Hybrid Reverse Monte Carlo (HRMC)

Version: 1.0.5

Contents

1 Overview

The Hybrid Reverse Monte Carlo (HRMC) program uses Monte Carlo based atomic moves to produce xyz models by fitting experimentally motivated constraints such as diffraction data in addition to minimizing some potential energy function. These constraints include the radial distribution function g(r), structural factor S(q), coordination (average or histogram), bond type fraction (histogram), bond length (average), pore volume, bond angle distribution, probability grids, 3-member ring constraints and others.

The original presentation of the HRMC methodology was published in Mol. Sim. in 2002[1]. A first version of the code, HRMC 1.0, was published in Comp. Phys. Comm. in 2008[2] and HRMC 2.0 in 2013[3] and finally HRMC 2.1 in 2014[4]. These papers, particularly HRMC 2.0, provide some methodological information along with a worked example.

In the following manual, bold text represents either filenames or simulation parameters from the simulation controlling file **HRMC.inp**.

The code along with citation information is located at the CSIRO Data Access Portal:

https://data.csiro.au/dap/landingpage?pid=csiro:24232

Please report all errors and issues to Dr George Opletal

george.opletal@data61.csiro.au

[1] G. Opletal, T. Petersen, B. O'Malley, I. Snook, D. G. McCulloch, N. A. Marks, I. Yarovsky, Mol. Sim. **28**, 927-938 (2002).

[2] G. Opletal, T. C. Petersen, B. O'Malley, I. K. Snook, D. G. McCulloch, I. Yarovsky, Comp. Phys. Comm. **178**, 777-787 (2008).

[3] G. Opletal, T. C. Petersen, I. K. Snook, S. P. Russo, Com. Phys. Comm., **184**(8), 1946-1957 (2013).

[4] G. Opletal, T. C. Petersen, S. P. Russo, Com. Phys. Comm., **185**(6), 1854-1855 (2014).

1.1 Licence

The HRMC code uses the CSIRO Open Source Software Licence (Based on MIT/BSD Open Source Licence).

This licence releases the Software as 'open source'. It reflects the terms of the BSD/MIT open source software licences, with a specific CSIRO liability disclaimer included. This means that the Software will be free for use, modification and distribution, for both commercial and non commercial purposes, provided it is attributed to CSIRO and subject to the additional terms that exclude CSIRO's liability.

The Software is licensed subject to any additional conditions notified as part of the download process, including under the Supplementary Licence distributed with the Software. The Supplementary Licence may identify third party components that are licensed separately (including where distributed as part of the Software, or required in order to operate the Software). You agree to comply with the separate terms of these third party licences as a condition for accessing the Software.

You are solely responsible for determining if the Software meets your own requirements for use. Subject to law, the Software is provided 'as-is'. CSIRO makes no representations, warranties or undertakings of any kind, including regarding accuracy, defects, title, merchantability, fitness for a purpose, or non-infringement.

If you require additional licence rights, you should contact the relevant area in CSIRO.

1.2 Installation

HRMC is written in Fortran 90 and has been tested for compilation using the GNU Fortran 6.3 (GFortran) compiler and the Intel Fortran Compiler. The code contains OpenMP parallelization of the g(r) change calculation, along with a serial version, which can represent a bottleneck for very large systems (over 50,000 atoms or so).

HRMC requires compilation of the **HRMC.f90** main program file and the **hrmc_data.f90** module file in addition to numerous input files. **HRMC.inp** must always be present and contains the simulation parameters. Additionally, various input files for constraint may be needed (see section 2). For the amorphous silicon example (section 5.1) since there is fitting of the radial distribution function, structure factor and bond angle distribution, the input files **input_gr.dat**, **input_sq.dat** and **input_bo.dat** are required to successfully run the program. All the input files need to be in the same directory and the generated output files write out to the same directory. Below are steps to compiling and running the program on a Unix based machine using the Intel Fortran Compiler.

- For use without OpenMP, type

  ```
  ifort hrmc_data.f90 hrmc.f90 –cpp -o HRMC.exe
  ```

- For use with OpenMP, type

  ```
  ifort hrmc_data.f90 hrmc.f90 –cpp –fopenmp -o HRMC.exe
  ```

The -cpp compilation flag enables C pre-processor commands that segment the OpenMP routines in the code. A successful compilation will create the executable file **HRMC.exe**.

1.3 OpenMP Usage

Various bottlenecks of the code are parallelized in OpenMP to speed up the execution of the program. In particular, the recalculation of the change in g(r) from an atomic move in the mc_move routine was found to be the major bottleneck (in needs to be calculated at each move and scales with the number of particles) in the code and has been parallelized.

The **do_chunk** parameter governs how many atoms each thread will loop over before checking if another thread has already rejected the move.

The way to control the placement (scatter/compact) depends on the compiler you are using to build the software.

INTEL COMPILERS

OpenMP thread affinity control uses the **KMP_AFFINITY** environmental variable. Detailed Intel documentation can be found here.

https://software.intel.com/en-us/articles/openmp-thread-affinity-control

2 File I/O

Files beginning with **input_** are input files required by the code. Files beginning with data_ are output files with simulation outputs. Files beginning with **xyz_** are configuration files that can be visualized and can be both input and output files. Commas in the description are spaces in the files while bold text represents files and input variables (input variable defined in section 3). Visualization has been tested using the Ovito visualization package found at https://www.ovito.org/

Table 1 Input and output files

FILE	DESCRIPTION
HRMC.inp	Input file for simulation variables
input_initial.xyz	xyz starting configuration file in the form, number of particles space or comment elemental symbol, location along x-axis, location along the y-axis, location along the z-axis of particle 1 elemental symbol, location along x-axis, location along the y-axis, location along the z-axis of particle 2 .. .. File is read in if **istart** = 1 or 2
input_restart.xyz	Restart configuration file used to restart simulations that have terminated without completion. File is read in if **istart** = 4
input_gr.dat	Experimental g(r) in the form of a two dimensional list (r, g(r)) File is read in if **gr_option** = 1 or 2
input_sq.dat	Experimental S(q) in the form of a two dimensional list (q, S(q)) File is read in if **sq_option** = 1 or 2
input_bo.dat	Normalized (integrates to unity) bond angle distribution in the form (Θ, g3(Θ)) where Θ should increase from 1 to 180 degrees in one degree increments. File is read in if **bo_option** = 1 or 2
input_pore.dat	A list of the location of porosity grids points in the form, Number of pore grids x1,y1,z1 where x1,y1,x1 are integer values of the grid locations that represent a pore x2,y2,z2 … File is read in if **pore_option** = 1
input_prob.dat	A list of the probability values of probability grids in the form, Loop over x

	Loop over y Loop over z Probability of grid(x,y,z) Thus, the first grid value is (1,1,1), then the second value is (1,1,2) etc. The loops are all out to **po_res** (assumes cubic cell). File is read in if **po_option** = 1 or 2
data_errors.dat	A list of simulation conditions which is periodically output (every **ierror** steps) in the form: • Step: step number of attempted MC moves • Temp(K): temperature/energy weighting • E/atom(eV): average energy per atom when using energy constraint • Vol_Frac: volume fraction of occupied grids containing at least one atom • ERRtotal: total cost function value • ERRenergy: cost function value of energy constraint • ERRgr: cost function value of g(r) constraint • ERRsq: cost function value of S(q) constraint • ERRvo: cost function value of the pore volume fraction constraint • ERRpo: cost function value of porosity grid constraint • ERRco1_A, ERRco1_B, ERRco1_C, ERRco1_T: cost function values associated with the average coordination constraints where the _A, _B, _C, _T labelling corresponds to the constraints on the coordination histograms of elements A, B, C and the total coordination histograms of all the atoms • ERRco2_A, ERRco2_B, ERR_co2_C, ERRco2_T: cost function values associated with the coordination histogram constraints where _A, _B, _C, _T labelling corresponds to the constraints on the coordination histograms of elements A, B, C and the total coordination histogram of all the atoms • ERRbt_A, ERRbt_B, ERRbt_C: cost function values associated with the bond type constraint for elements A, B and C. • ERRbl: cost functon value for the bond length constraint • WFenergy, WFgr, WFsq, WFvo, WFbo, WFpo, WFco_A, WFco_B, WFco_C, WFco_T, WFbt_A, WFbt_B, WFbt_C, WFbl: the weighting factors for the various constraints • ERRenergy∗wf, ERRgr∗wf, . . .: these values are the constraint cost function multiplied by the weighting factors which represents the unweighted squares of the differences between the desired and system values for each constraint • 3ring_acpt_P: the three member ring acceptance probability (between 0 and 1) • Stepsize: length of maximum possible displacement an atom can perform in Angstrom • Move_Acpted: total number of moves accepted
data_coord.dat	Periodically outputs (every **ierror** steps) the coordination histograms of the atomic network and contains the following information: • Step: step number of attempted MC moves • A0-A16, B0-B16, C0-C16, T0-T16: coordination histogram of elements A, B, C and all the atoms (total) from 0 to 16 first nearest neighbours. The values sum to unity for each element and the total • Aavg, Bavg, Cavg, Tavg: the average coordination number for elements A, B, C and entire system • Bt A-A, Bt A-B, .. Bt C-C: The bond type fractions of the six combinations of the three elemental types • Bl A-A, Bl A-B, Bl C-C: The average bond length for the six combinations of the three elemental types
data_summary.dat	A summary of the simulation parameters used.
data_bofinal.dat	Model bond angle distribution after fitting. Format as **input_bo.dat**. Averaged over data stage.
data_grfinal.dat	Model radial distribution function after fitting. Format as **input_gr.dat**. Averaged over data stage.

data_sqfinal.dat	Model structural factor after fitting. Format as **input_sq.dat**. Averaged over data stage.
data_xyzdetail.dat	Detailed local bonding information for the final configuration including: • Bond length list between all bonded atoms • Bond angle list of all bonded atomic triplets • Dihedral angle of all bonded atomic quadruplets
xyz_initial.xyz	Initial configuration xyz file
xyz_final.xyz	Final configuration xyz file
xyz_movie.xyz	xyz configurations output during the simulations every **iconfig** steps along with the energy per atom in the 5th column (might not have values if energy constraint is not used).
xyz_pore.xyz	xyz output of the porosity grids input file **input_pore.dat** scaled to the simulation dimensions. This allows the porosity grids to be displayed with the configurational xyz files in visualization apps.
xyz_dupl.xyz **xyz_dupl_r.xyz**	Files output when using the duplicate cells routine (**du_option** > 0). xyz_dupl.xyz labels atoms AA, BB and CC which represent the atoms that are allowed move (AA), static atoms (BB) and atoms that are removed (CC) from the simulation. xyz_dupl_r.xyz lists the removed atoms from the simulation which need to be inserted at the end of the simulation.

3 Input File

A sample test input file (**HRMC.inp**) is shown below with explanations. The value(s) at the start of the line is the variable that is read in before the code jumps to the next line while the text follows the values are ignored. The number of lines in the input file remains constant except for the **du_rx**, **du_ry**, **du_rz** variables at the end of the file which are added as needed. The rest is a description to assist the user including the name of the parameter at the end of each line in square brackets. A detailed explanation of each line follows this example file.

```
1           write out simulation detail to screen (0=OFF, 1=ON)                [scrwrite]
3           Starting config (1=initial.xyz with remove,2=initial…              [istart]

Si          Name of element A                                                  [configA]
C           Name of element B                                                  [configB]
Ge          Name of element C                                                  [configC]

400         No. of atoms of element A                                          [n_configA]
0           No. of atoms of element B                                          [n_configB]
0           No. of atoms of element C                                          [n_configC]

0           Element A random atoms removed(or added istart=2)…            [atom_remove(1)]
0           Element B ...                                                 [atom_remove(2)]
0           Element C ...                                                 [atom_remove(3)]

2.75        Define bond length A-A (Angstrom)                            [bondlength(1,1)]
2.85               bond length A-B                                       [bondlength(1,2)]
2.85               bond length A-C                                       [bondlength(1,3)]
2.85               bond length B-B                                       [bondlength(2,2)]
2.85               bond length B-C                                       [bondlength(2,3)]
2.85               bond length C-C                                       [bondlength(3,3)]

0.05        Reduced density (0.0006022 X density(kg/m3)/…                      [rho]
20.000      x cell length (Angstrom)  (input xyz must be positive)             [xl]
20.000      y cell length                                                      [yl]
20.000      z cell length                                                      [zl]
3           PBC option (0=no PBC, 1=x PBC, 2=xy PBC, 3=xyz PBC)                [pbc_option]
0.0         Fix atoms positions from 0 to fix_var in Z direction               [fix_var]

2000000     Number of main steps                                               [npre]
100000      Number of data steps                                               [irun]
100000      Number of steps b/w writing data to file                           [ierror]
100000      Number of steps b/w writing out to xyz file                        [iconfig]

0.5         Hard sphere diameter of minimum approach                           [hdcore]
133         Seed for random number generator (integer)                         [iseed]

0           Porosity grids option (0=OFF, 1=ON)                                [pore_option]
110         Cell grids in x direction                                          [po_res]

1           Potential constraint       (0=OFF, 1=ON, 2=VIEW)                   [en_option]
1           g(r) constraint            (0=OFF, 1=ON, 2=VIEW)                   [gr_option]
1           S(q) constraint            (0=OFF, 1=ON, 2=VIEW)                   [sq_option]
1           Coordination constraint    (0=OFF, 1=ON, 2=VIEW)                   [co_option]
0           Bond type constraint       (0=OFF, 1=ON, 2=VIEW)                   [bt_option]
0           Bond length constraint     (0=OFF, 1=ON, 2=VIEW)                   [bl_option]
0           Volume constraint          (0=OFF, 1=ON, 2=VIEW)                   [vo_option]
1           Bond angle dist.constraint (0=OFF, 1=ON, 2=VIEW)                   [bo_option]
0           Probability grid constaint (0=OFF, 1=ON, 2=VIEW)                   [po_option]
1           3 ring constraint          (0=OFF, 1=ON)                           [ring3_option]

0           Disallow coordination change (0=OFF, 1=ON)                         [nobond_option]

1           Quench type (1=linear quench, 2=exp quench)                        [quench_option]
0.1         Quench rate  T = (tempS-tempF)*quenchR^steps+tempF…                [quenchR]

5000        Temperature (LINEAR QUENCH - Stage 1 (stage1 values…               [tempK(1)]
0.16 0.16 0.20 0.00008 0.0001   gr,sq,…        [grerr(1),sqerr(1),voerr(1),boerr(1),poerr(1)]
0.20 0.20 0.20 0.0008           A,B,C,T, …     [coerr(1,1),coerr(1,2),coerr(1,3),coerr(1,4)]
0.0001 0.0001 0.0001            A,B,C bond…    [bterr(1,1),bterr(1,2),bterr(1,3)]
0.0001                          Bond length…   [blerr(1)]
1.00 0.80                       3 ring…        [ring3er(1),steper(1)]
```

```
0.25                            lengthstage1…  [lengthstage(1)]

2500                            Temperature (LINEAR QUENCH - Stage 2)
0.08 0.08 0.10 0.00004 0.0001   gr,sq,vo,bo,po weightings
0.10 0.10 0.10 0.0002           A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.50 0.40                       3 ring constraint acceptance probability, stepsize
0.50                            lengthstage2 - fraction of steps at the end of stage 2

1250                            Temperature (LINEAR QUENCH - Stage 3)
0.04 0.04 0.05 0.00001 0.0001   gr,sq,vo,bo,po weightings
0.05 0.05 0.05 0.0001           A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.25 0.20                       3 ring constraint acceptance probability, stepsize
0.75                            lengthstage3 - fraction of steps at the end of stage 3

300                             Temperature (LINEAR QUENCH - Stage 4…
0.01 0.01 0.02 0.000005 0.0001  gr,sq,vo,bo,po weightings
0.02 0.02 0.02 0.00003          A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.05 0.05                       3 ring constraint acceptance probability, stepsize

3           Potential (0=C EDIP, 1=Si EDIP, 2=SiC BO, 3=S-W_1C, 4=S-W_2C, 5=S-W_3C) [pot_type]

2000        Number of points in experimental g(r)                            [ntpsgr_expt]
641         Number of points in experimental S(q)                            [ntpssq_expt]
10          Min S(q) value fitted, qmin > 2*pi/xl                            [nqstart]
641         Max S(q) value fitted, qf = (nqfinish-1)*delq                    [nqfinish]
0.05        Spacing of exp S(q) points (delr=pi/(delq*(nqpts-1))             [delq]
4097        Number of points in FFT (nqpts= 2^n+1 where n is an integer)     [nqpts]

0.00 0.000 0.000 0.000…         Coordination histogram of element A       [coord_exp(1,1),..]
0.00 0.000 0.000 0.000…         Coordination histogram of element B       [coord_exp(2,1),..]
0.00 0.000 0.000 0.000…         Coordination histogram of element C       [coord_exp(3,1),..]
0.00 0.000 0.000 0.210…         Coordination histogram of all atoms       [coord_exp(4,1),..]

0           A atoms option for fitting (0=none, 1=avg coord…                 [co_opt(1)]
0           B atoms option ....                                              [co_opt(2)]
0           C atoms option ....                                              [co_opt(3)]
2           All atoms option ....                                            [co_opt(4)]

0.000 0.038 0.962               Element A bond…         [bt_exp(1,1),bt_exp(1,2),bt_exp(1,3)]
0.008 0.051 0.941               Element B bond…         [bt_exp(2,1),bt_exp(2,2),bt_exp(2,3)]
0.168 0.733 0.099               Element C bond…         [bt_exp(3,1),bt_exp(3,2),bt_exp(3,3)]

1          A atoms option for fitting (0=nothing, 1=bond type histogram)      [bt_opt(1)]
1          B atoms option for fitting (0=nothing, 1=bond type histogram)      [bt_opt(2)]
1          C atoms option for fitting (0=nothing, 1=bond type histogram)      [bt_opt(3)]

2.500 2.500 2.500 2.500…        Bond length average of A-A,…         [bl_exp(1),bl_exp(2),..]
1 1 1 1 1 1                     Bond length constraint…              [bl_opt(1),bl_opt(2),..]

1.0000      Grid occupancy where porosity=1-vol_exp (0=all void)             [vol_exp]

100         Loop granularity for move_rejected checks in OpenMP version       [do_chunk]

0           N^3 cells in supercell (0=OFF, 1=1 cell, 2=8…                    [du_option]
2           Duplicate or Shift mode (1=duplicate (single cell…               [du_option2]
2.0 4.0     du_Rmove, du_Rinter                                         [du_Rmove,du_Rinter]
1           Cell orientations (0=random, 1=user defined)                     [du_option3]
1 0 0       Rotations about x,y,z (90deg rotation multiples)     [du_rx(1),du_ry(1),du_rz(1)]
1 1 1
0 1 2
3 3 2
0 2 3
1 1 1
2 2 2
2 3 3
```

```
1           write out simulation detail to screen (0=OFF, 1=ON)               [scrwrite]
```

Option for write out to screen.

```
3           Starting config (1=initial.xyz with remove, 2=initial…             [istart]
```

Specifies whether the starting configuration should be input from a file (requires **input_initial.xyz**) with the possibility of adding or removing random atoms (options 1 or 2), randomly generated (option 3) or restarted from a previous configuration (requires **input_restart.xyz**) (option 4).

```
Si          Name of element A                                                  [configA]
C           Name of element B                                                  [configB]
Ge          Name of element C                                                  [configC]
```

One or two letter elemental symbol for the atom types. Note that for the use of potentials having more than a single element, alphabetical order needs to be used. As an example, for SiC, **configA** = C and **configB** = Si. In the above, silicon is before carbon only because the test case is using a silicon only potential. For any single element potentials, only **configA** is used. A maximum of three elemental types is supported.

```
400         No. of atoms of element A                                          [n_configA]
0           No. of atoms of element B                                          [n_configB]
0           No. of atoms of element C                                          [n_configC]

0           Element A random atoms removed(or added istart=2)…            [atom_remove(1)]
0           Element B ...                                                 [atom_remove(2)]
0           Element C ...                                                 [atom_remove(3)]
```

The number of atoms of each elemental type and the number of atoms to remove or add depending on the selection of **istart**. When using porosity grids (**pore_option** = 1), atoms are removed from regions designated as pores and the elemental populations decrease (as well as the density).

```
2.75        Define bond length A-A (Angstrom)                            [bondlength(1,1)]
2.85               bond length A-B                                       [bondlength(1,2)]
2.85               bond length A-C                                       [bondlength(1,3)]
2.85               bond length B-B                                       [bondlength(2,2)]
2.85               bond length B-C                                       [bondlength(2,3)]
2.85               bond length C-C                                       [bondlength(3,3)]
```

The various bonding distances cut-off combinations used to define whether pairs of atoms are bonded neighbours.

```
0.05        Reduced density (0.0006022 X density(kg/m3)/…                      [rho]
20.000      x cell length (Angstrom)  (input xyz must be positive)             [xl]
20.000      y cell length                                                      [yl]
20.000      z cell length                                                      [zl]
3           PBC option (0=no PBC, 1=x PBC, 2=xy PBC, 3=xyz PBC)                [pbc_option]
0.0         Fix atoms positions from 0 to fix_var in Z direction               [fix_var]
```

Reduced density (atoms/Å^3), x, y and z cell lengths and an option for the periodic boundary conditions. **fix_var** fixes the positions of all atoms in the initial configuration with a z coordinate smaller than this parameter. The simulation coordinates are all positive with the origin being the corner of the simulation cell.

```
2000000     Number of main steps                                               [npre]
100000      Number of data steps                                               [irun]
```

npre is the total number of attempted steps in the main part of the simulation. The main part of the simulation is where the constraint weightings are allowed to change. **irun** is the total number of attempted steps in the data part of the simulation which occurs after the main part. The data part is used to collect data under the assumption that the simulation has equilibrated and as such the constraint weightings cannot change.

```
100000      Number of steps b/w writing data to file                           [ierror]
100000      Number of steps b/w writing out to xyz file                        [iconfig]
```

ierror is the number of attempted steps between periodic outputs to the **data_coord.dat** and **data_errors.dat** files. Also, **ierror** specifies the number of steps between writing information to the screen if **scrwrite** = 1. **iconfig** is the number of attempted steps between periodic outputs of configuration to **xyz_restart.xyz** and **xyz_movie.xyz** files.

```
0.5         Hard sphere diameter of minimum approach                          [hdcore]
```

The minimum separation between two atoms below which all moves are rejected. Also used in the initial configuration generator.

```
133         Seed for random number generator (integer)                        [iseed]
```

The integer seed for the random number generator.

```
0           Porosity grids option (0=OFF, 1=ON)                               [pore_option]
110         Cell grids in x direction                                         [po_res]
```

The **pore_option** flag determines whether a porosity grid is read in from **input_pore.xyz**. All atoms within these grids are removed and moves into these grids are rejected throughout the simulation. The use of this option requires a cubic cell (**xl**=**yl**=**zl**). The cell length is divided by **po_res** to produce a three dimensional grid (containing **po_res**^{3} grids). Each grid has a set of three integer coordinate labels starting with (x=1,y=1,z=1) being the first grid with its corner on the origin. The file **input_pore.xyz** has a row list of these triplet values for any grids that make up pores. Clearly, all these coordinate values must be equal to or less than **po_res**.

A second use for the **po_res** option is with the volume constraint (**vo_option** = 1). Here the resulting grid and its occupancy is calculated and compared to a desired volume fraction.

```
1           Potential constraint        (0=OFF, 1=ON, 2=VIEW)                 [en_option]
1           g(r) constraint             (0=OFF, 1=ON, 2=VIEW)                 [gr_option]
1           S(q) constraint             (0=OFF, 1=ON, 2=VIEW)                 [sq_option]
1           Coordination constraint     (0=OFF, 1=ON, 2=VIEW)                 [co_option]
0           Bond type constraint        (0=OFF, 1=ON, 2=VIEW)                 [bt_option]
0           Bond length constraint      (0=OFF, 1=ON, 2=VIEW)                 [bl_option]
0           Volume constraint           (0=OFF, 1=ON, 2=VIEW)                 [vo_option]
1           Bond angle dist.constraint  (0=OFF, 1=ON, 2=VIEW)                 [bo_option]
0           Probability grid constaint  (0=OFF, 1=ON, 2=VIEW)                 [po_option]
1           3 ring constraint           (0=OFF, 1=ON)                         [ring3_option]
```

Activation of the various constraints within a simulation. A value of 1 uses the constraint and its contribution is added to the total cost function. A value of 2 calculates the constraint cost function but does not add it to the total cost function. This allows one to observe the behaviour of the cost function of a constraint without it influencing the simulation. This is useful for example in seeing how the total energy changes without actually minimizing the total energy. An explanation of each constraint is presented in Section 4.3.

```
0           Disallow coordination change (0=OFF, 1=ON)                        [nobond_option]
```

This constraint rejects all moves that change any local coordination of any atom. Thus, a simulation would only be allowed to adjust bond angles and bond lengths. Consequently, this constraint can be inconsistent with other constraints such as the coordination constraint.

```
1           Quench type (1=linear quench, 2=exp quench)                       [quench_option]
0.1         Quench rate  T = (tempS-tempF)*quenchR^steps+tempF…               [quenchR]
```

quench_option picks the type of constraint weighting quenching scheme to be used. A linear quench scheme divides the main part of the simulation into four stages and then linearly changes the various weighting factors in the main part of the simulation. An exponential quench scheme is also included which reduces the weighting factors as weighting factor = $(W_S – W_F)\text{quenchR}^{STEP}+ W_F$ where W_s and W_r are the starting and finishing values of the weighting factor, quenchR is the quench rate and STEP is the attempted MC step number in millions. The exponential quench uses the values from stage 1 for its starting values and stage 4 for its finishing values while the linear quench uses all the stage values. See Section 4.1 for a more detailed discussion.

```
5000        Temperature (LINEAR QUENCH - Stage 1 (stage1 values…             [tempK(1)]
0.16 0.16 0.20 0.00008 0.0001   gr,sq,…        [grerr(1),sqerr(1),voerr(1),boerr(1),poerr(1)]
0.20 0.20 0.20 0.0008           A,B,C,T, …     [coerr(1,1),coerr(1,2),coerr(1,3),coerr(1,4)]
0.0001 0.0001 0.0001            A,B,C bond…    [bterr(1,1),bterr(1,2),bterr(1,3)]
0.0001                          Bond length…   [blerr(1)]
1.00 0.80                       3 ring…        [ring3er(1),steper(1)]
0.25                            lengthstage1…  [lengthstage(1)]

2500                            Temperature (LINEAR QUENCH - Stage 2)
0.08 0.08 0.10 0.00004 0.0001   gr,sq,vo,bo,po weightings
0.10 0.10 0.10 0.0002           A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.50 0.40                       3 ring constraint acceptance probability, stepsize
0.50                            lengthstage2 - fraction of steps at the end of stage 2

1250                            Temperature (LINEAR QUENCH - Stage 3)
0.04 0.04 0.05 0.00001 0.0001   gr,sq,vo,bo,po weightings
0.05 0.05 0.05 0.0001           A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.25 0.20                       3 ring constraint acceptance probability, stepsize
0.75                            lengthstage3 - fraction of steps at the end of stage 3

300                             Temperature (LINEAR QUENCH - Stage 4…
0.01 0.01 0.02 0.000005 0.0001  gr,sq,vo,bo,po weightings
0.02 0.02 0.02 0.00003          A,B,C,T coordination weightings
0.0001 0.0001 0.0001            A,B,C bond type weightings
0.0001                          Bond length constraint weighting (variance)
0.05 0.05                       3 ring constraint acceptance probability, stepsize
```

These parameters control the weightings of the constraints over the course of the simulation. The four blocks represent the four stages of the main part of the simulation whose lengths (number of steps) are determined by the **lengthstage** parameters. These parameters are the fractional value of **npre** (total steps in the main part) at which the simulation should have the values of the stage parameters. In the above example, the parameter values begin with the first stage (first block of parameters) and when the simulation reaches 25% of **npre** steps, these values would linearly reach the values of the second stage values (second block of parameters) and so on. The fourth stage values become the values that are used in the data part of the simulation. Using a linear quench (**quench_option**=1), all four stages are used while the exponential quench (**quench_option**=2), only the first and last stage values are used along with the quench rate parameter, **quenchR**.

Looking at any individual stage, the first line is the temperature for the energy constraint (in Kelvin), the second contains the weightings for the g(r), S(q), volume, bond angle distribution and probability grids constraints. The third line then has the four coordination constraint weightings for elements A, B, C and for the totality of all elements. The fourth line has the three bond type constraint weightings for the three elements. The fifth line has the bond length constraint weighting. The sixth line has three member ring constraint acceptance probability and the maximum allowed step size of MC moves.

```
3         Potential (0=C EDIP, 1=Si EDIP, 2=SiC BO, 3=S-W_1C, 4=S-W_2C, 5=S-W_3C)    [pot_type]
```

Determines the type of empirical potential to be used with the energy constraint. Appropriate elemental labels are required for the potentials (carbon for EDIP carbon potential for example) and for multi-element potentials, they are input (**configA**, **configB**, **configC** parameters) in alphabetical order. The parameter based Stillinger-Weber potentials require parameter input files (see Section 4.3.1)

```
2000        Number of points in experimental g(r)                             [ntpsgr_expt]
641         Number of points in experimental S(q)                             [ntpssq_expt]
10          Min S(q) value fitted, qmin > 2*pi/xl                             [nqstart]
641         Max S(q) value fitted, qf = (nqfinish-1)*delq                     [nqfinish]
0.05        Spacing of exp S(q) points (delr=pi/(delq*(nqpts-1))              [delq]
4097        Number of points in FFT (nqpts= 2^n+1 where n is an integer)      [nqpts]
```

ntpsgr_expt and **ntpssq_expt** are the total number of points in the experimental g(r) (contained in input file **input_gr.dat**) and S(q) (contained in input file **input_sq.dat**). The parameters **nqstart** and **nqfinish** specify the range of experimental points (lines from the top to the bottom of the **input_sq.dat** file) between which the experimental S(q) will be fitted. This allows one to eliminate the fitting of low q noisy data if needed. **delq** determines the diffraction data spacing for the S(q) and also importantly the g(r) via Δr = π /(**delq**(**nqpts** -1)). Thus, even if only the g(r) constraint is used in the simulation, **delq** needs to be set correctly to match the g(r) spacing in **input_gr.dat**. **nqpts** is the number of points in the fast Fourier Transform and should be equal to an 2^n+1, where n is some integer value.

```
0.00 0.000 0.000 0.000…        Coordination histogram of element A       [coord_exp(1,1),..]
0.00 0.000 0.000 0.000…        Coordination histogram of element B       [coord_exp(2,1),..]
0.00 0.000 0.000 0.000…        Coordination histogram of element C       [coord_exp(3,1),..]
0.00 0.000 0.000 0.210…        Coordination histogram of all atoms       [coord_exp(4,1),..]

0           A atoms option for fitting (0=none, 1=avg coord…                  [co_opt(1)]
0           B atoms option ....                                               [co_opt(2)]
0           C atoms option ....                                               [co_opt(3)]
2           All atoms option ....                                             [co_opt(4)]
```

The block of 4 x 17 parameters are used to specify the coordination constraints. The rows represent the parameters for elements A, B, C and for the entire system while the columns represent coordination ranging from 0 to 16. Each value is the desired fraction at a particular coordination and each row should sum to unity. The **co_opt()** parameters then determine whether the constraint attempts to fit each elemental histogram or just its weighted average (average elemental coordination number). A value of zero omits that element from fitting.

```
0.000 0.038 0.962              Element A bond…         [bt_exp(1,1),bt_exp(1,2),bt_exp(1,3)]
0.008 0.051 0.941              Element B bond…         [bt_exp(2,1),bt_exp(2,2),bt_exp(2,3)]
0.168 0.733 0.099              Element C bond…         [bt_exp(3,1),bt_exp(3,2),bt_exp(3,3)]

1          A atoms option for fitting (0=none, 1=bond type histogram)         [bt_opt(1)]
1          B atoms option for fitting (0=none, 1=bond type histogram)         [bt_opt(2)]
1          C atoms option for fitting (0=none, 1=bond type histogram)         [bt_opt(3)]
```

Parameters for the bond type constraint. The first row represents the fraction of bonds between element A and its first nearest neighbour elements A, B and C with respect to the total number of bonds from element A. Similarly for elements B and C in the second and third rows. Each row needs to add to unity. The **bt_opt()** parameters then determine which elements are included in the constraint fitting.

```
2.500 2.500 2.500 2.500…       Bond length average of A-A,…          [bl_exp(1),bl_exp(2),..]
```

```
1 1 1 1 1 1                     Bond length constraint…              [bl_opt(1),bl_opt(2),..]
```

Parameters for the bond length constraint. The average bond length between elements A-A, A-B, A-C, B-B, B-C, C-C are specified (in Angstroms) are contained in the first line. The second line then specifies which elemental pairs are actually included in the fitting.

```
1.0000      Grid occupancy where porosity=1-vol_exp (0=all void)              [vol_exp]
```

Parameter for the volume constraint. **vol_exp** specifies the fraction of grids that should ideally contain at least one atom (occupied). The number of grids used is specified early via the parameter **po_res**.

```
100         Loop granularity for move_rejected checks in OpenMP version       [do_chunk]
```

do_chunk is an OpenMP parameter governing how many atoms each thread will loop over before checking if another thread has already rejected the move. A default value of 100 is recommended.

```
0           N^3 cells in supercell (0=OFF, 1=1 cell, 2=8…                     [du_option]
2           Duplicate or Shift mode (1=duplicate (single cell…                [du_option2]
2.0 4.0     du_Rmove, du_Rinter                                          [du_Rmove,du_Rinter]
1           Cell orientations (0=random, 1=user defined)                      [du_option3]
1 0 0       Rotations about x,y,z (90deg rotation multiples)     [du_rx(1),du_ry(1),du_rz(1)]
1 1 1
0 1 2
3 3 2
0 2 3
1 1 1
2 2 2
2 3 3
```

Parameters for the cell duplication routine which allows a single cell to be duplicated or N^3 model cells (N=2,3,4,..) to be merged into a single supercell and HRMC to be performed only on the cell edge regions. **du_option** determines the number of model cells to be used. **du_option2** determines if the **du_option** number of cells are derived from duplication of a single cell or come from **du_option** cubed number of unique model cells.

du_Rmove specifies the width of the edge of the cell within which atoms can move in a HRMC simulation. **du_Rinter** determines the width of the region within which atoms are kept static but required for nearest neighbour list interactions.

du_option3 determines if the model cell rotations are random or user defined. If they are user defined, then the remaining lines (number of lines equal to **du_option** cubed) specify the rotations. Each line gives the 90 degree multiplier of rotation around the x, y and z-axis in the clockwise direction. Thus in the above, model cell 3 will be rotated 0 degrees around the x-axis, 90 degrees around the y-axis and 180 degrees around the z-axis in that order. Further details are found in Section 4.4.

4 Theory/Operation

The HRMC algorithm is a hybrid of the Metropolis Monte Carlo (MMC) method, where a simulation displaces atoms to minimize a total energy and the Reverse Monte Carlo (RMC) method which stochastically minimizes the discrepancy between experimental and computed diffraction data. The motivation to merge these techniques stems from the lack of unique solutions to the inversion problem for pair correlation functions within RMC, which results in unphysical structures correctly fitting the experimental diffraction data. This typically shows up as large three member ring populations within the resulting atomic network and broad bond-angle distributions. These issues are particularly problematic for systems with inter-atomic interactions that are not pair-wise additive, such as amorphous solids with short range order dictated by covalent bonding. The inclusion of a three body term contained within some empirical potential penalizes the formation of such small unphysical angles and has been shown to address the issue.

A HRMC simulation starts from some arbitrary configuration and selects an atom at random. We label the position of this atom as OLD. The simulation then attempts to move this atom to a new random position (within some maximum step size distance) which we label as NEW. The total cost functions for the old and new positions are calculated as a sum of individual cost functions for each constraint. The cost function in the old position is given by equation 1.

$$\chi_{\mathrm{OLD}} = \sum_{\mathrm{i}} \sum_{\mathrm{j}} \frac{(\mathrm{A_i(j)_{EXP}} - \mathrm{A_i(j)_{OLD}})^2}{\mathrm{W_i}} + \frac{\mathrm{E_{OLD}}}{\mathrm{kT}} \tag{1}$$

where the sum i is over the number of constraints and j over the number of fitted data points within the i$^{\mathrm{th}}$ constraint. The individual cost functions are the squared differences between the experimental or desired values $\mathrm{A_i(j)_{EXP}}$ and the simulation values $\mathrm{A_i(j)_{OLD}}$, scaled by a weighting factor $\mathrm{W_i}$. These constraint weighting factors are effective variances related to the standard errors by $\mathrm{W_i} = \sigma_{\mathrm{i}}^2$. Within HRMC, these weighting factors are constant for the sum over j in equation 1 but can change as a function of attempted MC steps to vary the influence of individual constraints over the course of the simulation. An equivalent expression exists for these constraints in the new position χ_{NEW}.

Ultimately, atomic moves where $\chi_{\mathrm{NEW}} < \chi_{\mathrm{OLD}}$ are accepted, otherwise, moves are accepted with a probability given by,

$$\mathrm{P} = \mathrm{e}^{\chi_{\mathrm{OLD}} - \chi_{\mathrm{NEW}}} \tag{2}$$

A simulation is typically run until the total cost function oscillates about some value. Ensemble averages of various quantities are then accumulated in the data part of the HRMC simulation.

4.1 Simulation Stages and Quench Schemes

A simulation is divided into two parts, a main part and a data part. In the main part of the simulation, constraint weightings are allowed to vary while in the data part, they remain constant for the purpose of collecting statistics through ensemble averaging.

The code has two quenching approaches by which the temperature and other constraint weightings can be varied, a linear quench scheme and an exponential quench scheme. The linear quench scheme divides the main part into four stages as illustrated in an example in figure 1.

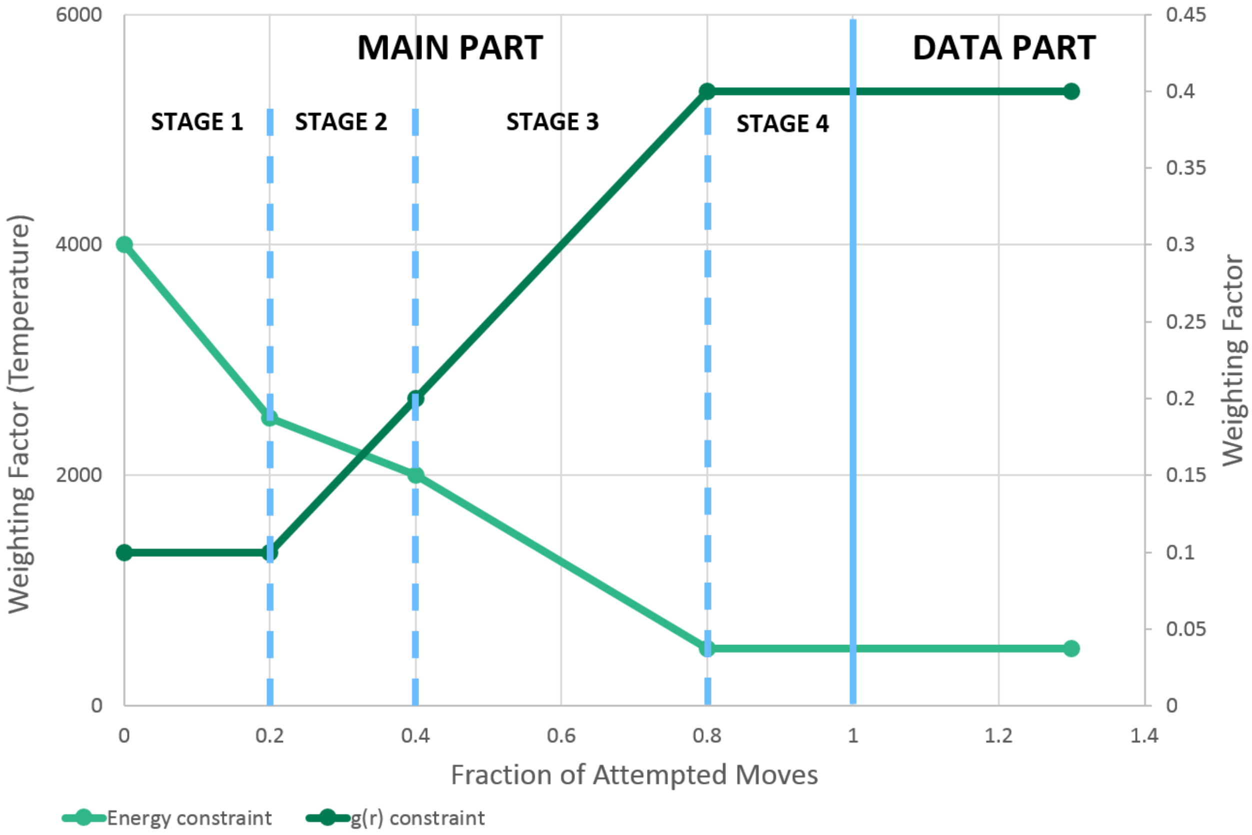


Figure 1 The main stage is divided up into a MAIN part where simulation parameters can be varied and quenched and a DATA part where these parameters are kept constant. The MAIN part is further divided into four stages

The fractions of the main part total steps (**npre** parameter) at which each stage ends are shown in italics and are given by the parameters **stagelength1**, **stagelength2** and **stagelength3** in the input file. Illustrated are the temperature and some other arbitrary constraint weighting factors and their values at the beginning of each stage. These values (variances) are specified in the input file and the code calculates a linear gradient between the subsequent stage values. The gradient for stage four is always set to zero inside the code which allows the system to equilibrate over stage 4 before passing into the data collecting part of the simulation. The generality of this quench scheme allows for the energy and other weightings to be decreased in proportion to each other as well as in other more complicated ways.

An example of the exponential quench scheme is illustrated in figure 2.

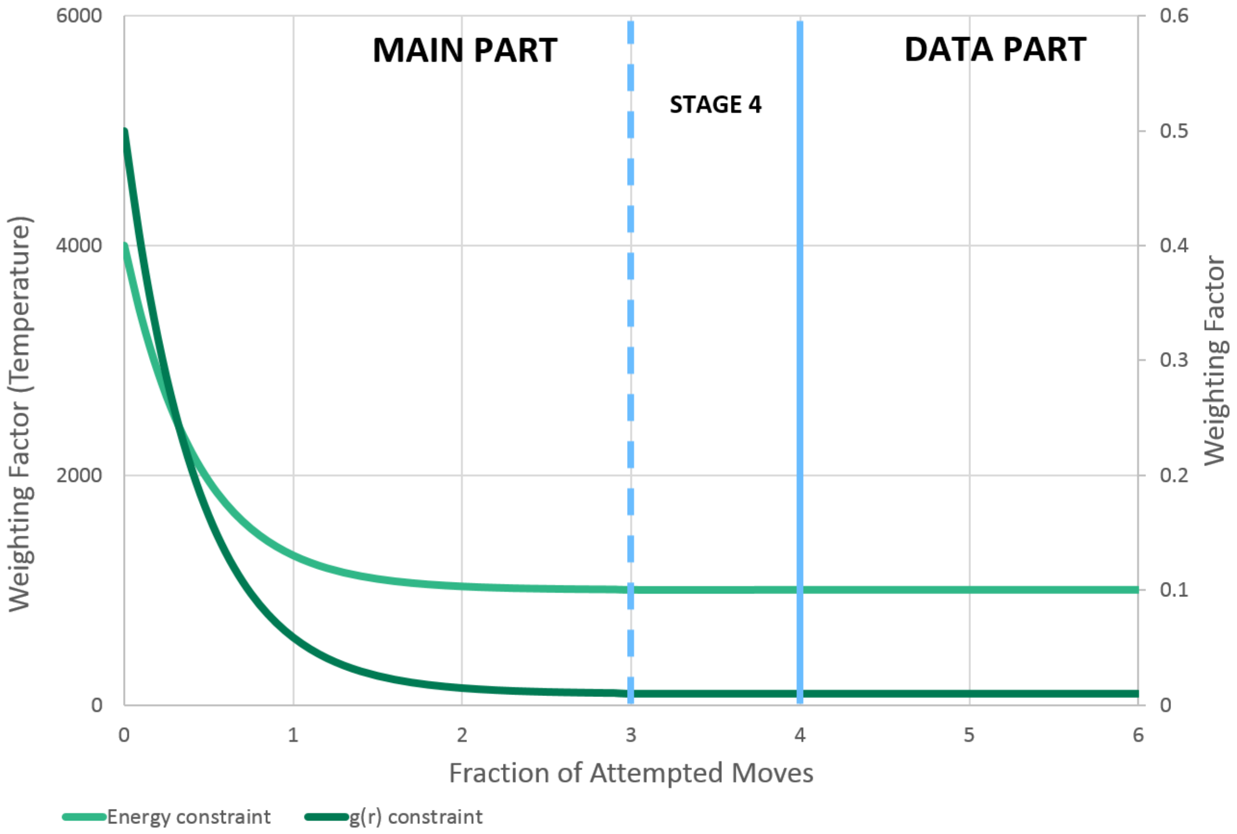


Figure 2 An exponential quench illustration is contrasted against the linear quench in figure 1

Using exponential quenching the temperature and each dynamic constraint weighting is altered via,

$$\mathrm{W(step)} = (\mathrm{W_S} - \mathrm{W_F})\mathrm{quenchR}^{\mathrm{step}} + \mathrm{W_F} \tag{3}$$

where $\mathrm{W_S}$ and $\mathrm{W_F}$ are the starting and finishing values of the constraint weighting, **quenchR** is the quench rate and step is the attempted MC move step number of the simulation in millions. Thus, only the parameters from stage 1 and 4 are used from the input file. As can be seen in figure 2, the weightings only became constant in the data part of the simulation. Thus, care must be taken in picking an appropriate quench rate (**quenchR** parameter) along with the total number of steps in the main part of the simulation to ensure that the weightings have settled down to a minimal gradient by the end of stage 4.

In addition to the variation of constraints throughout the simulation, the simulation maximum step size and the three member ring acceptance probability can also be varied in linear and exponential ways. This attempts to balance a number of effects. With regards to step size, porous models may require large step sizes to cross over empty void space while later in the simulation, smaller step sizes may be needed to increase the acceptance probability when the simulation is close to equilibrium. The three member ring constraint design also takes into account a number of issues. On the one hand, three member ring formation in RMC is an inherent, unphysical artifact in the simulation and should be minimized. However, it is also a very efficient pathway to fit other constraints such as the system g(r). Thus, one may address both issues by accepting three member ring formation at the start of the simulation while rejecting it in the larger stages. While activated with an acceptance probability of zero, no new three member rings are allowed to form while any moves breaking them up are accepted resulting in a gradual reduction of their population.

4.2 Initial Cell Setup

The code requires a starting structure to begin. This may be read in from a file (**input_initial.xyz**) or be generated internally (**istart** picks between these options). The initial structure generator randomly places atoms within an orthorhombic cell ensuring that no two atoms fall within a separation smaller than the hard core minimum distance parameter (**hdcore**). One needs to specify the elemental labels (**configA**,...) and the number of atoms of each of these element types (**n_configA**,...). When reading in the initial structure from file, additional options exist to add or remove atoms at random from the structure prior to the simulation.

The simulation cell dimensions (**xl,yl,zl**), density (**rho**) and periodic boundary condition type (**pbc_option**) need also be input, along with the cut-off distances that define the first nearest neighbours (**bond_length1**,...). These cut-off distances should ideally be the distances of minimal value between the first and second peaks in the appropriate partial radial distribution functions. The choice of the cell dimensions along with the number of atoms of each elemental type should be consistent with the cell density. If one has the sample density D (in g/cm^3) of a multi-element compound, the reduced density (in atoms/Å^3) required for the simulation is given by,

$$\mathbf{rho} = 0.60221415\mathrm{D}/\textstyle\sum_{\mathrm{i}} \mathrm{f}_{\mathrm{i}}\mathrm{W}_{\mathrm{i}} \qquad (4)$$

where the sum is over the number of element types, f_{i} is the fraction of atoms of element type i and W_{i} *i*s the atomic weight of element i. The fractions should sum to unity and the sum represents the average atomic weight of the sample.

An additional constraint can be imposed on the simulation volume via the use of porosity grids (**pore_option**). This option allows various parts of the simulation volume to be inaccessible to the moving atoms. The option can only be used with cubic periodic boundaries (**xl**=**yl**=**zl**). The simulation cell lengths are divided by the **po_res** parameter, to establish a grid (containing [**xl**/**po_res**]^3 cubic grids). The file called **input_pore.xyz** is read in and any moves into grids locations specified within that file are always rejected. This allows the simulation to wrap itself around an arbitrary pore geometry. The geometric construction of the file **input_pore.xyz** is not included in this code. A spherical overlap model fitting internal surface area and pore volume has been previously used to make these porosity grids in an external program to the HRMC code[5].

Some considerations about the starting structure must be taken into account when using porosity grids. If the starting structure is produced by the random generator, then the atom insertion never falls inside a porosity grid. Care must be taken on deciding which simulation density to use however. If the initial structure is read in from a file, then any atoms that fall within porosity grids are removed and the simulation density will be reduced.

[5] G. Opletal, T. Petersen, I. Snook, D. McCulloch, J. Chem. Phys. **126** (2007) 214705.

Finally, the positions of a substrate of atoms can be fixed via the use of the **fix_var** parameter. This parameter specifies the position in the z-direction below which all atoms have their moves rejected. These atoms still contribute like any other atom to the cost functions of all the active constraints but their positions remain static.

If a simulation needs to be restarted due to premature termination, the restart option can be used. This requires the **xyz_restart.xyz** file to be renamed to **input_restart.xyz** and the simulation will continue where it was terminated. The code outputs information into **xyz_restart.xyz** every **iconfig** number of steps which includes information on the current temperature and weightings at the current step number.

4.3 Constraints

The HRMC code contains numerous constraints, many of which have associated cost functions for minimization. Most constraints can be selected to be either on, off or viewed. Viewed constraints calculate the cost functions for that constraint but do not add that constraint contribution to the total cost function and thus have no influence. Constraint inputs generally have four values for the four stages in the main part of the simulation. A previous study of constraint combinations and their effects may assist the user[6]. Below is a brief overview of each constraint.

4.3.1 Potential Energy

The potential energy constraint allows for different types of empirical interatomic potentials for different elements. The parameters of these potentials are found in the **INIT_PARAMETERS** routine. The weighting (in Kelvin) for the potential energy constraint is the temperature given by the **tempK** parameter. The HRMC code contains a number of empirical potentials as listed below:

CARBON
Environment Dependent Interatomic Potential (EDIP)[7] (**pot_type** = 0)
Gives potential energy of -7.36eV for graphite and -7.34eV for diamond.

SILICON
Environment Dependent Interatomic Potential (EDIP)[8] (**pot_type**=1)
Gives potential energy of -4.65eV for diamond.

Stillinger-Weber[9] (**pot_type**=2)
Gives potential energy of -4.34eV for diamond.
SILICON CARBIDE

[6] G. Opletal, T. C. Petersen, A. S. Barnard, S. P. Russo, J. Phys. Chem. **38**(17), 1547(2017).
[7] N.A. Marks, Phys. Rev. B **63**, 1(2000).
[8] A.Z. Bazant, E. Kaxiras, J.F. Justo, Phys. Rev. B **56**, 8542(1997).
[9] F. Stillinger, T. Weber, Phys. Rev. B **31**, 5262(1985).

Bond-order[10] (**pot_type**=2)

For potentials with more than a single element, the order of the elemental label inputs for **configA**, **configB** and/or **configC** is in alphabetical order. Thus for example for SiC, **configA** is C while **configB** is Si.

STILLINGER-WEBER POTENTIAL

The Stillinger-Weber potential has been included with the ability to read in user defined potential parameters. The potential can handle up to three elements (labelled A, B and C below) with an input file selection of **pot_type**=3 for single element, **pot_type**=4 for two element and **pot_type**=5 for three element systems.

The parameters for this potential is input by the user via either the **input_SW1.dat**, **input_SW2.dat** and **input_SW3.dat** files, the choice of which depends on how many element types are being used. The formalism of the variable multibody Stillinger-Weber potential is adapted from LAMMPS.

$$E = \sum_i \sum_{j>i} V_2(r_{ij}) + \sum_i \sum_{j\neq i} \sum_{k>j} V_3(r_{ij}, r_{ik}, \theta_{ijk})$$

$$V_2(r_{ij}) = A_{ij}\varepsilon_{ij}\left[B_{ij}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}}\right] exp\left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right)$$

$$V_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk}\varepsilon_{ijk}\left[\cos\theta_{ijk} - \cos\theta_{0ijk}\right]^2 exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right) exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}}\right)$$

For a single element system, only a single line of parameters needs to be specified.

Elements	ε_{ijk}	σ_{ij}	a_{ij}	λ_{ijk}	γ_{ij}	$\cos\theta_{0ijk}$	A_{ij}	B_{ij}	p_{ij}	q_{ij}
A-A-A										

For a two element system, parameters for 8 elemental combination need to be specified.

Elements	ε_{ijk}	σ_{ij}	a_{ij}	λ_{ijk}	γ_{ij}	$\cos\theta_{0ijk}$	A_{ij}	B_{ij}	p_{ij}	q_{ij}
A-A-A										
A-A-B										
A-B-A										
A-B-B										
B-A-A										
B-A-B										
B-B-A										
B-B-B										

For a three element system, there are 27 elemental parameters needing specification

Elements	ε_{ijk}	σ_{ij}	a_{ij}	λ_{ijk}	γ_{ij}	$\cos\theta_{0ijk}$	A_{ij}	B_{ij}	p_{ij}	q_{ij}
A-A-A										
A-A-B										
A-A-C										
A-B-A										

[10] P. Erhart, K. Albe, Phys. Rev. B **71**, 035211(2005).

A-B-B										
A-B-C										
A-C-A										
A-C-B										
A-C-C										
B-A-A										
B-A-B										
B-A-C										
B-B-A										
B-B-B										
B-B-C										
B-C-A										
B-C-B										
B-C-C										
C-A-A										
C-A-B										
C-A-C										
C-B-A										
C-B-B										
C-B-C										
C-C-A										
C-C-B										
C-C-C										

The files **input_SW1.dat**, **input_SW2.dat** or **input_SW3.dat** containing the parameters for the one, two and three element systems above only require a specification of the block coloured in blue above (the labelling is left out of the files). As explained in the LAMMPS methodology, values for two body parameters (in both the two and three body terms) come from entries where the j and k labels are the same (for example for two elements, 1-1-1, 1-2-2, 2-1-1, 2-2-2 where 1-2-2 and 2-1-1 should be the same). All other values for the two body parameters are unused and could be set to any value.

Note that the generic elements in the tables above (labelled A, B and C) should correspond to different atomic labels for the variables **configA**, **configB** and **configC** in the input file. Generally, an alphabetical ordering is used for these in HRMC (for example C before Si in SiC with **configA** = C, **configB** = Si). HRMC will not work if pure carbon (single element type) is run with a two element SiC parameters set. For such situations, the single element Stillinger-Weber potential would be used.

A collection of potentials if found in the /Resources/SW Potential Parameters directory.

More discussion can be found on the LAMMPS discussion on the potential.

http://www.afs.enea.it/software/lammps/doc17/html/pair_sw.html

4.3.2 Diffraction Data

The g(r) and S(q) constraints fit experimental diffraction data and use **grerr** and **sqerr** as the weighting constant from the input file. Both constraints require input files to be read in if used (**input_gr.dat** and **input_sq.dat**). The data within these files needs to be linearly spaced, starting at r=q=0. The spacing of the S(q) is specified by the parameter **delq** and even if there is only fitting to an experimental g(r), the spacing of the experimental input g(r) data must be equal to Δr = π /(**delq**(**nqpts** -1)) where **nqpts** is the number of points used in the Fast Fourier Transform (FFT).

The range of the g(r) fitted is limited by the cell half length (of the shortest cell side length) automatically while its lower range can be controlled via the use of the **hdcore** parameter. This rejects any atoms moving within a specific minimum separation and thus produces a zero value in the simulation g(r) up to that distance. This parameter is typically set to the final zero value before the first peak in the experimental g(r). In general, the number of points in the FFT (**nqpts**) is greater than the number of g(r) experimental points (**nptsgr_expt**) since padding is required to ensure that the number of FFT points relates to a suitable power of two. To automatically pad any points beyond the experimental data range, the program assumes an asymptotic value of one for both S(q) and g(r).

The parameters **nqstart** and **nqfinish** specify the minimum and maximum points between which the experimental S(q) data is fitted (these are also equal to the line numbers in the **input_sq.dat** file). These parameters allow for the exclusion of low q noise from the fitting procedure. In many simulations, the use of just the g(r) constraint is sufficient since it avoids the time consuming FFT transformations at each attempted move.

4.3.3 Bond Angle Distribution

The bond angle distribution constraint reads in angular data from the **input_bo.dat** file which contains a probability distribution averaged over all atoms of the angle between a central atom and two if it's first nearest neighbours. The associated weighting is given by **boerr**. The input data must be linearly spaced starting at 1 degree and extending in range to 180 degrees. The distribution should be normalized to unity.

If no potential energy constraint is used, a bond angle distribution constraint can be used to control the formation of small ring populations.

4.3.4 Coordination

The coordination constraints attempt to produce a model with average number of first nearest neighbours around a particular element. The constraint values to be fitted are given by **coord_exp** and the constraint weighted by the parameters **coerr**.
There exist a total of up to four coordination constraints for a three component system, with one for each element type and one for the entire system. Each constraint can operate in two different modes and the parameters **co_opt** decide which constraints are used and in which mode. In the first mode (equal to **co_opt** = 1), the simulation fits only the average coordination which is calculated from the input coordination histogram as,

$$\mathrm{C_{avg}} = \sum_{\mathrm{i}=0}^{16} \mathrm{i}\,\mathrm{F_i} \qquad (5)$$

where the sum is from coordination zero to sixteen and $\mathrm{F_i}$ is the fraction of atoms that have a coordination of i. The sum of $\mathrm{F_i}$over i must equal one in the input file. Thus, when using these constraints in average coordination mode, the details of the coordination histogram are irrelevant as only the average coordination is fit.

In the second mode, a fit is made to the entire coordination histogram for each elemental type and/or all of the atoms. Due to the generality of these constraints, some care must be taken when specifying combinations of these constraints together. For example, when fitting a total average coordination of all the atoms alongside individual constraints on each elemental type, the input values between these constraints must be consistent.

4.3.5 Bond Type

The bond type constraint is applicable for systems containing more than a single element. The constraint attempts to fit the average fraction of bonds for an element type to all the other elemental type combinations. The constraint values are given by **bt_exp** while the weighting factors are specified by the **bterr** parameters. For each element, the fraction of bonds to all the other elements needs to sum to unity. The constraint is irrelevant to systems having only a single element type. For binary systems, two constraints (one for each element type) can attempt to fit the bond fraction to themselves and the second element. For ternary systems, three constraints can exist each with bond fractions to themselves and the two other elements. The **bt_opt** parameters determine which elements are to be included in the fitting.

4.3.6 Bond Length

The bond length constraint fits an average distance between two element types for the first nearest neighbours only. The constraint values are given by **bl_exp** while the weighting factors are specified by the **blerr** parameters. The **bl_opt** parameters decide which elemental pair distances are fitted.

4.3.7 Volume

The volume constraint fits a grid occupancy fraction, namely how many grids contain at least one atom from the total number of grids. The simulation divides the x, y and z lengths of the simulation cell by **po_res** to produce a grid containing (**po_res** x **po_res** x **po_res)** rectangular volumes. The constraint input parameter **vol_exp** specifies the fraction of grids that contain at least one atom. The **voerr** parameters specify the associated weighting values. The simulation cell can be rectangular. If the value of **po_res**^{3} is larger than the total number of atoms in the system, then it becomes impossible to reach a full occupancy.

4.3.8 Probability Grids

The probability grids constraint overlays a three dimensional probability field over the simulation cell to bias movement into specific regions. The **po_res** parameter is used in a similar way as the volume constraint, to divide the simulation cell into grids (the exception here is that only a cubic simulation cell is allowed). Each grid has a value between 0 and 1 and these are input from the **input_prob.inp** file. This file should have (**po_res** x **po_res** x **po_res**) lines and is read in looping wise as loop over x(loop over y(loop over z))).

The cost function is calculated by looping over atoms and summing the values of the probability grids they occupy (SUMOLD). If an atom then moves into a region of higher grid probability, then new total sum (SUMNEW) would have a higher value. The change in the cost function is SUMOLD

– SUMNEW which will be a negative number and thus a reduction in the cost function occurs from this move. For a move that puts an atom into a grid of lower grid probability, the cost function will increase. The associated weighting factor is given by **poerr**.

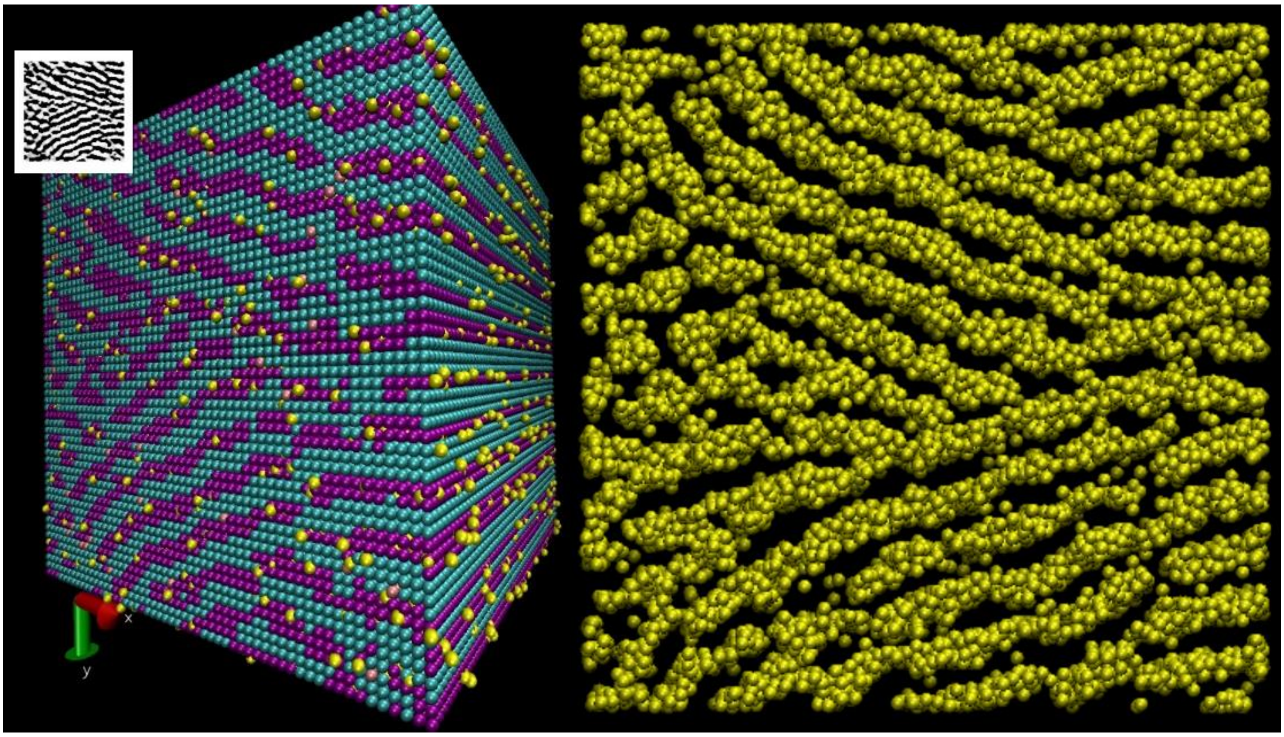


Figure 3 Probability grids shown on the left (from **xyz_prob.xyz** file) derived from a graphitic TEM image (top left overlayed figure). Regions of high grid probability are shown in purple while low probability grids shown in blue and atoms in yellow. The resulting model is shown on the right

The file **xyz_prob.xyz** illustrates the probability grid values graphically in an xyz file by assigning labels based on the grid probability. Their labels are given by:

Table 2 Labels for probability grids in the **xyz_prob.xyz** file

LABEL	PROBABILITY GRID VALUE
AA	P > 0.99
BB	0.80 > P ≤ 0.99
CC	0.60 > P ≤ 0.80
DD	0.40 > P ≤ 0.60
EE	0.20 > P ≤ 0.40
FF	0.00> P ≤ 0.20
GG	P = 0.0

4.3.9 Porosity Grids

The porosity grids constraint is similar in effect to the probability grids constraint but instead outright rejects moves into grids designated as pores. The simulation cell is similarly divided into grids by dividing the cell length (constraint requires a cubic system) by the **po_res** parameter to derive the grid size. The constraint then reads in data from the **input_pore.xyz** file which contains the locations labels in a list of grids that are designated as pores. Any moves into porosity grids are rejected and this allows the simulation to wrap itself around an arbitrary pore geometry. The geometric construction of the file **input_pore.xyz** is not included in this code. A spherical overlap model fitting internal surface area and pore volume has been previously used to make these porosity grids in an external program to the HRMC code[5].

Whether the starting configuration is generated randomly or read in from a file, any atoms that are initially found within a porosity grid, are removed. Porosity grids have no cost function associated with it.

4.3.10 Three-member ring

This constraint checks if an attempted atomic move results in a three membered ring and can then rejected the move probabilistically. The quench scheme in HRMC allows the three ring acceptance probability to be varied as a function of step number (given by the parameter **ring3er**). This is implement because whilst these small rings are generally energetically unfavourable, the RMC methodology uses these ring structures as an efficient pathway to structural optimization. Ideally, the bond angle constraint or the potential energy constraint are used instead of this constraint.

4.3.11 No Bond Change

The **nobond_option** specifies if any moves leading to a coordination change should be rejected. This allows a HRMC simulation that optimizes bond length and angles without any coordination changes.

4.4 Cell Duplication

The HRMC code allows for cell duplication to produce supercells and thus increase the size of small HRMC models into larger models. The idea is to be able to create a larger supercell out of smaller existing model cells and for the HRMC simulation to fit and relax the initially conflicting boundary regions by 'stitching' together the smaller models.

The routine has two modes, **Duplicate** and **Shift**.

The Duplicate mode requires a single xyz model in the **input_initial.xyz** file to be read in and it is then duplicated to produce a supercell.

The Shift mode requires multiple xyz models in the **input_initial.xyz** file to be read and then those individual models are shifted into appropriate positions. The order of the cells in the xyz input files depends on their location in the supercell. For a 2 x 2 x 2 supercell (made up of 8 cells), the atomic coordinates of the 8 models are read into the HRMC code and inserted into the supercell in the following order

1 1 1, 1 1 2, 1 2 1, 1 2 2, 2 1 1, 2 1 2, 2 2 1, 2 2 2

where the values are the x, y, z cell positions for the 8 cells. Thus in terms of looping its loop x (loop y (loop z))) with z being the innermost loop. The **input_initial.xyz** file should have the total number of atoms in the supercell in the first line, a space or comment and the coordinates of the atoms in all the models following (as standard xyz format demands). All the models must have the same cubic cell dimensions and the same number of atoms.

In both modes, whether duplicating a single cell (model) or shift multiples cells (models), the user has the option of either randomly rotating the cells or specifying their rotation. Due to the cubic stacking of the cells, only 0, 90, 180 and 270 degree rotations are allowed.

du_Rmove specifies the width of the edge of the cell within which atoms can move in a HRMC simulation. **du_Rinter** determines the width of the region within which atoms are kept static but required for nearest neighbour list interactions. A visualization of the range of these parameters on the cell is shown in the figure below and can be found in the **xyz_dupl.xyz** file. In that file, the labels AA, BB and CC represent the moving atoms (red in figure), static atoms (blue) and atoms that are removed from the simulation (yellow).

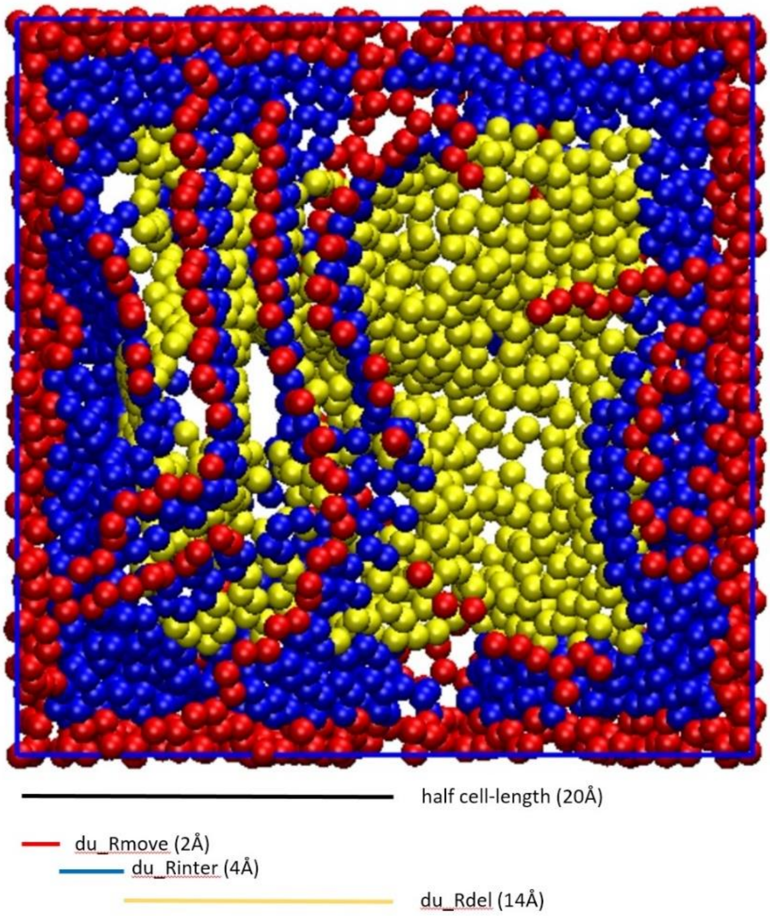


Figure 4 Illustration of the length variables used to control the cell duplication. Single input xyz model where yellow atoms are removed, blue atoms are fixed in place by present and red atoms are allowed to move in the simulation. The blue atoms are required for neighbour list calculations. **du_Rdel** = **xl**/2 – **du_Rmove** - **du_Rinter**

The operation of the routine using duplicate mode is shown below,

i) A cubic xyz model is read in

ii) The model cell is duplicated in each positive axis direction for either 2 x 2 x 2=8 or 3 x 3 x 3=27 copies. Each duplicate cell is rotated by either 0, 90, 180 or 270 degrees.

iii) A specification of the variables **du_Rmove** and **du_Rinter** determines which atoms are removed from the cell for computation efficiency.

iv) HRMC is performed on the atoms found in the **du_Rmove** region (see figure above). The removed atoms are found in the **xyz_dupl_r.xyz** and should be added to the final xyz from the HRMC.

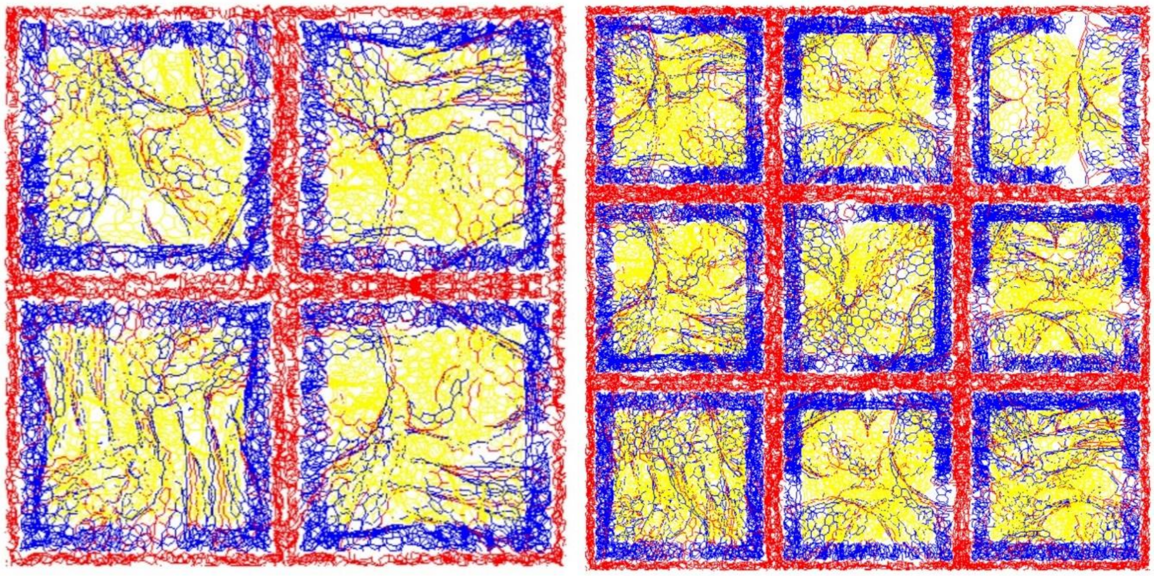


Figure 5 80Å 2 x 2 x 2 cell duplication (left) and a 120Å 3 x 3 x 3 cell duplication. HRMC occurs in the red regions while the blue regions are fixed atoms required for the neighbour lists of the blue region atoms. Yellow regions are removed entirely and need to be added back in after the simulation

A few notes,

- Following a HRMC simulation of the edge regions, the final xyz configuration (**xyz_final.xyz**) will be missing the removed atoms (yellow atoms in the figures). This will need to be added to the final configuration and these removed atoms are found in the **xyz_dupl_r.xyz** file.
- If the selection of the **du_Rinter** width is too small, the simulation will fail to conserve quantities such as the atomic energies and coordinations. It is recommended that small tests are performed to find this value (in the above using the carbon EDIP potential, 4Å was sufficient to avoid errors). Conservation errors are calculated at the end of the main stage prior to the data stage where an expensive n^2 calculation checks how well quantities were conserved over the course of the simulation.
- The g(r) is calculated and fitted out to a range equal to **du_Rinter**. This short range is required to avoid fitting void space.
- Atomic moves are confined within the red region and any moves out of this region are rejected.
- The energy per atom is higher than expected during a simulation and for the final configuration due to the blue atoms on the edge of the void space missing their neighbours. The energy will still minimize and will drop significantly when the yellow region atoms are merged with the simulation atoms at the end of the simulation (this has to be done manually outside of HRMC).

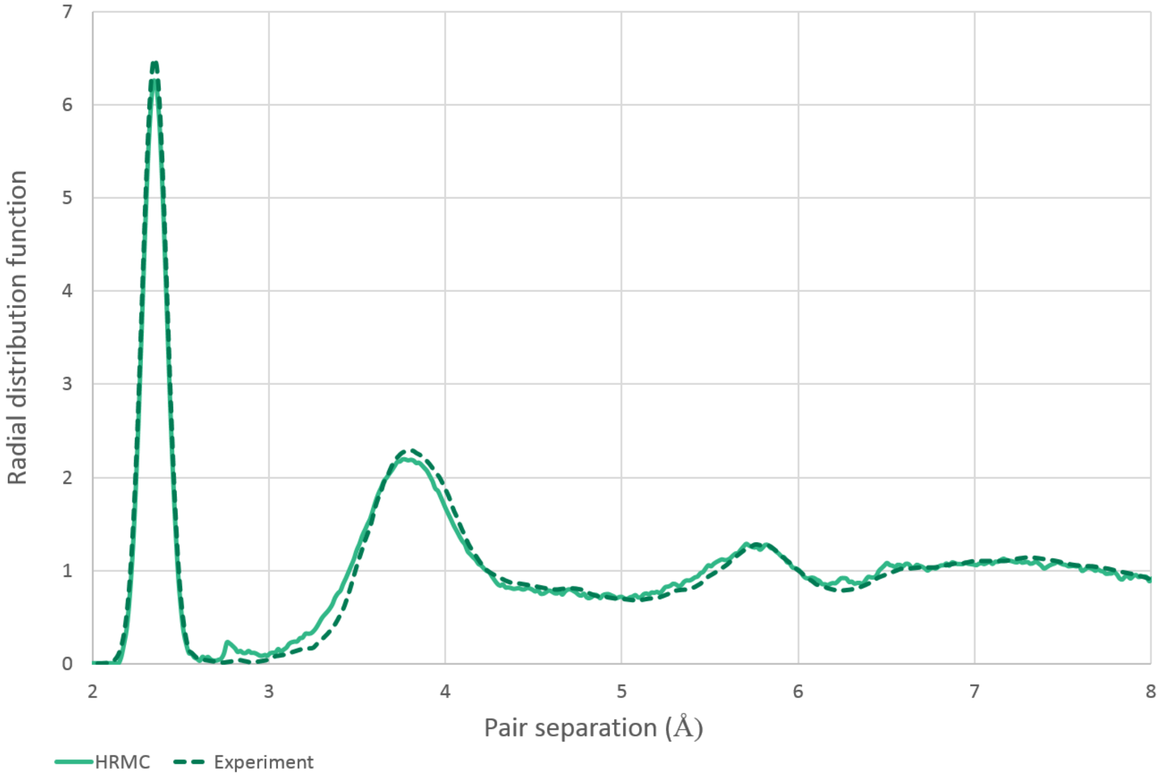
5 Example

The following example is included to aid the user.

5.1 Amorphous Silicon

Found within the **/Example/Amorphous Silicon** directory, is the input files for a HRMC simulation of amorphous silicon, starting from a random configuration of 400 atoms, using the modified Stillinger-Weber Potential[11] for the energy constraint (the parameters are contained in the **input_SW1.dat** file). Other fitting parameters include the g(r) and S(q) experimental data of Laaziri *et al*.[12], an average coordination distribution and a Gaussian bond angle distribution from previous a-Si HRMC models[13]. Geometric constraints such as a hard-core radius and penalties for three-member ring formation were also used. The weighting factors in this example were chosen to balance the various constraint influences with a bias towards the experimental data and potential energy cost functions, whilst maintaining an acceptance ratio above 10%. All constraints were minimised in four stages, using the linear option to smoothly connect weightings within each stage. When repeating this simulation, the results below should be similar, though not exact, to that of the user's results for this test system.

Figure 6 and 7 below show the HRMC fits to the diffraction data obtained over 2 million steps.



[11] R.L.C. Vink, G.T. Barkema, N. Mousseau, W.F. van der Weg, J. Non-Cryst. Solids **282** 248 (2001).

[12] K. Laaziri, S. Kycia, S. Roorda, M. Chicoine, J. L. Robertson, J. Wang, S. C. Moss, Phys. Rev. Lett. **82**, 3460(1999).

[13] G. Opletal, T. Petersen, I. Snook, D. McCulloch, J. Chem. Phys. **126** 214705(2007).

Figure 6 Fit comparison between final simulation and experimental radial distribution function g(r)

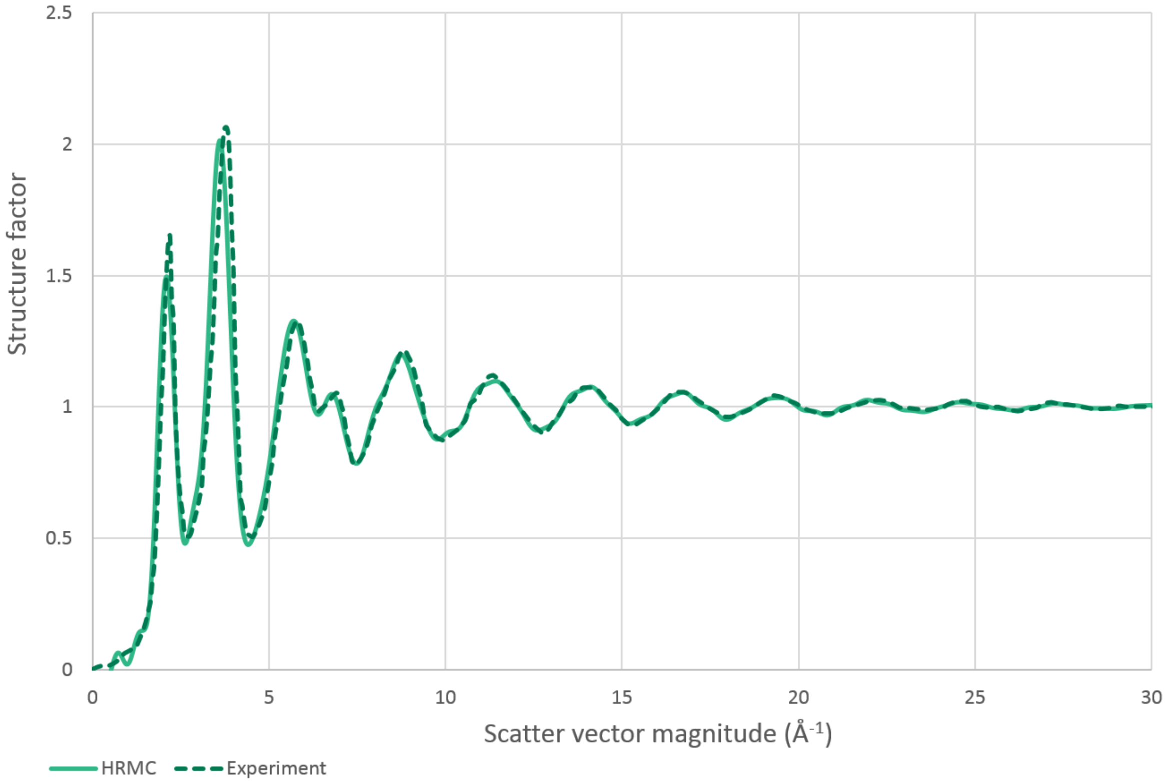


Figure 7 Fit comparison between final simulation and experimental structure factor S(q)

The input and fitted bond angle distribution functions are shown in figure 8.

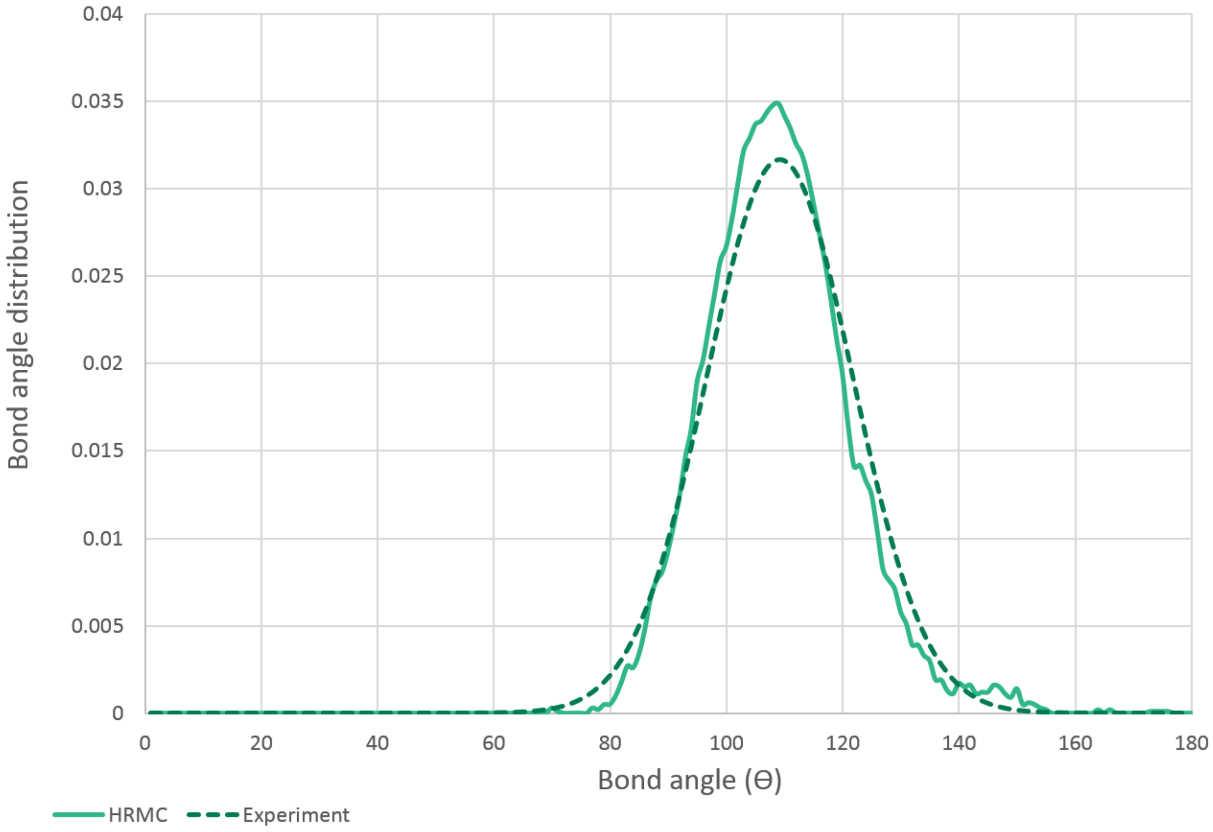


Figure 8 Fit comparison between final simulation and fitted bond angle distribution function

Similarly, the output percentage of three- and four-fold coordinated atoms was 0.20 and 0.79 respectively, close to the input values of 0.21 and 0.79 for the coordination constraint.

The final energy output of -2.85 eV per atom was obtained (this modified potential has a silicon diamond potential energy of -3.30eV/atom which is significantly lower than the -4.34eV/atom value for the regular Stillinger-Weber parameterized potential). The evolution of the cost functions is shown in figure 9.

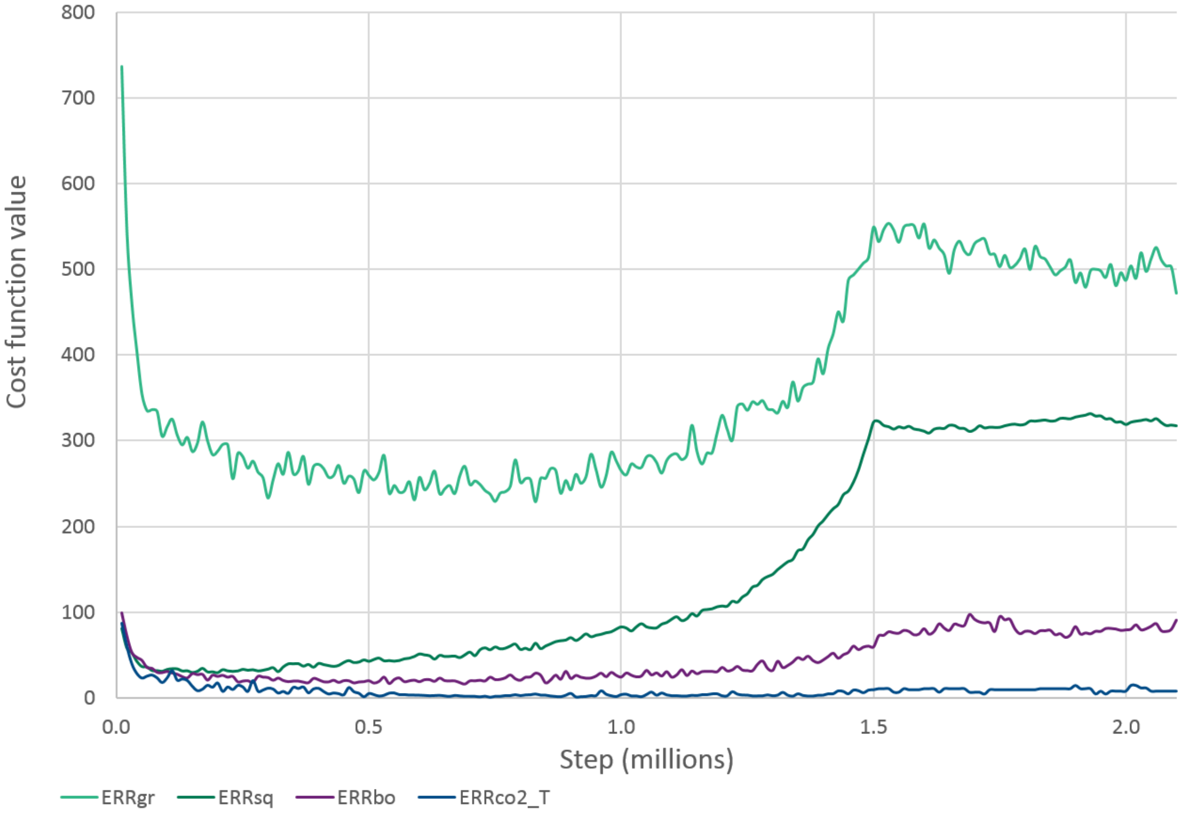


Figure 9 The evolution of the errors associated with the g(r), S(q), bond angle distribution and coordination constraints. The increase in errors near the end of the simulation is caused by the temperature quench which increases the influence of the empirical potential

The increases the diffraction and bond angle constraints towards the end of the minimisation stage shown in figure 9 are due to the energy constraint, which began to dominate as the temperature was decreased from 5000K to 300K.

CONTACT US

t 1300 363 400
+61 3 9545 2176
e csiroenquiries@csiro.au
w www.data61.csiro.au

FOR FURTHER INFORMATION

Dr George Opletal
e george.opletal@data61.csiro.au
w https://research.csiro.au

WE DO THE EXTRAORDINARY EVERY DAY

We innovate for tomorrow and help improve today – for our customers, all Australians and the world.

WE IMAGINE
WE COLLABORATE
WE INNOVATE