Preconditioned Geometry Optimisers for the CASTEP and ONETEP codes

James Kermode (University of Warwick), Christoph Ortner (University of Warwick), Nicholas Hine (University of Warwick), Matt Probert (University of York), Letif Mones (University of Warwick) and Jolyon Aarons (University of Warwick)

We implemented two recently developed preconditioners for enhancing LBFGS-based geometry optimisations of chemical and material systems into the CASTEP and ONETEP DFT packages. The implementations were tested for a wide range of systems and depending on the size and quality of the systems we could gain at least a two-fold decrease in the required number of optimisation steps and corresponding computational cost compared to the standard LBFGS method. The implementation was carried out in an extensible way that gives the possibility to straightforwardly add other general or specific preconditioners and apply these preconditioners to enhance other optimisation techniques implemented in the geometry optimisation module.

Introduction

Density functional theory (DFT) is one of the most widely used quantum chemistry method to investigate chemicals and materials by performing calculations that explicitly treat (at least the chemically important part of) electronic degrees of freedom. The CASTEP [1] and ONETEP [2] DFT packages are UK flagship codes and are heavily used on ARCHER (making up 8 and 1% respectively of recent usage).

For a given configuration DFT can provide the ground-state energy of the system and also the forces on atoms (i.e. the negative gradient of the ground-state energy with respect to the Cartesian coordinates of atoms). This gives the possibility to perform geometry optimisation to obtain optimal configuration(s), where the ground-state energy is minimal (and therefore the force and stress components are zero). Optimal configurations are essential for several advanced analysis such as the examination of vibrational modes for molecules in gas phase and phonon or electronic bandstructure calculations for material systems.

The most widely used geometry optimisers in the field of computational chemistry are based on quasi-Newton methods that try to find local stationary points by an iterative manner using only the energy and forces of the system. They approximate the required Hessian matrix based on gradient information of the actual and previous steps. One of the most successfully used approaches in computational chemistry is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm and its limited memory version (LBFGS). A specific version of both the BFGS and LBFGS methods that can simultaneously relax the fractional coordinates and lattice parameters [3] is implemented to CASTEP and ONETEP [4].

There are several chemical and material systems, however, that have a rather anisotropic shape of potential energy surface around their minima that leads to slow convergence of BFGS/LBFGS methods. It is widely known that preconditioners that redefine the metric through a coordinate transformation (and therefore make the local shape of the surface more isotropic) speed up the convergence. Our aim was to implement two recently developed preconditioners [5, 6] into the ONETEP and CASTEP codes, whose efficiency has been already demonstrated with a speedup factor of 2-10x compared to standard LBFGS [7].

The Exp preconditioner [5] is based on the continuous generalisation of the Laplacian matrix. It is a very simple but effective preconditioner that uses only local environmental information around the atoms and therefore its computation is pretty straightforward. The FF preconditioner [6] includes rather specific information about the local bonding based on some universal force field.

Application of preconditioners requires solving a linear system, which is the most expensive part of their application. However, our aim was to keep the computational cost of the construction of the preconditioners and the solution of linear system below 5% of the cost of the electronic structure problem. Keeping the overhead small can ensure that the computational gain using preconditioners is comparable to the speed up factor of the convergence they can provide.

Achievement of objectives

Below we list the objective of the project with the corresponding success metric and actual achievement.

1. Produce a robust serial implementation of the preconditioner of Packwood et al. [*J. Chem. Phys.* **144**, 164109 (2016)] in both CASTEP and ONETEP.

Success metric: reduction of 2-10x in the number of force evaluations required to optimise the geometry of ~1k atom inorganic crystals in both CASTEP and ONETEP compared to current implementation of LBFGS (Limited-memory Broyden Fletcher Goldfarb Shanno) scheme, with precise factor depending on system size and bonding complexity.

Achievement: several systems were tested including single organic molecule, molecular crystal, carbon nanotube with defect and material system. The reduction of force evaluation was at least two-fold even for smaller systems and depending on the bonding complexity the gain increased with system size as expected.

2. Optimise and parallelise the construction and application of the preconditioner sufficiently that it does not become a bottleneck in large systems for typical parallel partition sizes used on ARCHER.

Success metric: reduce overhead of preconditioned geometry compared to current LBFGS implementation to less than 5% of electronic minimisation time for systems containing ~1k atoms running on ~1k cores on ARCHER. Combined with 2-10x reduction in number of steps required from Objective 1, this will deliver a significant speedup in the overall runtime.

Achievement: based on careful tests the overhead of preconditioned geometry optimisation was less than 5% of electronic minimisation time and given the significant reduction in number of force evaluations it had a negligible computational cost.

3. Evaluate the necessity for the use of sparse matrices and/or an algebraic multigrid solver to speed up preconditioner assembly and application, considering both existing sparse matrix infrastructure within the codes and additional external libraries. Only required if Objective 2 cannot be met with dense linear algebra.

Success metric: use sparse linear algebra to reduce overhead of preconditioned geometry compared to current implementation to less than 5% of electronic minimisation time for systems containing ~1k atoms running on ~1k cores on ARCHER.

Achievement: as the current implementation of the construction of the preconditioners and solution of the linear equations have an overhead less than 5% this objective was not required.

4. Provide experimental support for new preconditioners (prototypes in unpublished work) to allow further speedup e.g. by taking into account multiple bond types, which need to be preconditioned differently.

Success metric: implementation of novel preconditioners completed and speed up of at least 2x relative to current unpreconditioned LBFGS demonstrated for geometry optimisation of molecular crystals.

Achievement: a simple but highly efficient recently introduced force field-based preconditioner was also implemented. For some systems this preconditioner outperformed the Exp preconditioner. Also, we implemented an automatic scaling method to adjust strain components of the preconditioner for variable cell optimisations that resulted in an additional ~30% improvement.

5. Leave behind the infrastructure and expertise to expand preconditioned geometry optimisation in both codes for specific application areas that may require bespoke preconditioners.

Success metric: code implementing objectives 1-4 made available in main CASTEP and ONETEP codebases, documented and supported through user tutorials, and in use by at least two of the early user research groups identified in Section 7.

Achievement: the implementation was carried out in a modular way that supports any additional extension of the code using general or specific preconditioners.

Implementation details

In CASTEP the geometry optimisation module (*Source/Functional/geometry.f90*) is a serial Fortran90 code that includes several optimisation techniques and line search methods. In this work we implemented and tested the preconditioners in combination with the LBFGS method, as this is the most widely preferred optimisation technique in the field. We note, however, that the preconditioners can be combined with other optimisation methods that may be the subject of some future work (e.g. preconditioning the Barzilai-Borwein gradient method).

The preconditioners were added into the geometry optimisation module and the existing LBFGS method was slightly modified to accommodate the applicability of the preconditioners. We note that the identity preconditioner (ID) corresponds to an unpreconditioned optimisation model. For validation purposes, we implemented the ID preconditioner as well and we made sure that this provided exactly the same results as the original (unpreconditioned) LBFGS method.

The construction of the preconditioners and solution of the linear system were implemented as serial codes. After careful tests we concluded that even for relatively large systems (~10k atoms) thank to the sparsity of the preconditioner matrices and the optimised Cholesky factorisation of LAPACK the serial implementation had only a marginal (< 5%) overhead relative to the computational cost of the electronic structure. Therefore we decided that neither the parallel implementation nor the application of sparse linear algebra / iterative solvers were required.

ONETEP's geometry optimisation module (*src/geometry_optimiser_mod.F90*) is derived from the one of CASTEP that made the transfer of the preconditioners straightforward.

The current implementation of the preconditioners also supports linear constraints and fixed atomic positions in both CASTEP and ONETEP.

A glossary with the available keywords and their default values are presented in Table 1.

Keyword	Description	Values	Remark
`geom_precond_type`	type of preconditioner	"NONE"	default; no preconditioner, equivalent to LBFGS,
		"ID"	identity preconditioner, equivalent to LBFGS, testing purpose only
		"EXP"	exponential preconditioner
		"FF"	force field based preconditioner

geom_precond_exp_A	Parameter A for the Exp preconditioner	REAL	must be >= 0.0, default is 3.0
geom_precond_exp_c_stab	stabilisation constant for EXP preconditioner	REAL	must be > 0.0, default is 0.1
geom_precond_exp_r_NN	nearest neighbour distance for EXP preconditioner	REAL	must be >= 0.0, default is 0.0 [ang] => computed automatically based on initial structure
geom_precond_exp_r_cut	cutoff distance for EXP preconditioner	REAL	must be >= 0.0, default is 0.0 [ang] => computed automatically based on initial structure
geom_precond_exp_mu	mu parameter for EXP preconditioner	REAL	must be >= 0.0, default is 0.0 [eV/ang**2] => computed automatically based on initial structure
geom_precond_ff_c_stab	stabilisation constant for FF preconditioner	REAL	must be > 0.0, default is 0.1 [eV/ang**2]
geom_precond_ff_r_cut	cutoff distance for FF preconditioner	REAL	must be > 0.0, default is 1.7 [ang]
geom_precond_scale_cell	scaling strain part of preconditioner	LOGICAL	default is .false.

Table 1. List of available keywords of the implemented preconditioners in CASTEP and ONETEP

Performance results

1. Preliminary results for the Exp preconditioner using empirical potential

Beside DFT CASTEP provides some empirical potentials that gave us the possibility to perform quick tests of the implementation on relatively cheap potentials.

The first test system was bulk silicon with 512 atoms using Stillinger-Weber (SW) potential. We tested both fixed and variable cell optimisations. We compared the performance of the geometry optimisations to those previously implemented in ASE as well as previously implemented preconditioners in CASTEP. The convergence criterion was $|F_{max}|$ = 0.0001 eV/Å. Results are presented in Table 2. We note that both the unpreconditioned (denoted as simple LBFGS) and Exp-preconditioned (denoted as LBFGS / Exp) geometry optimisations require significantly smaller number of steps in CASTEP than in ASE. The reason is that CASTEP has a much more sophisticated and stable line search algorithm. We also note that since ASE is written in Python its performance is significantly slower compared to CASTEP (which is written in Fortran 90) and we therefore omitted its CPU times.

In CASTEP, compared to the unpreconditioned LBFGS method using the Exp preconditioner resulted in **2.2** and **1.3 fold decrease** in CPU times for fixed and variable cell optimisations, respectively. Since the computation of the SW potential is negligible, for these optimisations it is a better to compare the number of optimisation steps, where the gain is **2.9** and **1.8 fold** for fixed and variable cell optimisations, respectively.

We also compared the performance of the Exp preconditioner to other preconditioners already implemented in CASTEP (i.e. Housholder, Identity and Scaled identity methods). For fixed cell optimisations the Exp preconditioner significantly outperforms the others in the required number of steps, while for variable cell optimisation it has a moderate gain compared to the Scaled identity preconditioner. In order to improve the convergence of the Exp preconditioner we introduced a new version of it in which we rescaled the magnitude of the diagonal strain components of the preconditioner to be adjusted to the fractional coordinates' components. Using this simple rescaling we were able to improve the convergence by an additional ~30%.

Method	Fixed cell optimisation		Variable cell optimisation	
	# of steps	Wall time (s)	# of steps	Wall time (s)
ASE / LBFGS	72	-	196	-
ASE / Precond-LBFGS / Exp	18	-	59	-
CASTEP / LBFGS	46	16.92	64	24.86
CASTEP / LBFGS / Housholder	46	16.97	64	23.71
CASTEP / LBFGS / Identity	79	28.73	80	28.72
CASTEP / LBFGS / Scaled identity	34	**7.32**	40	**14.94**
CASTEP / Precond-LBFGS / Exp	**16**	7.76	**35**	19.42
CASTEP / Precond-LBFGS / Exp (scaling)	-	-	**26**	15.33

Table 2. Comparison of number of steps of convergence and wall times of different geometry optimisation techniques with fixed and variable cell for bulk silicon using Stillinger-Weber pair potential. Convergence criterion was $|F_{max}| = 0.0001$ eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz.

2. Testing the Exp preconditioner for linear constraints

Both the CASTEP and ONETEP codes support linear constraints during geometry optimisations. We implemented the preconditioners to support these linear constraints and tested its performance on the silicon bulk system by fixing the single outermost layer of the simulation cell. All other properties of the system were the same as in the previous section. The results are collected in Table 3.

Unlike initial preconditioners, the Exp preconditioner significantly reduces both the required number of optimisations steps and associated CPU times for constrained optimisations as well. The computational gains are **3.0** and **2.1 fold** for the number of steps and CPU times, respectively.

Method	Fixed cell optimisation	
	# of steps	Wall time (s)
CASTEP / LBFGS	45	19.30
CASTEP / LBFGS / Housholder	45	19.46
CASTEP / LBFGS / Identity	57	23.61
CASTEP / LBFGS / Scaled identity	59	22.87
CASTEP / Precond-LBFGS / Exp	**15**	**8.99**

Table 3. Comparison of number of steps of convergence and wall times of different geometry optimisation techniques with linear constraints and fixed cell for bulk silicon using Stillinger-Weber pair potential. Convergence criterion was $|F_{max}|$ = 0.0001 eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz.

3. Testing the performance of the Exp preconditioner with increasing system size

As it has been shown, significant computational gain can be achieved by the Exp preconditioner for large systems so we tested the silicon bulk system with and without the preconditioner using 3 sizes (Table 4).

Method	Wall times (h)					
	Fixed cell			Variable cell		
	512 atoms	4096 atoms	10648 atoms	512 atoms	4096 atoms	10648 atoms
CASTEP / LBFGS	0.005	4.98	30.29	0.007	9.79	>48
CASTEP / Precond-LBFGS / Exp	**0.002**	**1.04**	**10.70**	**0.005**	**0.44**	**19.98**

Table 4. Comparison of wall times of geometry optimisations with and without the Exp preconditioner for fixed and variable cell for bulk silicon using Stillinger-Weber pair potential and different system size. Convergence criterion was $|F_{max}|$ = 0.0001 eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz.

4. Testing the Exp preconditioner for molecular crystals

We also tested how the Exp preconditioner perfroms on a molecular crystal taken from the CSP Blind Test crystal-structure prediction competition (system XXII.). We applied a PBE functional with 800 eV cutoff energy and fixed cell optimisations. From Table 5 it seems that the computational gain is again ~2-fold, although the system is still relatively small (60 atoms).

Method	Fixed cell optimisation	
	# of steps	Wall time (h)
CASTEP / LBFGS	70	10.13

CASTEP / Precond-LBFGS / Exp	**40**	**5.11**

Table 5. Comparison of number of steps and wall times of LBFGS geometry optimisations without and with the Exp preconditioners for a molecular crystal (XXII, atoms) on DFT potential. Convergence criterion was $|F_{max}|$ = 0.001 eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz, 128 cores.

5. Testing the FF preconditioner

Beside the Exp preconditioner we also implemented a more specific force field based preconditioner that has been shown to be effective for molecules and molecular crystals. The implementation is based on simple a general force field [7] that does not require any external file or parameter specification.

We compared the performance of unpreconditioned, Exp- and FF-preconditioned LBFGS optimisations for two systems using DFT potentials. The first system was a simple molecular system, 5-nitrobenzisoxazole (only 16 atoms) where the Exp preconditioner is not expected to perform much better than the unpreconditioned optimisation. The PBE functional with 400 eV cutoff energy was used. From Table 4 it is clear that although the improvement using the Exp preconditioner is relatively moderate (~30%), the FF preconditioner provided an approximately 6-fold improvement in CPU time compared to the standard LBFGS method even for such a small system.

Method	Fixed cell optimisation	
	# of steps	Wall time (h)
CASTEP / LBFGS	87	0.30
CASTEP / Precond-LBFGS / Exp	61	0.21
CASTEP / Precond-LBFGS / FF	**20**	**0.05**

Table 4. Comparison of number of steps and wall times of LBFGS geometry optimisations without and with the Exp and FF preconditioners for a simple molecular system (5-nitrobenzisoxazole, 16 atoms) on DFT potential. Convergence criterion was $|F_{max}|$ = 0.001 eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz, 16 cores.

The second system was a *z*-axial defect in a carbon nanotube [9] using PBE functional with 380 eV cutoff energy with two different system sizes. Compared to the unpreconditioned optimisation the Exp preconditioner enhances the convergence by about a factor of 2 for both sizes, while the FF preconditioner has an even better performance that increases with system size (~3 and ~4 fold gain in wall times for the smaller and larger systems, respectively).

Method	143 atoms		287 atoms	
	# of steps	Wall time (h)	# of steps	Wall time (h)

CASTEP / LBFGS	59	2.82	52	21.89
CASTEP / Precond-LBFGS / Exp	22	1.46	32	12.84
CASTEP / Precond-LBFGS / FF	**16**	**0.92**	**15**	**5.13**

Table 6. Comparison of number of steps and wall times of LBFGS geometry optimisations without and with the Exp and FF preconditioners for a carbon nanotube on DFT potential. Convergence criterion was $|F_{max}| = 0.001$ eV/Å. CPU: Intel Xeon E5-2630 v3 2.4 GHz, 16 cores.

6. Testing the ONETEP implementation

All of the above CASTEP systems except bulk silicon were retested with the ONETEP implementation, and comparable speed ups were obtained with both Exp and FF preconditioners. In addition, new tests were carried out on the Bisphenol A (BPA) organic molecule. The speed up obtained for BPA with Exp and FF preconditioners is illustrated in Fig. 1 below, demonstrating around 2x speedup for Exp preconditioner and approximately 5x speedup for FF.

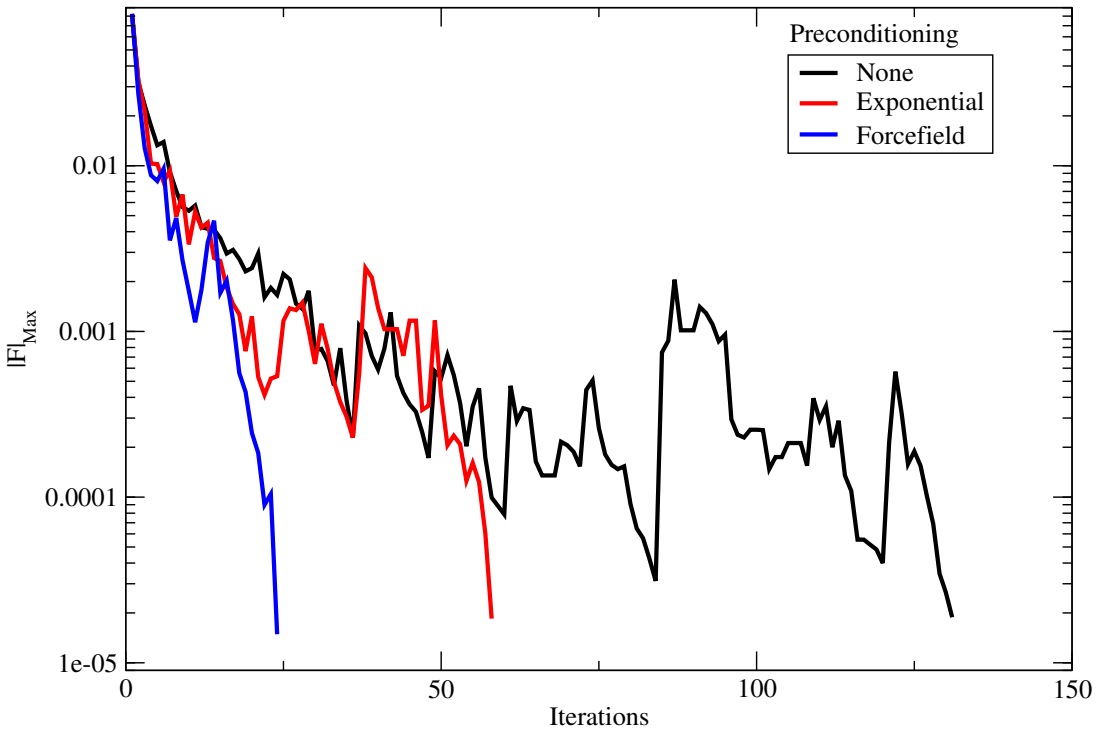

Figure. 1. Maximum absolute force component during optimisation, in Ha/Bohr, for BPA with three different LBFGS optimisation schemes. Black: previous implementation, no preconditioner; red: exponential preconditioner; blue: force field preconditioner.

Conclusion

In this technical report we demonstrated the successful implementation of the Exp and FF preconditioners in the CASTEP and ONETEP codes. We carried out a robust serial implementation of the preconditioners in combination of the LBFGS optimiser already implemented in the program packages. Based on careful tests of the performance we found that the overhead of preconditioners is less than 5% of the electronic minimisation time and concluded that no additional code optimisation (i.e. parallelisation and application of sparse matrices/algebraic multigrid solver) is required. Using several test systems we showed that the preconditioners efficiently reduce the number of optimisations steps and so the computational cost at least by a factor of 2 even for small systems, with the gain increasing with system size.

Acknowledgements

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

References

[1] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. J. Probert, K. Refson, 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] B. G. Pfrommer, M. Cote, S. G. Louie, and M. L. Cohen, *J. Comput. Phys.* **131**, 233 (1997).

[4] N. D. M. Hine, M. Robinson, P. D. Haynes, C.-K. Skylaris, M. C. Payne and A. A. Mostofi, Phys. Rev. B 83, 195102 (2011).

[5] D. Packwood, J. R. Kermode, L. Mones, N. Bernstein, J. Woolley, N. Gould, C. Ortner, and G. Csányi, *J. Chem. Phys.* **144**, 164109 (2016).

[6] L. Mones, G. Csanyi and C. Ortner https://arxiv.org/abs/1804.01590

[7] A. Larsen et al., *J. Phys. Condens. Matter* (2017)

[8] R. Lindh, A. Bernhardsson, G. Karlstrom, P.-A. Malmqvist, *Chem. Phys. Lett*. **241**, 423 (1995)

[9] M. R. C. Hunt and S. J. Clark *Phys. Rev. Lett.* **109**, 265502 (2012)