Technical Report for the eCSE project
A pinch of salt in ONETEP's solvent model

by Chris-Kriton Skylaris[1], Jacek Dziedzic[1,2], and Lucian Anton[3]

1. University of Southampton, School of Chemistry, Southampton, UK.
2. Gdansk University of Technology, Faculty of Applied Physics and Mathematics, Poland.
3. Scientific Computing Department, STFC, Daresbury Laboratory, UK.

1.1 Abstract

Chemical reactions, drug-protein interactions, and many chemical and physical processes on surfaces are examples of technologically important processes that happen in the presence of solvents. The inclusion of electrolytes (salt) in solvents such as water is crucial for biomolecular simulations, as most processes (e.g. protein-protein or protein-drug interactions or DNA mutations) take place in saline solutions. In this project we aimed to develop the capability to model electrolyte-containing solvents in quantum-mechanical simulations of materials from first principles.

The above aim has been achieved by extending the functionality and performance of two existing computer codes in tandem – the DL_MG multigrid solver and the ONETEP linear-scaling density functional theory package. The modifications to DL_MG were extensive in scope and included implementing an inexact-Newton solver, revisions in the treatment of boundary conditions, with simultaneous careful consideration of numerical stability in the presence of severe nonlinearities, and with continued excellent parallel performance in mind. Modifications to ONETEP involved comparatively less implementation, with significant focus on the physically correct capturing of relevant energy terms that arise in the presence of Boltzmann ions.

1.2 Introduction

The sustained exponential growth of computer power over the last decades now offers the capability to study quantum models of large molecules interacting in the presence of an ionic solute, which are of fundamental importance in a number of scientific fields and technological applications. One of most widely used equations to describe the solute and ion effects on the electrostatic potential is the Poisson-Boltzmann Equation (PBE):

$$\nabla \cdot (\varepsilon[n_{\mathrm{el}}](\mathbf{r})\nabla\phi(\mathbf{r})) = -4\pi\, n_{\mathrm{tot}}(\mathbf{r}) - 4\pi \sum_i q_i\, c_i\, \mathrm{e}^{-\beta(q_i\phi(\mathbf{r})+V(\mathbf{r}))}$$

where n_{el} is the electronic density, n_{tot} is the combined electronic and nuclear density, ϕ is the electrostatic potential, and ε is the dielectric permittivity, which is a functional of n_{el} in ONETEP, q_{i} are the charges of the ions that are in solution, each with a bulk concentration of c_{i}, $V(r)$ is the steric potential, and $\beta = 1/kT$ is the usual Boltzmann factor.

This project constituted a continuation of the dCSE project "*Multigrid solver module for ONETEP, CASTEP and other codes*" (1) which delivered the multigrid solver DL_MG to ONETEP and CASTEP developers. In the first release of DL_MG the Poisson Equation solver was fully tested and integrated in ONETEP and a basic implementation of the non-linear solver was produced as well. The work in the current project aimed to complete the implementation of the non-linear solver for PBE (WP1), its integration with ONETEP (WP2) and to release DL_MG to the wider community (WP3).

In the following sections we describe the main steps of the development for each of the work packages (WPs) and the new features made available.

1.3 WP1: Poisson-Boltzmann Equation solver

The first version of the nonlinear PBE solver was implemented with the Full Approximation Scheme (FAS) algorithm (2). Although this solution is elegant from a formal point of view, as it uses the multigrid elements developed for the Poisson equation solver, it converges very slowly or not at all for the values of concentration and temperature that are in the range of physical relevance. Consequently, we have moved to an inexact-Newton algorithm devised specifically for PBE of solutes described in detail in Ref. (3). In short, in this algorithm the linear multigrid is used at each step to find an approximate solution of the linearised equation. A damping factor is also computed for the linear correction in order to ensure global convergence. From the implementation point of view the work performed consisted in the following main steps: i) writing the subroutine for the inexact-Newton algorithm and the auxiliary subroutines for the computation of the damping factor, ii) writing a suite of tests to verify the correctness of the implementation, iii) extending the application programing interfaces and iv) testing and validation.

The parallel performance of the Poisson solver was presented in detail in (1). We should note here that the parallel scaling of the non-linear solver is not affected by the addition of the non-linear algorithm, as the majority of operations are local. If anything, the parallel scalability improves as the ratio of FLOPs to data transfers increases significantly in the non-linear section of the solver due to the computationally intensive floating point computations which now involve the (capped) exponential function, see Figure 1. To allow for full flexibility and efficiency, both the hard-core and soft-core potentials were implemented in separate functions. For the same reason the linearised PBE solver was implemented separately. During this development work the boundary conditions for the solver were extended to include Dirichlet, periodic and mixed for all cases of the solver.

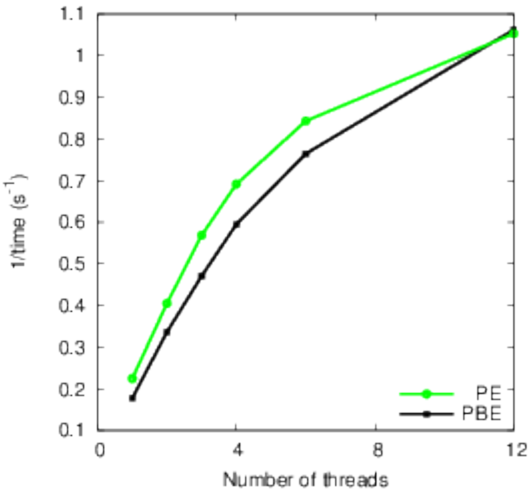

Figure 1 Comparing the scaling of the PBE solver vs PE solver for a spherical charge problem. The grid sizes, 449x545x609, are typical for ONETEP runs. The grid is distributed to 128 MPI ranks along one dimension, again typical for ONETEP computational setup.

1.4 WP2: Extend the ONETEP solvent model to saline solutions

The ONETEP solvent model has been suitably extended to support implicit solvation calculations in the presence of Boltzmann ions. A simple user interface has been provided, where an additional section in the input file can be used to specify species, charges and molar concentrations associated with the ions. The interface between ONETEP and DL_MG has been generalised to allow calculations where the linearised or full Poisson-Boltzmann equation is solved, in addition to calculations with the pure Poisson equation that had previously been supported.

Boundary conditions for implicit solvent calculations in saline solutions are now automatically set up to use the Debye-Hückel formulation, with a user option to override the default setting.

Both the hard-core and the soft-core steric potentials have been implemented in ONETEP. The hard-core potential strictly prevents the Boltzmann ions from entering spherically-shaped regions centred on atoms (the sphere radii are given through the input file), while the soft-core potential mimics damped Pauli repulsion with a term proportional to $A\,\mathrm{erf}(\alpha r)/r^{12}$, with A and α specified through the input file, thereby repelling the ions with a continuous potential. The steric potential is truncated at a user-specified distance to retain ONETEP's linear scaling, the discontinuity at the cutoff distance is prevented by an automatic shifting of the potential.

The defect-correction procedure has been extended to the case of non-zero salt concentrations. Suitable defect-correction equations have been re-derived and implemented through a concerted effort on both sides of the DL_MG and ONETEP interface. The defects in both the Poisson and Boltzmann terms are printed separately in the output to aid debugging in problematic situations.

The total energy expressions become non-trivial in the presence of Boltzmann ions, with the full PBE case necessitating particular attention. Following literature review (4) (5) (6), five energy terms have been identified and implemented in the full PBE scheme. In addition to the straightforward and previously implemented solvation of the solute in zero-ionic-strength solvent these are

1. the electrostatic work of placing the solute in the already organized ion atmosphere,
2. the electrostatic self-energy of the ion atmosphere,
3. the entropy of organizing the ions,
4. the so-called *osmotic term* arising from the excess osmotic pressure of the mobile ion cloud.

Furthermore, the above terms have been generalised in two ways beyond what is presented in established literature. First, the use of the capped exponential function in practical PBE calculations introduces a number of difficulties that are absent in purely-mathematical derivations of the above energy terms – with certain cancellations only taking place for uncapped exponentials. Second, the use of smooth ion accessibility functions (steric weights) complicates certain expressions. Both of these difficulties have been carefully addressed in this project.

The correctness of the implemented developments has been validated against carefully set up calculations with APBS (7), which is a well-known package for performing electrostatic calculations in the presence of solvent. While APBS is geared towards classical (point-charge) calculations, in one of its modes of operation it admits suitable quantities (charge density, permittivity, boundary conditions, ionic accessibility) on a Cartesian grid, which allows mimicking the set up of ONETEP solvation calculations. We demonstrated the agreement between the electrostatic energies in ONETEP and APBS on a simple test-case of a hydroxyl radical solvated in 0.1M NaCl for varying characteristics of hard-core and soft-core steric potentials. Below we reproduce plots that demonstrate very good agreement between APBS and ONETEP for both the linear and full formulation of the Poisson-Boltzmann equation and the unsuitability of point-charge solvation for meaningful comparisons.

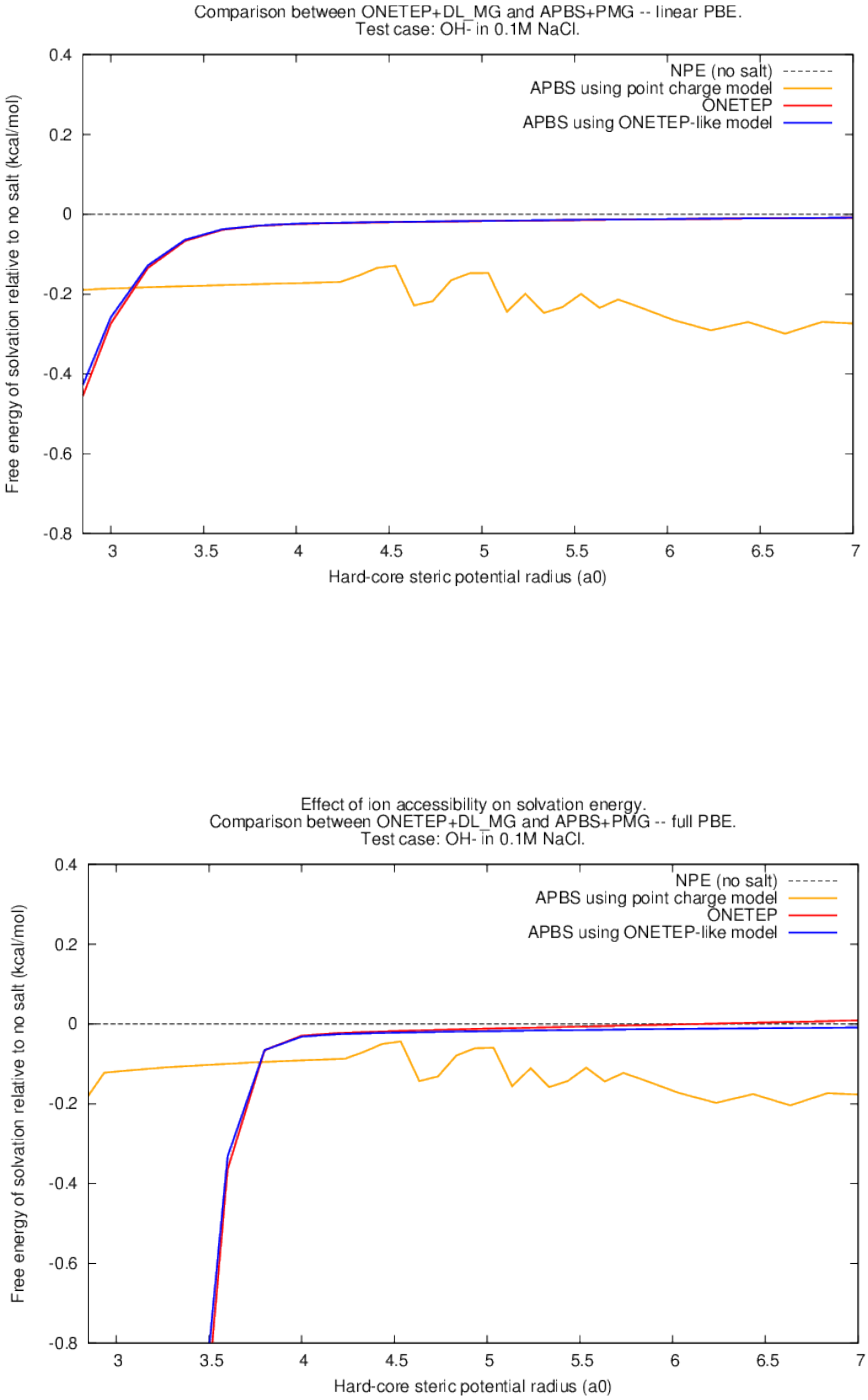

Fig. 2: Free energy of solvation of a hydroxyl radical in 0.1M NaCl, as computed by ONETEP with the DL_MG solver and APBS with the PMG solver. Linearised (top panel) and full (bottom panel) formulations have been tested. While APBS's point-charge model (yellow) is seen to be unsuitable for direct comparison against our formulation, a carefully set up distributed-charge APBS calculation (blue) reproduces our results (red) very well.

1.5 WP3: Usability improvements and first public release

The addition of the non-linear solver and the linearised PBE solver in DL_MG has significantly increased DL_MG's code complexity, and with it the number of input parameters. In order to avoid setup errors or inconsistencies, the initialisation subroutines now use a set of assertions to test the validity of the inputs. Reporting in the log file was improved to cover the new cases. The public subroutines can now return an error flag to the calling application, a number of specific error codes with associated explanatory messages were created and made available via an application programing interface that handles error messages. The set of test cases was extended to cover the linear and non-linear PBE. A test script was developed to run a suite of synthetic cases on a large number of parallel execution models: varying MPI topologies, varying number of OpenMP threads both in hybrid mode and standalone. This is the first defense barrier against code changes that could accidentally introduce regressions.

An option to output the values of crucial quantities, at every step of the multigrid defect-correction procedure, for a user-defined cross section through the simulation cell has been added. This greatly helps user-side debugging, should the calculation be set up incorrectly. The quantities (optionally) printed include the electronic density, electrostatic potential, permittivity, steric potential and steric weight. The produced outputs are can be directly plotted using e.g. gnuplot.

The documentation, which contains a general description of the algorithms employed and the details of all public interfaces, was integrated with the source code using Doxygen.

The solver was extensively tested on ARCHER but also on other systems which are available to UK scientists (e.g. Iridis4) and with all major compilers and releases of MPI.

DL_MG source code releases and the online documentation are available through CCP-Forge at http://ccpforge.cse.rl.ac.uk/gf/project/dl-mg/

1.6 Conclusions

Through extensive changes to the DL_MG solver and localised changes to ONETEP, with careful consideration of the underlying theory and numerical issues, we enabled the capability for density functional theory calculations with implicit solvent and non-zero salt concentrations through a physically sound, elegant and numerically efficient and scalable approach.

1.7 Acknowledgments

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

1.8 References

1. **Anton, Lucian, et al.** [Online] 2013. http://www.hector.ac.uk/cse/distributedcse/reports/onetep/onetep.pdf.
2. **Trotterberg, U, Oosterlee, C and Schüller, A.** *Multigrid.* s.l. : Academic Press, 2001.
3. **Holst, M. and Saied, F.** Numerical solution of the Poisson-Boltzmann equation: Developing more robust and efficient methods. *J. Comput. Chem.* 1995, pp. 337-364.
4. **Sharp, Kim A. and Honig, Barry.** Calculating total electrostatic energies with the nonlinear Poisson-Boltzmann equation. *J. Phys. Chem.* 94, 1990, 19, pp. 7684-7692.
5. **Fogolari, F., Brigo, A. and Molinari, H.** The Poisson–Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition.* 15, 2002, 6, pp. 377-392.
6. **Zhou, Huan Xiang.** Macromolecular electrostatic energy within the nonlinear Poisson-Boltzmann equation. *J. Chem. Phys.* 100, 1994, 4, pp. 3152-3162.
7. **Baker, N. A., et al.** Electrostatics of nanosystems: application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA.* 98, 2001, pp. 10037-10041.