

The Guide 2.0 to CASTEP (covering version 4.2)

Phil Lindan

Computational Materials Science Group,
Daresbury Laboratory, Warrington WA4 4AD, UK

From 2001:

Department of Physical Sciences,
University of Kent at Canterbury,
Canterbury, Kent Ct2 7NR, UK

`ukcp@dl.ac.uk`
`http://www.cse.clrc.ac.uk/Activity/UKCP`

October 19, 2000

About this Guide

This Guide covers the **CASTEP 4.2** code and is part of the distribution to UK academia. The Guide is divided into three parts:

Part I provides a brief introduction to first-principles simulation and the history of **CASTEP**.

Part II tells you how to install and run **CASTEP 4.2**

Part III contains some useful reference material

In this version the scope is deliberately limited to explaining the basics of running the code. In later revisions I hope to include more detail of the theory, methodology and the code itself.

License terms and copyright

The entire academic **CASTEP 4.2** distribution, that is, the **CASTEP 4.2** code, associated library files and data, utility programs and this Guide, is strictly copyright and is distributed under license as part of an Agreement between CLRC Daresbury Laboratory, the United-Kingdom Car-Parrinello Consortium (UKCP) and Molecular Simulations Inc. (MSI). You must sign a License Agreement to obtain and use any part of the academic **CASTEP 4.2** distribution. Please note in particular that the code is provided for use *solely in UK academic institutions*, and only for non-commercial use.

Prerequisites and minimum requirements

CASTEP is a scientific research code under constant development. It is implicit therefore that its users know rather a lot about the science and methodology, enough about the computers they use, and that these computers are suitable for large-scale simulations. To be explicit, one cannot run **CASTEP** sensibly without at least:

- Some knowledge of solid-state theory
- A UNIX workstation or LINUX PC with ≥ 1 GB free disk space and preferably at least 100 MB available memory
- A working knowledge of UNIX and the ability to edit text files

By far the most convenient way to use **CASTEP** is through the Cerius² User Interface (UI), available at reasonable cost to UK academics from MSI. If

you cannot obtain the Cerius², this Guide describes the essential features of the input files and some aspects of the code. Please bear in mind that CASTEP has been developed to run under the interface. Running the code without the interface requires considerable extra work and preparation of the files "by hand".

When all else fails

This Guide is by no means exhaustive, but it should help you to get started. You will also find that looking at the code is a good way (indeed, the only way) to understand fully what's going on. If you are at a complete loss, send a message to `ukcp@d1.ac.uk` describing your problem in detail. Whilst at this time we cannot provide full user support we will do our best to find a solution.

Credits and thanks

The CASTEP project was initiated by Mike Payne of Cambridge University. CASTEP 4.2 is the product of the collaborative efforts of dozens of researchers, students, postdocs and software professionals. Literally 100's of person-years' effort has gone into the code. Without naming names, which would inevitably result in omissions and unintentional offence, sincere thanks must go to all who have contributed over the years, and those who are actively developing the code right now.

Research and publications benefitting from the use of CASTEP 4.2 (and earlier academic releases, CASTEP 3.9) should include the following citation:

CASTEP 4.2 academic version, licensed under the UKCP-MSI Agreement, 1999; *Rev. Mod. Phys.* **64** (1992) 1045

If you use the K290 code (see Section 7.5) please cite the following reference: "Symmetry and Special Points Program K290", K. Kunc, R. J. Needs, O. H. Nielsen and R. M. Martin (to be published).

The information herein was derived from many sources. My thanks go to: Victor Milman of MSI for detailed basic information; Matt Probert of York University for molecular dynamics documentation; Matt Segall of Cambridge University for material on population analysis, included almost verbatim; Ben Jesson for information on aspects of the parallel coding; and Rob Allan for porting information and proof-reading. Errors and omissions are solely this author's fault!

Contents

About this Guide	1
I Overview	7
1 Introduction	7
1.1 UKCP	7
1.2 First-principles simulation	7
2 CASTEP History	8
2.1 Beginnings	8
2.2 CETEP: Meeting the Grand Challenges	8
2.3 CASTEP Now	9
2.4 Parallel CASTEP	9
II Using CASTEP	10
3 The basics	10
3.1 The distribution	10
3.2 Installation	10
4 Running CASTEP 4.2	11
4.1 The file “seed”	11
4.2 Examining the output	11
4.3 Types of calculation and associated files	12
4.4 Input files	13
4.5 Output files	13
4.6 Restarting a calculation	15
4.6.1 MD restarts	15
4.7 Similarities between CASTEP 4.2 and CETEP files	15
5 Running the test cases	16
5.1 Checking	18

6	Compiling CASTEP 4.2	18
6.1	Software resources, libraries	18
6.2	Targets	19
6.3	Compiler Options	19
6.3.1	Absoft Fortran v1.01 on Linux Pentium Platforms	19
6.3.2	Compaq	20
6.3.3	Cray	20
6.3.4	Fujitsu	21
6.3.5	Lahey	21
6.3.6	HP	21
6.3.7	Hitachi	21
6.3.8	IBM Fortran v6.1 and v7.1	22
6.3.9	NEC	22
6.3.10	Pacific Sierra Research	22
6.3.11	Portland Group Compiler on Linux Platforms	22
6.3.12	SGI	23
6.3.13	SUN	23
6.4	Serial or parallel?	24
6.5	SGI shared-memory machines	24
6.6	Running on parallel machines with MPI	24
6.7	MPI-Parallel distribution	25
6.8	Replicated data	26
7	Detailed description of some CASTEP 4.2 files	27
7.1	Geometry Information: <code>.geom</code>	27
7.2	Control parameters: <code>.rundat</code>	29
7.3	Runtime parameters: <code>.param</code>	42
7.4	The pseudopotentials: <code>.recpot</code> and <code>.realpot</code>	44
7.4.1	Preparing <code>.recpot</code> when using ultrasoft pseudopotentials	44
7.4.2	Preparing <code>.recpot</code> when using norm-conserving potentials	46

7.4.3	Using ultrasoft and norm-conserving pseudopotentials together	48
7.4.4	The real-space pseudopotential file <code>.realpot</code>	48
7.5	Symmetry information: <code>.symm</code>	48
7.6	The <code>.optim</code> file	49
7.7	Linear constraints within MD: the <code>.constr</code> file	50
7.8	Population analysis (and the <code>.batom</code> file)	51
7.8.1	Introduction	51
7.8.2	Theory	52
7.8.3	The LCAO Basis Set	53
7.8.4	Input Parameters	54
7.8.5	Interpreting the Results	54
III	For Reference	59
8	Publications about methodology	59
8.1	Scientific reviews and overviews	59
8.2	Density-functional theory	59
8.3	Functionals	59
8.4	Methodology	60
8.5	k-points	60
8.6	Pseudopotentials	60
8.7	Density mixing	61
8.8	Special techniques for metals	61
8.9	Molecular dynamics	61
8.10	Using parallel computers	61
9	Some UKCP publications	62

List of Tables

1	What's in the distribution	10
2	Relationships between <code>CASTEP 4.2</code> and <code>CETEP</code> files	16

3	The test cases	17
4	Choice of minimiser when using ultrasoft pseudopotentials . .	38
5	Pseudopotential files	44

Part I

Overview

1 Introduction

Welcome to CASTEP. The code has been made available to UK academia through a unique agreement between the UK Car-Parrinello Consortium (UKCP), Molecular Simulations Inc. (MSI) and CLRC Daresbury Laboratory. As well as providing you with the code, the agreement benefits from an energetic and co-ordinated CASTEP development programme. The agreement rests on a long history of CASTEP development and groundbreaking applications. CASTEP 4.2 is the result of much hard work and is a valuable resource, so please respect the terms of the license you signed.

1.1 UKCP

The United-Kingdom Car-Parrinello Consortium, or UKCP, was formed in 1990 with the aim of exploiting parallel supercomputers for first-principles simulation. This has been achieved with remarkable success, as witnessed by the large number of top-class publications from the UKCP research groups (see Section 9. UKCP has become much more than a computer consortium though. It is now a powerful collaborative network bringing together academic researchers, high-performance computing, MSI's software expertise and vast experience of Car-Parrinello code development and application. It is the focus for the UK's plane-wave DFT community of which you are now a part.

Currently there are twelve groups in the Consortium, based in the Universities of Bath, Cambridge (2), Durham, Edinburgh, Oxford (2), Queen's Belfast, University College London, University of Kent at Canterbury, York University and the Computational Materials science group at Daresbury Laboratory. The Engineering and Physical Sciences Research Council funds support for UKCP, which is provided by Daresbury.

1.2 First-principles simulation

CASTEP is a first-principles simulation code. First-principles calculations involve no adjustable parameters, and are based on an explicit quantum treatment of the electrons in a model system, which means solving Schrödinger's equation to find the electronic ground state. There are still approximations in the solution, but these are of an extremely general kind. The calcula-

tions yield the total energy of assemblies of atoms, and forces on the atoms. This knowledge enables one to calculate a vast range of properties with near chemical accuracy.

Density Functional Theory (DFT) is a powerful, general theory relating the total energy of a system of interacting electrons in an external potential to the electron density. Many first-principles calculations employ DFT to describe the valence electrons. The expansion in the scope of such calculations has been breathtaking in the last fifteen years or so, and they are now routinely used to study fundamental aspects of the physics and chemistry of condensed matter in every conceivable guise, ranging from catalysts to proteins, minerals to liquid metals. The great importance of DFT and its application in simulation was recognised by the award of the 1998 Nobel Prize in chemistry to Walter Kohn and John Pople.

The DFT plane-wave pseudopotential method, was pioneered in the 80's by Roberto Car and Michele Parrinello. This approach has transformed the first-principles landscape, making possible calculations far beyond previous approaches. Throughout this time the UK has played a leading role in the development and application of the methods, an effort founded on Mike Payne's work with CASTEP.

The method relies on a plane-wave basis, pseudopotentials and the use of density-functional theory to describe the valence electrons in a model system. Other ingredients include fast-Fourier transforms and minimization of the total energy rather than matrix diagonalization.

2 CASTEP History

2.1 Beginnings

The CAMbridge Serial Total Energy Package or CASTEP was developed in the mid-'80s by Mike Payne, with numerous contributions from the students, postdocs and academic staff of many research groups. As mentioned, CASTEP is a DFT plane-wave pseudopotential code, and it calculates the total energy, forces and stresses in a 3D-periodic system. Having ionic forces allows for efficient structural optimisation and for molecular dynamics calculations. The review article *Rev. Mod. Phys.* **64** (1992) 1045 sets out in detail the theory and technical implementation of CASTEP.

2.2 CASTEP: Meeting the Grand Challenges

It was recognised early that realistic and accurate simulations would inherently be on a grand scale, and that the use of parallel machines was essen-

tial. Out of this came CETEP, a highly efficient parallel version of CASTEP in which the grids spanning real- and reciprocal-space were decomposed and distributed among the nodes of the parallel machine. Once again, credit goes to many research groups and individuals for their contributions to CETEP. UKCP's ability to exploit parallel computers has consistently enabled it to tackle and solve the most difficult and demanding of problems of the day - real Grand Challenges. This in turn relies heavily on the UK's investment in supercomputing, from the Meiko Computing Surface at the University of Edinburgh in the early days through to the latest UK high-performance computer service, CSAR, at the Manchester Computing Centre.

2.3 CASTEP Now

Since 1993 MSI have developed and consolidated CASTEP as a commercial product, while CETEP remained the workhorse of UKCP academics. However, within the new UKCP-MSI agreement these strengths are combined. CASTEP is now a state-of-the-art plane-wave code, incorporating all the latest and best technology. Key Features include:

- Ultrasoft or norm-conserving pseudopotentials with non-linear core corrections
- Complete pseudopotential library
- Comprehensive range of minimisation methods: Density Mixing, RM-DIIS, Conjugate Gradients, band-by-band and all-bands
- Full structural relaxation and molecular dynamics capabilities
- Local Density and Generalised Gradient approximations, spin-polarisation
- Population analysis
- Supported by 100's of successful, published applications
- Compatible with MSI Cerius² user interface giving automatic setup and full visualisation facilities

2.4 Parallel CASTEP

For much of the functionality available in this release, CASTEP runs in parallel under MPI. However, some modes of execution are serial only. At this release the most important restriction is that the all-bands minimiser is serial only.

Part II

Using CASTEP

3 The basics

3.1 The distribution

The CD contains the following items.

```
POTENTIALS/ baselines/ extras/ quick_test/ utilities/
README      castepexe/  guide   test_cases/
```

The contents of the files and directories are described in Table 1

POTENTIALS	Library of ultrasoft pseudopotentials
README	Text file with basic instructions
baselines	Reference output from the test cases
castepexe	Source code and object directories
extras/	Useful items contributed by UKCP members and CASTEP users
guide/	This document in several formats
quick_test/	Set of input files for a single quick test
test_cases/	Set of input files for the test cases
utilities/	Useful programs and scripts

Table 1: What's in the distribution

3.2 Installation

Copy all the files from the distribution CD to a suitable place on your workstation. For example, on an SGI machine:

```
cd
mkdir CASTEP
cd CASTEP
cp -r /CDROM/* .
```

For various purposes you will need to set the following environment variables:

```
setenv MACHINE <your_machine>
setenv PLATFORM x11
```

where the target `<your_machine>` must match the name of one of the sub-directories `castepexe/obj`, containing a Makefile suitable for your machine.

4 Running CASTEP 4.2

Check if a suitable executable is available in one of the sub-directories under `castepexe/obj`. If there is you are ready to run.

If you are using another type of machine you will have to compile the code yourself. Go to Section 6 before continuing this Section.

To place things in context, take a look at the files in the `quick_test` directory. This contains a copy of one of the test cases, which will be used for illustration.

4.1 The file “seed”

You will see that in `quick_test` there are several files with a common prefix, `BN`. For a given calculation, all CASTEP 4.2 files are preceded by a “seed”, which serves to label the input and output from a job. Here the seed is `BN`. Note that the default seed is `castep`, and typing the executable name alone causes CASTEP 4.2 to look for files with this prefix. Assuming you have a complete set of input files and an executable, the code is run with the command

```
<executable_name> <seed>
```

To run this test case from the `quick_test` directory, type

```
$castep_dir/castepexe/obj/<target>/castepexe BN
```

where `$castep_dir` is the path to the CASTEP 4.2 distribution, and `<target>` is the directory corresponding to your machine, as described above. After execution, the main CASTEP output is in the file `BN.cst`.

4.2 Examining the output

The contents of the `.cst` output file are largely self-explanatory. Firstly, the memory requirements for the job are stated. When running in parallel, the requirements are per node. The next Section of the file contains many of the control parameters and variables used in the calculation. In addition there is information about system geometry, pseudopotentials, k-points, the FFT grid and any advisory or warning messages.

During the calculation, information about convergence or progress is written to this file. For example, `BN.cst` will contain information about the convergence of the electronic minimisation, looking something like this:

SCF loop	Energy (eV)		Energy gain per atom	Timer (sec)	<-- SCF <-- SCF
	Initial	Final			
1	0.00000	-391.75302	195.876511	2.39	<-- SCF
2	-391.75302	-399.35995	3.803465	3.47	<-- SCF
3	-399.35995	-363.30127	-18.029340	4.63	<-- SCF
4	-363.30127	-352.18237	-5.559450	5.72	<-- SCF
5	-352.18237	-352.15629	-0.013041	6.82	<-- SCF
6	-352.15629	-352.16147	0.002588	7.92	<-- SCF
7	-352.16147	-352.16157	0.000054	9.08	<-- SCF
8	-352.16157	-352.16157	-0.000003	9.86	<-- SCF
TOTAL ENERGY IS		-352.1615687 (eV)			

It is obviously essential to check this information to see that the calculation has achieved the desired result. After the calculation is complete other results are written, such as the electronic eigenvalues and stress tensor, as well as the overall timings. In many cases you will probably want more information: change the value of IPRINT in the `.rundat` file to print in greater detail.

4.3 Types of calculation and associated files

There are three main types of calculation possible with CASTEP 4.2:

Single-point energy

Here the ground-state energy is calculated for fixed ionic positions. The BN calculation in `quick_test` is a single-point energy calculation. The calculation also yields the Khon-Sham eigenvalues at the set of k-points considered, the ionic forces and the stress tensor (if selected).

Geometry optimisation

Almost always one requires the total energy of a system in which the structural parameters have been optimised. This may involve finding the lattice vectors that minimise the total energy (cell optimisation), or relaxing the structure by moving the ions to positions of zero force, or both.

Molecular dynamics

Molecular dynamics (MD) involves the stepwise integration of Newton's equations from a given starting point. It is the most natural method of performing equilibrium statistical-mechanical calculations via simulation. Additionally, MD may be used to explore conformations, such as molecular

adsorption sites on a surface. If frictional forces are added to the equations of motion, Langevin dynamics becomes possible, as does geometry optimisation *via* damped MD. Finally, extended Lagrangian methods allow thermostatic control of the temperature.

4.4 Input files

The mandatory input files are:

`<seed>.ewdata` Tabulated data used for the Ewald summation. This file is identical for all calculations.

`<seed>.geom` Contains the geometry and k-point information.

`<seed>.param` Parameters for memory assignment.

`<seed>.recpot` The pseudopotentials in reciprocal space.

`<seed>.rundat` The switches and variables that control CASTEP execution.

Several files must be present only if certain options are selected in the `.rundat` file. These are:

`<seed>.optim` Contains the constraints and the initial Hessian. Needed if using BFGS geometry optimisation (`IION=3` and/or `IBOX=2`).

`<seed>.realpot` Contains the real-space non-local pseudopotentials. Needed if evaluation is in real space.

`<seed>.symm` Contains the symmetry operations of the structure described in `<seed>.geom`. Needed if using symmetry information (`ISYMM = 1`).

`<seed>.constr` Contains the constraints imposed during MD, i.e. when `IDYNAM=1` and `NEW_CONS = 1`.

4.5 Output files

The number of output files produced by CASTEP 4.2 varies according to the type of calculation performed. All classes of calculation produce the following files:

`<seed>.cst` The output file containing details of the calculation, convergence information and any runtime analysis.

`<seed>.gm*` Geometry and eigenvalues.

`<seed>.charge*` The electronic charge.

`<seed>.wavfun*` The wavefunctions.

`<seed>.cst_esp` The local potential.

`<seed>.cham` The Hamiltonian matrix.

Additional files are written if certain options are chosen.

`<seed>.occnum*` Contains the occupation numbers, written when performing calculations on metals.

`<seed>.bands` The electron bands. Written if applying subspace rotation (`ISBROT` $\neq 0$) or performing a bandstructure calculation.

`<seed>.cst_dos` Density-of-states. Written if “enough” k-points are in use.

Files whose names are marked with an asterisk (*) are the binary files which must be present to re-start a calculation (see Section 4.6).

Both MD and geometry optimisation calculations generate additional output files. Both write these two files:

`<seed>.coord` Sequence of ionic co-ordinates.

`<seed>.force` Sequence of ionic forces.

Geometry optimisation calculations write these additional files:

`<seed>.optim_out` Like `<seed>.optim` but with the final Hessian.

`<seed>.cell` Evolution of the cell ($a, b, c, \alpha, \beta, \gamma$), internal and external stress tensors. Also written if the stress tensor is calculated (`ISTRESS` = 1)

MD calculations also produce these files (and others to do with wavefunction extrapolation)

`<seed>.veloc` Sequence of ionic velocities ($\text{\AA}/\text{ps}$)

`<seed>.temper` Temperature (K)

`<seed>.hamilt` MD constant of motion (eV)

`<seed>.snose` Nosé thermostat s

`<seed>.sderiv` Velocity of Nosé thermostat s

Several other intermediate and output files may be written depending on the settings in the `.rundat` file.

4.6 Restarting a calculation

Calculations may be restarted from the binary output files. The files required are:

`<seed>.gm` Geometry and eigenvalues.

`<seed>.charge` The electronic charge.

`<seed>.wavfun` The wavefunctions.

`<seed>.occnum` Contains the occupation numbers,

In addition, you must change the value of `ISTART` to 1 in the `.rundat` file. Note carefully that **CASTEP 4.2** will both read from and write to these files. Therefore, if you wish to keep intermediate results, such as charge densities, you must copy the relevant files before making a restart.

The other output files, including the `.cst` file, are treated differently. A restarted job will append its output to these files.

You may wish to stop a calculation without abandoning it, with the restart files written before the program stops “gracefully”. You can do this by creating a file `<seed>.runinf` in the working directory. At the end of each iteration (as defined by `NITER` in `.runinf`) **CASTEP 4.2** checks this file. You can request a clean stop by entering the value “1” on the first line of the `.runinf` file (actually, this overrides the value of `NITER`). As part of the shutdown, **CASTEP 4.2** will remove the `.runinf` file.

4.6.1 MD restarts

To restart an MD run both the `ISTART` and `ISTDYN` switches in `.rundat` must be set to 1. If `ISTDYN = 0` upon restart, the ionic velocities are assigned random values as for a new run. `NITER` should be set to the number of steps required in the restarted run. Note that **CASTEP 4.2** will perform an extra electronic minimisation on restart. This extra calculation (`ITERATION 0`) results in duplicate entries in the “trajectory” files (`.coord`, `.temper` etc.); the results of the calculation are appended to these files.

The wavefunction extrapolation files `.drhom`, `.drho2m`, `.wfm` and `.wf2m` are not required on restart.

4.7 Similarities between CASTEP 4.2 and CETEP files

If you are familiar with CETEP it is worth knowing that there are distinct relationships between **CASTEP 4.2** and CETEP input files. These are sum-

marised in Table 2. Examination of the subroutines in the `cst_inp_geom.f` file should enable you to cross-match precisely the two sets of files.

CETEP	CASTEP 4.2	Remarks
<code>fort.10</code>	<code><seed>.ewdata</code>	Identical.
<code>fort.11</code>	<code><seed>.recpot</code>	“Non-local” projector for the local potential not necessary in CASTEP file.
<code>fort.14</code>	<code><seed>.rundat</code>	Many parameters and variables in common, but formats differ.
<code>fort.18</code>	<code><seed>.realpot</code>	Identical.

Table 2: Relationships between CASTEP 4.2 and CETEP files

5 Running the test cases

Several test cases are included in the `test_cases` directory. The major options selected and functionality used for each test case is summarised in table 3. It is strongly recommended that you run all the test cases to validate your copy of the code and its installation. In addition, they provide a range of examples of calculations that are possible with CASTEP 4.2 . You may run any of the cases as described in Section 4.1.

Some simple scripts are provided with the test cases. The cases are split into small and big calculations, and scripts are provided to run these either in serial or parallel. The scripts run the test cases and `diff` their output with that contained in the `baselines/` directory. The output of the script is directed to a file of your choice. To use a serial script, type

```
run_small_serial <output_file_name> <executable_name>
```

For example, this might be

```
run_small_serial test.out $castep_obj_dir/castepexe
```

where `$castep_obj_dir` is the path to the CASTEP 4.2 object directory. The parallel scripts take two additional arguments:

```
run_small <output_file_name> <executable_name> <num_procs> <distribution>
```

Here, `<num_procs>` and `<distribution>` are the number of processors and the parallel distribution (K, G, or GK) respectively (see Section 6.7).

The script `clean_all` removes all output files generated by running the test cases, except the `.cst` files.

The `silicon*` test cases read files containing the charge density (`.charge`)

#	Sys	Sym	Fn	Mth	BSC	Ions	box	MD	PP	Space
1a	Si		LDA	0	•	SD			NL	k
1b	Si		GGA	0	•	SD			NL	k
1c	Si		GGA	0	•	BFGS			NL	k
1d	Si		GGA	1	•	BFGS			NL	k
1e	Si		GGA	0				•	NL	k
2a	Si	•	GGA	0	•				NL	k
2b	Si	•	GGA	0	•				NL	r
2c	MgCl		GGA	0		BFGS			NL	k
2d	Al	•	LDA	0	•		•		NL	k
3a	PuO2	•	GGA	0					NL	k
3b	PuO2	•	GGS	0					NL	k
3c	PuO2	•	GGG	0					NL	r
4a	Fe	•	LSDA	1					NL	k
4b	Fe	•	GGG	1					NL	k
5a	BN	•	GGA	2	•		•		US	k
5a BS	BN	•	GGA	0	•		•		US	k
5b	BN	•	GGA	1	•		•		US	k
5c	BN	•	GGA	2	•		•		NL	k
6a	BN	•	GGG	2	•		•		US	k
6a BS	BN	•	GGG	0	•		•		US	k
6b	BN	•	GGG	1	•		•		US	k
6c	BN	•	GGG	2	•		•		NL	k
7a	Pt		GGA	2					US	k
7b	Pt/Al		GGA	2					US/NL	k
DOS	Si		GGA	0					NL	k
BS	Si		GGA	0					NL	k
OPT	Si		GGA	0					NL	k
8a	TiO ₂	•	GGG	2					US	k
9a	N ₂		LDA	2		CG			US	k
9b	N ₂		LDA	2					US	k
9c	N ₂		LDA	2					US	k
9d	N ₂		LDA	2		Damp			US	k
9e	N ₂		LDA	2					US	k
9g	N ₂		LDA	1					US	k
9h	N ₂		LDA	2					US	k
9i	N ₂		LDA	2					US	k
9j	N ₂		LDA	1		Damp			NL	k
9k	N ₂		LDA	0					NL	k
10a	Al		GGA	2					US	k
10b	Al		GGA	2					US	k
11	CuC ₂ H ₄		GGG	2					US	k
12	Ru		GGA	2	•		•		US	k
13	N		LDA	2			•		US	k
14	CO:Cu		GGA	2		BFGS	•		US	k
15	Li:TiO ₂		GGG	2			•		US	k
16	Si		GGA	2		Damp	•		US	r
17	Chab		GGA	2			•		US	k
18	Pb	•	GGA	2		BFGS	•		US	k
19	MgSiO ₃	•	GGA-PW91	2					US	k
20	FeS ₂	•	GGA-PW91	2					US	k
21	TiN		GGA-PW91	2					US	k

Table 3: The test cases. The headings are (with `.rundat` parameters in parenthesis): #, the test case number (with BS, DOS and OPT denoting band structure, densities-of-states and optics, respectively); Sys, the system; Sym, whether symmetry is exploited (ISYM); Fn, the exchange-correlation functional used (Theory level); Mth, the method used to minimise the electronic energy (METHOD); BSC, basis-set correction computed (BAS_CORR); Ions, ionic relaxation (IION); box, cell relaxation (IBOX); MD, molecular dynamics; PP, the pseudopotential type, either non-local or ultrasoft (IVPTYP); Space, how the pseudopotential is evaluated, in real (r) or reciprocal (k) space (NPOT).

and geometry (`.gm`) from a previous calculation. Since these are binary files they are machine-specific. The files provided are SGI binaries.

5.1 Checking

Remember that there are bound to be small differences in the answers when the code is run on different machines. When checking the differences pay particular attention to total energies. These should agree to within the tolerance specified for the calculation. If they agree for all the test cases you may assume that everything is working properly.

The spin-polarised test case calculations need a special mention. Because these systems have multiple energy minima, it is possible that slight differences in initial conditions, caused by execution on different machines, will result in the location of different minima. This may arise, for example, if the sets of random numbers used to initialise the wavefunctions differ. This applies to test cases 3b, 3c, 4a-b and 6a-c. Comparison of the various baselines for these cases shows that the answers are somewhat machine-dependent. Therefore, if you compare your results for these cases with the baselines from a different machine, you may observe discrepancies.

6 Compiling CASTEP 4.2

Academic CASTEP 4.2 is distributed ready-compiled for several serial and parallel platforms. Here is some information to help with compilation on any platform, though please note that at present Daresbury can only offer full support for CASTEP 4.2 running on the Cray T3E at CSAR in Manchester. You may wish to use the shared executable mounted there, which is compiled from the most up-to-date code. Send a note to ukcp@dl.ac.uk to request access.

6.1 Software resources, libraries

Serial CASTEP 4.2 requires few ancilliary software packages, and should work on an “average” workstation. Compilation requires both FORTRAN90 and C compilers. There is scope for the use of the `blas` and `lapack` libraries in CASTEP 4.2, though these are not essential since software versions of the relevant routines are included in the distribution. However, the parallel version of the code does rely on an MPI library being present.

6.2 Targets

The directory `castepexe/obj` contains several sub-directories for different machines:

<code>Fujitsu_VPP.x11</code>	<code>irix32.x11</code>	<code>linux86_pgi_lammpi.x11</code>
<code>crayt3e.x11</code>	<code>irix6464.x11</code>	<code>linux86_pgi_mpich.x11</code>
<code>decalpha.x11</code>	<code>irix6464mpi.x11</code>	<code>solaris.x11</code>
<code>fujitsu_vpp.x11</code>	<code>irix64mpi.x11</code>	<code>solaris_mpich.x11</code>
<code>ibmrios.x11</code>	<code>irix64n32.x11</code>	<code>sparcv9.x11</code>
<code>irix.serial</code>	<code>linux86_lahey.x11</code>	

Each sub-directory contains a Makefile for the machine, and an executable compiled using that Makefile.

CASTEP 4.2 will compile on the machines for which there is a subdirectory to `obj/` (In fact, it will compile on others too but you would have to do a little more work.) Assuming your machine is covered, set the following environment variables if you have not done so already.

```
setenv MACHINE <your_machine>
setenv PLATFORM x11
```

Then, to (re-)compile CASTEP 4.2, run the script `castepexe/Make`. The script soft-links the source files to the appropriate directory, compiles and links them to produce `castepexe`. Note that although compilation may well generate warnings, these should be benign.

6.3 Compiler Options

Here is a list of some of the compilation options which have been found to be useful, and references to further information on the compilers. Other useful information can be found from various Web sites on which benchmark results are published, e.g. SPECfp95, see <http://www.specbench.org>. In the main, our default compilation options are similar to the SPECfp95 baseline options. Some further performance may however be possible in certain cases, but careful experimentation is required.

Note that we only reproduce below differences from the template makefile given in [?].

6.3.1 Absoft Fortran v1.01 on Linux Pentium Platforms

Options:

```
CPP= gcc -E
F90= f90 -c
```

The compiler requires free-form source to be in files with extension `.f90`. If modules (with the `.mod` extension) are present in another directory the `-p` flag should be used. However there was a bug which prevented this working correctly in the version tested.

See `www.absoft.com`.

For use on parallel systems, e.g. Beowulf clusters, either the LAM-mpi from Ohio State University or MPICH from Argonne National Laboratory may be installed. We will assume they are installed in a standard place such as `/usr/local/mpich` or `/usr/local/lam`.

Linking with LAM-mpi

```
LIBS= -L/usr/local/lam/lib -lmpi -largS -ltstdio -ltrillium -lt -lblas
```

Linking with MPICH

```
LIBS= -lpgftnrtl -L/usr/local/mpich/lib -lmpich -lmpichf -lblas
```

6.3.2 Compaq

Software tested on Columbus.rl.ac.uk, a Jura v1.0_050 system installed on 10/5/2000 running Digital UNIX V5.0 Rev. 910. Compaq Fortran V5.3-915 and Compaq C V6.1-015.

A serial install was also done on a Compaq DS20 tca16.dl.ac.uk Compaq Tru64 UNIX V5.0A Rev. 1094. Compaq Fortran V5.3-915 and Compaq C V6.1-019.

```
CC= cc -c -std
F90=f90 -c -free
F90FLAGS= -O5 -fast -arch ev6 -tune ev6 -recursive -non_shared
# enable parallelism
PFLAGS=-omp
# OpenMP, threads and DMXL optimised library if available
LIBS=-lpthread -ldmxlp -lmpi -lelan
```

The `-lelan` is only required for the QSW switch Elan driver.

6.3.3 Cray

Software tested on turing.cfs.ac.uk a Cray T3E-1200E running unicosmk 2.0.5.33. Cray CF90 Version 3.3.0.2 and Cray Standard C Version 6.3.0.2 with PrgEnv.33,craylibs.3.4.0.1 and mpt.1.4.0.0.

```
CPP= f90 -eP
CPPFLAGS= -Wp'-P'
F90= f90 -c -ffree
```

```
F90FLAGS= -dp -O3,unroll2,pipeline2
LDFLAGS=blas.cld
LIBS=
```

Note that the SciLib library, which contains BLAS, LAPACK and ScaLAPACK routines and also the SHMEM and MPI libraries are loaded automatically. We also note that Cray consider 64-bit arithmetic to be the default and all their BLAS and other numerical routines are supplied with “single precision” names, such as SAXPY. These actually apply to what are more commonly referred to as “double precision” data. Thus a code which calls DAXPY will not work on the Cray. Calling names of routines can however be changed in the link step by supplying a file “blas.cld” along with any libraries required. This file contains a list of names with the format:

```
equiv(DAXPY)=SAXPY
```

6.3.4 Fujitsu

Options:

```
CC= ccpx -c
CFLAGS= +p -DNDEBUG -DFUNCPROTO
F90= frptx -c
F90FLAGS= -Kfast,V8PFMADD,prefetch,gs -Oe -KVPP700 -Wv,-m1 -X7 -Pf -Sw
-Elmipue -Z $*.list
LD= /usr/local/bin/vppld -o
LDFLAGS= -Wl,-J,-P -dn
LIBS= -L/usr/lang/mpi2/lib -lmpi -lmp -lpx -lcvp -lgen -lssl2vp
```

6.3.5 Lahey

6.3.6 HP

Options:

```
F90= f90 -c
F90FLAGS= +Oall +Odataprefetch -Wl,+pd,64K
LDFLAGS= -Wl,-aarchive
```

6.3.7 Hitachi

Options:

```
F90= xf90 -c
F90FLAGS= -W0,'opt(o(3)),langlvl(hf(77),save(0))'
```

6.3.8 IBM Fortran v6.1 and v7.1

Software tested on splogin2.dl.ac.uk running AIX 4.3. Xlf Fortran compiler 6.1.0.0 and vac C compiler 4.4.0.0 with ppe.poe 2.4.0.11.

```
F90= mpixlf_r -c -qfree=f90
F90FLAGS= -O4 -qhot -qipa=level=2:partition=large -qarch=pwr3
-qtune=pwr3 -qnosave
# switch on processing of OpenMP directives
PFLAGS= -qsmp=nested_par
INFO= -qreport=smpelist
LD= mpixlf_r -o
LDFLAGS= -qipa=level=2:partition=large -qsmp -bmaxdata:2000000000
# optimised IBM library if available
LIBS= -lesslsmpl
```

We found that the `-qreport=smpelist` flag produced internal errors with some codes. We also found that the `-qipa` option was not reliable and produced incorrect code in some rare cases. Explicit code inlining could be used instead via `tt -qipa=inline=xxx` where `xxx` is a list of files to inline.

6.3.9 NEC

6.3.10 Pacific Sierra Research

PSR offer a F90 compiler for Linux systems, a free version is available for personal use. See www.psrv.com/lxf90.html.

6.3.11 Portland Group Compiler on Linux Platforms

Software tested on a Pentium III 450 MHz Beowulf, beowulf1.dl.ac.uk running Red Hat Linux v6.1. Portland Group Fortran compiler pgf90 3.1-3 and C compiler pgcc 3.1-3 with MPICH 1.2.0 and LAM 6.3.1.

See www.pgi.com.

Linking with LAM-mpi compiled with the GNU compilers

```
F90= pgf90 -c -Mfree
INCLUDES= -I/usr/local/lam/include
F90FLAGS= -fast -Knoieee -Malign -Msecond_underscore
LD= pgf90 -o
LIBS= -L/usr/local/lam/lib -lmpi -largS -ltstdio -ltrillium -lt -lblas
```

Linking with MPICH compiled with the PGI compilers

Note that if MPICH is compiled with the GNU compilers it will not work with a PGI F90 application.

```

CPFLAGS= -traditional -P
CC= pgcc -c
F90= pgf90 -c -Mfree
INCLUDES= -I/usr/local/mpich/include
F90FLAGS= -fast -Knoieee -Mdalign
LD= pgf90 -o
LIBS= -lpgftnrtl -L/usr/local/mpich/lib -lmpich -lmpichf

```

For linking to other libraries compiled with the GNU compiler set there is a compatibility flag to include on the LDFLAGS line: `-lg77libs`. This searches for the library `/usr/lib/libf2c.a` which should be a link to `/usr/local/gnu/i386-redhat-linu8/egcs-1.66.99/pibg2c.a`

6.3.12 SGI

Software tested on `fermat.cfs.ac.uk`, a 10-processor Origin2000 ip27 system running IRIX64 v6.5. Fortran 90 and C are MIPSpro Compilers Version 7.2.1 with `mpt.1.3`.

```

F90= f90 -c -freeform
F90FLAGS= -Ofast -OPT:0limit=0 -LNO:prefetch_ahead=1:auto_dist=on
# swith on automatic parallelisation
PFLAGS= -pfa -mplist
LDFLAGS= -mp
LIBS= -lblas -lmpi -lmp -lmpc -lftn -lbsd -lm

```

6.3.13 SUN

Software tested on a SUN sun4u sparc SUNW,Ultra-5_10 `tcs7.dl.ac.uk` running SunOS 5.7 (Solaris). WorkShop Compilers 5.0 99/06/24 FORTRAN 90 2.0 Patch 107356-02 and WorkShop Compilers 5.0 98/12/15 C 5.0 with MPICH 1.2.0.

```

F90= f90 -c
INCLUDES= -I/usr/local/mpich/include
F90FLAGS=-fast -xprefetch -dalign
PFLAGS= -xparallel
LD= f90 -o
LIBS= -L/usr/local/mpich/lib -lmpich -lmpichf -lnsl -lsocket -xlic_lib=sunperf

```

The loader reference `-xlic_lib=sunperf` links in the optimised SunPerf library if available.

As an alternative to MPICH the SunWhpc library MPI could be used with

```

LIBS= -L/opt/SUNWhpc/lib -lmpi -Bstatic -L/opt/SUNWspro/lib -lfsu
-lfui -lfai -lfai2

```

Beware that some SUN systems have a number of different compilers and

libraries installed. Fortran 90 compilers prior to SC5.0 do not perform well, and the library should be chosen compatible with the chipset.

6.4 Serial or parallel?

The most important choice you need to make at compile time is whether to compile for a serial or a parallel machine. The same source code works for both, but must be compiled appropriately, and that means either with or without the MPI library and the necessary pre-processing. For some machines we have provided both serial and parallel target directories. If you are not using one of these, and you are adapting a parallel Makefile, change the `-DMPI CPP` directive to `-DSERIAL`.

6.5 SGI shared-memory machines

The code will also execute in parallel on SGI shared-memory multiple-processor machines such as the Origin series. (This shared-memory execution is different than MPI execution.) On such machines, several costly parts of the calculation are shared among the processors, achieving some degree of parallelism. The environment variable `MP_SET_NUMTHREADS` must be set, for example:

```
setenv MP_SET_NUMTHREADS "min( 4,all)"
```

will select the smaller number from 4 and the total number of processors on the machine. Note that this shared-memory strategy is not a massively-parallel one, and useful speedup will only be achieved on a small number of processors.

6.6 Running on parallel machines with MPI

Running the code in parallel is essentially the same as for the serial code, except that the number of processors must be specified either on the command line or in a batch script. A typical batch script for the T3E might look like this:

```
#QSUB -r castep_job
#QSUB -q normal
#QSUB -A <your_account>
#QSUB -l mpp-p=64
#QSUB -l mpp-t=2:00:00
cd CASTEP/bigsystem
mpirun -np 64 /CASTEP/castepexe/obj/crayt3e.x11/castepexe si_surface
```

6.7 MPI-Parallel distribution

The largest data structures that CASTEP 4.2 deals with are the wavefunction and associated quantities, such its derivatives. The size of such a structure is proportional to

1. The number of k points
2. The granularity of the FFT grid (i.e. the number of plane-waves employed)
3. The number of bands
4. A factor related to the energy cutoff ($\sim 1/16$ or greater)

For example, a large calculation on a metal surface may employ 10 k points, a $50 \times 50 \times 200$ FFT grid and 700 bands, a total of about 3.2 Gigabytes, so it is imperative that on a distributed memory architecture data structures of this size are distributed among the processors. CASTEP 4.2 can perform this distribution in three different ways:

1. **By k-point:** Each processor stores the whole wavefunction for a subset of the required k points. For example if calculation that uses 8 k points is run on 8 processors, each processor would work on a single k point. This distribution is best when the number of k points is large, as might be required for a metal. This distribution involves very little inter-processor communication, and is therefore very efficient and scalable.
2. **By G-Vector:** Each processor stores a part of the wavefunction for all the k points. Thus, for a 2 k-point calculation on 8 processors, each processor would store 1/8 of the wavefunction at each k point. (This strategy will be familiar to CETEP users.) This approach works best for large supercells with few k points, but does require appreciably more inter-processor communication than the k-point strategy.
3. **By k-point and G-Vector:** This is a hybrid of the previous two methods, best explained by example. Consider the same calculation as in the G-vector case. Here, initially, the k points are distributed amongst the processors, in this case the first four processors get k point 1 and the second four k point 2. Then, for a given k-point, the wavefunction is distributed, as in the G-vector case, amongst the processors dealing with that k point.

You need not concern yourself with which distribution to use since the code will automatically choose a suitable one. However, the CASTEP 4.2 executable takes a second argument which forces one of the distributions:

```
castepexe <seed> <distribution>
```

where <distribution> is one of G, K or GK.

It is possible to re-start a calculation on a different number of nodes and with a different distribution strategy than was used originally.

6.8 Replicated data

Some data structures are not distributed among the nodes during parallel execution. This is because either it is not necessary, for example because they are very small, or because their parallel requirements are in conflict with the distributions described above. This is known as *replicated data*, and it can become a major limitation for large systems. This is because increasing the number of nodes will not reduce the memory required for the replicated data. The worst culprits are the arrays

```
cbec ( $N_B, N_\beta, N, N_{\mathbf{k}}N_{spin}$ )
cbecw3 ( $N_B, N_\beta, N, N_{\mathbf{k}}N_{spin}, 3$ ) ,
```

where N_B is the number of bands, N_β the number of ultrasoft projectors, N the number of ions and $N_{\mathbf{k}}N_{spin}$ the product of the numbers of \mathbf{k} -points and spins. For the metal surface example just given these arrays take several hundred Megabytes *per node*. In CASTEP 4.2 they are parallelised over \mathbf{k} -points, but even so one can appreciate the potential difficulties.

As already mentioned, the MPI code is currently under development, and although it is not a “beta” release, you may encounter some bugs. If you do, please let us know.

7 Detailed description of some CASTEP 4.2 files

In this Section the key CASTEP 4.2 files are described with reference to the files in the `quick_test` directory. Those files relate to the unit cell of Boron Nitride, which contains one B atom and one N atom.

CASTEP 4.2 relies on the information about different species appearing in the same order in the input files. For example, this means that the pseudopotentials for B and N appear in that order in the `BN.recpot` file, the positions of B atoms precede those for N atoms in `BN.geom`, and the specifications of core charge, numbers and masses in `BN.rundat` are in the same order.

7.1 Geometry Information: `.geom`

The file contains geometry information: cell parameters, atomic coordinates, k-points coordinates and weights. The whole file looks like this:

2.0871212231	0.0000000000	0.0000000000	
0.7379087850	2.2137263550	0.0000000000	
1.2780955070	1.2780955070	2.5561910140	
2.0871212231	0.0000000000	0.0000000000	
0.7379087850	2.2137263550	0.0000000000	
1.2780955070	1.2780955070	2.5561910140	
0.0000000000	0.0000000000	0.0000000000	0.0
0.0000000000	0.0000000000	0.0000000000	
0.2500000000	0.2500000000	0.2500000000	0.0
0.2500000000	0.2500000000	0.2500000000	
0.4000000000	0.4000000000	0.4000000000	
0.4000000000	0.4000000000	0.2000000000	
0.4000000000	0.4000000000	0.0000000000	
0.4000000000	0.4000000000	-0.2000000000	
0.4000000000	0.4000000000	0.6000000000	
0.4000000000	0.2000000000	0.2000000000	
0.4000000000	0.2000000000	-0.2000000000	
0.2000000000	0.2000000000	0.2000000000	
0.2000000000	0.2000000000	0.0000000000	
0.0000000000	0.0000000000	0.0000000000	
0.0640000000			
0.1920000000			
0.0480000000			
0.0960000000			
0.1920000000			
0.0960000000			
0.1920000000			
0.0640000000			
0.0480000000			
0.0080000000			

The first six lines are two sets of lattice vectors (in Å):

```

2.0871212231    0.0000000000    0.0000000000
0.7379087850    2.2137263550    0.0000000000
1.2780955070    1.2780955070    2.5561910140
2.0871212231    0.0000000000    0.0000000000
0.7379087850    2.2137263550    0.0000000000
1.2780955070    1.2780955070    2.5561910140

```

The format is columnar:

```

Ax           Bx           Cx
Ay           By           Cy
Az           Bz           Cz

```

Usually the two sets of lattice vectors are identical. For historical reasons it is possible to attempt using output files from one run to “restart” for a slightly different cell. In this case the first three lines give the new lattice vectors, and the next three give the old ones. This functionality is no longer supported.

The next section of the file contains the fractional positions of the ionic cores. In this example there are two species, B and N, and one atom of each species. The coordinates are sorted by species, and again for historical reasons, the set of positions for the ions in a given species appears twice.

```

0.0000000000    0.0000000000    0.0000000000    0.0
0.2500000000    0.2500000000    0.2500000000    0.0
0.2500000000    0.2500000000    0.2500000000

```

The line

```

0.0000000000    0.0000000000    0.0000000000    0.0

```

is the fractional coordinates of the atom of species 1, which is boron. The last number is the moveability switch: 0 - fixed, 1 - moveable atom. The number of lines for each species is determined by the NIONSP parameter in the `.rundat` file. The coordinates of this atom are repeated on the next line:

```

0.0000000000    0.0000000000    0.0000000000

```

This format is repeated for all species, so that in this example the fractional coordinates of the atom of species 2, nitrogen, appears next:

```

0.2500000000    0.2500000000    0.2500000000

```

After this coordinates section there might be a section with atomic velocities (only if the task is Molecular Dynamics). Velocities would be ordered by ion within species, one line of three numbers per atom. Note that velocities are with reference to the Cartesian frame ($\text{\AA}/\text{ps}$).

The last part of the file contains the fractional coordinates of the k-points

that are used for sampling the Brillouin zone. The number of lines is the same as the NKPTS parameter in the `.param` file.

```

0.4000000000    0.4000000000    0.4000000000
0.4000000000    0.4000000000    0.2000000000
0.4000000000    0.4000000000    0.0000000000
0.4000000000    0.4000000000   -0.2000000000
0.4000000000    0.4000000000    0.6000000000
0.4000000000    0.2000000000    0.2000000000
0.4000000000    0.2000000000   -0.2000000000
0.2000000000    0.2000000000    0.2000000000
0.2000000000    0.2000000000    0.0000000000
0.0000000000    0.0000000000    0.0000000000

```

The weights of the k-points, in the same order as the coordinates, appear last.

```

0.0640000000
0.1920000000
0.0480000000
0.0960000000
0.1920000000
0.0960000000
0.1920000000
0.0640000000
0.0480000000
0.0080000000

```

7.2 Control parameters: `.rundat`

This file contains settings and switches for various operation modes, plus certain data specific to atomic species in the model. The whole file looks like this:

3	!	NITER	! Number of iterations
1	!	NPRINT	! Forces are printed every NPRINTth iteration
300.00000	!	ENMAX	! Kinetic energy cut-off (eV)
0	!	NPOT	! Potential: 0-reciprocal, 1-real space
5	!	NITMAX	! Number of electronic CG steps
1	!	NDELAY	! Dummy starting from version 2.0
0	!	ISTART	! Mode: 0-new run, 1-restart
1	!	ISBROT	! Rotation: 0-NO,1-strict, 2-fast METALS only
0	!	IOCCUP	! Partial occupation: 0-no, 1-yes
0.20000	!	DELMIN	! Minimum smearing width (eV)
4.00000	!	DELMAX	! Starting smearing width (eV)
6	!	NDEL	! Smearing width is halved every NDEL SCF loops
0	!	IION	! Ionic relaxation: 0-no, 1-CG, 2-simple, 3-BFGS, 4-damped MD
0	!	IBOX	! Unit cell relaxation: 0-no, 1-uniform, 2-BFGS
0	!	IPRINT	! Amount of output
1	!	NIONCG	! Number of CG steps for ions
1	!	NITFIX	! Keep 1 (sort of dummy)
1	!	INRAND	! WFs initialization: 0-diamond 8at. 1-random
1	!	ICLOCK	! Timer: 0-off, 1-on
1.00000	!	SITIM	! Dummy (unit cell time step)
0.20000	!	SIDAMP	! Dummy (unit cell damping)
100.00000	!	SIMASS	! Dummy (unit cell mass)
0.01000	!	SIDISP	! Dummy (allowed unit cell change)
0.00020	!	POTIM	! Timestep for ionic dynamics (ps)
0.05000	!	PODISP	! Maximum displacement of ions
3	!	ICHARG	! The charge on atom species 1 (B)
5	!	ICHARG	! The charge on atom species 2 (N)
10.81000	!	POMASS	! The mass of atom species 1 (B)
14.00670	!	POMASS	! The mass of atom species 2 (N)
1	!	NIONSP	! The number of atoms of species 1 (B)
1	!	NIONSP	! The number of atoms of species 2 (N)
2	!	IVPTYP	! Type of potential(0-L,1-NL,2-USP)species 1 (B)
2	!	IVPTYP	! Type of potential(0-L,1-NL,2-USP)species 2 (N)
2001	!	NPSPTN	! Number of rec.space data points, species 1 (B)
2001	!	NPSPTN	! Number of rec.space data points, species 2 (N)
1.15734	!	SCAL	! Dummy
1	!	ISYMM	! Symmetrization switch: 0-off,1-on
0	!	IBANS	! Band structure mode: 0-no,1-yes
0	!	ICNSTR	! External constraints: 0-no,1-yes
0	!	IDYNAM	! MD mode: 0-off,1-on
0.0000050	!	EPSIL	! Convergence tolerance (eV/at)
5	!	NEXTWR	! Backup every NEXTWRth iteration
0	!	ISTDYN	! For MD: 0-new run, 1-restart
100	!	NPREC	! 2 extra electronic steps every NPREC iterations
1	!	ISTAT	! For MD: 0-NVE, 1-Nose NVT, 2-Langevin NVT
0.10000	!	QNOSE	! Nose thermostat parameter
273.00000	!	TEMPER	! Temperature (K)
1	!	NVEL	! Velocity initialization (0-file,>0-random)
0	!	NMDCG	! Keep 0 (sort of dummy)
CASTEP calculation	!		! comment
1	!	ISTRESS	! Stress output : 0-no, 1-yes
GGA	!		! Theory level
2	!	METHOD	! 0-band_by_band, 1-all_bands, 2-density mixing
20	!	ISEED	! 0-random, otherwise-pseudorandom wavefunctions
0	!	IWRCHD	! If nonzero, formatted charge density is created

Continued...

```

30      !MAX_CYCL! Maximum number of SCF cycles at each step
1       !BAS_CORR! Finite sampling correction: 0-no, 1-yes
0       !MODE_COR! Mode of sampling correction: 0-auto, 1-manual
0.00000 !DE_DlogE! Derivative of Etot w.r.t. log(Ecutoff)
0.00000 !PRESSURE! External pressure (GPa) and stress (next line)
0.000   0.000   0.000   0.000   0.000   0.000
3       !NUM_CUT ! Points for num. diff. for DE_DlogE
288.00000 ! ECUT  ! Cutoff # 1 for sampling correction (eV)
294.00000 ! ECUT  ! Cutoff # 2 for sampling correction (eV)
300.00000 ! ECUT  ! Cutoff # 3 for sampling correction (eV)
0.00100 !DISP_TOL! Tolerance for RMS displacement
0.05000 !RMSF_TOL! Tolerance for RMS force
0.10000 !RMS_STR ! Tolerance for RMS stress
0.0000200 !ETOT_TOL! Minimizer tolerance for total energy
0       ! OPTICS ! Calculate matrix elements: 0-no, 1-yes
4       !NEL_UP  ! Number of electrons with spin up
4       !NEL_DOWN! Number of electrons with spin down
0       !SPIN_FIX! Spin is fixed (0) or varied
0       !INIT_OCC! Init. occupancy: 0-lowest, 1-all,-1-file
0.20000 ! ROTMIX ! Mixing factor for WF rotation
0.50000 ! OCCMIX ! Mixing factor for occupation numbers
0.50000 !EL_STEP ! Step for all-bands minimizer
7       !MIX_HIST! Density mixing history (0:non-Pulay)
0.80000 ! CH_AMP ! Mixing amplitude for charge
1.50000 ! CH_Q   ! Cutoff wavevector for charge
2.00000 !SPIN_AMP! Mixing amplitude for spin density
1.50000 ! SPIN_Q ! Cutoff wavevector for spin density
0       !RM_DELAY! CG steps before RMM starts (0 is CG)
2       !DENS_INI! Use starting atomic densities (2)
0       !MULLIKEN! Mulliken analysis: 0-no, 1-yes
3.00000 ! CUTOFF ! Cutoff for bond order calculation
4       ! NORBS ! Number of atomic orbitals, species 1 (B )
4       ! NORBS ! Number of atomic orbitals, species 2 (N )
0       ! NPDOS ! PDOS set: -1:all,0:none,>0 - see below
1       ! IEXTRAP ! Wavefunction extrapolation method (0,1,2,21)
0       ! IDAMP  ! Auto-damping calculation for damped MD (iion=4)
30      !RECALC ! MD timestep between idamp recalculation (iion=4)
0.050  !TDAMP  ! Langevin damping time-constant [ps] (istat=2)
0       !NEW_CONS! New linear constraints (1) in MD

```

CASTEP 4.2 does not read the comments part of the lines in this file, indicated by a !. They are included as a brief reminder of the purpose of the parameters and variables. Many of the names used in the comments are the same as the data names in the code itself. CASTEP 4.2 relies on the fixed order of lines in this file, and on the type of the number on a line, i.e. floating point or integer, being correct. This means that certain variables are purely historic and have no meaning in the current version (dummy variables). Dummy variables will not be discussed further.

The meaning and purpose of each variable is as follows.

```

3       ! NITER  ! Number of iterations

```

Iteration has the meaning of a complete self-consistent calculation of the electronic wavefunctions and charge density for a fixed atomic geometry. `NITER` specifies the maximum number of iterations that `CASTEP 4.2` is allowed to make. Typically the single-point energy calculation just requires one iteration, and geometry optimisation should converge in 10-20 iterations. `CASTEP 4.2` stops when the required tolerance is achieved, for both single-point or geometry optimisation runs, so in these cases `NITER` is an upper limit. Molecular dynamics runs use `NITER` as the requested number of MD timesteps. The UI uses default values of `NITER=3` for single-point energy, and `51` for geometry optimisation. `CASTEP 4.2` only allows a limited number of SCF loops within one iteration (see `MAX_CYCL`). This means that for some systems with poor electronic convergence `CASTEP 4.2` might need more than one iteration for the single-point energy run, which is why the default is 3, not 1.

```
1      ! NPRINT ! Forces are printed every NPRINTth iteration
```

`NPRINT` is really an historical variable. If the output mode is brief (i.e., `IPRINT=0`), then the `.cst` output file will only display forces on all atoms at the end of the run. The forces, however, will be stored in a separate file, `.forces`, which can be used to monitor forces as a function of number of iterations. If `NPRINT` is set to 1, the `.forces` file will contain forces at each iteration, otherwise, at each `NPRINT`-th iteration. If the output mode is verbose (`IPRINT >0`), then the `.cst` file will also display forces for each `NPRINT`-th iteration.

```
300.00000      ! ENMAX ! Kinetic energy cut-off (eV)
```

`ENMAX` sets the energy cutoff for wavefunctions, E_c , in eV

```
0      ! NPOT ! Potential: 0-reciprocal, 1-real space
```

Norm-conserving pseudopotentials can be used either in reciprocal or real space representation (it's not possible to mix the two). In either case, the `.recpot` input file is always required. Real space potentials require an additional file, `.realpot`.

```
5      ! NITMAX ! Number of electronic CG steps
```

`NITMAX` has the meaning of the number of conjugate-gradients steps in any CG-based electronic structure algorithm, or the history depth for the RMM-DIIS scheme. In either case, a value of 5 should be sufficient.

```
0      ! ISTART ! Mode: 0-new run, 1-restart
```

`ISTART` has to be set to 1 when the run is either a direct continuation of the previous one, or for a band structure run (`IBANS=1`).

```
1      ! ISBROT ! Rotation: 0-NO,1-strict, 2-fast METALS only
```

This switch is only relevant for norm-conserving pseudopotentials. It is recommended to set it to 1. Subspace rotation at the end of the conjugate gradients loop is necessary to obtain eigenvalues, and is thus required for metallic systems. There is no good reason not to use the subspace rotation (ISBROT=0). The ISBROT=2 option is not thoroughly tested.

```
0      ! IOCCUP ! Partial occupation: 0-no, 1-yes
```

IOCCUP determines whether the system is treated as a metal (IOCCUP=1) or as an insulator (IOCCUP=0). Metallic system should have NBANDS set to a number bigger than half the number of valence electrons. The occupation numbers for the bands are then determined by introducing artificial Gaussian smearing on each band and finding the area of the smeared level which falls below the Fermi energy. IOCCUP=0 case means that each level has an occupancy of either 0 or 1.

```
0.20000      ! DELMIN ! Minimum smearing width (eV)
4.00000      ! DELMAX ! Starting smearing width (eV)
6            ! NDEL   ! Smearing width is halved every NDEL SCF loops
```

The width of the smearing function, which is used in metals calculations, is determined by these three numbers. The width is taken to be DELMAX for the first SCF loop, then it is halved every NDEL loops or when the total energy is converged at a given smearing, whichever happens first. The width is halved until it becomes equal to DELMIN. The DELMIN value should not be made too small, as this can lead to instabilities. A large DELMIN might help to converge metallic systems faster, but there will be an unaccounted-for error in atomic forces and stresses. Experience shows that a value of 0.2 provides a good intermediate solution. DELMAX should be set ≈ 4.0 unless the density mixing scheme is used for electronic minimization, when DELMAX should be ≈ 0.4 — in this case the algorithm is sufficiently stable not to require long quenching.

```
0      ! IION   ! Ionic relaxation: 0-no, 1-CG, 2-simple, 3-BFGS, 4-damped
MD
```

```
0      ! IBOX   ! Unit cell relaxation: 0-no, 1-uniform, 2-BFGS
```

IION=0 means that the fractional coordinates of atoms are fixed. IION=3 is the properly supported mode, and 1 and 2 can only be used when the cell parameters are fixed. IION=3 implies that the `.optim` input file is required, containing an initial guess for the Hessian. The updated `.optim_out` file is written at the end of the run. IION=4 in conjunction with IDYNAM = 1 gives a damped MD calculation. See also IDAMP, RECALC and TDAMP.

N.B: the use of ultrasoft pseudopotentials places restrictions on the choice of IION to 0, 3 or 4. This is because the CG and simple methods, after

moving the ions, update only the data associated with norm-conserving pseudopotentials.

`IBOX=0` means that the cell parameters are fixed. Full geometry optimisation (cell + internal degrees of freedom) requires `IBOX=2` and `IION=3`. `IBOX=1` only makes sense for cubic crystals; in this case `CASTEP 4.2` changes the cell constants uniformly on the way to the energy minimum, and at the end it also produces the bulk modulus and its pressure derivative.

```
0      ! IPRINT ! Amount of output
```

`IPRINT` controls the amount of detail in the `.cst` file: 0 produces brief output, and 1 gives more information about the SCF process and prints components of stresses and forces. It is rarely necessary to use higher settings, unless you need a very detailed description to help track down a problem with the code.

```
1      ! NIONCG ! Number of CG steps for ions
```

`NIONCG` is only relevant when performing conjugate gradients geometry optimisation (`IION=1`). The conjugate gradients search is in the combined space of electronic and atomic degrees of freedom. `NIONCG` is the number of conjugate search directions examined before re-setting to the steepest-descent search vector. Therefore, 1 means that the steepest descents algorithm is used. Values of `NIONCG` between 3 and 10 are recommended. Higher values will probably not improve the search efficiency.

```
1      ! INRAND ! WFs initialization: 0-diamond 8at. 1-random
```

An historical variable, but has to be set to 1.

```
1      ! ICLOCK ! Timer: 0-off, 1-on
```

`ICLOCK=0` means that only the total timing will be included in the file. `ICLOCK=1` provides timings after each SCF loop and in some other important places in the code. Note that the timings reported are calculated from CPU time, not wall time.

```
0.00100      ! POTIM ! Timestep for ionic dynamics (ps)
```

MD calculations employ a fixed time step in picoseconds, defined by `POTIM`. The total length of the simulation is then `POTIM×NITER`

```
0.05000      ! PODISP ! Maximum displacement of ions
```

`PODISP` has different meanings within different geometry optimisation modes. When `IION` is 1 or 2, `PODISP` limits the step length in every cartesian component for every atom. When `IION` is 3, i.e. when performing BFGS minimisation, `PODISP` limits the relative change in bond length between any two

atoms. A value of 0.05 works quite well in either case.

```
3      ! ICHARG ! The charge on atom species 1 (B )
5      ! ICHARG ! The charge on atom species 2 (N )
```

ICHARG is number of valence electrons for each atomic species. This is not the nominal valence, but the number of electrons treated as valent during pseudopotential generation (these might include semicore states). The numbers can be extracted from the pseudopotential files: see section 7.4.

```
10.81000      ! POMASS ! The mass of atom species 1 (B )
14.00670      ! POMASS ! The mass of atom species 2 (N )
1             ! NIONSP ! The number of atoms of species 1 (B )
1             ! NIONSP ! The number of atoms of species 2 (N )
2             ! IVPTYP ! Type of potential(0-L,1-NL,2-USP)species 1 (B )
2             ! IVPTYP ! Type of potential(0-L,1-NL,2-USP)species 2 (N )
```

The parameters POMASS are the atomic masses for each species, and are only relevant for molecular dynamics. NIONSP is the number of atoms of each kind in the unit cell. The pseudopotential type IVPTYP is usually 1 (norm-conserving) or 2 (ultrasoft), but may also be 0 for a completely local norm-conserving potential.

```
2001        ! NPSPTN ! Number of rec.space data points, species 1 (B )
2001        ! NPSPTN ! Number of rec.space data points, species 2 (N )
```

Defines the number of grid points used for storage of reciprocal space potentials.

```
1          ! ISYMM ! Symmetrization switch: 0-off,1-on
```

ISYMM=0 means that the structure is assumed to have no symmetry elements. If ISYMM=1, then the `.symm` input file is required, containing a list of symmetry operations. Note that the set of k-points in `.geom` should be consistent with the symmetry setting, i.e. if symmetry is exploited the k-point set should be symmetrised.

```
0          ! IBANS ! Band structure mode: 0-no,1-yes
```

A band structure calculation is performed when IBANS=1, determining the electronic energies in a fixed potential, i.e., it is the setting for a non self-consistent run. For single-point energy, geometry optimisation or MD runs, IBANS is set to 0. If IBANS=1, ISTART should be set to 1 as well, since CASTEP 4.2 will attempt to read the charge density from a previous calculation. The number of bands and the number/coordinates of k-points for the band structure run are unrelated to the settings of the original run.

```
0          ! ICNSTR ! External constraints: 0-no,1-yes
```

The setting ICNSTR=0 should be used. Any additional constraints may be

added via the `.optim` file (if using BFGS, `IION = 3`), or the `.constr` file (if using constrained dynamics, `IDYNAM = 1 NEW_CONS=1`).

```
0      ! IDYNAM ! MD mode: 0-off,1-on
```

This is the MD switch. Note that the MD style is determined by the settings of `IION` and `ISTAT`.

```
0.0000050      ! EPSIL ! Convergence tolerance (eV/at)
```

This parameter specifies the accuracy of electronic structure optimisation, in eV per atom. Atomic motion (either for MD or for geometry optimisation) only starts when this tolerance is satisfied.

```
5      ! NEXTWR ! Backup every NEXTWRth iteration
```

CASTEP 4.2 produces a full set of backup files at regular intervals. Unless the job is killed while writing one of those files, it should be possible to restart later by simply changing `ISTART` to 1.

```
0      ! ISTDYN ! For MD: 0-new run, 1-restart
```

Setting `ISTDYN=0` allows you to continue a previous MD run using the final atomic positions but with re-generated thermal velocities. The value of `ISTDYN` is only relevant if `ISTART=1` (restart).

```
100     ! NPREC ! 2 extra electronic steps every NPREC iterations
```

Not used.

```
0      ! ISTAT ! For MD: 0-microcanonical,1-Nose,2-Langevin NVY
0.00007      ! QNOSE ! Nose thermostat parameter
```

The switch `ISTAT` turns the thermostat on or off. `ISTAT=0` generates the NVE ensemble, `ISTAT=1` the NVT with the Nosé thermostat, and `ISTAT=2` generates Langevin dynamics. For the latter see also `TDAMP`.

The Nosé-Hoover thermostat generates a deterministic dynamics, with the temperature controlled by a fictitious additional coordinate s added to the Lagrangian of the system. The thermostat employs a feedback loop between the instantaneous kinetic energy and the required temperature. The rate of feedback is determined by the "mass" parameter Q (`QNOSE`). This should be chosen so that the natural oscillation frequency of the Nosé coordinate is the same as the characteristic frequency of the real system.

The mass parameter is related to the thermostat relaxation time by

$$(g/2)k_B T \tau^2 = Q/2 \quad (1)$$

where g = the number of degrees of freedom (usually $3N - 3$, where N is the number of atoms), k_B is Boltzman's constant, T is the thermostat

temperature, and τ is the relaxation time as used in some implementations of the thermostat. In CASTEP 4.2 Q is in units of eV/ps².

The Langevin thermostat generates a stochastic dynamics, via a Brownian motion model, implemented using the fluctuation-dissipation theorem. It is controlled by the "damping time" parameter TDAMP, which controls the rate of decay of fluctuations in the temperature. In practice, this should be chosen to be larger than the time scale of any motions of interest.

```
273.00000      ! TEMPER ! Temperature (K)
```

TEMPER is used to initialize velocities during MD. The actual temperature fluctuates around TEMPER, even for NVT runs.

```
1      ! NVEL      ! Velocity initialization (0=file,>0=random)
```

Setting NVEL=0 means that velocities are taken from the .geom file, while NVEL=1 will force CASTEP 4.2 to generate initial velocities using the TEMPER value.

```
CASTEP calculation      ! comment
```

The title line: it is reproduced in the .cst file

```
1      ! ISTRESS! Stress output : 0-no, 1=yes
```

ISTRESS=1 forces CASTEP 4.2 to calculate the stress tensor even when the cell parameters are fixed (IBOX=0).

```
GGA      ! Theory level
```

Selects the exchange-correlation functional. Allowed values and corresponding functionals are: LDA and LSDA (Ceperly-Alder), GGA, LDA+Gradient Corr., LDA+GC, GGA-PW91, GGS, GGS-PW91 (synonyms for Perdew-Wang II GGA and GGS), GGA-PBE, GGS-PBE (Perdew-Becke-Ernzerhof GGA and GGS, PRL **77** (1996) 3865), and GGA-RPBE, GGS-RPBE (revised PBE GGA and GGS, Phys. Rev. B **59** (1999) 7413).

```
2      ! METHOD ! 0-band_by_band, 1-all_bands, 2-density mixing
```

METHOD=0 selects the original Payne-Teter-Allan band-by-band conjugate gradients scheme when using norm-conserving potentials, or employs a slightly modified CUSP code from David Vanderbilt when using ultrasoft potentials. METHOD=1 only applies to norm-conserving potentials, and it selects conjugate gradients minimisation applied to all bands simultaneously. METHOD=2 selects the recommended mode, where a self-consistent density mixing scheme is used. This operates with potentials of either class.

All methods of minimisation work with norm-conserving pseudopotentials. With ultrasoft pseudopotentials there are restrictions, which are summarised

in table 4.

METHOD	Minimiser	Insulators?	Metals?
0	Band-by-band CG		
1	All-bands CG	•	
2	Density mixing	•	•

Table 4: Choice of minimiser when using ultrasoft pseudopotentials. All methods work with norm-conserving pseudopotentials.

```
20      ! ISEED ! 0-random, otherwise-pseudorandom wavefunctions
```

ISEED is normally set to zero, and it results in different random initial wavefunctions each time the code is run. A non-zero value may be useful, since each nonzero ISEED generates a deterministic and reproducible pseudo-random sequence.

```
0      ! IWRCHD ! If nonzero, formatted charge density is created
```

CASTEP 4.2 usually outputs the charge density as a binary file only. If you require a formatted file (e.g., for easy transfer from a Cray machine) then IWRCHD=1 should be used. The formatted file is only created at the very end of the run, while the binary file is backed-up regularly during the calculation. A formatted .gm file is also written.

```
30      !MAX_CYCL! Maximum number of SCF cycles at each step
```

Each CASTEP 4.2 iteration can only make MAX_CYCL number of SCF loops. A value of 30 is usually sufficient, but if SCF convergence is not achieved in MAX_CYCL loops, then CASTEP 4.2 still moves on to the next iteration (out of NITER). Note that because in general metals need more steps than insulators, CASTEP 4.2 uses 5*MAX_CYCL as the number of SCF steps for metals (i.e. when partial band occupation is allowed).

```
1      !BAS_CORR! Finite sampling correction: 0-no, 1-yes
0      !MODE_COR! Mode of sampling correction: 0-auto, 1-manual
0.00000 !DE_DlogE! Derivative of Etotal w.r.t. log(Ecutoff)
```

BAS_CORR=1 means that the finite basis set correction will be used. This correction effectively allows one to get away with a k-point set and energy cutoff that are not high enough for absolute convergence, and still perform cell optimisation.

MODE_COR=0 means that CASTEP 4.2 will calculate the finite basis set correction during the run. The value that needs to be calculated is the derivative of the total energy with respect to $\log E_c$. MODE_COR=1 assumes that this value is already known (e.g., from the previous run performed for the same system with exactly the same run parameters). The value of $\partial E / \partial \log E_c$,

DE_DlogE (in eV), is required when MODE_COR=1 is used.

```
0.00000    !PRESSURE! External pressure (GPa) and stress (next line)
0.000      0.000      0.000      0.000      0.000      0.000
```

This is the external applied stress (pressure is the trace of the tensor, and the tensor is given in a short notation as xx,yy,zz,yz,xz,xy).

```
3          !NUM_CUT ! Points for num. diff. for DE_DlogE
288.00000 ! ECUT   ! Cutoff # 1 for sampling correction (eV)
294.00000 ! ECUT   ! Cutoff # 2 for sampling correction (eV)
300.00000 ! ECUT   ! Cutoff # 3 for sampling correction (eV)
```

When MODE_COR=0, $\partial E/\partial \log E_c$ is calculated automatically by finding the total energy at a number of reference cut-off energies. The value of NUM_CUT, specifies the number of energies which are given on the following NUM_CUT lines. The last ECUT should be the same as ENMAX. A default setting of NUM_CUT=3 and a step between the reference energies of 2% of ENMAX should work in most cases. Note that if the difference between the reference energies is so small that the number of plane waves used does not change from one ECUT to the next, the calculation will not make sense.

```
0.00100    !DISP_TOL! Tolerance for RMS displacement
0.05000    !RMSF_TOL! Tolerance for RMS force
0.10000    !RMS_STR ! Tolerance for RMS stress
0.0000200  !ETOT_TOL! Minimizer tolerance for total energy
```

There are several parameters controlling geometry optimisation: It will continue until the ionic mean-square displacement is less than DISP_TOL (Å), or until the mean-square force on movable ions is less than RMSF_TOL. Geometry optimisation continues until the mean-square stress is below RMS_STR. Only components that correspond to variable cell parameters are taken into account (e.g., only diagonal terms for a cubic cell). Finally, optimisation continues until the change of the total energy between successive configurations is less than ETOT_TOL. NOTE: All the relevant criteria must be satisfied for the run to stop, i.e., there are four criteria to satisfy when both cell and internal degrees of freedom are allowed to relax.

```
0          ! OPTICS ! Calculate matrix elements: 0-no, 1-yes
```

Running with OPTICS=1 is only possible when IBANS=1. A calculation with OPTICS=1 produces a .cst_ome file, containing matrix elements for optical transitions, after the band structure run. It is expected that the Monkhorst-Pack set of k-points is used when OPTICS=1.

```
4          !NEL_UP   ! Number of electrons with spin up
4          !NEL_DOWN ! Number of electrons with spin down
0          !SPIN_FIX ! Spin is fixed (0) or varied
```

The use of a spin-polarized exchange-correlation functional is required when the numbers of spin-up and spin-down electrons differ. `SPIN_FIX` only affects runs using spin-polarized exchange-correlation functionals. If its value is zero, then the spin of the system is fixed. If its positive, then the spin is fixed for the first `SPIN_FIX` SCF loops and is allowed to vary thereafter (which is required for magnetic metallic systems). A negative value of `SPIN_FIX` means that spin is allowed to vary from the very first loop.

```
0      !INIT_OCC! Init.  occupancy: 0-lowest, 1-all, -1-file
```

The parameter `INIT_OCC` specifies how occupation numbers are initialised:

- 0 — the lowest bands are filled, and the higher ones are empty if their number is greater than half of the number of electrons.
- 1 — all bands are filled equally, even if this creates partial occupancies
- 1 — exotic setting, `CASTEP 4.2` reads starting occupation from the `.occnum` file This might be useful, e.g., for simple atoms where user knows the degeneracies

```
0.20000      ! ROTMIX ! Mixing factor for WF rotation
0.50000      ! OCCMIX ! Mixing factor for occupation numbers
0.50000      !EL_STEP ! Step for all-bands minimizer
```

These parameters are only used by the all=bands minimiser. Recommended values are 0.2, 0.5 and 0.5 respectively, and these should not be altered.

```
7           !MIX_HIST! Density mixing history (0:non-Pulay)
0.80000     ! CH_AMP ! Mixing amplitude for charge
1.50000     ! CH_Q   ! Cutoff wavevector for charge
```

`MIX_HIST` specifies the depth of the history stack in the Pulay scheme of the density mixing method (DIIS history). A value of 0 means that the density mixing scheme is not using the Pulay algorithm. `CH_AMP` specifies the mixing amplitude for the charge density. It might be useful to decrease it if the convergence is not sufficiently stable. `CH_Q` specifies the cutoff wavevector: small- G components are mixed with smaller coefficients as prescribed by Kerker. A value of zero means that linear mixing is used, rather than Kerker mixing.

```
2.00000     !SPIN_AMP! Mixing amplitude for spin density
1.50000     ! SPIN_Q ! Cutoff wavevector for spin density
0           !RM_DELAY! CG steps before RMM starts (0 is CG)
```

`SPIN_AMP` and `SPIN_Q` have the same meanings as `CH_AMP` and `CH_Q`, but refer to the spin density (these variables are irrelevant for non spin-polarised calculations). `RM_DELAY` specifies whether to use the conjugate gradients scheme

in the density mixing approach (RM_DELAY=0) or the DIIS-based residual minimization. Setting RM_DELAY>0 means that the RMM-DIIS algorithm is used, but the first RM_DELAY steps are still done by conjugate gradients.

```
2      !DENS_INI! Use starting atomic densities (2)
```

Set to 2 for an initial density composed from atomic densities, or to 1 otherwise.

```
0      !MULLIKEN! Mulliken analysis: 0-no, 1-yes
```

Switches population analysis on (1) or off (0).

```
3.00000      ! CUTOFF ! Cutoff for bond order calculation
```

Cutoff in Å for bond (overlap) population calculation. Populations will only be output for bonds with a length less than or equal to this.

```
4      ! NORBS ! Number of atomic orbitals, species 1 (B )
```

```
4      ! NORBS ! Number of atomic orbitals, species 2 (N )
```

One entry per species. This controls the number of atomic orbitals used in the LCAO basis set for this species. See Section 7.8 for further details.

```
0      ! NPDOS ! PDOS set: -1:all,0:none,>0 - see below
```

The number of PDOS projectors to calculate. If 1, the weights for all PDOS projectors will be calculated. If 0, no PDOS calculation will be performed. If greater than 0, selected PDOS projections will be performed as listed in the subsequent lines of the file.

```
1      !IEXTRAP ! Wavefunction extrapolation method (0,1,2,21)
```

Sets the kind of wavefunction extrapolation used in MD: 0, no extrapolation; 1, first order Velocity-Verlet best-fit extrapolation; 2, second order Velocity-Verlet best-fit extrapolation; 21, alternating 1st / 2nd / 1st etc; -1, first order constant extrapolation, and equivalents for -2 and -21.

Try IEXTRAP = 1 first.

```
0      !IDAMP ! Auto-damping calculation for damped MD (iion=4)
```

Sets the kind of damping applied when IION=4: IDAMP= 0 means no damping; IDAMP= 1 uses the independant modes algorithm for automatic calculation of the optimal damping coefficient; IDAMP= 2 uses the coupled modes algorithm for automatic calculation of the optimal damping coefficient; IDAMP= 3 applies steepest descents damping. Notice that IDAMP=-1 or -2 gives the corresponding automatic damping with automatic timestep selection.

Try IDAMP=1 first.

```
30      !RECALC ! MD timestep between idamp recalculation (iion=4)
```

This controls the interval between recalculations of the damping coefficient when IDAMP=1 or 2. This should not be done too often as it is costly: 30 steps is a useful default.

```
0.050      !TDAMP ! Langevin damping time-constant [ps] (istat=2)
```

See ISTAT.

```
0      !NEW_CONS! New linear constraints (1) in MD
```

Set to 1 to apply linear constraints during MD (IDYNAM = 1). The constraints must be specified in the .constr file.

7.3 Runtime parameters: .param

This file contains parameters used to define the dimensions of various CASTEP 4.2 arrays.

```
12      * NGX      * FFT grid dimension
12      * NGY      * FFT grid dimension
12      * NGZ      * FFT grid dimension
```

These parameters define the FFT grid, whose size depends on the cutoff energy. The grid should be fine enough to prevent Fourier aliasing. This occurs when the period in reciprocal space is too small to prevent the sphere of plane waves overlapping with its periodic images. To completely avoid aliasing, the following prescription should be used:

$$NGX = 4L\sqrt{\frac{2m_e E_c}{h^2}}, \quad (2)$$

and similarly for the other two dimensions. Here, L is the length of the real-space lattice vector, m_e is the electron mass, E_c the cutoff energy and h is Planck's constant. With L in Å and E_c in eV this becomes

$$NGX = 0.32615L\sqrt{E_c}. \quad (3)$$

It is possible to run a plane-wave code with a grid smaller than the recommended one. For example, the Cerius2 interface typically sets the dimensions to 3/4 of those given by the above equation. The motivation for this is increased execution speed, but be warned, one must quantify errors that arise from using a smaller grid. Note also that ultrasoft pseudopotentials and gradient-corrected functionals (GGA) both require a finer FFT grid than is needed without them, assuming all other variables are the same.

```
2001      * NPSPTS * Max. dimension of reciprocal space data
```

This is the maximum size of the reciprocal-space dataset for the potentials being used, and is the largest of the NPSPTN values in the `.rundat` file. It is almost always 2001.

```
160      * NRPLWV * Max. number of plane waves accepted
```

This is the maximum number of plane waves needed to describe ψ_k . The number of plane waves is different for different k , and it depends on the spacing between points in reciprocal space and the relative sizes of k and $G_{max} = \sqrt{2m_e E_c / \hbar^2}$. In the limit of fine reciprocal-space grid (large simulation box) and large E_c , $NRPLWV \approx NGX * NGY * NGZ / 16$. This assumes that a full-size, zero-overlap grid is used. If all grid dimensions are reduced by a factor f , then $NRPLWV \approx NGX * NGY * NGZ / (16f^3)$. Note that the number of plane waves is constant even if the cell optimisation is performed.

```
1      * NIONS * Max. number of atoms of any kind
2      * NSPEC * Number of different atomic species
2      * NIONST * Total number of atoms
4      * NBANDS * Number of electron bands
10     * NKPTS * Number of special k-points
```

NIONS is the largest of all NIONSP values in the `.rundat` file. NSPEC and NIONST are self-explanatory NBANDS should be at least a half of the number of electrons as given in the `.rundat` file (`NEL_UP + NEL_DOWN`). It is necessary to include some empty bands in calculations on metals. NKPTS should agree with the contents of `.geom` and `.symm` (where applicable): CASTEP 4.2 will attempt to read NKPTS lines from those files.

```
1      * NRRGPT * Dimension of arrays for real-space potentials
```

This is usually calculated by the Cerius interface, but if set too small, CASTEP 4.2 will increase it at runtime. This value can be also increased at runtime as a result of atomic motion during MD or geometry optimisation.

```
1      * MXRLNL * Number of real-space nonlocal projectors
1      * MXRLSH * Number of real-space spherical harmonics
1      * NRLPTS * Max. dimension of real-space data
```

These parameters must be set if real-space pseudopotentials are used. In the BN example the potentials are evaluated in reciprocal space and these parameters are set to 1. However, if using the real-space option, their values might be: `MXRLNL=2` if all the potentials have only two nonlocal components, e.g., s and p ; `MXRLSH = 1 + 3 = 4` (1 for s , 3 for p); `NRLPTS=2001`, having the same meaning as NPSPTS, but for the real-space radial grid. The latter is usually 2001.

```
0      * NCOORD * Degrees of freedom for atomic coordinates
```

0 * N_CELL * Degrees of freedom for cell vectors

The parameter `N_COORD=3` \times number of *moveable* atoms, i.e. those with a non-zero moveability switch in `.geom`. `N_COORD=0` if `IION=0` (in `.rundat`). `N_CELL` is the number of independent degrees of freedom for cell variables. E.g., `N_CELL=1` for cubic cells with cell optimisation switched on, i.e. `IBOX>0` in `.rundat`. Both `N_COORD` and `N_CELL` are only used for BFGS optimisation, that is, when `IION=3` and and/or `IBOX=2`.

7.4 The pseudopotentials: `.recpot` and `.realpot`

Each CASTEP 4.2 calculation needs a single `.recpot` file, and if the pseudopotentials are to be evaluated in real space, a single `.realpot` file. These files contain data taken from the pseudopotential library files, or generated from them. The distribution contains the library `POTENTIALS`. Note that norm-conserving potentials are no longer distributed, and some of what follows is included for compatability purposes only. Table 5 gives the meaning of all the suffices that pseudopotential files may have. Note that the CASTEP 4.2 input file containing the reciprocal space potentials, and the library files containing a *single* norm-conserving potential, share the same suffix, `.recpot`. You should note that ultrasoft pseudopotentials and similar technologies have been chosen as the preferred development route for CASTEP 4.2.

<code>.usp</code>	Vanderbilt ultrasoft pseudopotentials
<code>.uspc</code>	As above but with core charge
<code>.recpot</code>	Norm-conserving reciprocal space pseudopotential in Kleinman-Bylander form
<code>.psp</code>	Teter pseudopotential
<code>.pspnc</code>	Troullier-Martins pseudopotential

Table 5: Pseudopotential files

A full description of the pseudopotentials will not be given here. However, you will need to know some details of the structure of the input files `.recpot` and `.realpot`, so that you can create them for your calculations.

7.4.1 Preparing `.recpot` when using ultrasoft pseudopotentials

All calculations require a `.recpot` file, though its contents depend on the types of potential you are using. In all cases the complete reciprocal-space pseudopotential for each species must be in this file, in the same order as the species appear in the `.geom` and `.rundat` files. You must also tell

CASTEP 4.2 which types of potentials are being used by setting the IVPTYP parameters in the `.rundat` file.

The overall structure of `.recpot` is as follows. The first line contains two numbers which denote the CASTEP version number for which the potentials were generated¹. Next comes the pseudopotential for the first species, with a line containing the number “1000” marking the end of the data. Then comes the pseudopotential for species 2, and so on, with each data set terminated by “1000”.

The library files contain ancilliary information which must be removed before they are assembled into a `.recpot` file. It is easier to understand this by working through the building of the `.recpot` file for the BN example. In this case, both pseudopotentials are ultrasoft ones. The boron potential is taken from the `POTENTIALS/B_00.usp` file. In this file you will see that there is a comments section, delimited by `START COMMENT` and `END COMMENT` lines. All the `.usp` and `.recpot` files contain a comments section, in which the construction and testing of the pseudopotential is described. In addition there are recommended plane-wave cutoff energies for `COARSE`, `MEDIUM` and `FINE` grids. The entire comments section, including the delimiting lines, must be removed when inserting the pseudopotential data into `.recpot`.

The line after the `END COMMENT` line contains the version information for this potential. However, in `.recpot`, the first line contains version information for **all** the pseudopotentials. If `.recpot` contains a single potential there is no ambiguity. However, for systems with more than one species, the highest version among the potentials should be taken and inserted into `.recpot` as the first line. I’m afraid this does not guarantee success, and it is better not to mix potentials with different versions.

In `B_00.usp` and for all ultrasoft potentials, the next line gives the core charge for the potential. This is the value to which `ICHARG` should be set in the `.rundat` file. However, it is not required in the `.recpot` file. All data after this line is inserted into `.recpot`.

Gathering this together, the structure of the ultrasoft pseudopotential files is as follows.

Comments Section delimited by `START COMMENT` and `END COMMENT` lines, containing generation and testing information

Version Line with the version of CASTEP for which the potential was made

Core Charge Line with the pseudopotential core charge

Pseudopotential Section containing all data for the pseudopotential

¹Versions earlier than 3.0 did not require this line in `.recpot`. If the version line is missing, CASTEP 4.2 assumes that `.recpot` conforms to this older format.

The stages in preparing `.recpot` are as follows.

1. Determine the highest version number of the potentials you are using from the Version Lines and insert it as the first line of `.recpot`
2. Set the `ICHARG` parameters in `.rundat` to the core charges appearing on the Core Charge Lines.
3. Remove the Comments Section, Version Line and Core Charge Line from all the potentials.
4. Insert all the potentials into the `.rundat` file, in the same order as the species appear in the other input files.

7.4.2 Preparing `.recpot` when using norm-conserving potentials

The norm-conserving potentials have the `.recpot` suffix. (Remember, both the CASTEP 4.2 input file `.recpot` and the norm-conserving potential files share the same suffix.) Broadly, the foregoing description relating to constructing the CASTEP 4.2 input file `.recpot` when using ultrasoft pseudopotentials carries over to norm-conserving ones. There are three differences. Firstly, an extra line should be inserted in the file containing information about nonlinear core corrections. Secondly, the norm-conserving potentials do not contain the Core Charge line. Thirdly, each norm-conserving pseudopotential must be terminated by a line containing the integer 1000

After the Comments Section, the first few lines of the norm-conserving pseudopotential files have the following format:

```

3 5
100.0000000000
  2.9612498283      -217140.6718750000      -54282.9492187500
-24124.1113281250      -13568.5205078125      -8682.7910156250
```

Line one is the version line, line two is the value of `PSGMAX`, the maximum wavevector of the pseudopotential in $1/\text{\AA}$, and subsequent lines define the local part of the potential. The nonlinear core correction line is inserted after `PSGMAX`, so that the file becomes:

```

3 5
100.0000000000
  0.0000000000          0.0000000000
  2.9612498283      -217140.6718750000      -54282.9492187500
-24124.1113281250      -13568.5205078125      -8682.7910156250
```

The two numbers are read into the variables `fcharg` and `rcharg`, the prefactor and core radius for the nonlinear core charge respectively. For all

the norm-conserving potentials in the `POTENTIALS_NC/` directory, distributed with CASTEP 3.9, these numbers should be zero.

Since there is no Core Charge Line you need to find the core charges for the pseudopotentials by other means. Usually the Comments Section of the potential file contains enough information to deduce the core charge. The configurations used in generating the potential indicate which states were treated as valence states. Subtraction of the number of core states from the atomic number gives the core charge.

A second method, which is always reliable, is based on the nature of the local potential V . As the magnitude of the wavevector $q \rightarrow 0$, corresponding to large distances, V tends to that of a point charge, given by

$$V = Z/q^2. \quad (4)$$

For a norm-conserving pseudopotential we can find the smallest wavevector and the corresponding local potential. The line after the Version Line contains the radial grid maximum `PSGMAX` (in $1/\text{\AA}$), with the potential tabulated on a grid of `(NPSPTN-1)` points. `NPSPTN` is almost always 2001, though some very early potentials used a value of 501. The smallest wavevector is therefore `PSGMAX/(NPSPTN-1)`. The value of the V at the smallest wavevector is the second number in the following line. Including the necessary conversion factors, the pseudopotential core charge is given by the nearest integer to

$$\text{ICHARG} = \frac{|V|}{180.95263} \left(\frac{\text{PSGMAX}}{\text{NPSPTN} - 1} \right)^2. \quad (5)$$

For the boron norm-conserving pseudopotential `POTENTIALS_NC/B_00.recpot` this becomes:

$$\text{ICHARG} = \frac{217140.671875}{180.95263} \left(\frac{100}{2000} \right)^2, \quad (6)$$

yielding `ICHARG = 3`.

The steps in preparing `.recpot` with norm-conserving pseudopotentials are thus:

1. Determine the highest version number of the potentials you are using from the Version Lines and insert it as the first line of `.recpot`
2. Set the `ICHARG` parameters in `.rundat` to the core charges as calculated or read from the potential files.
3. Remove the Comments Section and Version Line from all the potentials, insert the Nonlinear Core Correction line and append a line with the integer "1000" to them.
4. Insert all the potentials into the `.rundat` file, in the same order as the species appear in the other input files.

7.4.3 Using ultrasoft and norm-conserving pseudopotentials together

CASTEP 4.2 will run with a mixture of ultrasoft and norm-conserving pseudopotentials. The steps in preparing the `.recpot` file are an obvious merge of the instructions given in the previous two Sections.

7.4.4 The real-space pseudopotential file `.realpot`

This file contains the non-local pseudopotential in real space. There is a considerable advantage to evaluating the pseudopotential in real space for large systems, since then the calculation scales more favourably with system size. However, you will notice immediately that there are no `.realpot` files in the pseudopotential libraries. In fact, it is not possible to provide a single transformation of the reciprocal-space potential since the process is cutoff-dependent. Also, we are not able at present to provide a transformation code with the distribution. Cerius² users can access this facility when preparing CASTEP 4.2 jobs. CETEP users should note that the `.realpot` file has the same structure as the CETEP input file `fort.18`. Members of UKCP groups can obtain the pseudopotential generation code through their group's UKCP member.

7.5 Symmetry information: `.symm`

This file contains information on the symmetry of the structure specified by `.geom`, as well as a symmetrised set of k-points. Symmetry is used in the calculation, and hence this file read, when `ISYMM=1` in the `.rundat` file. For high-symmetry crystals this can result in a substantial speedup, but the benefit is less for low-symmetry crystals and surfaces. Using symmetry during geometry and cell optimisation preserves the initial symmetry, which while useful, may result in metastable configurations at the end of the optimisation. Symmetry cannot be exploited during MD runs.

From a given `.geom` file, the utility program `K290` in the `utilities/K290` directory will generate the appropriate `.symm` file. This program was kindly provided by Karel Kunk, and if you make use of it please cite the reference given in Section .

This utility is not supported as part of the distribution. However, you will find brief instructions in the file `sym.txt`. The control parameters are in the `fort.4` file. The input files `fort.14` and `fort.15` correspond to `<seed>.rundat` and `<seed>.geom` respectively. The output file `fort.3` is equivalent to the CASTEP 4.2 input file `<seed>.symm`

Note that the number of symmetry operations identified by `K290` can vary

with the orientation of the crystal with respect to the Cartesian axes.

7.6 The .optim file

This file contains information used by the BFGS minimiser, and is relevant therefore when `IION = 3` and/or `IBOX = 2`: if either of these options is set, the .optim file must be present.

The file looks something like this:

```

  1      2      3      4      5      6
  6      ! number of nonzero elements in constraints matrix
  1      1      1.000000000
  2      2      1.000000000
  3      3      1.000000000
  4      4      1.000000000
  5      5      1.000000000
  6      6      1.000000000
  1.0000000  0.0000000  1.0000000  0.0000000  0.0000000  1.0000000
  0.0000000  0.0000000  0.0000000  1.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  1.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  1.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  100.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  100.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  100.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
  100.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  100.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  100.0000000

```

Here, a 2-atom system is to undergo full relaxation of the cell and internal co-ordinates. The file contains a mapping of old-to-new coordinates for the ions, and an initial Hessian.

There are three sections in the file; the first section is the first line, and this gives information about cell relaxation. There are six integers on the line, and they govern what happens to the cell parameters a , b , c , α , β and γ . The rules are: a 0 indicates that the parameter is fixed, whereas any non-zero value means the parameter will be optimised; it is possible for some of the parameters to be equivalent (in fact, symmetry may require that they are), in which case their corresponding integers must have the same value; finally, the initial Hessian elements must be provided for the parameters to be optimised (see below).

The second section of the file gives information about constraints on the atoms. In the example above, this consists of the seven lines:

```

  6      ! number of nonzero elements in constraints matrix

```

```

1      1      1.000000000
2      2      1.000000000
3      3      1.000000000
4      4      1.000000000
5      5      1.000000000
6      6      1.000000000

```

The “number of nonzero elements in constraints matrix” is the number of (ionic) degrees of freedom. In this example, both atoms are free to move under optimisation, and there are therefore six degrees of freedom. Each of the six lines following the commented line refers to a degree of freedom: the numbers refer to the old co-ordinates (i.e. those without atomic constraints), the “new” co-ordinates, which are those free to change after constraints have been applied to the atoms, and a number that is equivalent to REMOVE in the .geom file. This becomes clearer when constraints are applied to the atoms, for example, if the first atom is held fixed, the corresponding lines in .optim become:

```

3      ! number of nonzero elements in constraints matrix
4      1      1.000000000
5      2      1.000000000
6      3      1.000000000

```

The third section of the file gives the lower triangle of the initial Hessian matrix. The matrix is $(N_{cell_vars} + N_{ionic_vars}) \times (N_{cell_vars} + N_{ionic_vars})$, where N_{cell_vars} is the number of cell degrees of freedom, and N_{ionic_vars} the number of ionic degrees of freedom. If we let $N = N_{cell_vars} + N_{ionic_vars}$, the total number of elements is $\frac{1}{2}N(N - 1) + N$. The initial matrix is diagonal, with the diagonal elements set to 1.0 for cell degrees of freedom, and 100.0 for ionic degrees of freedom. The values must appear in row order, i.e. $H_{11}, H_{21}, H_{22}, H_{31}$ etc. The formatting of these numbers is unimportant.

All of this is not straightforward at first glance. Users who want to understand better may need to delve into the source code. The .optim file is read in the routine `cst_input_bfgs_optim`, which is in `cst_input_geom.F`. The BFGS constraints are applied in the `cst_bfgs_min.F` routine and routines called therein.

7.7 Linear constraints within MD: the .constr file

This file specifies linear constraints defined by

$$\sum_{N_c} \sum_i \mathbf{A}_{N_c} \cdot \mathbf{R}_i = c . \quad (7)$$

The summations are over the N_c constraints and the i ions; \mathbf{R}_i is the Cartesian position of ion i , and c a constant. The constraints are obeyed within MD runs (`IDYNAM = 1`) and when `NEW_CONS = 1`.

In the file a constraint is specified by a line:

```
<constraint_number> <species> <ion> Ax Ay Az
```

For example, to fix the x -component of ion 2 of species 1:

```
1 1 2 1.0 0.0 0.0
```

To fix all components (equivalent to setting `RMOVE = 0.0` in `.geom`):

```
1 1 2 1.0 0.0 0.0
2 1 2 0.0 1.0 0.0
3 1 2 0.0 0.0 1.0
```

To fix the x -separation of ions 1 and 2 of species 1:

```
1 1 1 1.0 0.0 0.0
2 1 2 -1.0 0.0 0.0
```

To constrain an atom to a plane with a surface normal $\mathbf{n} = n_x\hat{\mathbf{x}} + n_y\hat{\mathbf{y}} + n_z\hat{\mathbf{z}}$ whose origin is the initial atomic position:

```
1 1 1 nx ny nz
```

Finally, for convenience the centre-of-mass may be constrained by the single keyword:

```
fix_com = true
```

At present it is not possible to vary the cell shape during MD runs; all cell parameters are fixed.

7.8 Population analysis (and the `.batom` file)

7.8.1 Introduction

A disadvantage of the use of a plane-wave (PW) basis set is that, due to the delocalised nature of the basis states, it provides no information regarding the localisation of the electrons in the system. In contrast, a Linear Combination of Atomic Orbitals (LCAO) basis set provides a natural way of specifying quantities such as atomic charge, bond population, charge transfer and so on.

This Section describes the implementation of population analysis within CASTEP 4.2. This analysis is performed using a projection of the PW states onto a localised basis set by the projection technique described by Sanchez-Portal *et al.* [1]. Population analysis of the resulting projected states is then performed using the Mulliken analysis formalism [2]. This technique is widely used in the analysis of electronic structure calculations performed

with LCAO basis sets.

7.8.2 Theory

The eigenstates $|\psi_\alpha(\mathbf{k})\rangle$, obtained from the PW calculation when sampling at a given wavevector \mathbf{k} are projected onto Bloch functions formed from a LCAO basis set $|\phi_\mu(\mathbf{k})\rangle$. In general, such a localized basis set will be neither orthonormal nor complete. Therefore, care must be taken when performing the projection.

The overlap matrix of the localised basis set, $S(\mathbf{k})$, is defined:

$$S_{\mu\nu}(\mathbf{k}) = \langle \phi_\mu(\mathbf{k}) | \phi_\nu(\mathbf{k}) \rangle . \quad (8)$$

The quality of the projection may be assessed by calculation of a *spilling parameter*,

$$\sigma = \frac{1}{N_\alpha} \sum_{\mathbf{k}} w_{\mathbf{k}} \sum_{\alpha} \langle \psi_\alpha(\mathbf{k}) | (1 - \hat{p}(\mathbf{k})) | \psi_\alpha(\mathbf{k}) \rangle . \quad (9)$$

Here, N_α is the number of PW states, $w_{\mathbf{k}}$ are the weights associated with the \mathbf{k} -points in the Brillouin zone, and $\hat{p}(\mathbf{k})$ is the projection operator of Bloch functions with wavevector \mathbf{k} formed from the atomic basis:

$$\hat{p}(\mathbf{k}) = \sum_{\mu} |\phi_\mu(\mathbf{k})\rangle \langle \phi^\mu(\mathbf{k})| . \quad (10)$$

In this expression $\langle \phi^\mu(\mathbf{k})|$ are the duals of the LCAO basis, such that

$$\langle \phi_\mu(\mathbf{k}) | \phi^\nu(\mathbf{k}) \rangle = \langle \phi^\mu(\mathbf{k}) | \phi_\nu(\mathbf{k}) \rangle = \delta_{\mu\nu} . \quad (11)$$

Explicitly,

$$\langle \phi^\mu(\mathbf{k}) | = \sum_{\nu} S_{\mu\nu}^{-1}(\mathbf{k}) \langle \phi_\nu(\mathbf{k}) | . \quad (12)$$

The spilling parameter varies between one in the case that the LCAO basis is orthogonal to the PW states and zero when the projected wavefunctions perfectly represent the PW states.

The density operator may be defined as,

$$\hat{\rho}(\mathbf{k}) = \sum_{\alpha}^{occ} n_{\alpha} |\chi_{\alpha}(\mathbf{k})\rangle \langle \chi^{\alpha}(\mathbf{k})| . \quad (13)$$

where n_{α} are the occupancies of the PW states, $|\chi_{\alpha}(\mathbf{k})\rangle$ are the projected PW states $\hat{p}(\mathbf{k}) |\psi_{\alpha}(\mathbf{k})\rangle$, and $\langle \chi^{\alpha}(\mathbf{k})|$ are the duals of these states. From this the density matrix for the atomic states may be calculated as follows:

$$P_{\mu\nu}(\mathbf{k}) = \langle \phi^{\mu}(\mathbf{k}) | \hat{\rho}(\mathbf{k}) | \phi^{\nu}(\mathbf{k}) \rangle . \quad (14)$$

Inline:
swapped
ket to bra
 $\langle \chi^{\alpha}(\mathbf{k}) |$ -
ok?

The density matrix $P(\mathbf{k})$ and the overlap matrix $S(\mathbf{k})$ are sufficient to perform population analysis of the electronic distribution. In Mulliken analysis [2] the charge associated with a given atom A is given by

$$Q(A) = \sum_{\mathbf{k}} w_{\mathbf{k}} \sum_{\mu}^{onA} \sum_{\nu} P_{\mu\nu}(\mathbf{k}) S_{\nu\mu}(\mathbf{k}) , \quad (15)$$

and the overlap population between two atoms A and B is .

$$n(AB) = \sum_{\mathbf{k}} w_{\mathbf{k}} \sum_{\mu}^{onA} \sum_{\nu}^{onB} 2P_{\mu\nu}(\mathbf{k}) S_{\nu\mu}(\mathbf{k}) , \quad (16)$$

The weight of a band on a given orbital may be calculated by simply projecting the band onto the selected orbital. So, the weight of band α on orbital μ is given by

$$W_{\alpha\mu}(\mathbf{k}) = \langle \psi_{\alpha}(\mathbf{k}) | \phi_{\mu}(\mathbf{k}) \rangle \langle \phi^{\mu}(\mathbf{k}) | \psi_{\alpha}(\mathbf{k}) \rangle . \quad (17)$$

The contribution of each band α to the density of states is multiplied by $W_{\alpha\mu}(\mathbf{k})$ to get the projected density of states for orbital μ .

7.8.3 The LCAO Basis Set

In this case, the natural choice of basis set is that of pseudo-atomic orbitals, generated from the pseudopotentials used in the electronic structure calculation. These are generated automatically when population analysis is performed.

By default, the basis set used on an atom of a given species is that of the orbitals in the closed valence shell of that species. This is usually a sufficiently good basis set for the purposes of population analysis and partial density of states projection. If the spilling parameter indicates that the basis set does not represent the PW states with sufficient accuracy, the number of orbitals in the atomic basis set may be increased. A spilling parameter of the order of a few percent or less is sufficiently good for population analysis and partial density of states projection. Care should be exercised in increasing the basis set dramatically, as this can lead to artificial transfers of charge.

The user may change the number of orbitals used in the basis set of a species by altering the NORBS parameter in the `.rundat` file, corresponding to the required element. The entry for a species should be such that it results in a complete set of angular momentum states. For example, to use a single s and single p state NORBS must be 4 ($1 \times 1 + 1 \times 3$). The orbitals will be chosen in order of energy, lowest first. If the number of orbitals requested does not correspond to a full set of orbitals for the lowest energy states, the highest energy state will not be used.

The orbitals are generated by solving for the lowest energy eigenstates of the pseudopotential in a spherical box, using a spherical Bessel basis set. Fine control may be exercised over this procedure using a file '<seedname>.batim', which may contain optional parameters. The available parameters are:

```
RBOX          ! size of spherical box (default 15 a.u.)
ATOM_NCONJG  ! The number of conjugate gradient steps used in minimisation
procedure (default 6).
ATOM_MIX     ! Mixing parameter of potential (default species dependent)
ATOM_THRESH  ! The convergence criterion (in Ryd)
ATOM_DEBUG   ! Print extra information.
DELTA_[SPDF] ! Change occupations (default experimental ground state).
```

The parameters may be set globally

```
25 RBOX
0.7 ATOM_MIX
```

or species-by-species,

```
25 RBOX_Li
10 RBOX_H
```

Species-by-species assignments take precedence over the global values and later entries in the file take precedence over earlier entries.

In some cases, for example an impurity atom in an interstitial region of a crystal, it may be necessary to compress the atomic orbitals in order to achieve a good representation of the orbitals of the impurity atom. This can be achieved by reducing the size of the spherical box used to solve for the orbitals by reducing RBOX for that species.

Please note that the .batim file is used only very rarely and care should be exercised.

7.8.4 Input Parameters

The parameters controlling population analysis and Partial Density of States are set in the .rundat file.

7.8.5 Interpreting the Results

It is widely accepted that the absolute magnitude of the atomic charges yielded by population analysis have little physical meaning, as they display a high degree of sensitivity to the atomic basis set with which they were calculated [3]. However, consideration of their relative values can yield useful information [4, 5] provided a consistent basis set is used for their calculation.

It should be ensured that the spilling parameter is of the order of a few

percent or less in order to ensure that the results are reliable. If this is not the case, the number of orbitals in the basis set should be increased with care. If this does not significantly improve the spilling parameter, it may be that atomic orbitals do not provide a good representation of some of the states in the system.

Here is an example of the output for the population analysis of a calculation performed on a water molecule.

```

Spilling parameter= 2.85%
Atomic Populations
-----
Species   Ion      s      p      d      f      Total Charge (e)
=====
O         1      1.85   5.38   0.00   0.00   7.23   -1.23
H         1      0.39   0.00   0.00   0.00   0.39   0.61
H         2      0.39   0.00   0.00   0.00   0.39   0.61
Overlap Populations
-----
=====
Bond      Population      Length
=====
O 1-- H 2      0.50      0.85000
O 1-- H 1      0.50      0.85000
H 1-- H 2     -0.17      1.38804
=====

```

The first line indicates the spilling parameter for the projection, in this case just under 3%. The following table lists the electronic population on each atom in the system. This is decomposed into the different angular momentum channels. The final two columns give the total electronic population on the atom and the net charge on the atom. The second table contains the bond, or overlap, populations between each pair of atoms separated by a distance less than the CUTOFF parameter in the .rundat file. A positive value indicates a bonding interaction between the atoms, a negative value anti-bonding.

If a spin-dependent calculation was performed, a spilling parameter for each spin component will be reported and values for the net spin population of each atom and bond will also be given.

In crystalline systems with small unit cells, care must be taken in interpreting the bond populations. If an atom interacts with more than one periodic image of another atom, the reported bond population will be the sum of the interactions. For example, compare the bond populations for a calculation performed on the 8-atom cubic cell of Silicon, with that of the primitive 2-atom cell.

Si, 8-atom unit cell			Si, 2-atom unit cell		
Overlap Populations			Overlap Populations		
-----			-----		
=====			=====		
Bond	Population	Length	Bond	Population	Length
=====			=====		
Si 1--Si 6	0.75	2.35156	Si 1--Si 2	3.03	2.35156
Si 2--Si 5	0.75	2.35156	=====		
Si 2--Si 6	0.75	2.35156			
Si 2--Si 7	0.75	2.35156			
Si 2--Si 8	0.75	2.35156			
Si 3--Si 6	0.75	2.35156			
Si 3--Si 8	0.75	2.35156			
Si 4--Si 5	0.75	2.35156			
Si 4--Si 6	0.75	2.35156			
Si 4--Si 7	0.75	2.35156			
Si 4--Si 8	0.75	2.35156			
Si 1--Si 5	0.75	2.35156			
Si 1--Si 7	0.75	2.35156			
Si 1--Si 8	0.75	2.35156			
Si 3--Si 5	0.75	2.35156			
Si 3--Si 7	0.75	2.35156			
=====					

Here we can see that the bond population in the 2-atom cell appears to be four times greater than that in the 8-atom cell. However, in the 2-atom cell, each atom is bonded to the other atom in the cell, plus three of its periodic images. Thus the true bond population must be divided by four, giving a bond population of 0.76 e , in good agreement with the result from the 8-atom cell.

In addition to providing an objective criterion for bonding between atoms, the overlap population may be used to assess the covalent or ionic nature of a bond. A high value of the bond population indicates a covalent bond, whereas a low value indicates an ionic interaction. A further measure of ionicity may be obtained from the effective ionic valence, which is defined as the difference between the formal ionic charge and the Mulliken charge on the anion species. A value of zero indicates a perfectly ionic bond, while values greater than zero indicate increasing levels of covalency. See Ref. [5] for more details.

The PDOS projection may be used to obtain a qualitative measure of the contribution of a selected set of orbitals to the density of states of a system. Any angular momentum channel on an atom, or combinations thereof, may be chosen. A local density of states projection, indicating the contribution of all states on a given atom to the density of states may be obtained by selecting all angular momentum channels on that atom.

References

- [1] D. Sanchez-Portal, E. Artacho and J.M. Soler, "Projection of plane-wave calculations into atomic orbitals". *Solid State Commun.* **95** (1995) 685-690
- [2] R.S. Mulliken. "Electronic population analysis on LCAO-MO molecular wave functions I". *J. Chem. Phys.* **23** (1955) 1833-1840
- [3] E.R. Davidson and S. Chakravorty, "A test of the Hirshfeld definition of atomic charges and moments". *Theor. Chim. Acta.* **83** (1992) 319-330
- [4] M.D. Segall, C.J. Pickard, R. Shah and M.C. Payne, "Population analysis in plane wave electronic structure calculations". *Mol. Phys.* **89** (1996) 571-577
- [5] M.D. Segall, R. Shah, C.J. Pickard and M.C. Payne, "Population analysis of plane wave electronic structure calculations of bulk materials". *Phys. Rev. B* **54** (1996) 16317-16320

Part III

For Reference

8 Publications about methodology

8.1 Scientific reviews and overviews

- M. Parrinello, “From silicon to RNA: the coming of age of *ab initio* molecular dynamics”, Sol. State Comm. **102** (1997) 107

8.2 Density-functional theory

- “Density-Functional Theory of Atoms and Molecules”, International Series of Monographs on Chemistry - 16 Robert G. Parr and Weitao Yang, Oxford University Press (1994, ISBN:0195092767)
- “A Chemist’s Guide to Density Functional Theory (An Introduction and Practical Guide to DFT Calculations)”, Wolfram Koch and Max C. Holthausen, Wiley (2000, ISBN:3527299181)
- R. Jones and O. Gunnarsson, Rev. Mod. Phys. **61** (1989) 689
- G. P. Srivastava and D. Weaire, Adv. Phys. **36** (1987) 463
- P. Hohenberg and W. Kohn, Phys. Rev. **136** (1964) B864
- W. Kohn and L. Sham, Phys. Rev. **140** (1965) A1133

8.3 Functionals

- D. M. Ceperley and B. J. Alder, Phys. Rev. Lett. **45** (1980) 566
- J. P. Perdew, Phys. Rev. B **34** (1986) 7406(E)
(Becke-Perdew GGA functional)
- J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, D. J. Singh and C. Fiolhais, Phys. Rev. B **46** (1992) 5571
(“Perdew-Wang II” GGA)
- J. P. Perdew, K. Burke and M. Ernzerhof, ACS Symposium series **629** (1996) 453
(“PBE” GGA)

8.4 Methodology

- M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias and J. D. Joannopoulos, “Iterative minimization techniques for *ab initio* total energy calculations: molecular dynamics and conjugate gradients”, *Rev. Mod. Phys.* **64** (1992) 1045
- D. K. Remler and P. A. Madden, “Molecular dynamics without effective potentials via the Car-Parrinello approach”, *Mol. Phys.* **70** (1990) 921
- R. Car and M. Parrinello, *Phys. Rev. Lett* **55** (1985) 2471
- Dominik Marx and Jürg Hutter, in “Modern Methods and Algorithms of Quantum Chemistry”, published by the John von Neumann Institute for Computing (NIC), pp 301 ff (2000). Available on-line at <http://www.fz-juelich.de/wsqc/proceedings/>

8.5 k-points

- H. J. Monkhorst and J. D. Pack, *Phys. Rev.* **B 13** (1976) 5188
- D. J. Chadi and M. L. Cohen, *Phys. Rev.* **B 8** (1973) 5747

8.6 Pseudopotentials

- Andrew M. Rappe, Karin M. Rabe, Efthimios Kaxiras and J. D. Joannopoulos, “Optimized pseudopotentials”, *Phys. Rev.* **B 43** (1990) 6796
- J. S. Lin, A. Qteish, M. C. Payne and V. Heine, “Optimized and transferable nonlocal separable *ab initio* pseudopotentials”, *Phys. Rev.* **B 47** (1993) 4174
- Steven G. Louie, Sverre Froyen and Marvin L. Cohen, “Nonlinear ionic pseudopotentials in spin-density-functional calculations”, *Phys. Rev.* **B 26** (1982) 1738
- David Vanderbilt, “Soft self-consistent pseudopotentials in a generalized eigenvalue formalism”, *Phys. Rev.* **B 41** (1990) 7892
- Kari Laasonen, Roberto Car, Changyol Lee and David Vanderbilt, “Implementation of ultrasoft pseudopotentials in *ab initio* molecular dynamics”, *Phys. Rev.* **B 43** (1991) 6796

- Kari Laasonen, Alfredo Pasquarello, Roberto Car, Changyol Lee and David Vanderbilt, “Car-Parrinello molecular dynamics with Vanderbilt ultrasoft pseudopotentials”, Phys. Rev. **B** 47 (1993) 10142

8.7 Density mixing

- G. Kresse and J. Furthmüller, “Efficient schemes for *ab initio* total-energy calculations using a plane-wave basis set”, Phys. Rev. **B** 54 (1996) 11169
- D. R. Bowler and M. J. Gillan, Chem. Phys. Lett. **325** (2000) 473-476

8.8 Special techniques for metals

- M. J. Gillan, “Calculation of the vacancy formation energy in aluminium”, J. Phys. Condens. Matter **1** (1989) 689
- N. Marzari, D. Vanderbilt and M. C. Payne, “Ensemble density-functional theory for *ab initio* molecular dynamics of metals and finite-temperature insulators”, Phys. Rev. Lett. **79** (1997) 1337

8.9 Molecular dynamics

- Daan Frenkel and Berend Smit, “Understanding Molecular Simulation: from Algorithms to Applications”, Academic Press, 1996
- M. P. Allen and D. J. Tildesley”, “Computer Simulation of Liquids”, Oxford University Press, 1987
- Dominik Marx and Jürg Hutter, in “Modern Methods and Algorithms of Quantum Chemistry”, published by the John von Neumann Institute for Computing (NIC), pp 301 ff (2000). Available on-line at <http://www.fz-juelich.de/wsqc/proceedings/>

8.10 Using parallel computers

- L. J. Clarke, I. Stich and M. C. Payne, “Large-scale *ab initio* total-energy calculations on parallel computers”, Comp. Phys. Comm. **72** (1992) 14
- C. Cavazzoni and G. L. Chiarotti, “A parallel and modular deformable cell Car-Parrinello code”, Comp. Phys. Comm. **123** (2000) 56

9 Some UKCP publications

1. I. Štich, M. C. Payne, R. D. King-Smith, J.-S. Lin and L. J. Clarke, “Ab initio total energy calculations for extremely large systems: application to the Takayanagi reconstruction of Si (111)”, *Phys. Rev. Lett.*, **68** 1351 (1992)
2. A. De Vita, M. J. Gillan, J.-S. Lin, M. C. Payne, I. Štich and L. J. Clarke, “Defect energetics in oxide materials from first principles”, *Phys. Rev. Lett.*, **68**, 3319 (1992)
3. L. J. Clarke, I. Štich and M. C. Payne, “Large-scale ab initio total energy calculations on parallel computers”, *Comput. Phys. Commun.*, **72**, 14 (1992)
4. M. C. Payne, I. Štich, R. D. King-Smith, J.-S. Lin and L. J. Clarke, “Ab initio total energy calculations on parallel computers: application to the Takayanagi reconstruction”, *Phys. Scripta*, **T 45**, 265 (1992)
5. A. De Vita, I. Manassidis, J.-S. Lin and M. J. Gillan, “The energetics of Frenkel defects in Li₂O from first principles”, *Europhys. Lett.*, **19**, 605 (1992)
6. A. De Vita, M. J. Gillan, J.-S. Lin, M. C. Payne, I. Štich and L. J. Clarke, “Defect energetics in MgO treated by first-principles methods”, *Phys. Rev. B*, **46**, 12964 (1992)
7. J. R. K. Bigger, D. A. McInnes, A. P. Sutton, M. C. Payne, I. Štich, R. D. King-Smith, D. M. Bird and L. J. Clarke, “The atomic and electronic structures of the 90° partial dislocation in silicon”, *Phys. Rev. Lett.*, **69**, 2224 (1992)
8. M. C. Payne, L. J. Clarke and I. Štich, “Role of parallel architectures in periodic boundary calculations”, *Phil Trans. Roy. Soc.*, A **341**, 211 (1992)
9. D. M. Bird, L. J. Clarke, R. D. King-Smith, M. C. Payne, I. Štich and A. P. Sutton, “First-principles calculation of the structure and energy of Si (113)”, *Phys. Rev. Lett.*, **69**, 3785 (1992)
10. I. Manassidis, A. De Vita and M. J. Gillan, “Structure and energetics of the (0001) surface of alpha-alumina from first-principles calculations”, *Surf. Sci. Lett.*, **285**, L517 (1993)
11. I. Manassidis, A. De Vita and M. J. Gillan, “Parallel computation and the structure of catalyst surfaces”, *Proc Int. Conf. ‘Physics Computing ’92’, Prague, August 1992* (Singapore, World Scientific 1993), p. 397

12. I. Štich, M. C. Payne, A. De Vita and M. J. Gillan, "First-principles dynamics used to study dissociative chemisorption", *Proc. Int. Conf. 'Physics Computing '92', Prague, August 1992* (Singapore, World Scientific, 1993), p. 477
13. M. C. Payne, I. Štich, R. D. King-Smith, J.-S. Lin, A. De Vita, M. J. Gillan and L. J. Clarke, "Ab initio simulations on parallel computers", *'Computer Aided Innovation of New Materials'*, ed. M. Doyama, J. Kihara, M. Tanaka and R. Yamamoto (Elsevier, 1993), p. 101
14. I. Štich, M. C. Payne, A. De Vita, M. J. Gillan and L. J. Clarke, "First-principles studies of semiconductor surfaces: reconstruction and dissociative chemisorption", *Proc. NATO ARW on 'Computations for the Nanoscale'*, NATO ASI Series E, Vol. 240 (1993), p. 63
15. A. De Vita, I. Štich, M. J. Gillan, M. C. Payne, R. D. King-Smith and L. J. Clarke, "Dynamics of dissociative chemisorption: Cl₂/Si(111)-2×1", *Phys. Rev. Lett.*, **71**, 1276 (1993)
16. J. Crain, G. J. Ackland, R. O. Piltz and P. D. Hatton, "Phase-transition-induced defect formation in III-V semiconductors", *Phys. Rev. Lett.*, **70**, 814 (1993)
17. M. J. Gillan, "Gulliver among the atoms", *New Scientist*, 3rd April 1993, p. 34
18. M. J. Gillan, "First applications of a new approach", *Europhys. News*, **23**, 176 (1992)
19. I. Štich, M. C. Payne, A. De Vita, M. J. Gillan and L. J. Clarke, "Chemically driven molecular decomposition at semiconductor surfaces", *Chem. Phys. Lett.*, **212**, 617 (1993)
20. I. Štich, A. De Vita, M. C. Payne, M. J. Gillan and L. J. Clarke, "Surface dissociation from first principles: dynamics and chemistry", *Phys. Rev. B*, accepted
21. I. Manassidis and M. J. Gillan, "The structure and energetics of alumina surfaces calculated from first principles", *J. Amer. Ceram. Soc.*, **77**, 335 (1994)
22. M. J. Gillan, I. Manassidis and A. De Vita, "The ab initio energetics of oxides", *Phil. Mag. B*, accepted
23. D. M. Bird, L. J. Clarke, M. C. Payne and I. Štich, "Dissociation of H₂ on Mg (0001)", *Chem. Phys. Lett.*, **212**, 518 (1993)
24. J. A. White and D. M. Bird, "Dissociative adsorption of H₂ on Cu (100)", *Chem. Phys. Lett.*, **213**, 422 (1993)

25. J. Crain, S. J. Clark, G. J. Ackland, M. C. Payne, V. Milman, P. D. Hatton and B. J. Reid, "Theoretical study of high-density phases of covalent semiconductors: I. Ab initio treatment", *Phys. Rev. B*, **49**, 5329 (1994)
26. E. S. Fois, J. N. Penman and P. A. Madden, "Self-interaction corrected density functionals and the structure of metal clusters", *J. Chem. Phys.*, **98**, 6352 (1993)
27. E. S. Fois, J. N. Penman and P. A. Madden, "Control of the adiabatic ground state in ab initio molecular dynamics", *J. Chem. Phys.*, **98**, 6361 (1993)
28. J. Crain, G. J. Ackland, P. D. Hatton, M. I. McMahon, R. J. Nelmes and R. O. Piltz, "Image plate studies of tetrahedral semiconductors under pressure: implications for experiment and theory", *Jap. J. Appl. Phys.*, **32**, 36 (1993)
29. H. Akbarzadeh, S. J. Clark and G. J. Ackland, "Theoretical study of selenium I under high pressure", *J. Phys.: Condens. Matter*, **5**, 8065 (1993)
30. J. Crain, G. J. Ackland, J. R. Maclean, R. O. Piltz, P. D. Hatton and G. S. Pawley, "Reversible pressure-induced structural transitions between metastable phases of silicon" *Phys. Rev. B* **50**, 13043 (1994).
31. F. Kirchhoff, J. Holender and M. J. Gillan, "The energetics and electronic structure of silver chloride", *Phys. Rev. B*, **49**, 17420 (1994)
32. I. Manassidis and M. J. Gillan, "First-principles study of SnO₂ (110)", *Proc. 10th Interdisciplinary Surface Science Conference, Liverpool, March 1994*, ed. S. Barrett, accepted
33. J. Crain, G. J. Ackland, S. J. Clark, Y. H. Nam, P. D. Hatton, J.-S. Lin, M. C. Payne and R. O. Piltz, "Tetrahedral structures and phase transitions in III-V semiconductors", *Phys. Rev. B*, **50**, 8389 (1994)
34. C. P. Ursenbach and P. A. Madden, "Cluster formation in alkali-doped zeolite-Y: an ab initio study", in preparation
35. J. A. White, D. M. Bird, M. C. Payne and I. Štich, "Surface corrugation in the dissociative adsorption of H₂ on Cu (100)", *Phys. Rev. Lett.* **73** 1404 (1994)
36. C. M. Goringe, L. J. Clark, M. S. Lee, M. C. Payne, I. Stich, J. A. White, M. J. Gillan and A. P. Sutton, "GaAs(001)-(2 × 4) Structure, Chemistry and Adsorbates" *J. Phys. Chem.* Invited feature article. In preparation.

37. S. Pugh and M. J. Gillan, "The energetics of NH_3 adsorption at the MgO (001) surface", *Surf. Science*, **320**, 331-43 (1994)
38. L. N. Kantorovich, J. M. Holender and M. J. Gillan, "The energetics and electronic structure of defective and irregular surfaces of MgO", *Surf. Sci.*, accepted.
39. S.J.Clark, G.J.Ackland, J.Crain and M.C.Payne, "Very low energy surface of silicon", *Phys. Rev. B* **50**, 5728 (1994)
40. G.J.Ackland, "Pressure induced symmetry breaking in tetrahedral networks" *Phys. Rev. B*, **50**, 7389 (1994).
41. S.J.Clark and G.J.Ackland, "Vibrational and elastic effects of point defects in silicon", *Phys. Rev. B* **48**, 10899 (1993).
42. J.Crain, G.J.Ackland, J.R.Maclean, R.O.Piltz, P.D.Hatton and G.S.Pawley, "Reversible pressure-induced structural transitions between metastable phases of silicon", *Phys. Rev. B* **50**, 13043 (1994).
43. S.J.Clark, G.J.Ackland and H.Akbarzadeh, "A theoretical study of pressure effects on selenium", *J.Phys.Chem. Solids* **56**, 329 (1995)
44. J.Crain, S.J.Clark and G.J.Ackland, "Dense tetrahedral structures of compound semiconductors", *J.Phys.Chem. Solids* **56**, 495 (1995)
45. R.O.Piltz, J.R.Maclean, S.J.Clark, G.J.Ackland, P.D.Hatton and J.Crain, "The Structure and Properties of Silicon XII: A Complex Tetrahedrally-bonded Phase", *Phys. Rev. B* **52**, 4072 (1995)
46. H-C Hsueh, J.R.Maclean, G.Y.Guo, M-H. Lee, S.J.Clark, G.J.Ackland and J.Crain, "Pressure-induced polymorphism in CuCl: An ab initio study", *Phys. Rev. B* **51** 12216 (1995)
47. H-C Hsueh, H.Vass, S.J.Clark, G.J.Ackland and J.Crain, "High-Pressure effects in the layered semiconductor Germanium Selenide", *Phys. Rev. B* **51**, 16750 (1995)
48. S.J.Clark, G.J.Ackland and J.Crain, "Theoretical stability limit of diamond at ultra-high pressure", *Phys. Rev. B* **52**, 15035 (1995)
49. S.J.Clark, G.J.Ackland and J.Crain, "A comparison of bonding in amorphous carbon and silicon", *Phys.Rev.B* **55** (1997) 14059
50. J.A White, D.M. Bird and M. C. Payne, "Dissociation of H_2 on W(100)" *Phys. Rev. B*, **53** 1667 (1996)
51. J.A. White and D.M. Bird, "Implementation of Gradient-Corrected Exchange-Correlation Potentials in Car-Parrinello Total Energy Calculations", *Phys. Rev. B* **50** 4954 (1994)

52. Ruben Perez and Michael C. Payne, "First-Principles Simulations of Silicon Nanoindentation", *Phys. Rev. Lett.*, **75** 4748 (1995)
53. F. Kirchhoff, J. M. Holender and M. J. Gillan, "Complex ion formation in liquid Ag-Se alloys", submitted to *Phys. Rev. Lett.*
54. F. Kirchhoff, J. M. Holender and M. J. Gillan, "The liquid Ag-Se system studied by ab initio simulations", *Proc. 9th Int. Conf. on Liquid and Amorphous Metals, J. Non-Cryst. Solids*, **205-207**, 79 (1996)
55. C.M. Goringe and A. P. Sutton, "Tight binding and local density approximation studies of semiconductor surfaces and small organic adsorbates" *Mat. Sci. Eng. B* **37** 150 (1996)
56. J. M. Holender, M. J. Gillan, M. C. Payne and A. D. Simpson, "Static, dynamic and electronic properties of liquid gallium studied by first-principles simulation", *Phys. Rev. B* **52**, 967 (1995).
57. Philip Lindan and Nic Harrison, "And the rest is chemistry...", HPCProfile **1** 4 (March 1995)
58. E. Hernandez, M. J. Gillan and C. M. Goringe, "Towards a linear-scaling DFT technique: the density matrix approach", *Phys. Rev. B*, **53**, 7147 (1996)
59. R. Shah, A. De Vita and M. C. Payne, "Ab initio study of tritium defects in lithium oxide", *J. Phys.:Cond. Matter*, **6**, 6891 (1995)
60. R. Shah, J. D. Gale and M. C. Payne, "Acid-base catalysis in zeolites from first principles", *Int. J. Quantum Chem.*, **61**, 393 (1997)
61. M. Kay, G. R. Darling, S. Holloway, J. A. White and D. M. Bird, "Steering Effects in Non-Activated Adsorption", *Chem. Phys. Lett.*, **245** 311 (1995)
62. P. A. Gravil, J. A. White and D. M. Bird, "Chemisorption of O₂ on Ag(110)", *Surf. Sci.*, **352-354** 248 (1996)
63. R. Shah, A. De Vita, V. Heine and M. C. Payne, "The mechanism of tritium diffusion in lithium oxide", *Phys. Rev. B*, **53** 8257 (1996)
64. P. J. D. Lindan, N. M. Harrison, J. Holender, M. J. Gillan and M. C. Payne, "The TiO₂ (100) 1x3 reconstruction: Insights from ab initio calculations", *Surf. Sci.* **364** 431 (1996)
65. sP. J. D. Lindan, N. M. Harrison, J. Holender and M. J. Gillan, "First-principles molecular dynamics simulation of water dissociation on TiO₂ (110)", *Chem. Phys. Lett.* **261** 246-252 (1996)

66. P. J. D. Lindan, N. M. Harrison, J. A. White and M. J. Gillan, "First-principles spin-polarised calculations on the reduced and reconstructed TiO₂ (110) surface", *Phys. Rev. B* **55** (1997) 15919
67. R. Shah, M. C. Payne, M. H. Lee and J. D. Gale, "Understanding the catalytic behaviour of zeolites: first principles study of adsorption of methanol", *Science*, **271**, 1395 (1996)
68. R. Shah, J. D. Gale and M. C. Payne, "Methanol adsorption in zeolites - a first principles study" *J. Phys. Chem.*, **100**, 11688 (1996)
69. J. Crain, G. J. Ackland and S. J. Clark, "Exotic Structures of tetrahedral semiconductors" (Invited review article), *Rep. Prog. Phys.*, **58**, 705 (1995)
70. M. C. Warren and G. J. Ackland, "Ab initio studies of structural instabilities in magnesium silicate perovskite", *Phys. Chem. Minerals* **23**, 107 (1996).
71. H. C. Hsueh, M. C. Warren, H. Vass, G. J. Ackland, S. J. Clark and J. Crain, "Vibrational properties of the layered semiconducting germanium sulfide under hydrostatic pressure", *Phys. Rev. B*, **53** 14806 (1996)
72. B. B. Karki, L. Stixrude, S. J. Clark, M. C. Warren, G. J. Ackland and J. Crain, "Structure and elasticity of MgO at high pressure", *American Mineralogist* **82** (1997) 52
73. S. J. Clark, C. J. Adam, J. A. White, G. J. Ackland and J. Crain, "Properties of liquid crystal molecules by first-principles calculation", *Liquid Crystals*, in press.
74. G. J. Ackland, M. C. Warren and S. J. Clark, "Practical methods in ab initio lattice dynamics", *J. Phys. Condens. Matt.* **9** (1997) 7861
75. R. Shah, J. D. Gale and M. C. Payne, "Comparing the acidities of zeolites and SAPOs from first principles", *Chem. Comm. Chem. Commun.*, **1**, 131 (1997)
76. R. Shah, J. D. Gale and M. C. Payne, "The active sites of microporous solid acid catalysts" *Phase Transitions B*, submitted for publication, (1996)
77. I. Dawson, P. D. Bristowe, M. H. Lee, M. C. Payne, M. D. Segall and J. A. White, "A first principles study of a tilt grain boundary in rutile", *Phys. Rev. B* **54**, 13727 (1996)
78. M. Segall, R. Shah, C. J. Pickard, M. C. Payne, I. Dawson and P. D. Bristowe, "Population analysis of plane wave electronic structure calculations of bulk materials", *Phys. Rev. B* in press (1996)

79. P. A. Gravil, D. M. Bird and J. A. White, "Adsorption and Dissociation of O₂ on Ag(110)", *Phys. Rev. Lett.*, **77**, 3933 (1996)
80. D. M. Bird and P. A. Gravil, "First Principles Calculations of Molecular Dissociation at Surfaces", *Surf. Sci.*, submitted
81. R. Perez, M. C. Payne, I. Stich and K. Terakura, "First Principles Simulations of Nanoindentation and Atomic Force Microscopy on Silicon Surfaces", *Materials Theory, Simulations and Parallel Algorithms*, Editors: E. Kaxiras, J. Joannopoulos, P. Vashishta and R. K. Kalia, MRS Symposia Proceedings, Vol. **409** (MRS, Pittsburgh, 1996). (ISBN: 1-55899-311-8)
82. R. Perez, M. C. Payne, I. Stich and K. Terakura, "Role of Tip-Surface Covalent Interactions in Non-Contact Atomic Force Microscopy on Reactive Surfaces", *Phys. Rev. Lett.*, submitted
83. M. J. Gillan, L. N. Kantorovich and P. J. D. Lindan, "Modelling of oxide surfaces", *Current opinion in solid-state and materials science* **1** 820 (1996)
84. P. J. D. Lindan, N. M. Harrison and M. J. Gillan, "Paint by numbers: Oxide surface chemistry using first-principles molecular dynamics", *Psi-k Network newsletter* **17** (December 1996)
85. F. Kirchhoff, J. Holender and M. J. Gillan, "The structure, dynamics and electronic structure of liquid Ag-Se alloys investigated by ab initio simulation", *Phys. Rev. B*, **54**, 190 (1996)
86. L. N. Kantorovich, M. J. Gillan and J. A. White, "Adsorption of atomic oxygen on the MgO (100) surface", *J. Chem. Soc. Faraday Trans.*, **92**, 2075 (1996)
87. L. N. Kantorovich and M. J. Gillan, "Adsorption of atomic and molecular oxygen on the MgO and CaO (001) surfaces", *Proc. 13th Int. Conf. on Defects in Insulating Materials*, ed. R. T. Williams and G. E. Matthews, *Materials Science Forum*, **239-241**, 637 (1997)
88. C. M. Goringe, L. J. Clarke, M.-H. Lee, M. C. Payne, I. Stich, J. A. White, M. J. Gillan and A. P. Sutton, "GaAs (001)-(2 × 4) structure, chemistry and adsorbates", *J. Phys. Chem.*, submitted (invited feature article)
89. C. M. Goringe and A. P. Sutton, "Tight binding and local density approximation studies of semiconductor surfaces and small organic adsorbates", *Mat. Sci. Eng. B*, **37**, 150 (1996)

90. M. J. Gillan, "Quantum simulations of catalysis", *Physics World*, July 1996, p. 31
91. I. Dawson, P. D. Bristowe, M. C. Payne and M.-H. Lee, "An ab initio investigation of a grain boundary in a transition-metal oxide", *Mat. Res. Soc. Proc.*, **408**, 271 (1996)
92. I. Dawson, P. D. Bristowe, J. A. White and M. C. Payne, "First-principles simulation of the defect chemistry of rutile TiO₂" *Mat. Res. Soc. Proc.*, **409**, 453 (1997)
93. G. J. Ackland and M. C. Warren, "Soft-mode phase transitions from first principles", *Phase Transitions*, accepted.
94. B. B. Karki, S. J. Clark, M. C. Warren, G. J. Ackland and J. Crain, "Ab initio elasticity and lattice dynamics of AgGaSe₂", *J. Phys. Condens. Matter*, **9**, 375 (1997)
95. S.J.Clark, C.J.Adam, G.O.Bodammer, J.A.White, G.J.Ackland and J.Crain, "Structural and vibrational properties of large molecules from first principles: application to 5CB", *Liquid Crystals* **22** (1997) 469
96. C. M. Goringe, E. Hernández, C. M. Goringe and I. J. Bush, "Linear-scaling DFT-pseudopotential calculations on parallel computers", *Comput. Phys. Commun.*, submitted.
97. E. Hernández, C. M. Goringe and M. J. Gillan, "Basis functions for linear-scaling first-principles calculations", *Phys. Rev. B*, accepted
98. P. J. D. Lindan, N. M. Harrison and M. J. Gillan, "Mixed dissociative and molecular adsorption of water on the TiO₂ (110) surface", *Phys. Rev. Lett.* **80**, 762 (1998)
99. M. J. Gillan, P. J. D. Lindan, L. N. Kantorovich and S. P. Bates, "Molecular processes on oxide surfaces studied by first-principles calculations" *Mineral. Mag.* **62** (1998) 669
100. B.B.Karki, M.C.Warren, L.Stixrude, G.J.Ackland, and J.Crain, "Ab initio studies of high-pressure structural transformations in silica" *Phys. Rev. B* **55** (1997) 3465
101. S.J.Clark, C.J.Adam, G.J.Ackland, and J.Crain, "Conformation energy surface for liquid crystal molecules from first principles: Application to 5CB" *Molecular Crystals and Liquid Crystals Science and Technology A* **299** (1997) 39
102. G.J.Ackland and M.C.Warren, "Soft-mode phase transitions from first principles" *Phase Transitions* **61** (1997) 215

103. S.J.Clark and G.J.Ackland, "Ab initio calculations of the self interstitial in silicon", *Phys.Rev.B* **56** (1997) 47
104. C.J.Adam, S.J.Clark, G.J.Ackland and J.Crain, "Conformation-dependent dipoles of liquid crystal molecules and fragments from first principles", *Phys.Rev.E* **55** (1997) 5641
105. B.B.Karki, L.Stixrude, S.J.Clark, M.C.Warren, G.J.Ackland and J.Crain, "Elastic properties of orthorhombic MgSiO_3 perovskite at lower mantle pressures", *Am. Mineral.***82** (1997) 635
106. S.J. Clark, C. Adam, G.J. Ackland and J. Crain, " Conformation energy surface for liquid crystal molecules from first principles: Application to 5CB", *Mol. Cryst. and Liq. Cryst. Sci & Tech* **299** (1997) 39
107. S.J.Clark, G.J.Ackland, and J.Crain, "Electronic structure calculations of liquid crystal molecules: Application to chiral solutes", *Liquid Crystals: Physics, Technology and Applications*, 1998, Vol.3318, Ch.99, pp.166-170 (1998)
108. B.B.Karki, G.J.Ackland and J.Crain, "Elastic instabilities in crystals from ab initio stress-strain relations", *J.Phys. Condens. Matt.* **9** (1997) 8579
109. A.A.Kelsey, G.J.Ackland and S.J.Clark, "Stability and electronic structure of the cinnabar phase in GaAs", *Phys.Rev.B* **57** (1998) R2029
110. D.R.Allan, A.A.Kelsey, S.J.Clark, R.J.Angel and G.J.Ackland, "High pressure semiconductor - semimetal transition in TiS_2 " *Phys.Rev.B* **57** (1998) 5106
111. G.J.Ackland, "Embrittlement and the bistable crystal structure of zirconium hydride", *Phys.Rev.Lett.* **80** (1998) 2233
112. S.J.Clark, G.J.Ackland, and J.Crain, "Ab initio molecular polarisabilities of liquid crystals: Application to DOBAMBC and 5CB", *Europhysics Letters* **44** (1998) 578
113. C.J. Adam, S.J. Clark, M.R. Wilson, J. Crain and G.J.Ackland, "Transferability of liquid crystal intra-molecular potentials" *Molecular Physics*, **93** (1998) 947
114. H.C.Hsueh, R.K.Chen, H.Vass, S.J.Clark, G.J.Ackland, W.C-K.Poon, and J.Crain, "Compression mechanisms in quasimolecular XI3 (X = As, Sb, Bi) solids", *Physical Review B* **58** (1998) 14812

115. D.R.Allan, S.J.Clark, M.J.P.Brugmans, G.J.Ackland and W.L.Vos, "Structure of crystalline methanol at high pressure" *Phys.Rev.B* **58** (1998) 11809
116. A.A.Kelsey and G.J.Ackland, "Simulation of microscopic strain in semiconducting heterojunctions", *Modelling and Simulation in Materials Science and Engineering* **7** (1999) 311
117. C.J.Adam, A.Ferrarini, M.R.Wilson, G.J.Ackland and J.Crain, "A first principles and mean field investigation of the conformational properties of 5CB", *Mol.Phys.* **97** (1999) 541
118. S.J.Clark, P.R.L.Markwick, G.J.Ackland and J. Crain, "Short time reorientational processes in molecular fluids", *Europhys.Lett* **49** (2000) 34
119. S.K.Reed and G.J.Ackland, "Theoretical and computational study of high pressure structures in barium", *Phys Rev Letters* **84** (2000) 5580
120. A.A.Kelsey and G.J.Ackland, "Computational analysis of the high pressure structures of InSb", *J. Phys. Condens. Matt.* **12** (2000) 32
121. S.P. Bates, G. Kresse, M.J. Gillan, "The Adsorption of Methanol on TiO₂ (110): a First-Principles Investigation", *J. Phys. Chem. B* **102** (1998) 2017-2026
122. S.P. Bates, G. Kresse, M.J. Gillan, "The Adsorption and Dissociation of ROH Molecules on TiO₂ (110)", *Surf. Sci.* **409** (1998) 336-349
123. P.J.D. Lindan, J. Muscat, S.P. Bates, N.M. Harrison, M.J. Gillan, "The *Ab Initio* Simulation of Molecular Processes on Oxide Surfaces" *Faraday Discussions* **106** (1997) 135-154
124. S.P. Bates, G. Kresse, M.J. Gillan, "A Systematic Study of the Surface Energy and Structure of TiO₂ (110) by First-Principles Calculation" *Surf. Sci.* **385** (1997) 386-394
125. Philip J. D. Lindan, "Water adsorption on SnO₂ (110): the role of inter-adsorbate interactions and surface geometry", *Chem. Phys. Lett.* (in press)
126. G. Materzanini, P. Saalfrank and Philip J. D. Lindan, "Quantum size effects in thin metal films: a first-principles study of Pb/Cu(111)", *Phys. Rev. B.* (submitted)
127. Philip J. D. Lindan, "The philosophy and practise of first-principles simulation and its application in surface science", in "Computational

- and visualisation methods in surface analysis”, ed. Kam Chana, Marcel-Dekker Inc. (in press)
128. Philip J. D. Lindan and N. M. Harrison, “The structure of the TiO₂ (100) 1x3 reconstruction”, *Surf. Sci. Lett.* (submitted)
 129. G.J.Ackland, “First principles molecular dynamics”, *New Models and Numerical Codes for Shock Wave Processes in Condensed Media* pub. AWE Hunting Brae (1997) 360
 130. G.J.Ackland, “Elastic properties as a pointer to phase transitions”, *RIKEN review* **29** (2000) 34
 131. G.J.Ackland, “High Pressure Phases of Group IV and III-V Semiconductors”, *Rep.Prog.Phys* (submitted)
 132. D.C.Swift, G.J.Ackland, A.Hauer, J.S.Wark, D.H.Kalantar, G.A.Kyrala and R.Kopp, “Ab initio predictions of material properties compared with time resolved crystallography of laser-shocked silicon” *Proc. 40th APS Plasma Physics Group meeting* New Orleans, USA (1998)
 133. D C Swift, “Ab initio polymorphic equations of state for silicon”, *Proc APS Topical Conference on Shock Compression of Condensed Matter*, Snowbird CO USA (1999)
 134. J. A. Chisholm and P. D. Bristowe, “A first principles investigation of stacking faults in wurtzite materials”, *J. Phys. Cond. Matt.* **11** (1999) 5057
 135. J. A. Chisholm and P. D. Bristowe, “Classical and quantum simulations of In and Al incorporation in GaN”, *Physica Status Solidi (a)* **176** (1999) 257
 136. J. A. Chisholm and P. D. Bristowe, “Stacking fault energies in Si doped GaN: a first principles study”, *Appl. Phys. Lett.* **77**(2000) 534
 137. Jose Antonio Mejias, Andrew Berry, Keith Refson and Donald G. Fraser, “The Kinetics and Mechanism of MgO dissolution”, *Chem. Phys. Lett.* **314** (1999) 558
 138. John P. Brodholt and Keith Refson, “An ab initio study of hydrogen in forsterite and a possible mechanism for hydrolytic weakening”, *J. Geophys. Res.* **105** (2000) 18997
 139. Steven D. Kenny, J. Desmond C. McConnell and Keith Refson, “The stability of low temperature Al/Si ordered albite, NaAlSi₃O₈”, *Amer. Mineral.* **85** (2000) (in press)