Tutorial in Force Field Simulation of Materials with GULP

Julian D. Gale

Nanochemistry Research Institute, Department of Chemistry, Curtin University, Perth, WA 6845, Australia

This tutorial for the Winter School is aimed at illustrating how one can simulate materials using force field methods. This tutorial will use the program GULP, but similar calculations can be performed with other software, including METADISE (Prof. Steve Parker, University of Bath, UK), SHELL (Prof. Neil Allan, University of Bristol, UK) and THBREL (Dr Maurice Leslie, Daresbury Laboratory). The reason for using GULP (apart from personal bias!) is that it is freely available to those in academic institutions from the web site:

http://www.ivec.org/GULP/

Both the source code and executables, examples, documentation, etc can all be found at this web site. To look up any commands you encounter, or to read more about the background theory, go to the page;

https://projects.ivec.org/gulp/help/manuals.html

and select the appropriate link for version 3.4 that will be used in this tutorial.

The results of GULP calculations can be visualised using a number of programs including Materials Studio (Accelrys Inc) and GDis (http://gdis.sf.net).

Tutorials: Overview

Calcium carbonate ($CaCO_3$) is a common mineral that forms through the presence of carbon dioxide into the oceans. At ambient conditions, it exists as three polymorphs, calcite, aragonite and vaterite. In addition, there are high-pressure phases, such as calcite II and III.

As a series of tutorial examples, we will explore the structures, properties, and stabilities of these polymorphs, as well as the surface properties of calcite.

Tutorial 1: Optimisation of polymorphs of calcium carbonate

In the first tutorial we introduce how to run a GULP calculation and look at the results of an optimisation for the two most common polymorphs of calcium carbonate, calcite and aragonite.

1) Input file

The aim of this tutorial is demonstrate how force field methods can be used to optimise the structure and calculate the properties of a material. The input file, tutorial 1.gin, consists of 4 parts:

Part 1: Keywords

GULP can perform many types of simulation and calculate many things. The first line (excluding comments that begin with "#") specifies what type of run GULP should perform. Abbreviations are accepted, and so usually the first four letters of a keyword is sufficient. In this example the keywords are:

optimise Optimise the structure
conp Constant pressure – this selects what to optimise
compare Print a comparison of the initial and final structures
property Compute the properties of the optimised structure
phonon Compute the phonons of the optimised structure

Part 2: Structures

GULP can perform calculations on molecules (0-D), polymers (1-D), surfaces and slabs (2-D) and solids (3-D). Hence, there are quite a lot of different ways of specifying the structure. In this example we use crystallographic input for a solid consisting of 3 option words:

cell Specifies the lattice parameters, a, b, c, α , β , and γ .

fractional Specifies the atomic coordinates in fractions of the unit cell

space Gives the space group number or symbol

Because the space group is given, GULP only requires the input of the atoms in the asymmetric unit. If "space" is not given, then all atoms would have to be input explicitly.

To specify an atom, GULP expects the atomic symbol (optionally followed by a number if you want to label atoms separately, e.g. Ca1, Ca2) and a particle type (which is usually either "core" or "shel"). By default, all atoms are of type "core". In this example, each oxygen is represented by a matching pair of a core and a shell as the force field uses the shell model of Dick and Overhauser. In this model, the shell acts to mimic the dipolar polarizability of an atom by having two particles with different charges, coupled together with a harmonic spring.

GULP will accept multiple structures in the same input file and so we can include both calcite and aragonite in the structure input.

Part 3: Force field

Having specified the structures we are interested in, the next thing is to describe how the atoms interact. This is the basis of the force field. The parameters of the force field can be input directly in the file, or, as is the case in this example, read from a library file. If running lots of calculations then a library file avoids cutting and pasting lots of parameters between inputs (with the chance of a mistake). Here the line;

Library nodump rohlwrightgale

tells GULP to read the force field from a file called "rohlwrightgale.lib" (the file type extension is automatically added), which is provided as part of the examples. The sub-option word "nodump" requests that the parameters are not explicitly dumped to the restart file.

If you want to see what the force field looks like for this run then open the library file to inspect them.

Part 4: Other options

There are many other options for GULP, including those that control what the program writes out. In this example, we have only one option, "dump", that requests that a restart file called "tutorial1.res" is written at "every" step of the optimisation. The restart file is just a copy of your input, updated to reflect the result of the calculation. In this case, the restart file contains the optimised structures for calcite and aragonite so that you can use them for further GULP calculations.

2) Running the calculation

GULP can be run in either serial or parallel mode, depending on the run type and how it is compiled. For simplicity, we will just run in serial for this tutorial, as the calculations are fast. To run gulp and send the output to a file then just type:

gulp < tutorial1.gin > tutorial1.got

If you want to see the output as the job runs then you can also use:

gulp < tutorial1.gin | tee tutorial1.got</pre>

3) Output file

There is a lot of output in the file tutorial 1.got, but hopefully most of it is selfexplanatory. The first half of the output just repeats the input structures and force field, as well as printing a few properties of the initial structures. The user will see that GULP prints out 2 warning messages – just ignore these. The force field being used here has a non-standard feature and GULP is designed to highlight possible sources of error in input when it finds something unusual. If you are sure you are right (as we are here) then we can carry on without worrying about these warnings.

The interesting parts begin with the line:

"Output for configuration 1: calcite"

After printing the energy contributions for the initial structure then the optimisation of calcite begins. Only 2 steps are required to converge the energy to better than 1×10^{-6} eV and the gradient norm to 1×10^{-3} . Because GULP uses analytical second derivatives, optimisations can often converge very quickly and to very high precision if a reasonable initial starting structure is given. The comparison of the initial (which was the experimental structure) and final configurations show that the largest change in a cell parameter is only -0.19%, which is why the optimisation converges so rapidly.

Having optimised the structure then the following properties can be found in the output file:

- Density
- Born effective charge tensors
- Elastic constant tensor
- Elastic compliance matrix
- Bulk modulus
- Shear modulus
- S- and P-wave velocities
- Compressibility
- Young's moduli
- Poisson's ratios
- Piezoelectric strain and stress tensors
- Static and high frequency dielectric constant tensors
- Static and high frequency refractive indices
- Phonon frequencies at the gamma point
- Zero point energy
- Phonon density of states (crude picture!)

Following this, there is all the same information for the next structure of aragonite.

Question 1: Which of these two polymorphs of calcium carbonate is more stable and what is the energy difference?

Question 2: Which polymorph is the hardest of the two?

Question 3: Why are there 3 phonons with frequencies of zero and what does this tell us about the structures?

Tutorial 2: Lattice dynamics and free energy

Having found the optimised structure of two structures and compared their stability at 0 K, we now consider the influence of temperature. There are two common approaches to including temperature in the modelling of a solid:

- 1) <u>Lattice dynamics</u>: This method computes the free energy using the statistical mechanics of the phonon energy levels. It correctly allows for the zero point energy and quantisation of phonons and is therefore accurate in the low temperature limit. At high temperature, anharmonic effects become important and so the (quasi-)harmonic approximation used to compute the phonons begins to break down. This method is therefore typically best for crystalline materials at temperatures up to half the melting point (this is only a guideline).
- 2) <u>Molecular dynamics</u>: This method solves Newton's equations of motion to generate an ensemble of configurations by propagating the system in time. It treats nuclei as classical particles and neglects the quantisation of phonons and zero-point motion. However, it does capture anharmonicity. Therefore molecular dynamics is typically best for systems above the Debye temperature and when there is disorder in the structure (provided the barrier between states is of the order of thermal energy).

In this tutorial we will consider *lattice dynamics* and use it to calculate the free energy difference between the two polymorphs at moderate temperature.

When calculating the free energy using the phonons of a system it is important that the structure is optimised first since the harmonic approximation is only valid about a stationary point in the energy. For the next tutorial you can use the restart file from tutorial1 (tutorial1.res) and copy this to create the input for tutorial2 (tutorial2.gin). Since the structure is already optimised, you can now change the keyword "opti" to "grad" so that just a single point gradient calculation is performed. You should also remove the keyword "comp" and add "noden" to make the output cleaner for the next step. Your keyword line should now look like:

grad conp prop phon mole

The keyword "mole" was inserted by the force field and can be left there or deleted. It directs GULP to look for molecules (in this case the carbonate anion) and to use Coulomb subtraction within molecules.

To calculate the thermodynamic properties due to the phonons at a given temperature a line must insert after each structure to tell the program what temperature is required for this structure (note that there is no requirement that the value is the same for both structures, though it should be in this tutorial). For example, the following line will lead to standard conditions:

temperature 298.15

If you now run GULP with this input to create an output file tutorial2a.got, then you should find each structure now has several new quantities output:

- Entropy
- Helmholtz free energy
- Mean kinetic energy
- Heat capacity at constant volume

From this you are now in a position to calculate the free energy difference between the two structures at 298.15 K. However, there is one further thing that needs to be considered. The phonon frequencies are actually a function of the position in the Brillouin zone (k-point) and so far we have only used the gamma point. To obtain a converged free energy difference it is important to integrate across the Brillouin zone. The easiest way to do this in GULP is using a so-called Monkhorst-Pack grid (i.e. an evenly space grid of points is used to perform a numerial integration). The finer the grid, the more precise the calculation will become. To set this grid in GULP uses the "shrink" option (short for shrinking factor which is another name for the parameters that specify this grid). The shrinking factor has 3 numbers – one for each crystallographic direction. For calcite the following input should lead to a converged calculation for this phase:

shrink 888

Note the values are the same, because the primitive rhombohedral cell is used and so it is equivalent in all lattice vector directions. For aragonite, the cell lengths are different and so each value must be converged separately.

Question 4: What shrinking factors are needed to converge the free energy difference between calcite and aragonite to within 0.01 eV at 298.15 K?

Question 5: What is the free energy difference at 1000 K?

Optional: Free energy minimisation

If you have time/interest then you may like to explore the more challenging area of free energy minimisation. In the above tutorial we have assumed that the structure remains fixed with increasing temperature. However, we know that this is a poor approximation due to thermal expansion. GULP includes the possibility to compute the analytic forces based on the Helmholtz free energy and therefore the structure can be optimised as a function of temperature.

To explore this, you will need to replace the "grad" keyword with "opti free zsisa". Here "free" indicates that the free energy, rather than the internal energy is used during optimisation, while "zsisa" is the name of an approach for achieving more robust free energy minimisations.

NB: Free energy minimisation is much more expensive than a normal calculation so be prepared to wait. Also, the method breaks down at higher temperatures where the approximations are no longer valid.

Tutorial 3: The basal surface of calcite

Although surface calculations will not be covered until later in the school, it is worth giving one example to illustrate the capabilities of GULP in this area. For the material calcite, there is a single surface that usually dominates the morphology of the pure crystal, which is the (10-14) surface (referred to as the basal plane). This tutorial will show this surface can be modelled.

The approach to surface simulation used in GULP involves a 2 region method. The first region includes the atoms near the surface that will relax when the surface is cleaved from the bulk. The second region is held fixed and is used to describe the bulk below the surface and the interactions it generates with the surface region. Both regions are treated using 2-D boundary conditions, repeating in the plane of the surface. To save time, the surface has been generated using 4 layers in both region 1 and region 2 using the program GDis, based on the optimised structure from tutorial 1.

The input for this tutorial (tutorial3.gin) has a few new options:

svectors This options specifies the 2-D lattice vectors in the xy-plane. The atomic coordinates are now specified in Cartesian form.

Kpoints Specifies particular k-points for phonon calculation.

Dhkl Specifies the spacing between lattice planes for a layer sbulkenergy Bulk energy for the material, scaled to the size of region 1.

The surface structure can also be specified in fractional coordinates using other options. GULP assumes that the surface normal is parallel to the z-axis. Note that for the phonon k-points, only 2 fractional values are now needed since we have a 2-D Brillouin zone.

The quantities "dhkl" and "sbulkenergy" may seem to be irrelevant at first, but they are needed to calculate two properties of the surface:

- 1) <u>Surface energy</u>: This is the energy per unit area to create the surface from the bulk, usually expressed in Jm⁻². This is a measure of the thermodynamic stability of a surface.
- 2) <u>Attachment energy</u>: This is the energy of the top layer of the surface, known as the growth slice, and should be a negative energy. It is a measure of the energy released when a new layer is added and is often considered as representative of the rate at which a surface will grow.

After having run this tutorial you can examine the phonons for the surface at the gamma point and the special point of (1/2,0). Here the frequencies are only calculated for the relaxed region 1 at the surface and not the rigid region 2.

Question 6: Why is the lowest frequency at the gamma point not equal to zero?

Question 7: What is the significance of the first mode at the special point?