta G_{\text{non-es}} = \gamma_{\text{eff}} S[n_{\text{elec}}]. \quad (8)$$

Since the solute-solvent interface in the MPSM is a smooth transition, the surface area of the cavity, $S[n_{\text{elec}}]$, is defined as the volume of a thin film around n_0 , divided by the thickness of this film. The surface area of the film can be calculated straightforwardly by integration of an appropriate integrand over the volume of the simulation cell—see section II.C of Ref. 10 for details.

The solution of the NPE (Eq. 1) requires the imposition of boundary conditions (BCs). The MPSM was originally developed for modelling isolated solutes, so open BCs were adopted for the vacuum and

solvent calculations required to calculate ΔG_{sol} . In practice, the BCs are calculated approximately using a charge coarse-graining approach and assuming a homogeneous dielectric permittivity equal to the bulk permittivity, ϵ_∞ , i.e.

$$\phi_{\text{es}}^{\text{BC}}(\mathbf{r}) \approx \frac{1}{\epsilon_\infty} \sum_i^{N_{\text{CG}}} \frac{n_{\text{tot}}^{\text{CG}}(\mathbf{R}_i)}{|\mathbf{r} - \mathbf{R}_i|} \quad \text{for } \mathbf{r} \in \partial\Omega, \quad (9)$$

where the BCs are evaluated on simulation cell faces, $\partial\Omega$, as the potential due to a set of point charges, $n_{\text{tot}}^{\text{CG}}(\mathbf{R}_i)$, which provide a coarse-grained representation of the charge density in the cell (see Ref. 7 for further details).

In vacuum DFT calculations it is typical to use point charges to represent nuclear/ionic core charges. This representation is not well-suited for use in multigrid calculations (section 2.2), where the total (electronic and nuclear) charge density is a continuous quantity represented on a real-space grid. The MPSM follows Refs. 10, 17 and uses a distributed charge or ‘‘smeared ion’’ representation of ionic core charges when calculating the electrostatic potential, i.e.

$$n_{\text{si}}(\mathbf{r}) = \sum_I^{N_{\text{ions}}} n_I(\mathbf{r}), \quad (10)$$

where the smeared ionic charges are represented by Gaussians, normalized to the charge of the ionic core, Z_I , with an adjustable width parameter, σ :

$$n_I(\mathbf{r}) = -\frac{Z_I}{(\sigma\pi^{1/2})^3} \exp\left(-\frac{|\mathbf{r} - \mathbf{R}_I|^2}{\sigma^2}\right). \quad (11)$$

The total electrostatic potential (Eq. 2) can then be obtained by solving the NPE (Eq. 1) for the total solute charge density (Eq. 4, with $n_{\text{ionic}}(\mathbf{r})$ given by $n_{\text{si}}(\mathbf{r})$) and corresponding density-dependent dielectric function (Eq. 6).

The use of smeared ionic core charges to construct $n_{\text{tot}}(\mathbf{r})$ for use in Eq. 1 modifies the form of the ion-ion and ion-electron interactions compared to the representation used in pseudopotential-based DFT calculations. To restore the usual representation of these interactions, it is necessary to introduce correction terms. The total electrostatic energy (Eq. 5), corrected for smeared ion interactions is then,

$$E_{\text{es}}[n_{\text{tot}}] = \frac{1}{2} \int d\mathbf{r} n_{\text{tot}}(\mathbf{r}) \phi_{\text{NPE}}[n_{\text{tot}}](\mathbf{r}) + (E_{\text{locps}}[n_{\text{elec}}] - E_{\text{si-elec}}[n_{\text{elec}}]) + (E_{\text{ion-ion}} - E_{\text{si-si}}), \quad (12)$$

where the usual local pseudopotential, $E_{\text{locps}}[n_{\text{elec}}]$, and ion-ion, $E_{\text{ion-ion}}$, energies have been added and the corresponding smeared ion-electron, $E_{\text{si-elec}}[n_{\text{elec}}]$, and smeared ion-smeared ion energies have been subtracted. When applying the electrostatic potential in an SCF procedure, the smeared ion-electron interaction may be replaced by the local pseudopotential-electron interaction in the same way, so that

$$\phi_{\text{es}}^{\text{corr}}(\mathbf{r}) = \phi_{\text{NPE}}[n_{\text{tot}}](\mathbf{r}) + (V_{\text{locps}}(\mathbf{r}) - V_{\text{si}}(\mathbf{r})). \quad (13)$$

See the appendices of Refs. 7, 10 for further details on the smeared ion representation and the correction terms introduced.

In practical calculations, the MPSM can be applied with fixed or self-consistently optimized cavity. In a fixed-cavity calculation, the dielectric function (Eq. 6) is constructed using the ground state density of the solute in vacuum and kept fixed over the course of the calculation in solvent. In a self-consistent cavity calculation the dielectric function is updated during the SCRF process in response to changes in the electron density. This introduces additional potential terms to the one-electron Hamiltonian, which are not present in the fixed-cavity variant. These terms originate in the dependence of the dielectric function $\epsilon[n_{\text{elec}}](\mathbf{r})$ and cavity surface area $S[n_{\text{elec}}]$ on the electron density and appear in the functional derivatives of E_{es} and $\Delta G_{\text{non-es}}$ with respect to n_{elec} (see Refs. 6, 8, 10).

The use of a self-consistent cavity introduces numerical noise to the evaluation of the functional derivative $\delta E_{\text{es}}/\delta n_{\text{elec}}$, necessitating the use of finer grids and higher-order finite differences and thus

increasing computational cost. It has been demonstrated that the error introduced by using a fixed cavity, rather than a self-consistently optimized cavity, is modest⁶ and this has therefore become the preferred mode of operation for the MPSM in ONETEP. In this report, only the fixed-cavity variant of the MPSM is considered.

2.2 Multigrid solver

The multigrid method¹⁸ is a technique for solving a broad range of linear and non-linear problems in which a hierarchy of progressively coarsened grids is employed to improve rates of convergence over single-grid approaches. By transforming a problem to coarser grids, a multigrid solver can rapidly remove low frequency modes in the error which would otherwise cause single-grid solvers to converge very slowly to a solution. Multigrid solvers have been successfully applied in many fields of study. For example, in the field of electronic structure theory, the multigrid approach has been employed to efficiently solve the Kohn-Sham equations in real-space implementations of DFT—see Refs. 19, 20.

The multigrid approach is based on two key observations:²¹ (i) iterative methods (such as the Jacobi or Gauss-Seidel methods) can rapidly attenuate high frequency components of the error but are slow to remove low frequency components; and, (ii) smooth errors on a fine grid can be well approximated on a coarser grid. These observations may be applied to accelerate convergence via three basic operations:

Smoothing

Apply an iterative method to remove higher frequency components of the error on a given grid.

Restriction

Transfer the defect computed on a finer grid to a coarser grid.

Prolongation

Transfer the error computed on a coarser grid to a finer grid.

Using these operations, lower frequency components of the error can be removed by applying smoothing operations on coarser grids, where these lower frequency components appear as higher frequency components.

For the general linear equation

$$\hat{A}u = f, \quad (14)$$

a simplified two-grid procedure with a fine grid spacing of h and coarse grid spacing of $2h$ has the form:

1. Apply an iterative method to solve $\hat{A}_h u_h = f_h$ with the current guess u_h^m to obtain a new guess, \bar{u}_h^m , with smoothed error.
2. Compute the defect for the smoothed solution on the fine grid, $\bar{r}_h^m = f_h - \hat{A}_h \bar{u}_h^m$.
3. Transfer the defect to the coarse grid using the restriction operator, i.e. $\hat{I}_h^{2h} \bar{r}_h^m = \bar{r}_{2h}^m$.
4. Apply an iterative method to solve the defect equation on the coarse grid, $\hat{A}_{2h} e_{2h} = r_{2h}$, yielding the error, e_{2h}^m .
5. Transfer the error from the coarse grid to the fine grid using the prolongation operator, i.e. $\hat{I}_{2h}^h e_{2h}^m = e_h^m$.
6. Update the current guess on the fine grid using the error, $u_h^{m+1} = \bar{u}_h^m + e_h^m$.
7. Smooth the error in the updated solution using an iterative method.

This procedure may be repeated iteratively until a convergence criterion is satisfied. In practice, multiple levels of progressively coarser grids can be employed, with recursive application of the two grid procedure outlined above. See Ref. 21 for a detailed exposition of the multigrid approach.

To solve the NPE for use in the MPSM both CASTEP¹ and ONETEP² employ DL_MG, a multigrid solver library developed in previous HeCTOR dCSE and ARCHER eCSE projects.^{12,13} The multigrid scheme implemented in DL_MG was based on standard recommendations — technical details are available in the original technical report¹² and in documentation accompanying the software.²² For the purposes of this project, it is sufficient to note the following key aspects of the solver:

- Uses a second-order discretization of the differential operator $\nabla \cdot \varepsilon \nabla$ with a 7-point finite difference stencil.
- Capable of solving the NPE (Eq. 1) and Poisson-Boltzmann equation (Eq. 3) to second order in fully periodic, fully open and mixed open/periodic BCs.
- Utilizes MPI parallelism based on a 3-dimensional cuboidal distribution of data among MPI ranks.
- Utilizes OpenMP parallelism to distribute work among threads for each MPI rank.
- Combined MPI and OpenMP parallelism scales effectively to thousands of cores.

During this project, further capabilities were added to the solver library, most notably the incorporation of the defect correction scheme.

2.3 Defect correction

The approximate solution of the NPE (Eq. 1), is associated with an error,

$$e = \phi - \phi', \quad (15)$$

where ϕ is the exact solution of the NPE and ϕ' is the approximate solution. The defect (or residual),

$$r = f - \hat{A}\phi' \quad (16)$$

is related to the error by

$$\begin{aligned} \hat{A}(\phi - \phi') &= f - \hat{A}\phi' \\ \hat{A}e &= r, \end{aligned} \quad (17)$$

where we have used simplified notation for the terms in the NPE: $\hat{A} \equiv \nabla \cdot \varepsilon[n](\mathbf{r})\nabla$ and $f \equiv -4\pi n(\mathbf{r})$. This relationship can be used to iteratively correct an approximate solution obtained with a lower-order discretization of \hat{A} towards a higher-order solution—this is the “high-order defect correction” method (or simply “defect correction”). The key steps in the defect correction method are outlined below. For additional details, see Refs. 7 (appendix A), 21 (ch. 5) and 23.

In the implementation of the MPSM in CASTEP and ONETEP, the high-order defect correction method is used to correct the second-order solution to the NPE produced by DL_MG to higher orders. In this case, the defect correction procedure starts with the solution of the second order NPE,

$$\hat{A}_2\phi^{(0)} = f, \quad (18)$$

by DL_MG, where \hat{A}_2 is the second-order discretization of the operator \hat{A} on the fine grid used by the multigrid solver.* An approximation to the high-order defect is then computed using a higher-order discretization of \hat{A} , i.e.

$$r_d^{(i)} = f - \hat{A}_d\phi^{(i)} \quad (19)$$

*Note that the meaning of the subscripts for \hat{A} , r and e used here differ from those used in section 2.2. In section 2.2, the subscripts of these quantities refer to the spacing of the grid on which the quantity is represented, while in this section, the subscripts refer to the order of discretization used in finite differences.

where $d > 2$. This high-order defect can then be used to solve Eq. 17 for a second-order approximation to the error, $e_{2,d}^{(i)}$, again using DL_MG:

$$\hat{A}_2 e_{2,d}^{(i)} = r_d^{(i)}. \quad (20)$$

Finally, using Eq. 15, the approximate error is used to correct the potential:

$$\phi^{(i+1)} = \phi^{(i)} + e_{2,d}^{(i)}. \quad (21)$$

The updated $\phi^{(i+1)}$ then becomes the starting point for the next iteration of the defect correction loop, being used to construct the residual $r_d^{(i+1)}$ using Eq. 19. This process can be repeated until a convergence criterion is met (e.g. when $|\phi^{(i)} - \phi^{(i+1)}|$ is reduced to below some threshold), allowing control of the accuracy of the potential.

In the original implementation of the MPSM in ONETEP,^{6,7} the defect correction loop was implemented in ONETEP, calling routines from DL_MG to provide the initial second-order solution (Eq. 18) and then to solve the defect equation (Eq. 20) in each iteration of the defect correction. As part of this project, the defect correction was implemented within DL_MG, enabling defect-corrected solutions to the NPE to be obtained with a single call to the library—see section 3 for details. Incorporating the defect correction into DL_MG simplified the subsequent implementation of the MPSM within CASTEP (section 5), avoiding the need to reimplement the defect correction.

2.4 CASTEP

CASTEP¹ is an electronic structure software package based on the plane-wave pseudopotential formulation of DFT. The code is highly parallelized, combining distributed and shared memory parallelism (MPI and OpenMP) to scale effectively to thousands of cores. It is also under active development, with a growing list of capabilities which supplement or extend the core plane-wave DFT functionality—a list of features, with relevant publications and documentation, is available on the CASTEP website.²⁴ The academic distribution of the software is free to UK academics and it is available to industrial users as part of BIOVIA's Materials Studio package.²⁵

The plane-wave pseudopotential DFT approach has been comprehensively covered in numerous books and reviews and will not be described in detail here. In the following we provide only a brief account of the key features of the method, as relevant to the implementation of the MPSM—curious readers are referred to Refs. 26–28.

To solve the NPE (Eq. 1) for the electrostatic potential, in vacuum or solution, the charge of the system (Eq. 4) is required. The electron density in CASTEP is given by

$$n_{\text{elec}}(\mathbf{r}) = \sum_{m,\mathbf{k}} f_{m\mathbf{k}} |\psi_{m,\mathbf{k}}(\mathbf{r})|^2, \quad (22)$$

with occupancies $f_{m\mathbf{k}}$ and where the summation is over eigenfunctions of the one-electron Hamiltonian, m , for each wavevector, \mathbf{k} . The one-electron eigenfunctions are represented as expansions in a periodic plane-wave basis set,

$$\psi_{m,\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{G}m\mathbf{k}} \exp(i(\mathbf{G} + \mathbf{k}) \cdot \mathbf{r}), \quad (23)$$

with expansion coefficients, $c_{\mathbf{G}m\mathbf{k}}$. The quality of the plane-wave basis set is determined by a single parameter, the cutoff energy, E_c , which determines the highest energy plane-wave available in the basis.

The nuclei and core electronic states are represented by pseudopotentials, with which the one-electron wavefunctions (Eq. 22) interact. The ionic charge, $n_{\text{ionic}}(\mathbf{r})$, is thus represented as a set of point charges centred on the nuclei, each with the combined charge of the nucleus. As mentioned in section 2.1, this representation of the ionic charge is problematic for multigrid operations, and the MPSM introduces a smeared ion representation (Eq. 10).

The ground state energy of a system is determined by self-consistently optimizing the plane-wave expansion coefficients, $c_{\mathbf{G}m\mathbf{k}}$, for the electronic states, to produce the electron density, $n_{\text{elec}}(\mathbf{r})$, which

minimizes the total energy functional, $E_{\text{tot}}[n_{\text{elec}}]$. Of particular relevance to the MPSM is the electrostatic energy component of the total energy, $E_{\text{es}}[n_{\text{tot}}]$ (Eq. 12), through which the electrostatic potential obtained by solving the NPE enters into the total energy expression.

CASTEP offers several methods for self-consistently obtaining the ground state energy and density of a system (see the CASTEP documentation for details²⁴), but for the purposes of this report it is sufficient to note that two slightly different total energy expressions are used depending on the method. In one expression, each term contributing to the total energy is computed separately and summed:

$$E_{\text{tot}}[n_{\text{elec}}] = E_{\text{kin}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{nl}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{xc}}[n_{\text{elec}}] + \{E_{\text{Hart}}[n_{\text{elec}}] + E_{\text{locps}}[n_{\text{elec}}] + E_{\text{ion-ion}}\}, \quad (24)$$

where the individual contributions are the kinetic, E_{kin} , non-local pseudopotential, E_{nl} , exchange-correlation, E_{xc} , Hartree, E_{Hart} , local pseudopotential, E_{locps} , and ion-ion interaction, $E_{\text{ion-ion}}$, energies. While the total energy is a functional of the electron density, the kinetic energy and non-local pseudopotential terms depend on the electron density via the one-electron wavefunctions (Eq. 23) and the ion-ion interaction depends only on the ionic centres and charges. In the other expression,

$$E_{\text{tot}}[n_{\text{elec}}] = E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{xc}}^{\text{corr}}[n_{\text{elec}}] - E_{\text{Hart}}[n_{\text{elec}}] + E_{\text{ion-ion}}, \quad (25)$$

the total energy is computed by correcting the band structure energy, $E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}]$, which is a sum over eigenvalues of the one-electron Hamiltonian and includes contributions from the Hartree potential and local pseudopotential. Of particular relevance to this work is the double-counting of the Hartree energy in E_{BS} , which must be corrected in the total energy expression by subtracting E_{Hart} .

The modified electrostatic energy used in the MPSM (Eq. 12) can be considered to replace the three electrostatic terms in Eq. 24 (these are grouped in curly braces). In Eq. 25, the situation is more complicated, since some electrostatic terms are included in the band-structure energy and some must be corrected for double counting—see section 5.3.

2.5 ONETEP

ONETEP² is a linear-scaling density-matrix DFT package, focused on the application of high-accuracy DFT to extended systems containing thousands of atoms. Such large systems present difficulties for conventional DFT approaches (as used in CASTEP), where the computational cost asymptotically scales as $O(N^3)$, with N a measure of the size of the system. In linear-scaling DFT methods, $O(N)$ scaling is achieved by exploiting the locality of electronic interactions, making such large scale calculations practical.

ONETEP combines distributed and shared memory parallelism (MPI and OpenMP)^{29,30} with efficient sparse-matrix algebra routines³¹ to scale across thousands of cores. Using ONETEP, linear-scaling DFT calculations on systems with tens of thousands of atoms are practical using tier-1 computing resources, such as ARCHER. As with CASTEP, ONETEP is under active development with an extensive and growing list of features—see the ONETEP website for an up-to-date summary of recent methodological developments.³² The software is available to academic users via an inexpensive license or collaboration agreement and can be purchased, alongside CASTEP, as part of BIOVIA’s Materials Studio package.²⁵

Since the linear-scaling DFT formalism used in ONETEP has been previously described in a number of publications,^{2,29,33} we will outline only the aspects of the formalism which are directly relevant to the MPSM and its implementation in ONETEP.

The central quantity in the linear-scaling DFT formalism used in ONETEP is the density matrix,

$$\rho(\mathbf{r}, \mathbf{r}') = \sum_{\alpha\beta} \varphi_{\alpha}(\mathbf{r}) K^{\alpha\beta} \varphi_{\beta}^*(\mathbf{r}'), \quad (26)$$

which is constructed as a product of strictly localized non-orthogonal orbitals, $\{\varphi_{\alpha}\}$ and a density kernel, \mathbf{K} . The locality of electronic interactions is exploited by truncation of the density kernel, such that it is

zero when the distance between localized orbitals, $R_{\alpha\beta}$, exceeds a cutoff value, r_{cut} , i.e.

$$R_{\alpha\beta} > r_{\text{cut}} \implies K^{\alpha\beta} = 0. \quad (27)$$

The strictly localized orbitals used in ONETEP are called “non-orthogonal generalized Wannier functions” (NGWFs) and are constructed from an underlying basis of psinc (periodic cardinal sine) functions centred on a regular real-space grid.^{34–36} A key feature of the psinc basis is that it is expressible in terms of a finite sum over plane waves—this allows ONETEP to make use of the mathematical machinery of plane-wave pseudopotential DFT.

The electron density, part of the total charge required to solve the NPE (Eq. 1) in the MPSM, is obtained from the density matrix by setting $\mathbf{r} = \mathbf{r}'$:

$$n_{\text{elec}}(\mathbf{r}) = \sum_{\alpha\beta} \varphi_{\alpha}(\mathbf{r}) K^{\alpha\beta} \varphi_{\beta}^*(\mathbf{r}). \quad (28)$$

The use of strictly localized orbitals in ONETEP ensures that the electronic charge cannot exist outside localization regions of the NGWFs—this contrasts with the situation in CASTEP, where electronic charge is not constrained and may exist in any region of the cell.

As in CASTEP (section 2.4), ONETEP uses pseudopotentials to represent nuclei and core electronic states and thus the ionic charge, $n_{\text{ionic}}(\mathbf{r})$, corresponds to a set of nucleus-centred point charges. To overcome the difficulties associated with this representation of the ionic charge when solving the NPE using DL_MG, the implementation of the MPSM in ONETEP uses the smeared ion representation described in section 2.1.

In density-matrix DFT, the ground state energy and electronic density for a given system can be obtained by minimizing the total energy with respect to the density matrix. In ONETEP, this is achieved by two nested self-consistent energy minimizations, i.e.

$$E_{\text{min}} = \min_{\{\varphi_{\alpha}\}} \left(\min_{\mathbf{K}} E[\mathbf{K}, \{\varphi_{\alpha}\}] \right). \quad (29)$$

In the inner self-consistent cycle, the total energy is minimized with respect to the elements of the density kernel, \mathbf{K} , with fixed NGWFs, while in the outer self-consistent cycle, the NGWFs themselves are optimized (see Ref. 34 for further details).

The total energy expression used in ONETEP is similar to Eq. 24, but differs in that the total energy is formally a function of the density matrix, $\rho(\mathbf{r}, \mathbf{r}')$ and the terms depending on CASTEP’s one-particle wavefunctions, $\psi_{m,\mathbf{k}}$ now depend on the density kernel and NGWFs, i.e.

$$E_{\text{tot}}[\rho] = E_{\text{kin}}[\mathbf{K}, \{\varphi_{\alpha}\}] + E_{\text{nl}}[\mathbf{K}, \{\varphi_{\alpha}\}] + E_{\text{xc}}[n_{\text{elec}}] + \{E_{\text{Hart}}[n_{\text{elec}}] + E_{\text{locps}}[n_{\text{elec}}] + E_{\text{ion-ion}}\}. \quad (30)$$

The terms grouped in the curly braces represent the electrostatic contributions that are replaced by $E_{\text{es}}[n_{\text{tot}}]$ (Eq. 12) in the MPSM.

3 WP1: DL_MG defect correction

3.1 Key outcomes

- The defect correction code from ONETEP was successfully ported to DL_MG.
- The defect correction code in DL_MG was modified to support generalized periodic and mixed BCs and a general 3-D parallel decomposition of data.
- The halo communication and high-order finite difference components of the defect correction code in DL_MG were optimized, yielding a 1.2–1.5× speed-up for the combined defect correction and multigrid solver component of large scale ONETEP calculations.

- The convergence criteria for the second-order solver and defect correction components of DL_MG were consolidated in a single module providing default values and a routine for setting alternative values.
- New convergence criteria for the defect correction based on the norms of the potential and defect were established.

3.2 Limitations

- Alternative iterate change criteria for the multigrid solver were not implemented.
 - No significant issues with convergence of the multigrid solution were encountered during this project, so it was determined that new criteria were not currently required.

3.3 Implementation details

In the original implementation of the MPSM in ONETEP, the defect correction procedure made use of routines across several Fortran modules. This was ported to DL_MG as two major new modules: (i) `dl_mg_defco`, containing the routine for performing the main defect correction loop, `dl_mg_defco_defect_corr_solver`; and (ii) `dl_mg_defco_fd`, containing routines for performing high-order finite difference operations. The original code in ONETEP made use of many routines which provide supporting functionality, such as performing real-space integration on a grid. Where this functionality was not available in DL_MG, replacement routines were implemented—these were placed in a further new module, `dl_mg_defco_utils`.

The defect correction and high-order finite differences code ported from ONETEP was initially restricted to open BCs and a 1-D parallel decomposition of the grid (i.e. with real-space grid data divided along one direction and each MPI rank holding a “slab”, as described in Ref. 29). The ported code was modified to support the full range of BCs and the more general 3-D parallel decomposition of the grid used by the second-order solver component of DL_MG. This involved the creation of a new derived type, `fd_t`, in analogy to the existing `mg_t` type, to hold data required to perform high-order finite differences with a 3-D parallel data decomposition. An instance of this type is held on each MPI rank and is initialized with data describing the halo exchanges necessary to perform finite difference operations over the entire grid.

In addition to the modifications described above, the ported code was also optimized. To improve cache utilization, the loops computing the higher order derivative over the grid were restructured. The halo exchange subroutine needed for calculation of the derivative was also rewritten in order to take advantage of the information provided in newly created `fd_t` data structure and to reduce the latency in the case of halos distributed over several MPI ranks by using non-blocking MPI communications.

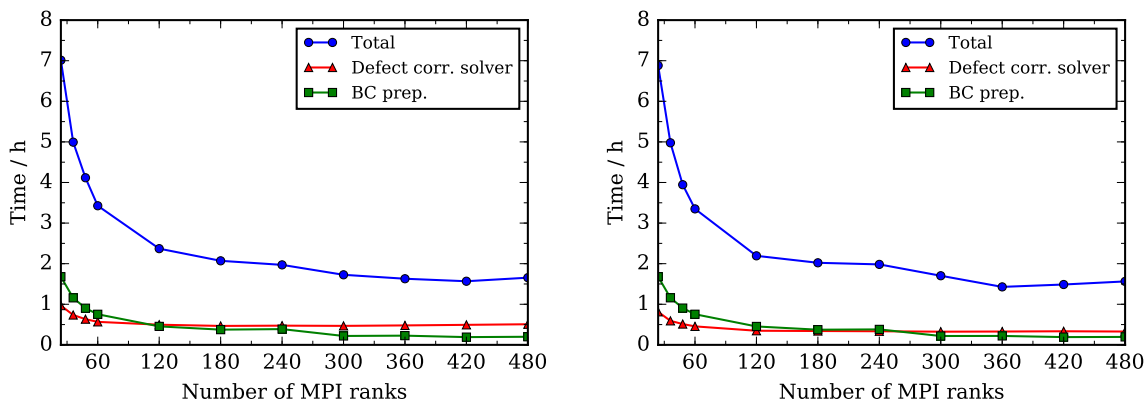
The high-order defect correction procedure (section 2.3) is an iterative procedure—to determine when this procedure is complete, some convergence criterion is required. The original implementation in ONETEP used the norm of the error, $|\phi^{(i)} - \phi^{(i+1)}|$, to determine when convergence was achieved. In the implementation within DL_MG a new criterion based on the norm of the defect, $|r_d^{(i+1)}|$, was added. Support for absolute and relative convergence criteria was also implemented. The high-order defect correction is now stopped when the following conditions are met:

$$|\phi^{(i)} - \phi^{(i+1)}| < \max(\tau_\phi^{\text{abs}}, \tau_\phi^{\text{rel}} |\phi^{(i)}|) \quad (31)$$

$$|r_d^{(i+1)}| < \max(\tau_{r_d}^{\text{abs}}, \tau_{r_d}^{\text{rel}} |r_d^{(0)}|) \quad (32)$$

where the most-recently updated potential and defect are $\phi^{(i+1)}$ and $r_d^{(i+1)}$, the initial (uncorrected) defect is $r_d^{(0)}$, and where the absolute, τ^{abs} , and relative, τ^{rel} , convergence thresholds are user configurable parameters.

In addition to the work described above, a number of other changes were made to DL_MG during the course of this work package, notably:



(a) ONETEP defect correction

(b) DL_MG defect correction

Figure 1: Execution time for single point energy calculations of T4 lysozyme bound to catechol with increasing numbers of MPI processes (in vacuum with open BCs). The total time (blue, circles), time taken by the multigrid solver and defect correction procedure (red, triangles) and time taken to evaluate Dirichlet BCs (green, squares) are shown for the defect correction implemented within (a) ONETEP and (b) DL_MG. Further calculation details are provided in section 3.4.

Simplification of the interface to the library

The need to call a separate initialization routine for nonlinear variants of the Poisson equation was eliminated and an overloaded interface to `dl_mg_solver` was created for all variants of the Poisson equation.

Improvements to convergence parameter control

The convergence criteria governing the behaviour of the multigrid solver and defect correction were collected in the new `dl_mg_convergence_params` module and a routine was created for setting the parameters using values provided as arguments to `dl_mg_solver` or preset defaults.

3.4 Results

To investigate the performance and parallel scaling of the defect correction implemented in DL_MG, single-point energy calculations were performed on the T4 lysozyme protein bound to catechol (2615 atoms)[†] using ONETEP / DL_MG. The calculations were performed in vacuum and solvent, using the calculation settings described in appendix A, with a $129.5 a_0 \times 129.5 a_0 \times 129.5 a_0$ grid and $0.5 a_0$ coarse grid point spacing, yielding a kinetic energy cutoff of 826.827 eV. The fine grid used in ONETEP was $518 \times 518 \times 518$ points, truncated to $513 \times 513 \times 513$ for multigrid operations. The vacuum and solvent calculations were performed independently (i.e. not part of an autosolvation calculation), with solvent calculations initialized with the converged density kernel and NGWFs from the vacuum calculations. In both cases, DL_MG was used to compute the electrostatic potential, using the smeared ion representation (Eqs. 10 to 13) and coarse graining of the boundary conditions (Eq. 9).

The relative performance of the previous implementation of the defect correction in ONETEP and the new implementation in DL_MG was compared for calculations performed on ARCHER with varying numbers of MPI processes, as shown in Figs. 1 and 2. The timing data presented in these figures is for development versions of ONETEP and DL_MG, compiled using the settings described in appendix B. All jobs used 6 MPI processes per node, 3 MPI processes per NUMA region and 4 OpenMP threads per MPI process.

[†]The structure of the T4 lysozyme/catechol complex was provided by J. Dziedzic, and is the same structure used in solvation calculations presented in Ref. 6.

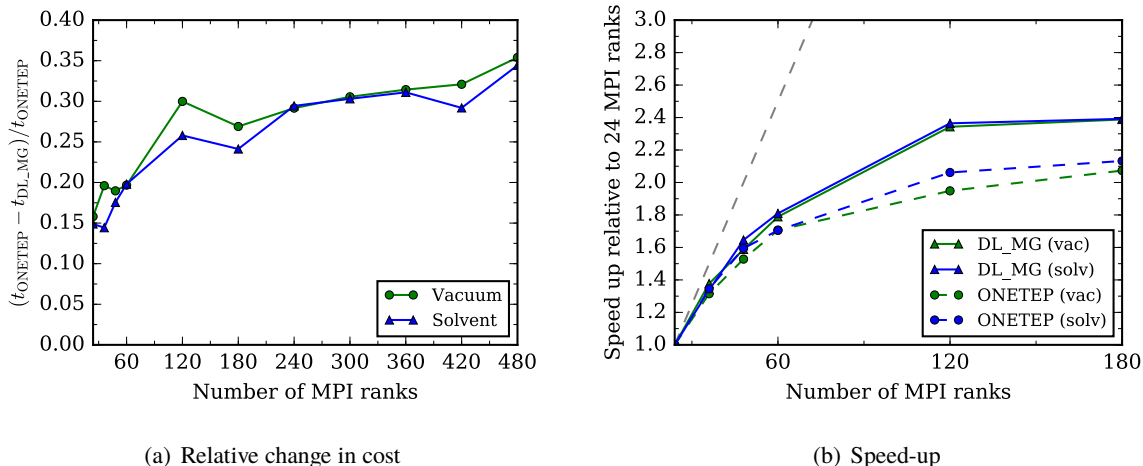


Figure 2: Comparisons of the computational performance of the defect correction implemented in ONETEP and DL_MG in vacuum (green) and solvent (blue) measured in single point energy calculations of T4 lysozyme bound to catechol. Plot (a) shows the relative change in total time spent in the defect correction and solver routines where the defect correction was performed in DL_MG (t_{DL_MG}) compared to the previous implementation in ONETEP (t_{ONETEP}). Plot (b) shows the speed-up relative to a calculation performed with 24 MPI ranks for the portion of the calculation spent in the defect correction and solver routines when the defect correction was performed in ONETEP (dashed, circles) and DL_MG (solid, triangles). The ideal speed-up for increasing numbers of MPI processes is shown as a grey dashed line. Further calculation details are provided in section 3.4

The execution times plotted in Fig. 1 for the vacuum calculations show that for more than 120 MPI ranks (120 MPI \times 4 OpenMP = 480 total cores), the execution time plateaus. This plateau is observed in the total execution time as well as the time spent in the defect correction and solver routines. The reduction in speed-up for larger core counts is a consequence of the 1-D parallel data decomposition used in ONETEP—as the number of MPI ranks increases the width of the 1-D slabs decreases. For 120 MPI ranks, the fine grid slabs along the z -direction are already on average four grid points wide, and this decreases to one grid point at 360 MPI ranks. Such narrow slabs are particularly problematic for the high-order finite differences used in the defect correction, leading to the need to exchange halos which extend over more than one MPI rank. The scaling up to 120 MPI ranks is respectable, with an overall (for the entire ONETEP calculation) $\sim 3\times$ speed-up relative to 24 MPI ranks achieved for both implementations of the defect correction (compared to an ideal $5\times$ speed-up).

It is clear that the 2615 atom T4 lysozyme complex used in these benchmarks is too small to take full advantage of the larger numbers of MPI ranks considered in Fig. 1. It should be noted, however, that ONETEP is capable of performing calculations on systems of tens of thousands of atoms and that the scaling to higher core counts would likely be more favourable in larger systems (see the results reported in Ref. 30, for example).

While Fig. 1 shows that the execution time spent within solver and defect correction routines plateaus for more than 120 MPI ranks regardless of which implementation of the defect correction is used, Fig. 2(a) demonstrates that the implementation in DL_MG offers a substantial performance improvement over the previous implementation at all core counts. For 24 MPI processes, the new implementation of the defect correction (within DL_MG) takes $\sim 15\%$ (vacuum and solvent) less time than the previous implementation, and this saving increases to $\sim 30\%$ (vacuum) and $\sim 25\%$ (solvent) with 120 MPI ranks. This improvement is also observed in the parallel speed-up shown in Fig. 2(b). With 120 MPI ranks, where the parallel performance starts to plateau, the speed-up for DL_MG’s defect correction (and solver routines) is $2.3\times$ (vacuum) and $2.4\times$ (solvent), compared to the defect correction implemented in

ONETEP which achieves $1.9\times$ (vacuum) and $2.1\times$ (solvent). Interestingly, despite the plateau observed in overall execution time for calculations using more than 120 MPI ranks for this test case, the defect correction in DL_MG continues to yield increasing cost savings over the defect correction in ONETEP as the number of MPI ranks is increased. This is likely to be a consequence of the introduction of non-blocking MPI communications for halo exchange in the DL_MG implementation.

4 WP2: ONETEP solvent model extensions

4.1 Key outcomes

- A framework for flexibly performing calculations with open, periodic and mixed BCs was implemented in ONETEP.
- The capability to compute the electrostatic potential in open, periodic and mixed BCs using the multigrid solver was added to ONETEP.
- The fully periodic Hartree energy computed using DL_MG rapidly converges towards the Hartree energy obtained by solving the homogeneous Poisson equation in reciprocal space as the order of finite differences is increased—for a 448 atom graphene sheet, the difference is $\sim 10^{-2}$ kcal mol $^{-1}$ with 12th-order finite differences.
- Smearing ion corrections to the electrostatic energy (Eq. 12) and potential (Eq. 13) in fully periodic BCs were implemented.
- Fully periodic BC calculations in vacuum and implicit solvent are now possible using ONETEP.
- The consistency of treatment of periodic BCs was demonstrated in calculations with (i) small molecules located in different positions in the simulation cell and (ii) graphene sheets with different unit cell sizes.

4.2 Limitations

- Support for mixed BC calculations was not implemented as mixed BC versions of the local pseudopotential, ion-ion and smeared ion correction terms were required.
 - None of these terms were implemented in CASTEP, while ONETEP lacked mixed BC smeared ion correction terms.
 - Derivation and implementation of these additional terms was not feasible within the timescale of the eCSE project.
- Support for fully periodic BCs for MPSM calculations was only implemented in ONETEP.
 - Implementation of the additional terms required for fully periodic BCs in CASTEP was not feasible within the timescale of the eCSE project.

4.3 Implementation details

In order to perform fully periodic BC calculations using the MPSM, each term in the electrostatic energy and potential expressions (Eqs. 12 and 13) must be evaluated with consistent, periodic, BCs. At the start of this project, the second-order multigrid solver component of DL_MG was already capable of evaluating the electrostatic potential in fully periodic and mixed open/periodic BCs. The changes made to DL_MG in WP1 (section 3) also allowed the application of the high-order defect correction under these BCs. Since the fully periodic BC variants of the local pseudopotential and ion-ion terms were also already available in ONETEP, only the smeared ion correction terms in Eqs. 12 and 13 required

implementation. The remaining electrostatic potential, local pseudopotential and ion-ion terms simply needed to be substituted with their periodic BC variants.

The fully open BC expressions for the smeared-ion-corrected electrostatic energy and potential are presented in appendix A of Ref. 7. In that case, the correction to the local pseudopotential, $V_{\text{si}}(\mathbf{r})$, is expressed as a sum over the individual potentials due to each smeared Gaussian charge distribution representing an ionic core, i.e.

$$V_{\text{si}}(\mathbf{r}) = \sum_I^{N_{\text{ions}}} V_I(\mathbf{r}) \quad (33)$$

where $V_I(\mathbf{r})$ is the analytic form for the potential of the smeared Gaussian of ionic core I (see Ref. 7, Eq. A2). To obtain the correction in periodic BCs, a different approach was used, leveraging the availability of full simulation cell FFTs in ONETEP to solve the homogeneous Poisson equation in reciprocal space for the total charge due to the smeared ionic charges. In reciprocal space, the fully periodic potential has the form

$$\tilde{V}_{\text{si}}(\mathbf{G}) = 4\pi \frac{\tilde{n}_{\text{si}}(\mathbf{G})}{|\mathbf{G}|^2} \quad (34)$$

where $\tilde{n}_{\text{si}}(\mathbf{G})$ is the Fourier transform of the smeared ion charge density, $n_{\text{si}}(\mathbf{r})$ (Eq. 10). Fourier transforming Eq. 34 yields a periodic real-space potential for the total smeared ionic charge, $V_{\text{si}}(\mathbf{r})$, which can be used in the periodic variant of Eq. 13.

Having obtained a periodic potential due to the smeared ions, it is simple to evaluate the smeared ion-smeared ion interaction energy by integration over the simulation cell in real-space:

$$E_{\text{si-si}} = \frac{1}{2} \int d\mathbf{r} n_{\text{si}}(\mathbf{r}) V_{\text{si}}(\mathbf{r}). \quad (35)$$

A similar integral can be used to obtain the smeared ion-electron interaction energy, $E_{\text{si-elec}}$ (Eq. 13), though in practice the energy of the interaction of the electron density with the smeared-ion-corrected local pseudopotential is calculated, i.e.

$$E_{\text{locps}}^{\text{corr}}[n_{\text{elec}}] = E_{\text{locps}}[n_{\text{elec}}] - E_{\text{si-elec}}[n_{\text{elec}}] = \int d\mathbf{r} n_{\text{elec}}(\mathbf{r}) V_{\text{locps}}^{\text{corr}}(\mathbf{r}) \quad (36)$$

where $V_{\text{locps}}^{\text{corr}}(\mathbf{r}) = V_{\text{locps}}(\mathbf{r}) - V_{\text{si}}(\mathbf{r})$.

As described in appendix A of Ref. 7, the open BC smeared ion-smeared ion interaction energy in ONETEP is evaluated as separate self-interaction and non-self-interaction contributions:

$$E_{\text{si-si}} = E_{\text{si-si}}^{\text{self}} + E_{\text{si-si}}^{\text{non-self}}. \quad (37)$$

In open BCs, the contributions are separated simply by splitting the summation of contributions from the interaction of pairs of smeared ionic cores into diagonal and non-diagonal contributions. The use of a single periodic potential (Eq. 34) to evaluate the smeared ion-smeared ion interaction (Eq. 35) precludes such a simple separation of self and non-self terms in the periodic BC implementation of the MPSM. To maintain the separation of the self and non-self terms in periodic BCs, the non-self-interaction energy is calculated by subtracting the self-interaction energy, which has the same form in open and periodic BCs (the first term in Eq. A11 of Ref. 7), from the total smeared ion-smeared ion interaction energy (Eq. 35), i.e.

$$E_{\text{si-si}}^{\text{non-self}} = E_{\text{si-si}} - E_{\text{si-si}}^{\text{self}}. \quad (38)$$

Most of the modifications necessary to implement the periodic BC MPSM in ONETEP were made to the `is_smeared_ions` module. The routine which generates the smeared ion charge density, `smeared_ion_generate_density`, was modified to support periodic BCs, using a minimum-image convention to “wrap around” smeared Gaussian charges at cell boundaries. New routines were created for evaluating the smeared ion non-self-interaction energy term (Eq. 38) and potential due to the smeared ion charge density (Eq. 34) in fully periodic BCs. Additionally, the routine for evaluating the correction

to the local pseudopotential (Eq. 36), `smear_ion_apply_vloc_corr`, was modified to use the periodic smeared ion potential when required.

It was also necessary to modify the code associated with determining the dimensions for grids used in ONETEP and DL_MG. In the original fully open BC version of the code, the dimensions of the grid used by DL_MG were determined by rounding down the dimensions of ONETEP's fine grid to the nearest multiple of an even number, m , and adding one ($m = 2^n$ with $n = 3$ by default in ONETEP). This ensured that DL_MG's grid size constraints were satisfied (open BCs require an odd number of grid points) and ensured the solver employed a sufficient number of multigrid levels (see DL_MG's documentation for details²²). In this case, only the charge density in the truncated fine grid is used by DL_MG to compute the electrostatic potential—this is not an issue in open BCs with strictly localized orbitals, since the simulation cell can be set up to ensure no charge exists outside of the truncated region.

For periodic BCs, the grid used by DL_MG must have the same dimensions as the fine grid used elsewhere in ONETEP in order that the electrostatic potential output by DL_MG has the correct periodicity. Consequently, ONETEP's fine grid cannot be truncated as in the case of open BCs. To overcome this issue, the routine used to determine the size of the fine grid used throughout ONETEP, `cell_grid_distribute`, was modified to ensure that the number of grid points along each direction of the fine grid was rounded up to an even multiple of m when periodic BCs are requested (DL_MG requires an even number of grid points along periodic directions). An unfortunate consequence of increasing the size of the fine grid in this way is that additional work must be done by ONETEP to interpolate and filter between the fine grid and a slightly smaller "double grid", which is used by ONETEP for some operations (such as evaluation of the electronic charge density from the NGWFs and density kernel). This is normally avoided by making the fine grid and double grid identical in size, but when the fine grid is modified to satisfy DL_MG's grid constraints, this is no longer possible.

In the original implementation of the MPSM in ONETEP, fully open BCs would be automatically used in all implicit solvent calculations. These could also be manually activated in non-implicit-solvent calculations for the ion-ion, Hartree and local pseudopotential terms by assigning Boolean values to the keywords `openbc_ion_ion`, `openbc_hartree` and `openbc_pspot` in the input file. In order to allow users to flexibly control the BCs used in MPSM (and in vacuum calculations), a new set of input keywords were added to ONETEP: `multigrid_bc`, `ion_ion_bc`, `pspot_bc` and `smear_ion_bc`. These allow the user to specify the BCs used by the multigrid solver, and in computing the ion-ion interaction, local pseudopotential and smeared ion interaction terms along each simulation cell dimension as a three-character string. For each direction, "P" denotes periodic, "O" denotes open and "Z" denotes zero BCs.[‡] These new keywords were implemented in a manner which preserved the previous behaviour of ONETEP in cases where the BCs are not explicitly defined. In practice, since mixed periodic/open BCs were not implemented in ONETEP during this project, only the BCs corresponding to the specifications "OOO", "ZZZ" and "PPP" are currently supported.

The code for evaluating the non-electrostatic contribution to the total energy (Eq. 8) also required modification since the calculation of the surface area of the cavity requires the gradient of the electronic density (see Eq. 11 of Ref. 10). In the original implementation of the MPSM, this gradient was evaluated using ONETEP's `finite_differences` module, which only supports fully open BCs, leading to the use of different finite difference stencils at cell boundaries. In open BCs, this is not an issue, since the solute should be located far from the cell boundaries. To compute the gradient in periodic BCs, where the solute may be close to the cell boundaries, an alternative approach was adopted, using ONETEP's existing FFT capabilities to apply the gradient operator to the electronic density in reciprocal space. Since there was no existing routine for applying the gradient operator in reciprocal space to a function on the simulation cell grid, this was implemented as a general-purpose subroutine in the `services` module.

Table 1: Free energies of solvation in kcal mol⁻¹ for toluene (C₆H₅CH₃), deprotonated benzoic acid (C₆H₅COO⁻) and protonated aniline (C₆H₅NH₃⁺) in H₂O, computed using the MPSM in ONETEP with fully open BCs (OBC) and fully periodic BCs (PBC). The computational details of these calculations are described in section 4.4.

Molecule	Charge	OBC ^a	PBC ^a	PBC ^b
Toluene	0	1.666	1.664	1.664
Benzoic acid	-1	-56.556	-34.231	-34.232
Aniline	1	-72.386	-49.952	-49.954

^a Molecule located at the centre of the simulation cell.

^b Molecule located at the origin of the simulation cell.

4.4 Results

The free energies of solvation presented in Table 1 were computed using the MPSM in ONETEP with the solvent model parameters described in appendix A.1. In all cases, a fixed dielectric cavity was used, constructed using the converged ground state electronic density of the solute computed in vacuum. Calculations were performed in fully open BCs, with approximate coarse-grained Dirichlet BCs (Eq. 9), and fully periodic BCs. The three molecules studied were: toluene (C₆H₅CH₃), deprotonated benzoic acid (C₆H₅COO⁻) and protonated aniline (C₆H₅NH₃⁺). The geometries used in the calculations were gas-phase optimized geometries from the Minnesota solvation database (2009 version).^{37,38}

All calculations used a 20 Å × 20 Å × 20 Å simulation cell and norm-conserving pseudopotentials (see appendix A.2). The number of NGWFs and size of the localization region for each atom type in the calculations was as described in appendix A.3. The calculations were performed using a kinetic energy cutoff of 800 eV. For the open BC calculations, a 150 × 150 × 150 fine grid was used, truncated to 145 × 145 × 145 grid points for use in DL_MG. In periodic BCs, the fine grid scale was slightly increased to satisfy DL_MG’s grid constraints, yielding a 152 × 152 × 152 fine grid (used in both ONETEP and DL_MG).

To test the consistency of the treatment of periodic BCs, the free energy of solvation was computed with the solutes placed at the centre of the simulation cell (as in the open BC calculations) and at the origin of the simulation cell. Under periodic BCs, shifting the position of the molecule in the cell should not meaningfully affect the computed solvation energy. As can be seen in Table 1, the positioning of the molecule in the cell has no substantial impact on the computed periodic BC solvation energies. This is strong evidence that the energy and potential components associated with the solvation model have been correctly implemented for fully periodic BCs.

The comparison of open BC and periodic BC solvation energies in Table 1 reveals an interesting effect for the charged solutes. While the open and periodic BC solvation energies for the neutral solute, toluene (C₆H₅CH₃), differ by ~ 10⁻³ kcal mol⁻¹, this difference is ~ 10 kcal mol⁻¹ for the charged solutes, deprotonated benzoic acid (C₆H₅COO⁻) and protonated aniline (C₆H₅NH₃⁺). It is possible that this large difference in open and periodic BC solvation energies for the charged solutes is a physical effect, arising because the solvation of an isolated charged molecule and a periodic array of charged molecules are different processes. However, it is also possible that the discrepancy is due to additional effects that occur only for charged solutes and which are not currently treated for in the periodic version of the MPSM. As noted in Ref. 39, special care is required when dealing with the electrostatics of solvated systems in periodic BCs. Further investigation of this issue is necessary before the MPSM can be safely applied to periodic BC calculations with charged solutes.

Table 2 contains free energies of solvation computed for neutral graphene sheets under periodic BCs. As for Table 1, the solvation energies were computed using a fixed dielectric cavity constructed using the ground state electronic density obtained calculation in vacuum, and with the calculation settings

[‡]“Zero” BCs are open BCs where the potential is assumed to be zero at the simulation cell boundaries, and only apply to the multigrid solver.

Table 2: Free energies of solvation in kcal mol⁻¹ for graphene sheets with fully periodic BCs. The computational details of these calculations are described in section 4.4.

Material	N_{atoms}	Total	per C atom
Graphene	448	58.096	0.129 68
Graphene	224	29.073	0.129 79

(solvation model, pseudopotentials and NGWFs) described in appendix A. A 448 atom graphene sheet, automatically generated based on a bond length of 1.43 Å, was used, with a 224 atom sheet created by cutting along one of the periodic directions.[§]

The 448 atom sheet calculation was performed in a cell with dimensions 34.31 Å × 34.67 Å × 31.75 Å, with the sheet periodic in the xy -plane. A large z -dimension was used to reduce unwanted interactions between periodic images of the sheet. The 224 atom sheet simulation cell was identical to the 448 atom sheet cell, but halved along the x -direction. In both cases, the kinetic energy cutoff was set as 800 eV, yielding fine grid sizes of 256 × 264 × 240 and 136 × 264 × 240 grid points for the 448 atom and 224 atom sheets, respectively (used in ONETEP and DL_MG).

Since graphene is naturally periodic in two directions, it can be used to test that the treatment of periodic BCs is consistent by comparing solvation energies obtained for different periodic unit cells. In Table 2, we compare per-C-atom solvation energies computed for the 448 and 224 atom sheets. As expected, the per-C-atom solvation energy is effectively independent of the size of the periodic unit cell, differing by $\sim 10^{-4}$ kcal mol⁻¹ for the two sheets.

Despite the currently unexplained discrepancy between open and periodic BC solvation energies for charged solutes, we can be confident that periodic BCs are consistently treated *within* the existing model. This is evidenced by the excellent agreement between periodic BC solvation energies obtained for molecular solutes located at different positions in the simulation cell (Table 1) and per-atom solvation energies for different periodic unit cells (Table 2).

The numerical accuracy of the calculation of the electrostatic energy and potential in periodic BCs using DL_MG is further confirmed by the results presented in Fig. 3. This plot demonstrates the agreement between solutions to the homogeneous Poisson equation obtained using DL_MG’s defect-corrected multigrid approach and the reciprocal space solution usually used in ONETEP. For the 448 atom graphene sheet described above, the Hartree energy was calculated using DL_MG with increasing finite difference order. These calculations were performed in vacuum and without smeared ions, corresponding to solving the homogeneous Poisson equation (Eq. 1 with $\varepsilon = 1$ and $n(\mathbf{r}) = n_{\text{elec}}(\mathbf{r})$). Solving the homogeneous Poisson equation for the electron density allows direct comparison with the reciprocal space approach which is usually employed by ONETEP in periodic BCs. The absolute error plotted in Fig. 3 is with respect to the Hartree energy obtained using this reciprocal space approach.

All calculations presented in Fig. 3 were performed self-consistently, starting with an initial guess density kernel and NGWFs. The same 34.31 Å × 34.67 Å × 31.75 Å cell was used with a kinetic energy cutoff of 800 eV and a 256 × 264 × 240 fine grid. Although the reciprocal space approach is not subject to DL_MG’s grid size constraints, this fine grid was used in all calculations in order to avoid introducing discrepancies from the use of different grid sizes.

As the order of discretization of \hat{A}_d (see Eq. 19) used in the defect correction is increased, the Hartree energy computed using DL_MG rapidly converges towards the energy computed using the reciprocal space approach. With second-order finite differences (i.e. no defect correction), the error is 485 kcal mol⁻¹, decreasing to $\sim 10^{-2}$ kcal mol⁻¹ for 12th-order finite differences.

[§]The structure of the 448 atom graphene sheet was originally generated by L. G. Verga.

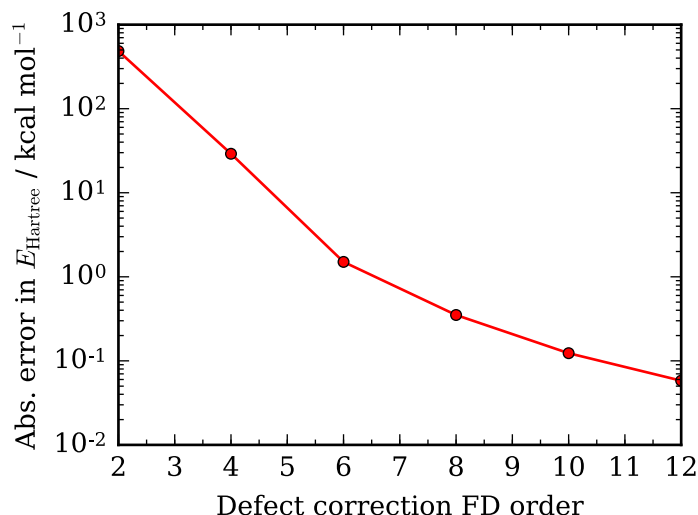


Figure 3: The absolute error in the Hartree component of the total energy of the 448 atom graphene sheet computed using DL_MG with increasing order of finite differences in the defect correction procedure (d in \hat{A}_d from Eq. 19). The error is with respect to the Hartree energy computed by solving the Poisson equation in reciprocal space. All calculations were performed in vacuum with periodic BCs. Further computational details are provided in section 4.4.

5 WP3: CASTEP solvation

5.1 Key outcomes

- The complete implicit solvation model was successfully implemented in CASTEP, enabling fully open BC implicit solvent calculations with a fixed dielectric cavity.
- A facility for “autosolvation” was implemented, providing a simple means of computing free energies of solvation in CASTEP.
- Fully open BC vacuum calculations were also made possible in CASTEP using components of the solvation model.
- Solvation energies computed using the MPSM in ONETEP and CASTEP demonstrate excellent numerical agreement—for the small molecules tested (neutral and charged) the free energies of solvation computed by ONETEP and CASTEP differed by $\sim 0.1 \text{ kcal mol}^{-1}$ or less.

5.2 Limitations

- Only the fixed cavity version of the solvent model was implemented in CASTEP.
 - The fixed cavity model is a first step towards the self-consistent cavity model.
 - For the computation of solvation energies, the use of a fixed cavity incurs very little additional error.⁶
 - Self-consistent cavity calculations suffer from numerical instability due to additional dielectric-dependent terms in the electrostatic potential.
- The multigrid solver and computation of Dirichlet BCs currently run on a single MPI process.

- The multigrid solver uses contiguous 3-D data distributed in a Cartesian MPI topology while CASTEP stores real-space data in a non-contiguous representation.
- This necessitated gathering and reordering of real-space data prior to performing multigrid operations.
- The initial implementation was performed using shared-memory (OpenMP) parallelism for multigrid operations.
- Work to support contiguous distribution of real-space data in CASTEP suitable for fully MPI-parallelized multigrid operations is currently being undertaken as part of a PhD project supervised by P. J. Hasnip.
- Only energetic terms of the solvation model were implemented.
 - Force terms related to the fully open BC local pseudopotential and smeared ions were not implemented.
 - The implementation of accurate solvation energy calculation using the model was prioritized.
- Support for ionic solutions was not implemented.
 - The addition of an ionic solution significantly increases the complexity of energy expressions and requires the implementation of steric potentials.
 - The implementation of accurate solvation energy calculation in pure solvent was prioritized.

5.3 Implementation details

To implement the full MPSM in CASTEP, a large number of new modules and routines were created, consisting of a combination of new code and code ported from ONETEP. The integration of the new functionality into the CASTEP codebase also necessitated significant modifications to existing code. The following is a summary of the most significant of these changes and additions:

CASTEP/DL_MG interface

A new module, `multigrid_dlmg`, which provides wrappers for calling DL_MG library routines was created. The module deals with initialization/deinitialization of the DL_MG library and provides a simplified interface to the solver routines via the `multigrid_dlmg_calculate_hartree` routine. The `multigrid_dlmg_calculate_hartree` takes CASTEP data structures as arguments and converts these to a DL_MG-compatible representation. Dirichlet BCs are calculated by an internal module routine, based on a similar routine ported from ONETEP, but modified for use in CASTEP. The `multigrid_dlmg` module is the only part of CASTEP which explicitly makes use of DL_MG routines and variables. To allow CASTEP to be compiled without DL_MG, a “stub” module was created which replicates the interface of `multigrid_dlmg` to satisfy other module dependencies without the need for DL_MG itself.

Open BC local pseudopotential

A new module, `openbc_locps`, was created, providing routines for the evaluation of the local pseudopotential in open BCs. The code in this module was largely derived from a similar `openbc_locps` module in ONETEP, but modified to make use of the data structures used in CASTEP. In particular, the `openbc_locps_calculate_potential` routine was designed to replicate the interface of the CASTEP’s existing routine for calculating the local pseudopotential in periodic BCs, `locps_calculate_potential`. For more details on the theoretical method used to evaluate the open BC local pseudopotential, see appendix B of Ref. 40.

Open BC ion-ion interaction

The `ion_ion` module was created to provide a general interface for calculating the ion-ion interaction energy in fully open and mixed periodic/open BCs. Currently, the `ion_ion_calculate_openbc_energy`

routine can only compute the ion-ion interaction energy with fully open BCs, though the routine is designed to be extensible so that the mixed BC case can be implemented at a later date. Under fully open BCs, the ion-ion interaction energy is calculated by summation over the interaction energies of pairs of ionic cores (Eq. A10 of Ref. 7).

Smeared ion representation

The new `smeared_ions` module provides routines associated with the generation of the smeared ion charge density (Eq. 10) and calculation of the corrections to the electrostatic energy (Eq. 12) and potential (Eq. 13). The correction to the potential is applied directly to the local pseudopotential, as in ONETEP (Eq. 36). The module also provides a routine for computing the electrostatic potential and energy due to the combined electronic and smeared ion density, `smeared_ions_calculate_es_potential`, using `DL_MG` to solve the NPE. The code in this module is mostly derived from ONETEP's `is_smeared_ions` module, adapted to make use of CASTEP's data structures and interface conventions.

Non-uniform dielectric permittivity

The `dielectric_permittivity` module provides the `dielec_perm` derived type, which represents the non-uniform dielectric permittivity, $\epsilon(\mathbf{r})$, which features in the MPSM (Eq. 6). The module features routines for initialization and deallocation of the dielectric permittivity object based on pre-defined parameterizations, which can be flexibly defined using extended types. A routine for calculating the surface area of the solute cavity in the dielectric medium, $S[n_{\text{elec}}]$ (Eq. 8), is also provided. The module was designed to be sufficiently general to allow alternative functional forms for $\epsilon(\mathbf{r})$ (other than Eq. 6) to be implemented in future.

Non-electrostatic solvation contribution

The `implicit_solvent` module was created to contain routines that specifically deal with implicit solvation (many of the other new modules have functionality which can be applied in other contexts). One of these routines computes the non-electrostatic contribution to the solvation energy (Eq. 8) using the dielectric cavity surface area from `dielectric_permittivity_cavity_surface_area`.

Addition of autosolvation task

A new autosolvation task was implemented, in which the free energy of solvation (Eq. 7) is computed automatically by performing the vacuum and solvent calculations in a single CASTEP run. This can be activated by setting the task keyword to "AUTOSOLVATION" in the CASTEP input file. To achieve this, a new routine was added to the main `castep.f90` file, `castep_autosolv_vac`, which performs the initial vacuum calculation and copies the relevant calculation results (converged charge density, wavefunction and energy) to an instance of CASTEP's `model_state` derived type, which is then used to perform the calculation in the presence of solvent. The results of the autosolvation calculation are reported at the end of the CASTEP run by calling `implicit_solvent_autosolvation_report`, a routine from the `implicit_solvent` module.

Modification of model data structure

CASTEP's `model_state` derived type is a container for information which has been about computed about a system. Properties associated with implicit solvation and the solution of the NPE (Eq. 1), such as the non-uniform dielectric permittivity and non-electrostatic energy contributions (Eq. 8) were added to `model_state`. These additions were particularly important in the implementation of the autosolvation task, which involves copying relevant data between instances of `model_state` corresponding to vacuum and solvent calculations.

Modification of total energy expressions

The expressions used to evaluate the total energy of the system in the electronic energy minimization procedure were modified to incorporate the smeared ion energy correction terms (Eq. 12) and the non-electrostatic energy contribution (Eq. 8), when required. This involved the modification of several routines in the electronic module featuring both the energy expressions described

in section 2.4. Additionally, code was added to the routine for outputting the energy components, `electronic_write_energies`, to output the additional energy components involved in an MPSM calculation.

Integration of DL_MG into CASTEP build system

Compilation of DL_MG was incorporated into the CASTEP build system, so that the library could be distributed with CASTEP and easily compiled into the binary. The method used to integrate other libraries distributed with CASTEP (e.g. `spglib`⁴¹) into the build system was adopted for DL_MG. The value of the DL_MG variable in the main CASTEP Makefile determines if and how DL_MG is linked to CASTEP: “none” for compilation without DL_MG (using the stub module); “compile” to compile and link the version of DL_MG distributed with CASTEP. The capability to use an alternative version of DL_MG by setting DL_MG to “system” is planned, but not yet implemented.

New MPI communicator for multigrid operations

DL_MG requires an MPI communicator with a Cartesian topology which defines the relationships between local data held by each rank. To allow the creation of communicators with appropriate Cartesian topologies, the `comms_cart_create` routine was added to the `comms` module. Although the current implementation of the DL_MG interface only supports a single MPI process, a flexible framework for defining the parallel strategy to be used in multigrid operations was implemented in the `comms` module. This was based on CASTEP’s existing parallel framework, where parallelism can be distributed based on **k**-points, bands and/or **G**-vectors. The multigrid MPI communicator is formed as a subset of the **G**-vector communicator, which is natural since the real-space charge density is distributed across members of the **G**-vector group. This framework should form the foundation for future work to allow DL_MG to make use of multiple MPI processes.

As mentioned in section 5.2, the multigrid operations and the calculation of Dirichlet BCs are currently only executed on a single MPI rank. The remainder of the CASTEP calculation can use an arbitrary number of ranks, but the calculation of the electrostatic potential using DL_MG (via the `multigrid_dlmng` module) uses only one of these MPI ranks. The fundamental reason for this is that CASTEP distributes the real-space charge density among MPI ranks in a non-contiguous form—columns of data along the *z*-direction of the simulation cell are distributed among MPI ranks in round-robin fashion to balance the load across the **G**-vector communicator.

DL_MG requires that the real-space charge density local to each MPI rank is contiguous along the *x*-, *y*-, and *z*-directions. The Dirichlet BC calculation code ported from ONETEP also requires this contiguous distribution of data. Consequently, it was necessary to reorder the real-space charge data held by MPI ranks in the **G**-vector communicator before passing this to the BC calculation routines and DL_MG. It was similarly necessary to reorder the contiguous electrostatic potential output by DL_MG to match CASTEP’s non-contiguous distribution of the real-space charge density.

In the current implementation, the real-space charge density is simultaneously gathered from MPI ranks in the **G**-vector communicator and reordered (to a contiguous distribution). The result is held on the single MPI rank in the multigrid communicator, which performs the calculation of Dirichlet BCs and solves the NPE using DL_MG. The resulting electrostatic potential is then simultaneously scattered back to the **G**-vector communicator and reordered (back to the original non-contiguous distribution). This reordering is not necessary in ONETEP because real-space quantities are always contiguous along each simulation cell dimension.

A more complex scheme for gathering, reordering and scattering could have been implemented to allow the calculation of Dirichlet BCs and multigrid operations using multiple MPI ranks. Given the additional communication overhead this would engender, it is unclear whether this would yield a practical performance benefit. The decision to use the simpler single MPI rank implementation was taken because it is likely that it will be possible to have CASTEP use contiguous real-space data natively in the near future—this is being investigated as part of a PhD project supervised by P. J. Hasnip.

The representation of real-space quantities used by DL_MG in CASTEP differs from the representation in ONETEP in another significant way. As mentioned in section 4.3, in open BCs the fine

grid on which ONETEP represents the real-space charge density is truncated to satisfy DL_MG's grid constraints. This poses no difficulties in open BCs—provided that the system is positioned to avoid the NGWF localization regions from extending beyond the truncated grid, the strict localization of the NGWFs (see section 2.5) ensures that no charge is removed from the cell during truncation. In CASTEP, the one-electron wavefunctions (Eq. 23) extend over the entire simulation cell and the corresponding real-space charge density (Eq. 22) is not guaranteed to exist within the truncated grid. This is a particular issue at the start of the SCF procedure, where the wavefunctions used to construct the charge density are generally initialized with random coefficients. This also causes a problem when calculating Dirichlet BCs as the potential at the cell boundaries can become very large when $|\mathbf{r} - \mathbf{R}_i| \rightarrow 0$ in Eq. 9.

To ensure that all the real-space charge density is included in the grid passed to the multigrid solver and avoid numerical issues when computing Dirichlet BCs, CASTEP uses a padded grid for multigrid operations. The size of this grid along each dimension is given by

$$n_i^{\text{pad}} = \text{round_up}(n_i + 2p, m) + 1, \quad (39)$$

where the $\text{round_up}(a, b)$ function rounds a up to the nearest multiple of b , n_i is the number of grid points along direction i , p is an integer determining the degree of padding and m is an even integer (which has the default value $m = 2^n$ with $n = 3$, as described in section 4.3). CASTEP's real-space charge density is copied from the original $n_x \times n_y \times n_z$ grid to the $n_x^{\text{pad}} \times n_y^{\text{pad}} \times n_z^{\text{pad}}$ padded grid such that it is always separated from the padded cell boundaries by a padding region. The non-uniform dielectric permittivity is also placed on this padded grid, with $\varepsilon(\mathbf{r}) = \varepsilon_\infty$ in the padding regions. DL_MG's solver routines are passed these padded input quantities and return an electrostatic potential on the padded grid. The potential in the region corresponding to CASTEP's unpadded real-space grid is then extracted and scattered to the \mathbf{G} -vector group, as described earlier.

A further complication associated with the implementation of the MPSM in CASTEP is the use of two different total energy expressions: Eqs. 24 and 25. Particular care must be taken when applying the smeared ion energy corrections. In Eq. 24, the modified electrostatic energy expression of Eq. 12 can simply be substituted for the electrostatic terms in curly braces, i.e.[¶]

$$E_{\text{tot}}[n_{\text{elec}}] = E_{\text{kin}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{nl}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{xc}}[n_{\text{elec}}] + \left\{ E_{\text{NPE}}^{\text{tot}}[n_{\text{elec}}] + E_{\text{locps}}^{\text{corr}}[n_{\text{elec}}] + (E_{\text{ion-ion}} - E_{\text{si-si}}) \right\}, \quad (40)$$

where $E_{\text{locps}}^{\text{corr}}[n_{\text{elec}}]$ is the smeared-ion-corrected local pseudopotential (as in Eq. 36), and

$$\begin{aligned} E_{\text{NPE}}^{\text{tot}}[n_{\text{elec}}] &= E_{\text{NPE}}^{\text{elec}}[n_{\text{elec}}] + E_{\text{NPE}}^{\text{si}}[n_{\text{elec}}] \\ &= \frac{1}{2} \int d\mathbf{r} n_{\text{elec}}(\mathbf{r}) \phi_{\text{NPE}}(\mathbf{r}) + \frac{1}{2} \int d\mathbf{r} n_{\text{si}}(\mathbf{r}) \phi_{\text{NPE}}(\mathbf{r}). \end{aligned} \quad (41)$$

In Eq. 25, the smeared ion corrections must be applied differently, since $\phi_{\text{NPE}}(\mathbf{r})$ appears in the one-electron Hamiltonian which is used to evaluate the band structure energy, $E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}]$. Consequently, the interaction of the electronic density with the electrostatic potential due to the total charge density (smeared ion and electronic), ϕ_{NPE} , is included in $E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}]$, but is double-counted. To obtain a total energy expression with the correct overall electrostatic energy, we must correct for the double-counting of the electron- ϕ_{NPE} interaction energy in $E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}]$ while adding the missing smeared ion- ϕ_{NPE} interaction energy, i.e.

$$\begin{aligned} E_{\text{tot}}[n_{\text{elec}}] &= E_{\text{BS}}[\{\psi_{m,\mathbf{k}}\}] + E_{\text{xc}}^{\text{corr}}[n_{\text{elec}}] \\ &\quad + E_{\text{NPE}}^{\text{si}}[n_{\text{elec}}] - E_{\text{NPE}}^{\text{elec}}[n_{\text{elec}}] + (E_{\text{ion-ion}} - E_{\text{si-si}}), \end{aligned} \quad (42)$$

where $E_{\text{NPE}}^{\text{si}}[n_{\text{elec}}]$ and $E_{\text{NPE}}^{\text{elec}}[n_{\text{elec}}]$ are as defined in Eq. 41.

[¶]Note that the total energy in Eq. 40 is given as a functional of only the electronic density, n_{elec} , despite the formal dependence of the expression on the total (electronic and smeared ion) charge density—this is to emphasise that the smeared ion charge density, n_{si} , is constant during the SCF procedure, while n_{elec} is varied.

Table 3: Free energies of solvation in kcal mol⁻¹ for toluene (C₆H₅CH₃), deprotonated benzoic acid (C₆H₅COO⁻) and protonated aniline (C₆H₅NH₃⁺) in H₂O, computed using the MPSM in open BCs using ONETEP and CASTEP. The CASTEP calculations were performed with three different energy minimization schemes: density mixing (DM), conjugate gradient energy minimization with fixed occupancies (ALLBANDS) and ensemble DFT (EDFT). The computational details of these calculations are described in section 5.4.

Molecule	Charge	CASTEP			ONETEP
		ALLBANDS	DM	EDFT	
Toluene	0	1.695	1.696	1.695	1.666
Benzoic acid	-1	-56.401	-56.401	-56.401	-56.556
Aniline	1	-72.666	-72.664	-72.666	-72.386

The electronic module was modified to use Eqs. 40 and 42 for calculations using the smeared ion representation in vacuum or solvent. In the presence of solvent, the non-electrostatic solvation energy (Eq. 8) is also added to the expression—for the fixed cavity model, this is the same for both total energy expressions. If the self-consistent cavity model is implemented in future, the appearance of a non-electrostatic solvation contribution in the band structure energy will need to be accounted for.

5.4 Results

To validate the accuracy of the implementation of the MPSM in CASTEP, free energies of solvation computed in CASTEP were compared to energies obtained using the existing implementation of the MPSM in ONETEP. The results of these calculations are summarized in Table 3. These were computed in open BCs, using the solvent model parameters described in appendix A.1 and a fixed dielectric cavity, constructed using the converged ground state electronic density of each solute in vacuum. The three molecules studied were: toluene (C₆H₅CH₃), deprotonated benzoic acid (C₆H₅COO⁻) and protonated aniline (C₆H₅NH₃⁺), as in section 4.4. The geometries used in the calculations were gas-phase optimized geometries from the Minnesota solvation database (2009 version).^{37,38}

For all calculations presented in Table 3, a 20 Å × 20 Å × 20 Å simulation cell and norm-conserving pseudopotentials (see appendix A.2) were used. The number of NGWFs and size of the localization region for each atom type in the ONETEP calculations was as described in appendix A.3. The ONETEP calculations were performed using a kinetic energy cutoff of 800 eV, producing a 150 × 150 × 150 fine grid, which was truncated to 145 × 145 × 145 grid points for use in DL_MG. A kinetic energy cutoff of 528.792 eV was used in CASTEP (with a fine grid scale of 2.0) in order to obtain an identical 150 × 150 × 150 fine grid^{||}—this was then padded to 161 × 161 × 161 grid points for use in DL_MG.

The results presented in the Table 3 demonstrate excellent agreement between the implementations of the MPSM in ONETEP and CASTEP. For toluene, the free energies of solvation computed using ONETEP and CASTEP differ by only ~ 10⁻² kcal mol⁻¹, while for the two charged solutes the difference is ~ 10⁻¹ kcal mol⁻¹. This difference is an order of magnitude smaller than the ~ 1 kcal mol⁻¹ target originally proposed as a success metric for WP3. The results also demonstrate that the MPSM has been implemented consistently for CASTEP’s three energy minimization schemes, with maximum absolute differences between the solvation energies computed using the three methods of ~ 10⁻³ kcal mol⁻¹ or less for each solute.

^{||}The relationship between energy cutoff and grid point spacing is different in CASTEP and ONETEP, so different cutoff energies are required to obtain identical grid sizes.

6 Concluding remarks

This project has delivered significant new functionality for each of the software packages involved. The key goals of adding an implicit solvent model to CASTEP, generalizing the BCs available in ONETEP's implicit solvent model and implementing the high-order defect correction in DL_MG have all been achieved, with some limitations, as outlined in sections 3 to 5.

It is anticipated that this new functionality will be of great utility to the users of each software package. The work done for WP3 (section 5) will enable users to perform calculations in the presence of implicit solvent using CASTEP for the first time, while the work done for WP2 (section 4) will enable ONETEP users to treat the solvation of periodic materials. The improvements to DL_MG described in section 3 make the solver capable of producing solutions to the NPE which are sufficiently accurate to be used directly in electronic structure calculations. This should reduce the development work required to use DL_MG in electronic structure contexts by removing the need to implement the high-order defect correction or a similar correction scheme.

This project has also engendered a number of additional benefits, beyond the originally proposed objectives. For example, the open BC electrostatic energy and potential terms implemented in CASTEP as part of the MPSM (section 5) give CASTEP the capability to perform open BC calculations both with and without implicit solvent. The ability to perform open BC calculations in vacuum is valuable in itself, and was not previously available in CASTEP. Another additional benefit arising from this work was the simplification and extension of DL_MG's API (section 3), which removed some of the complexity of initializing the library, while allowing for more flexible specification of convergence criteria.

While the original objectives of this project have been broadly achieved, the outcomes are subject to some limitations and caveats. Additional work will be necessary to fully realise all the goals outlined in the project proposal. In particular:

Extension of the implementation of the MPSM in ONETEP to support mixed open/periodic BCs

During this project, support for fully periodic BCs was added to the MPSM in ONETEP, but mixed BCs were not. It emerged during the project that treatment of mixed BCs for all the electrostatic terms involved in the model requires careful theoretical consideration. While DL_MG is able to produce an electrostatic potential in mixed BCs, consistent application of mixed BCs requires that these BCs are also reflected in the ion-ion interaction, smeared ion correction and local pseudopotential terms. The implementation of a mixed BC version of the MPSM in ONETEP will require the derivation of mixed BC expressions for these terms.

Implementation of MPI parallelism for multigrid operations in CASTEP

The current implementation of the MPSM in CASTEP performs multigrid operations and the calculation of Dirichlet BCs on a single MPI rank. As described in section 5.3, this was a consequence of the non-contiguous distribution of real-space data among MPI ranks in CASTEP. To take advantage of MPI parallelization in DL_MG while avoiding the overhead of reordering the real-space data, CASTEP will need to provide real-space quantities in a contiguous form suitable for use by DL_MG. Work towards this goal is currently underway, as part of a PhD project supervised by P. J. Hasnip.

Saline solvent implementation and extension

It was originally envisaged that the extensions to the solvent modelling capabilities in ONETEP and implementation of the model in CASTEP would include support for saline solvents (using DL_MG to solve the Poisson-Boltzmann equation, Eq. 3). This was not possible within the timescale of the project, since support for the simpler non-saline case, where the NPE must be solved (Eq. 1), was prioritized. To add support for saline solutions in CASTEP will require the implementation of suitable steric potentials and additional energy components arising from the presence of ions in solution, as described in Ref. 13. Extending the saline solvent model in ONETEP to support periodic and mixed BCs will require that the types of steric potential currently implemented are generated subject to these BCs.

Atomic forces implementation and extension

In order to perform geometry optimizations and *ab initio* molecular dynamics calculations in the presence of solvent, the forces acting on atoms must be computed. At the start of this project, ONETEP was able to compute forces in open BC MPSM calculations using non-saline solvent, but not saline solutions. It was envisaged that support for forces in saline solutions would be added to ONETEP and that the new implementation of the MPSM in CASTEP would include support for atomic forces calculation. In the event, computation of accurate solvation energies was prioritized over forces, meaning that these tasks still remain to be tackled. In both cases, appropriate expressions for atomic forces in the presence of solvent will need to be implemented. In particular, care must be taken to ensure that the forces account for the dependence of smeared ionic core charges and (for saline solutions) steric potentials on atomic positions.

Throughout the project, care was taken to ensure that features were implemented in a way which supports future extensions, such as those outlined above. For example, although the current implementation of the MPSM in CASTEP only uses a single MPI process in DL_MG and to evaluate Dirichlet BCs, a flexible framework for defining the parallel strategy to be used in these operations was implemented, in anticipation of future work to support multiple MPI processes (see section 5.3). Similarly, the new input keywords added to ONETEP to allow specification of boundary conditions were developed specifically to allow specification of open, periodic and mixed BCs, despite the current implementation of the MPSM in ONETEP supporting only fully open and fully periodic BCs (see section 4.3).

As indicated in section 1.1, accurately modelling solvent effects is essential if electronic structure simulations are to be a useful tool in the design and development of new technologies. Towards this aim, this eCSE project has seen the implementation of a powerful implicit solvent model in CASTEP and the extension of the model in ONETEP. In addition, the underlying multigrid solver used in both codes has been made significantly more capable. It is hoped that these developments will support researchers currently studying electronic structure in solution and facilitate the application of electronic structure methods to new technologically relevant problems.

Acknowledgements

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

A Calculation details

A.1 Implicit solvent model settings

A common set of parameters were used for implicit solvent calculations using the MPSM presented in this report.

Physical parameters describing the solvent were set to values suitable for H₂O (these are the default values used when performing solvation calculations in ONETEP):

- Solvent bulk permittivity, $\epsilon_{\infty} = 78.54$
- Solvent surface tension, $\gamma = 4.7624 \times 10^{-5} E_h a_0^{-2}$

The non-electrostatic component of the solvation energy (Eq. 8) was computed using an effective surface tension, γ_{eff} , scaled by a constant to account for dispersion-repulsion ($\gamma_{\text{eff}} = 0.281075\gamma$, as recommended in Ref. 6).

The “low-beta” parameter set recommended in the ONETEP documentation (available at the ONETEP website³²) was used to construct the dielectric function (Eq. 6):

- Solvent-solute interface width parameter, $\beta = 1.3$

- Density at centre of solvent-solute interface, $n_0 = 3.5 \times 10^{-4} a_0^{-3}$

Some other significant parameters controlling the behaviour of the model were based on recommendations in Ref. 7:

- Charge coarse-graining block size for BC calculation (Eq. 9), $5 \times 5 \times 5$ grid points
- Smeared ion width parameter (Eq. 11), $\sigma = 0.8 a_0$

A.2 Pseudopotentials

All calculations presented in this report used norm-conserving pseudopotentials from the Rappe-Bennett pseudopotential library⁴² (GGA-optimized). The calculations on the T4 lysozyme complex presented in section 3.4 also made use of a sulfur pseudopotential from a suite of pseudopotentials generated by K. Refson to supplement the Rappe-Bennett library. ONETEP- and CASTEP-compatible versions of the Rappe-Bennett library and K. Refson’s supplementary set of pseudopotentials are available to download from the CASTEP project page on CCPForge.⁴³

A.3 NGWFs

The ONETEP calculations presented in this report used $7.0 a_0$ (WP1, section 3) or $8.0 a_0$ (WP2 and WP3, sections 4 and 5) localization sphere radii for all atom types. The number of NGWFs used to represent the valence electrons for each atom type was as follows: H, 1; C, 4; N, 4; O, 4; S, 4.

A.4 Exchange-correlation functional

All ONETEP and CASTEP calculations presented in this report were performed using the PBE exchange-correlation functional.^{44,45}

B ARCHER development environment

For calculations measuring the performance of ONETEP / DL_MG on ARCHER (section 3.4), the ONETEP executable was compiled using gfortran 5.1 with the “-Ofast” optimization flag, and linked to FFTW.^{46,47} The binary was compiled based on development versions of the ONETEP and DL_MG sources. The following key modules were loaded during compilation:

- PrgEnv-gnu/5.2.82
- fftw/3.3.4.10
- cray-mpich/7.2.6

References

- [1] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. I. J. Probert, K. Refson, and M. C. Payne, *Z. Kristallogr.* **220**, 567 (2005).
- [2] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, *J. Chem. Phys.* **122**, 084119 (2005).
- [3] S. Miertuš, E. Scrocco, and J. Tomasi, *Chem. Phys.* **55**, 117 (1981).
- [4] A. Klamt and G. Schüürmann, *J. Chem. Soc., Perkin Trans. 2*, 799 (1993).
- [5] J. Tomasi, B. Mennucci, and R. Cammi, *Chem. Rev.* **105**, 2999 (2005).

- [6] J. Dziedzic, H. H. Helal, C.-K. Skylaris, A. A. Mostofi, and M. C. Payne, *EPL* **95**, 43001 (2011).
- [7] J. Dziedzic, S. J. Fox, T. Fox, C. S. Tautermann, and C.-K. Skylaris, *Int. J. Quantum Chem.* **113**, 771 (2013).
- [8] J.-L. Fattebert and F. Gygi, *J. Comput. Chem.* **23**, 662 (2002).
- [9] J.-L. Fattebert and F. Gygi, *Int. J. Quantum Chem.* **93**, 139 (2003).
- [10] D. A. Scherlis, J.-L. Fattebert, F. Gygi, M. Cococcioni, and N. Marzari, *J. Chem. Phys.* **124**, 074103 (2006).
- [11] O. Andreussi, I. Dabo, and N. Marzari, *The Journal of Chemical Physics* **136**, 064102 (2012).
- [12] L. Anton, J. Dziedzic, C.-K. Skylaris, and M. I. J. Probert, Multigrid solver module for ONETEP, CASTEP and other codes, Technical report, dCSE, 2013.
- [13] C.-K. Skylaris, J. Dziedzic, and L. Anton, A pinch of salt in ONETEP's solvent model, Technical report, eCSE, 2014.
- [14] S. J. Fox, J. Dziedzic, T. Fox, C. S. Tautermann, and C.-K. Skylaris, *Proteins* **82**, 3335 (2014).
- [15] M. J. S. Phipps, T. Fox, C. S. Tautermann, and C.-K. Skylaris, *J. Chem. Theory Comput.* (2017).
- [16] T. J. Zuehlsdorff, P. D. Haynes, M. C. Payne, and N. D. M. Hine, *J. Chem. Phys.* **146**, 124504 (2017).
- [17] M. P. Merrick, K. A. Iyer, and T. L. Beck, *J. Phys. Chem.* **99**, 12478 (1995).
- [18] A. Brandt, *Math. Comp.* **31**, 333 (1977).
- [19] E. L. Briggs, D. J. Sullivan, and J. Bernholc, *Phys. Rev. B* **54**, 14362 (1996).
- [20] J. Bernholc, M. Hodak, and W. Lu, *J. Phys.: Condens. Matter* **20**, 294205 (2008).
- [21] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, 2001.
- [22] L. Anton, J. Womack, and J. Dziedzic, DL_MG multigrid solver, 2017, <https://ccpforge.cse.rl.ac.uk/gf/project/dl-mg/>.
- [23] S. Schaffer, *Math. Comp.* **43**, 89 (1984).
- [24] CASTEP version 17.2, <http://www.castep.org/>.
- [25] Dassault Systèmes BIOVIA, *BIOVIA Materials Studio 2017 R2*, San Diego: Dassault Systèmes, 2017, <http://3dsbiovia.com/>.
- [26] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, *Rev. Mod. Phys.* **64**, 1045 (1992).
- [27] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, 2004.
- [28] M. D. Segall, P. J. D. Lindan, M. J. Probert, C. J. Pickard, P. J. Hasnip, S. J. Clark, and M. C. Payne, *J. Phys.: Condens. Matter* **14**, 2717 (2002).
- [29] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, *Phys. Status Solidi B* **243**, 973 (2006).
- [30] K. A. Wilkinson, N. D. M. Hine, and C.-K. Skylaris, *J. Chem. Theory Comput.* **10**, 4782 (2014).

- [31] N. D. M. Hine, P. D. Haynes, A. A. Mostofi, C. K. Skylaris, and M. C. Payne, *Comput. Phys. Commun.* **180**, 1041 (2009).
- [32] ONETEP version 4.4, <http://www.onetep.org/>.
- [33] P. D. Haynes, C.-K. Skylaris, A. A. Mostofi, and M. C. Payne, *J. Phys.: Condens. Matter* **20**, 294207 (2008).
- [34] C.-K. Skylaris, A. A. Mostofi, P. D. Haynes, O. Diéguez, and M. C. Payne, *Phys. Rev. B* **66**, 035119 (2002).
- [35] A. A. Mostofi, C.-K. Skylaris, P. D. Haynes, and M. C. Payne, *Comput. Phys. Commun.* **147**, 788 (2002).
- [36] A. A. Mostofi, P. D. Haynes, C.-K. Skylaris, and M. C. Payne, *J. Chem. Phys.* **119**, 8842 (2003).
- [37] A. V. Marenich, C. P. Kelly, J. D. Thompson, G. D. Hawkins, C. C. Chambers, D. J. Giesen, P. Winget, C. J. Cramer, and D. G. Truhlar, *Minnesota Solvation Database – version 2009*, University of Minnesota, Minneapolis, 2009.
- [38] A. V. Marenich, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. B* **113**, 6378 (2009).
- [39] O. Andreussi and N. Marzari, *Phys. Rev. B* **90**, 245101 (2014).
- [40] N. D. M. Hine, J. Dziedzic, P. D. Haynes, and C.-K. Skylaris, *J. Chem. Phys.* **135**, 204103 (2011).
- [41] Spglib, a library for finding and handling crystal symmetries, <https://atztogo.github.io/spglib/>.
- [42] J. W. Bennett, *Physics Procedia* **34**, 14 (2012).
- [43] CASTEP project page on CCPForge, <https://ccpforge.cse.rl.ac.uk/gf/project/castep/>.
- [44] J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **77**, 3865 (1996).
- [45] J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **78**, 1396 (1997).
- [46] FFTW version 3.3.4.10, <http://www.fftw.org/>.
- [47] M. Frigo and S. G. Johnson, *Proc. IEEE* **93**, 216 (2005).