Multigrid solver module for ONETEP, CASTEP and other codes

Lucian Anton[1], Jacek Dziedzic[2&3], Chris-Kriton Skylaris[2] and Matt Probert[4]

[1]*Scientific Computing Department, STFC, Daresbury Laboratory WA4 4AD, UK*

[2]*School of Chemistry, University of Southampton SO17 1BJ, UK*

[3]*Faculty of Applied Physics and Mathematics, Gdansk University of Technology, Poland*

[4]*Department of Physics, University of York YO10 5DD, UK*

November 3, 2013

Abstract

Chemical reactions, drug-protein interactions, surfaces and materials-related interactions are examples of technologically important processes that happen in the presence of solvents. Solvent effects can be included within *ab initio* quantum chemistry simulations of such processes by implicit solvation models. The most accurate models of this kind solve directly the Poisson Equation (PE) or the Poisson-Boltzmann Equation (PBE) in 3D space, which is computationally very demanding. This report describes the implementation of DL_MG, a hybrid parallel (MPI+OpenMP) geometric multigrid solver for the PE and PBE with continuous charge distributions in a medium with non-uniform dielectric constant. DL_MG was interfaced with the linear-scaling code for quantum-mechanical calculations based on density-functional theory (DFT) ONETEP and the state-of-the-art DFT code CASTEP. DL_MG has enabled large-scale DFT calculations on molecules and materials in the presence of solvent as it can solve PE up to 16 times faster as compared to the previously-used multigrid code on the same number of cores. As a result, the computational cost of including solvent is no longer prohibitive and it has been made comparable to the cost of calculations in vacuum - enabling a host of applications on materials and biomolecules which are briefly outlined in this report. Furthermore, significant gains in performance are provided by the superior parallel scalability of the the new code, up to 1024 cores, due to its hybrid parallelism. We provide a description of the parallelisation strategy used to implement DL_MG as well as detailed performance tests on a number of supercomputers. DL_MG has been developed as a stand-alone library which makes it potentially useful outside ONETEP and CASTEP, for example in other electronic structure codes, in classical force field codes and in engineering codes which use multigrid solvers. While DL_MG has enabled calculations which were not computationally feasible up to now, important parts of its development and its integration within ONETEP and CASTEP were not possible to complete within the duration of this project and have been outlined here as aims for a future continuation of this project.

Contents

1 Introduction

1.1 Project description

The goal of this project was to develop in Fortran 95 a hybrid parallel (MPI+OpenMP) multigrid solver package scalable to at least 1000 cores for typical problems, tuned to the requirements of the ONETEP `http://www.onetep.org/` [1] and CASTEP `http://www.castep.org/` [2] codes: continuous charge distributions, variable permittivity function, Dirichlet, Neumann, periodic and mixed boundary conditions, linear Poisson equation (PE) and non-linear case as in the Poisson-Boltzmann equation (PBE).

1.2 Summary of the work done

The parallel multigrid solver DL_MG was developed according to the following specifications:

- The standard components of the multigrid solver were implemented for the linear and non-linear case, but only for Dirichlet and full periodic boundary conditions. More details are presented in Section 2.2.
- Hybrid parallelism was implemented for general 3D domain partition using MPI topologies and blocked loops for OpenMP threads. Good scaling, up to 1024 cores in hybrid mode, was achieved for the typical PE and PBE problems ONETEP and CASTEP need to solve, see Sections 2.3, A.
- The code was thoroughly tested and integrated in the ONETEP distribution. Basic tests were performed for CASTEP, more work is needed to achieve full integration, see Sec. 4.
- The solver source code was developed as a self-contained library (in its own directory) that can be used easily by any other applications.

1.3 Work left outstanding

The following subtasks named in the work packages were not finished in the allocated time. Justification is provided below for each outstanding item:

- CASTEP integration was done only for the serial version. Towards the end of the project it became apparent that the multiblock charge density grid decomposition used by CASTEP requires a more general interface than the 3D decomposition used by DL_MG. Also, CASTEP needs to parallelise the defect correction procedure before a fully parallel solvation calculation can be performed. However, as the typical grids used by CASTEP are relatively small (smaller than 129^3), practical calculation can be performed in OpenMP mode, or using MPI parallelism with a simple adaptation which gathers the charge density on one MPI rank and subsequently solves the PE or PBE in serial mode.
- The boundary conditions implemented for the solver at this stage only are full Dirichlet and full periodic for the PE and homogeneous Dirichlet for PBE. Mixed boundary conditions and Neumann boundary conditions will be added as the applications will move to calculations which require them.

1.4 General aspects of solvation calculations

Implicit solvation models provide a computationally affordable way to model solvent, by representing it as a dielectric continuum which surrounds and is polarised by the solute and reacts by back-polarising the solute. In first principles electronic structure calculations this is taken into account in the self-consistent process so that the electronic density is deformed accordingly by the presence of the solvent. Furthermore, a number of explicit solvent molecules can be included when necessary (e.g. water molecules forming hydrogen bonds with an exposed protein active

site). A large number of solvation modes have been reported in the literature [3]. The equations relevant to the solvent models in this work are the non-homogeneous Poisson equation:

$$\nabla \cdot (\varepsilon(\mathbf{r})\nabla\phi(\mathbf{r})) = -4\pi\rho_{\text{tot}}(\mathbf{r}), \tag{1}$$

and the Poisson-Boltzmann equation

$$\nabla \cdot (\varepsilon(\mathbf{r})\nabla\phi(\mathbf{r})) = -4\pi\rho_{\text{tot}}(\mathbf{r}) - 4\pi\sum_i q_i c_i \exp\left(-\beta(q_i\phi(\mathbf{r}) + V(\mathbf{r}))\right), \tag{2}$$

where ε is the non-homogeneous dielectric permittivity of the medium (solvent), $\rho_{\text{tot}}(\mathbf{r})$ is the total charge density, $\phi(\mathbf{r})$ is the sought potential generated by this density in the presence of a dielectric, q_i are the charges of the ions that are in solution, each with a bulk concentration of c_i, $V(\mathbf{r})$ is the steric potential, and $\beta = 1/kT$ is the usual Boltzmann factor.

Classical and quantum-based approaches differ in how they describe the charge density of the system, $\rho_{\text{tot}}(\mathbf{r})$. The former express it as a collection of point charges, represented by Dirac delta functions [7], the latter use continuous charge densities, typically represented on a Cartesian grid. This is an important distinction, as it necessitates implementing different numerical approaches in the solvers in order to tackle the corresponding difficulties. The multigrid solver described here is geared towards electronic structure calculations and thus assumes that all three quantities of interest (permittivity $\varepsilon(\mathbf{r})$, charge density $\rho_{\text{tot}}(\mathbf{r})$, and potential $\phi(\mathbf{r})$) are continuous and are represented on a Cartesian grid.

However, even in electronic structure methods, the ionic cores are internally represented as point pseudocharges and careful treatment is needed to correctly describe them in the context of a multigrid solver. The so-called smeared-ion formalism [8] can be used to replace the point pseudocharges with Gaussian ("smeared"), Coulombic representation of the cores. In this formalism, the total charge density is expressed as

$$\rho_{\text{tot}}(\mathbf{r}) = \rho(\mathbf{r}) + \sum_I^N \rho_I(\mathbf{r}), \tag{3}$$

where $\rho(\mathbf{r})$ is the electronic charge density, and $\rho_I(\mathbf{r})$ is a Gaussian density representing core I:

$$\rho_I(\mathbf{r}) = -\frac{Z_I}{\sigma_I^3}\pi^{-\frac{3}{2}} \exp\left(-\frac{|\mathbf{r} - \mathbf{R}_I|^2}{\sigma_I^2}\right), \tag{4}$$

where $\mathbf{R}_I$ is the position of core I, Z_I is its charge, and σ_I is a chosen smearing width in the order of $1\,a_0$. Appropriate correction terms to energy [8] and forces [9] are needed to ensure they remain unchanged. The dielectric permittivity can be treated in different ways, depending on the problem at hand. In solvation calculations it is usually set to a solvent bulk value outside of a suitably defined cavity around the solute and assumes a value close to unity inside the cavity. Slightly larger values $(2-5)$ are occasionally used to model screening effects within the solute. Two main approaches are employed to define the shape of the cavity. In the first one the cavity is constructed as a set of overlapping atom-centred spheres, whose radii are parametrised. In the second approach the permittivity is defined as a function of the electronic density: $\varepsilon = \varepsilon[\rho](\mathbf{r})$. This approach, first proposed by Fattebert and Gygi [10] and later extended by Scherlis *et al.* [9] and Dziedzic *et al.* [11] and Andreussi et al. [13] avoids a discontinuity at the cavity boundary, replacing it with a steep, but continuous change of the permittivity, which follows naturally from the change in electronic density. This is the approach used in ONETEP [11] and CASTEP [12]. Calculations in vacuum use a homogeneous permittivity of 1, whereby equation (1) simplifies to the homogeneous Poisson equation:

$$\nabla^2\phi(\mathbf{r}) = -4\pi\rho_{\text{tot}}(\mathbf{r}). \tag{5}$$

Consistent treatment of boundary conditions between calculations in vacuum and in solvent is crucial for obtaining meaningful solvation energies. Periodic boundary conditions often used

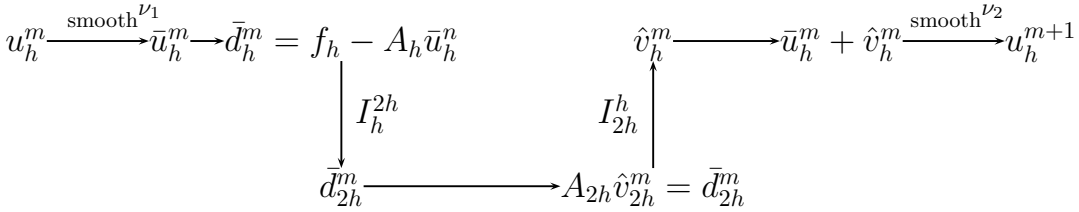

Figure 1: Two-grid iteration for $Au = f$ problem. The smoothing reduces the short wavelength error components of the initial guess u_h^m. After smoothing the defect $\bar{d}_h^m$ can be transferred to a coarser grid without loss of information (restriction); the error found on the coarse grid $\hat{v}_{2h}^m$ is transferred to the initial grid (prolongation). A powerful feature of this approach is that the solution of the problem formulated on the level $2h$ can be obtained recurrently with a further two-grid step.

in electronic structure calculations in vacuum do not offer a good description of the situation in solution due to the dielectric screening of the solvent. Zero boundary conditions can be used [10] in solution, as an approximation to open ("Coulombic") boundary conditions. Calculations in vacuum can then use one of many available approaches for the realisation of open boundary conditions [14] or can be performed under periodic boundary conditions and subsequently corrected [9]. A more elegant and physically sound approach consists in performing all calculations under open boundary conditions [11, 14], which usually requires changes to how electrostatic interactions (core-core and pseudocore-electronic) are handled in the electronic structure code. A further difficulty is the generation of the boundary conditions for the multigrid solver (i.e. the values of $\phi(\mathbf{r})$ on the faces of the simulation cell). An approach suitable for calculations in vacuum and in solution is outlined in Ref. [11] and the effect of the approximations employed is described in Ref. [8].

Solvation energies are sensitive to the accuracy of the solution of the Poisson equation, with second-order discretisation often proving inadequate [8]. Both higher-order methods [9] and a technique called defect-correction [8] have been used with considerable success. Both ONETEP and CASTEP employ the latter. Its main advantage is that a second-order multigrid solver is sufficient, and higher-order accuracy is obtained through an iterative procedure, where subsequent calls to the solver are used to "defect-correct" the initial solution. For more details the Reader is referred to Ref. [8].

2 Multigrid algorithm and implementation details

2.1 Introduction

Multigrid techniques provide optimal, or near-optimal, and highly efficient solvers for a broad set of linear and non-linear equations which are usually derived form several classes of partial differential equations. The optimality of the multigrid methodology results from the combination of iterative solvers with grid transformations and is based on two principles: (i) smoothing iterations: short-wavelength error components can be uniformly attenuated with a broad class of iterative procedures, (ii) grid coarsening: a smooth error can be transferred to a coarser grid without significant information loss. Fig. 1 illustrates the basic two-grid iteration which can be applied recursively in a variety of combinations.

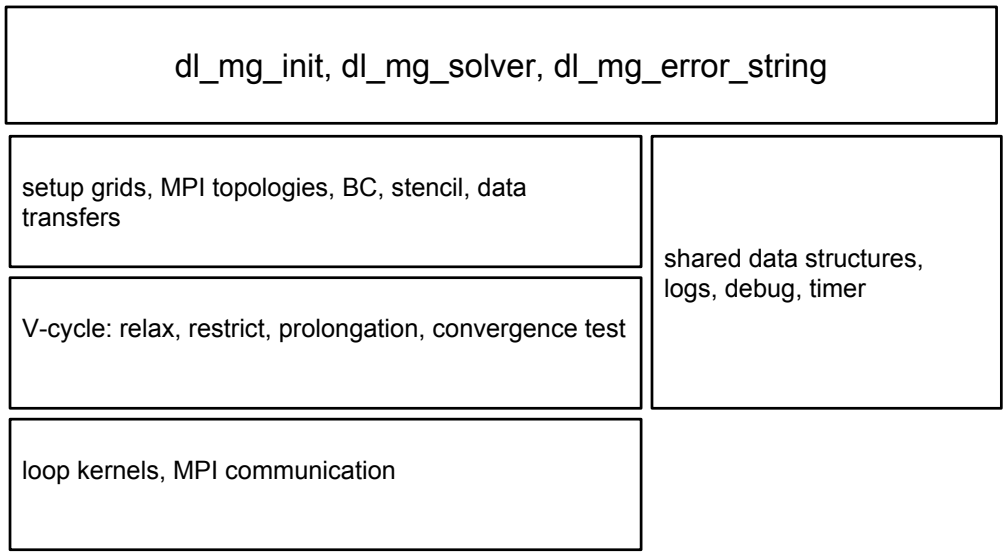

Figure 2: DL_MG code structure. The procedures from the top layer can be accessed by user applications with a `use dl_mg` statement.

Besides adaptivity, generality and high efficiency, another important feature of multigrid for todays computer architectures is the possibility to use algorithms with multiple levels of parallelism, e.g.: the grid over which the solution is sought can be partitioned to MPI tasks which exchange only halos with their topological neighbours, the update operation for smoothing, restriction and prolongation can be parallelised further with OpenMP threads [4].

2.2 DL_MG description

The main components of DL_MG were selected following the standard recommendations for the PE and PBE. The grid stencils are derived by discretisation of the differential operator at each level (geometric approach) with the finite differences method. In order to generate the multigrid levels, the global grid must have sizes given by $[q_1 2^{n_1} + 1, q_2 2^{n_2} + 1, q_3 2^{n_3} + 1]$ with $q_1 \approx q_2 \approx q_3 \leq 20$ for an efficient solution at the coarsest level. Coarsening is achieved by doubling the lattice constant in all dimensions, smoothing uses Gauss-Seidel red-black method, inter-grid transfers are performed with half-weight restriction and bilinear interpolation. The non-linear case (PBE) uses the full approximation scheme (FAS). With the above components one can build a close to optimal and efficient solver for the PE and PBE, provided that the models used for the permittivity and charge density are smooth and without strong anisotropies [4].

The V-cycle was selected for multigrid iterations as generally recommended for parallel computations [4, 5]. MPI parallelism is used for data decomposition; the cuboid global grid is distributed amongst MPI ranks using a 3D topology. As the coarse grids are derived from the fine grids by removing the points with even index coordinates in all directions, no data is exchanged between MPI ranks during inter-grid transfers. The number of MPI ranks can vary across multigrid levels because some ranks could be assigned zero grid points below a certain coarsening level, depending upon the global grid size and MPI topology at the finest level. MPI communication at each level is done in a separate communicator that includes only the active ranks. The solver is provided with the option to "agglomerate" the grid domains at a chosen level and thus to use only one MPI task for the levels below. Domain halos are exchanged with non-blocking send-receives, the edge and corner points of the local domains are transfered between ranks by using ordered communication along axes between nearest neighbours [4].

OpenMP parallelism is implemented using one parallel region that covers the V-cycle loop

and the subroutine which builds the stencil coefficients. The local domain is decomposed in thread blocks with an algorithm that ensures equal work for all threads even in the case of very thin local domains (situation encountered in ONETEP in certain use cases). Block sizes can be tuned also for better cache utilisation, if necessary. First touch policy is used to ensure optimal memory access on NUMA architectures. MPI communication inside OpenMP regions is handled by the master thread (the so-called funnelled mode), mainly for reasons of portability. Data transfers between MPI buffers and halos are done using one thread per local grid side with the help of "single" directives.

The software package is developed in Fortran 95, uses modules and derived data types for information encapsulation and for maintaining a hierarchal structure of the package. The software stack organisation is sketched in Fig. 2.

Besides the solver subroutines, several auxiliary procedures and programs were developed to ensure solver correctness, to assess its performance and to help its integration with user applications:

- In debug mode a detailed print of the residual at each multigrid level and iteration step can be enabled with a preprocessor macro.
- A grid dump subroutine is provided (for any multigrid component), which writes the global grid values in an unique logical layout with the help of MPI IO. The grid dumps can be compared using an auxiliary program which reports statistics of data differences between two runs with different MPI topologies or numbers of OpenMP threads.
- The solver is equipped with a timer module which can collect the time spent in computation per OpenMP threads and MPI communication at every multigrid level. The average, minimum and maximum time values per thread and multigrid level are reported in the log file.
- A test suite was developed that checks solutions obtained with the solver against analytical solutions across various grid sizes, boundary conditions, MPI topologies and numbers of OpenMP threads.
- The build system is based on a generic Makefile, the variables which depend on the particular system setup (such as: compiler, libraries, etc) are specified in a set of include files.

2.3 Testing and performance

DL_MG correctness and performance were thoroughly tested using a three stage procedure.

1. Basic correctness tests on a set of problems with known solutions. A systematic exploration of MPI topologies and numbers of OpenMP threads was done using the testing script. DL_MG average convergence rate (the average ratio of two consecutive residual norms) is close to 0.125 which is in agreement with the values reported in literature and it is independent of grid sizes.
2. Inclusion of DL_MG in the ONETEP test suite. The quantities computed with DL_MG coincided with the reference ones within the prescribed tolerance, i.e. agreement was achieved with the previously used solver.
3. The performance of DL_MG was measured in a solvation calculation performed with ONETEP for a realistic system (T4 lysozyme protein, 2615 atoms), with the system discretised over a grid with sizes $513 \times 513 \times 577$ [8]. Fig. 3 shows that at large core counts DL_MG is almost 16 times faster compared to the the previous solver. Using DL_MG ONETEP can perform this calculation with good scalability up to 128 MPI ranks, see Fig. 4.

An observations is now in order: DL_MG scales fine up to 1024 cores in hybrid mode, see Fig. 6, this will be useful for the next generation of ONETEP protein models that will use 5 to 10 times more atoms than the current models.

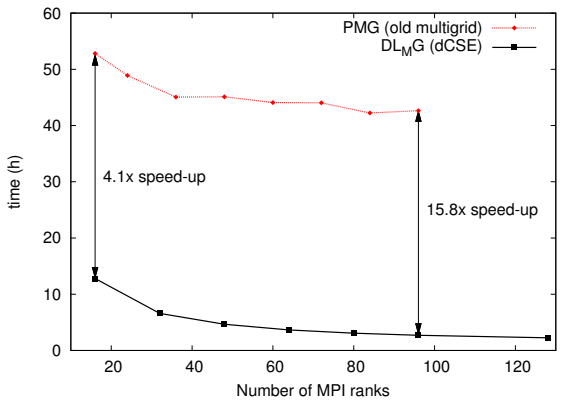

Figure 3: Speed comparison between DL_MG and the multigrid solver previously used in ONETEP. Data was collected on IRIDIS4 [17], the code was compiled with the Intel compiler. This is purely MPI scaling, with OMP to follow on subsequent pages.

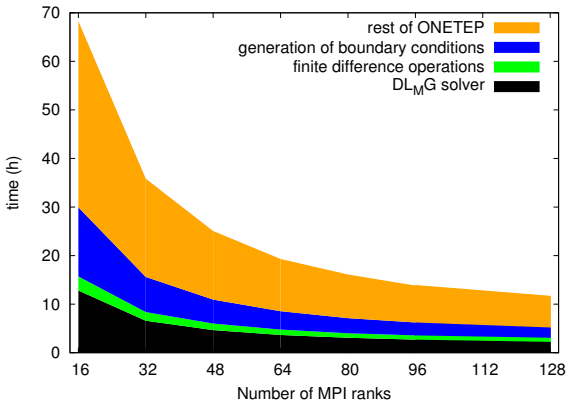

Figure 4: Comparison of time spent in components of ONETEP and in the solver during a solvation calculation for a practical problem (T4 lysozyme protein, 2615 atoms) [8]. Runs done on IRIDIS4 using 1 OpenMP thread.

DL_MG was incorporated in the serial version of CASTEP, the correctness test was passed, the computation efficiency was twice that of the previous multigrid solver. DL_MG non-linear branch for PBE has passed the elementary correctness tests across the various MPI topologies and number of OpenMP threads.

In Sec. A we present more performance data collected on HECToR [16] and IRIDIS4 [17] systems for hybrid parallel mode scaling.

3 Project impact

Since August 2013 the DL_MG code (with its updated versions that were developed until the end of this dCSE project) has been included in the ONETEP academic distribution. For the first time this has enabled practical calculations with the solvent model in ONETEP, which is now similar in computational performance to standard "vacuum" calculations, as opposed to being impractically slow before this dCSE project. This has enabled a number of leading research groups in the UK and internationally who use ONETEP to continue their (long overdue) projects which required the solvent model. A selection of these projects is:

- Simulations of materials for solar cell applications by the group of Dr Gilberto Teobaldi (currently on HECToR and due to continue also on ARCHER) at the University of Liverpool.
- Simulations of the interactions of proteins with carbon-based nanoparticles in order to investigate the toxicity of nanoparticles to biological tissues, by the group of Professor Irene Yarovsky at RMIT University (Australia).
- Drug optimisation applications by the group of Dr Chris-Kriton Skylaris at the University of Southampton in research projects supported by Boehringer Ingelheim (currently on HECToR and due to continue also on ARCHER).
- Simulations of enzyme catalysis by the groups of Professor Mike Payne (Cambridge) and Dr Chris-Kriton Skylaris (Southampton) in collaboration with Professor Adrian Mullholand (Bristol). The calculations are run on local supercomputing resources as well as on HECToR (and are due to continue also on ARCHER).

- Simulations of adsorption and chemical reactions on the surface of metallic nanoparticles by the group of Dr Chris-Kriton Skylaris (Southampton) in research projects supported by Johnson Matthey. The calculations are run on local supercomputing resources as well as on HECToR (and are due to continue also on ARCHER).

For the above applications alone, the usage of HECToR and ARCHER systems is expected to be in the region of 15 million AUs per year.

Over 15 participants from UK and abroad were trained at the use of the solvent model at the ONETEP Masterclass 2013 which was held in Cambridge between 27-30 August 2013.

In addition to the academic version of ONETEP, the DL_MG code and the solvent model will soon be made available in the commercial version of ONETEP which is marketed by Accelrys and used by researchers in industry.

Overall this project has proved extremely useful for the ONETEP developer and user communities. It is a further demonstration of the success of the HECToR dCSE scheme in enabling the re-engineering of complex algorithms in such a way that they can be run efficiently on supercomputers such as HECToR to benefit their users. While not yet applicable in calculations with CASTEP, this project has made significant progress towards this goal by developing most of the necessary groundwork that will be needed in the future completion of the integration of DL_MG with the solvent model within CASTEP.

The usefulness of this work is not limited to solvation models. An equally useful outcome of the availability of this code within ONETEP and CASTEP is that it also enables their standard (vacuum) calculations to be run under arbitrary (e.g. open) boundary conditions in the form of state-of-the-art real space-based electronic structure theory methods. Up to now the standard implementations of ONETEP and CASTEP relied on Fourier Transforms which necessarily impose periodic boundary conditions. The open boundary conditions make the simulations of molecular and nanoparticle systems more realistic by avoiding the supercell approximation.

Furthermore, beyond the realm of electronic structure theory, the DL_MG library (which will be made freely available via a BSD-type license) will find application in fields such as materials simulations with classical force fields (there are already plans for incorporating it into the DL_POLY package), engineering, fluid mechanics and climate modelling as a few possible examples of its wide applicability.

4 Conclusion and future work

We have delivered a software package thoroughly tested and integrated in the main branch of ONETEP. The performance gain offered by DL_MG with respect to previous solver, a factor of almost 16 at large core counts, allows ONETEP users to explore significantly larger or more complex systems that rely on implicit solvation or open boundary conditions, as detailed in the previous section.

In order help a wider user base, DL_MG is structured as a parallel library package using MPI+OpenMP with a general 3D domain decomposition, hence it can be used easily to solve PE and PBE type of equations in other applications, provided that they have specifications similar to ONETEP or CASTEP.

Nevertheless, more work is needed to complete the unfinished subtasks of this project and to support the computational requirements of future ONETEP and CASTEP projects. Henceforth we plan to continue our work in the following directions in order to enhance DL_MG capabilities and performance:

- DL_MG was incorporated only in the serial version of CASTEP, which is used currently for solvation calculations. Because a full parallel calculation with CASTEP depends on the development of parallel defect correction and of an interface to handle multi-block partition of data, we decided to delegate future CASTEP developments to a new project.
- Implement more boundary conditions (mixed periodic and Dirichlet, Neumann).

- Improve parallel scalability and serial performance: As seen in Figs. 7, 9, the MPI communication needs further optimisation; several new ideas which have appeared in literature recently [6] are a promising starting point. We plan to focus on the optimisation of the coarsest level solver and the red-black smoother.
- So far our calculations have only included the **non-homogeneous Poisson equation**. The ability to work with the **non-homogeneous Poisson-Boltzmann equation** will allow our solvation model to include also ionic strengths (i.e. to simulate salt solutions rather than pure water solutions). This is crucial for biomolecular simulations, as most processes (e.g. protein-protein or protein-drug interactions or DNA mutations) happen (in vivo and in vitro) in saline solutions. In order to accomplish this, the non-linear capability of the solver needs to be developed further, we plan to add more boundary conditions and a continuation algorithm that is needed for the PBE with strong nonlinearity.
- Improve user accessibility. DL_MG will be released through CCPForge [18] and we will use the software engineering facilities provided by this server to maintain the source code and documentation.

Acknowledgements

This project was funded under the HECToR Distributed Computational Science and Engineering (CSE) Service operated by NAG Ltd. HECToR – A Research Councils UK High End Computing Service - is the UK's national supercomputing service, managed by EPSRC on behalf of the participating Research Councils. Its mission is to support capability science and engineering in UK academia. The HECToR supercomputers are managed by UoE HPCx Ltd and the CSE Support Service is provided by NAG Ltd. `http://www.hector.ac.uk`.

References

[1] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, *Introducing ONETEP: Linear-scaling density functional simulations on parallel computers*, J. Chem. Phys. 122 084119 (2005).

[2] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. J. Probert, K. Refson, and M. C. Payne, *First principles methods using CASTEP*, Z. Kristallogr. 220 567 (2005).

[3] J. Tomasi, B. Mennucci, R Cammi, *Quantum Mechanical Continuum Solvation Models*, Chem. Rev. 105 2999 (2005).

[4] Ulrich Trottenberg, Cornelius Oosterlee and Anton Schüller, *Multigrid*, Academic Press, 2001.

[5] Edmond Chow, Robert D. Falgout, Jonathan J. Hu, Raymong S. Tuminaro, and Ulrike Meier Yang, "A Survey of Parallelization Techniques for Multigrid Solver" *Parallel Processing For Scientific Computing*, Heroux, Raghavan, and Simon, editors, SIAM, series on Software, Environments, and Tools (2006).

[6] Samuel Williams, Dhiraj D. Kalamkar, Amik Singh, Anand M. Deshpande, Brian Van Straalen, Mikhail Smelyanskiy, Ann Almgren, Pradeep Dubey, John Shalf, and Leonid Oliker, *"Optimization of Geometric Multigrid for Emerging Multi- and Manycore Processors"*, SC12, November 10-16, 2012, Salt Lake City, Utah, USA.

[7] N.A. Baker, *Poisson-Boltzmann Methods for Biomolecular Electrostatics*, Methods in Enzymology 383 (2004).

[8] J. Dziedzic, S.J Fox, T. Fox, C.S. Tautermann and C.-K. Skylaris, *Large-scale DFT calculations in implicit solvent – a case study on the T4 lysozyme L99A/M102Q protein*, Int. J. Quant. Chem. 113 6 (2013).

[9] D.A. Schelis, J.L. Fattebert, F. Gygi, M. Cococcioni and N, Marzari, *A unified electrostatic and cavitation model for first-principles molecular dynamics in solution*, J. Chem. Phys. 124 7 (2006)

[10] J.L. Fattebert and F. Gygi, *Density functional theory for efficient ab initio molecular dynamics simulations in solution*, J. Comp. Chem. 23 6 (2002).

[11] J. Dziedzic, H.H. Helal, C.-K. Skylaris, A.A. Mostofi and M.C. Payne, *Minimal parameter implicit solvent model for ab initio electronic-structure calculations*, Europhys. Lett. 95, 43001 (2011).

[12] H.H. Helal *Including solvent effects in first-principles simulations of biological systems: Development, implementation, and application of an implicit solvent model*, PhD thesis, University of Cambridge (2010).

[13] O. Andreussi, I. Dabo, and N. Marzari, *Revised self-consistent continuum solvation in electronic-structure calculations*, J. Chem. Phys. 136, 064102 (2012).

[14] N.D.M. Hine, J. Dziedzic, P. D Haynes and C.-K. Skylaris, *Electrostatic interactions in finite systems treated with periodic boundary conditions: Application to linear-scaling density functional theory*, J. Chem. Phys. 135, 204103 (2011).

[15] T. L. Beck, *Real-space mesh techniques in density functional theory*, Rev. Mod. Phys. 72, 1041 (2000).

[16] Cray XE-6 system, a compute node has 2 AMD Interlagos processors(32 cores), 32 GB RAM across 4 NUMA domains.

[17] University of Southampton HPC cluster, 16 2.6 GHz Intel Xeon E5-2670 cores per node, 64 GB RAM per node.

[18] `http://ccpforge.cse.rl.ac.uk`.

A Hybrid parallel performance

In this section we present more DL_MG performance and scaling data for hybrid parallelism in ONETEP and also for calculations over a grid partitioned in 3D MPI topologies which is representative for CASTEP.

ONETEP uses 1D MPI domain decomposition of its global grid, hence the number of MPI ranks that can be used for parallelism is limited by the number of grid points in the partitioning direction (denoted by z). Furthermore, the thickness of the local domain cannot be reduced to size 1 in the z direction without significant performance loss because of high-order discretisation needed for certain derivatives in the defect-correction procedure (which is a wrapper around DL_MG). In this context, the OpenMP parallelism inside each MPI domain is one of the options that can be used to increase computation parallelism.

Fig. 5 shows the hybrid scaling performance for ONETEP components used in solvation calculation: DL_MG, boundary condition and defect correction. This set of data was obtained on the IRIDIS4 system [17] using the Intel compiler, with 1, 2 and 4 OpenMP threads, respectively.

For the next benchmark the Poisson problem data were extracted from the ONETEP lysozyme model and solved separately with DL_MG on HECToR. Fig. 6 shows the total solver speed (1/time) for 3 MPI grid partitions to which OpenMP threads are added. The global grid sizes are: $449 \times 545 \times 609$. The solution was attained in 9 steps for a tolerance of $\approx 10^{-8}$, the convergence factor (the ratio of two consecutive residual norms) was close to 0.125. Data show that MPI and OpenMP scaling are very good up to 4 threads for all MPI partitions and acceptable with 8 threads and it stays positive across the whole NUMA node. DL_MG internal timer was used to measure the time spent in the multigrid components (smoother, restriction, prolongation) at each multigrid level for computation and MPI communication sectors. Fig. 7 shows the scaling of computation and communication for the smoother component, which takes the largest share of the DL_MG run time ($\approx 70\%$ amongst multigrid components). Two observations are in order: i) the communication time decreases with the number of threads inside of a NUMA node for a 1D grid partition, ii) the degradation of the total speed scaling in Fig. 6 is correlated with the loss of scaling for the communication sectors seen in Fig. 7. This gives us a hint where to focus our efforts for parallel performance improvements.

In the last test, done on HECToR as well, 3D MPI topologies were used for global grid sizes $129 \times 129 \times 129$ which is representative for CASTEP calculations. The MPI 3D topologies were build with the following rule

$$2^{n_x} \times 2^{n_y} \times 2^{n_z}$$

with n_y, n_y and n_z chosen such that $n_x \leq n_y \leq n_z$, $|n_i - n_j| \leq 1$ $i, j \in \{x, y, z\}$, the number MPI ranks being $2^{n_x+n_y+n_z}$. Figs. 8, 9 show that 3D MPI problems have a good scaling up to 256 cores (MPI ranks or OpenMP threads). Similarly to ONETEP, the loss of scalability is correlated with the crossover between the scaling computation and flat communication.

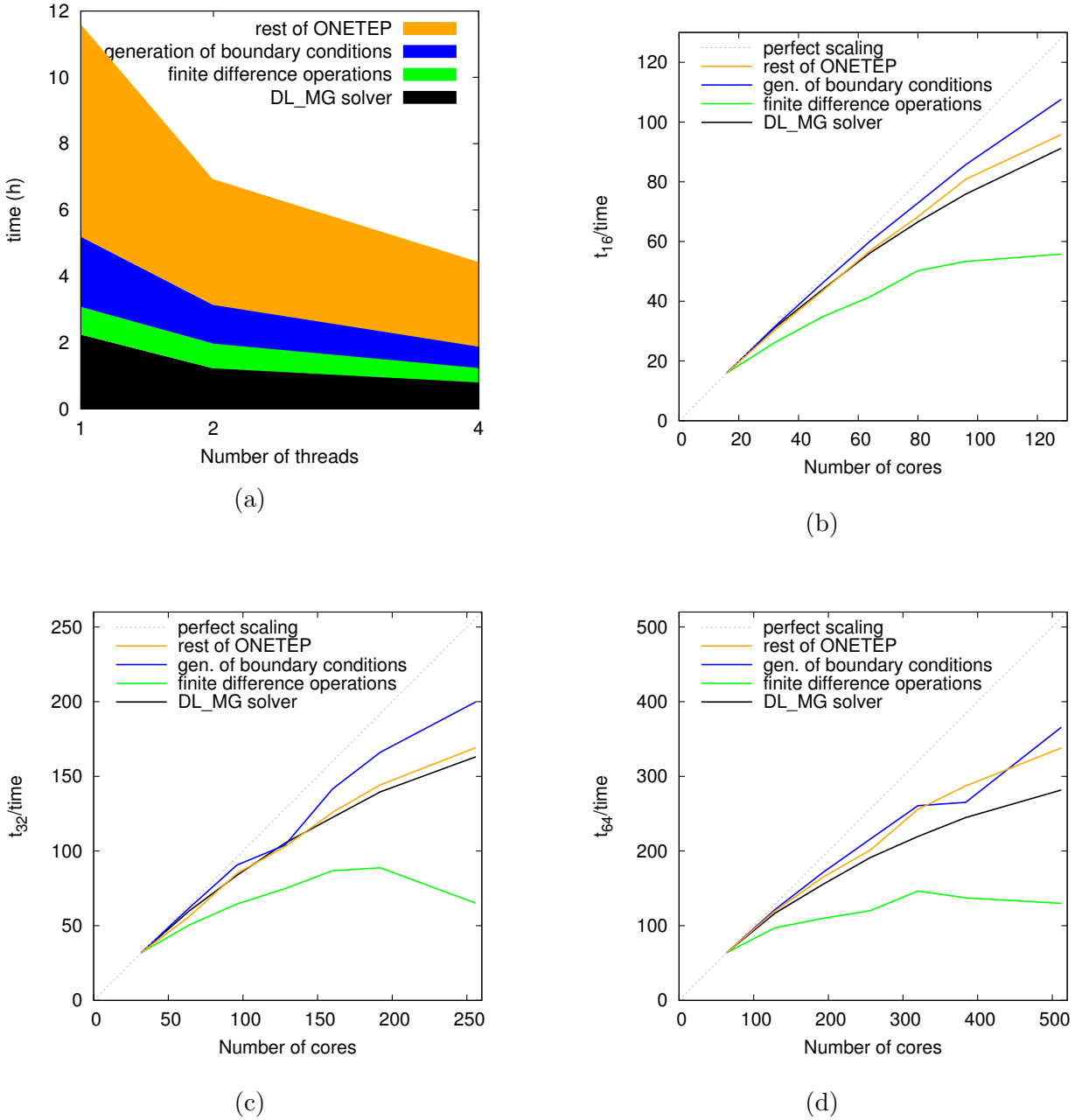

Figure 5: a) The partition of computing time amongst ONETEP components used in solvation components for 1, 2 and 4 OpenMP threads (128 MPI ranks). Hybrid mode scaling of the same components function of MPI ranks: b) 1 OpenMP thread (16 cores as references), c) 2 OpenMP threads (32 cores as reference), d) 4 OpenMP threads (64 cores as reference). Although the parallel performance of the finite difference operations is poor its weight in the ONETEP total run time is relatively small, see a).

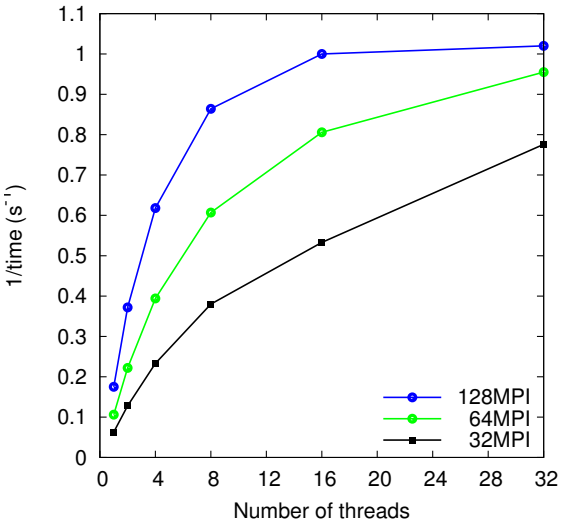

Figure 6: DL_MG scaling on HECToR for a PE representative for ONETEP. Global grid sizes are 449 × 545 × 609 partitioned in a 1D MPI topology. Please note that the hybrid mode is faster if the number the number of threads per MPI rank is less than 4, i.e. 32MPI×4threads > 64MPI×2threads > 128MPI×1thread.

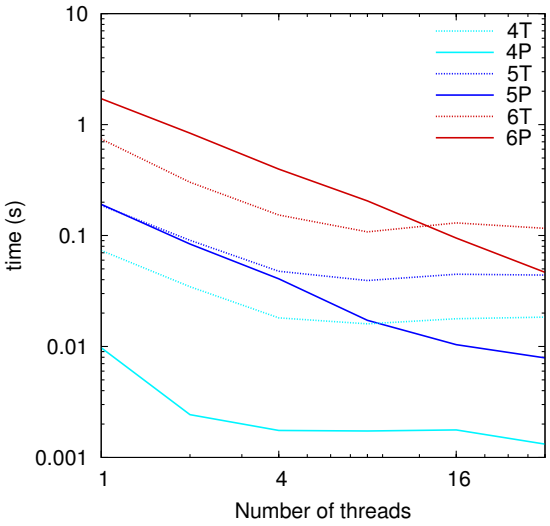

Figure 7: Smoother timings versus number of OpenMP threads for the top 3 multigrid levels (for the calculation shown in Fig. 6). In the key 'P' stands for computation time, 'T' for communication time. Run done with 128 MPI ranks. The degradation of OpenMP scaling observed in Fig. 6 for more than 8 threads is correlated with the crossover observed between the computation and communication time lines.

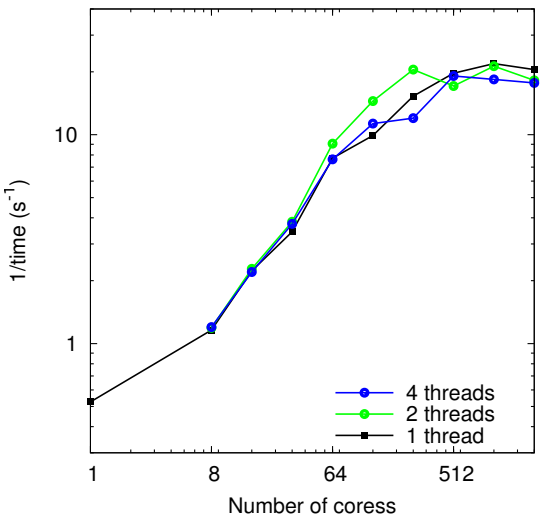

Figure 8: DL_MG scaling for a PE partitioned with a 3D MPI topology with 1, 2, and 4 OpenMP threads. Global grid sizes 129 × 129 × 129 are representative for CASTEP; runs done on HECToR.

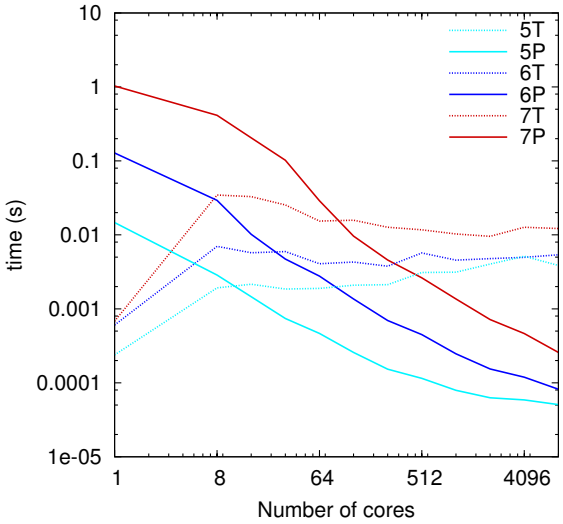

Figure 9: Smoother timings for the top 3 multigrid levels for 3D MPI topologies using 1 OpenMP thread (see Fig. 8). In the key 'P' stands for computation time, 'T' for communication time.