



Efficient stochastic generation of special quasirandom structures



A. van de Walle^{a,*}, P. Tiwary^a, M. de Jong^b, D.L. Olmsted^b, M. Asta^b, A. Dick^c,
D. Shin^d, Y. Wang^e, L.-Q. Chen^e, Z.-K. Liu^e

^a Brown University, United States

^b University of California, Berkeley, United States

^c Belarusian Institute of System Analysis, Belarus

^d Oak Ridge National Laboratory, United States

^e Pennsylvania State University, United States

ARTICLE INFO

Article history:

Received 8 February 2013

Received in revised form

30 April 2013

Accepted 24 June 2013

Available online 18 July 2013

Keywords:

Special quasirandom structures

Monte Carlo

Alloy theory

Ab initio methods

Disordered state

Solid solution

ABSTRACT

We present a new algorithm to generate Special Quasirandom Structures (SQS), i.e., best periodic supercell approximations to the true disordered state for a given number of atoms per supercell. The method is based on a Monte Carlo simulated annealing loop with an objective function that seeks to perfectly match the maximum number of correlation functions (as opposed to merely minimizing the distance between the SQS correlation and the disordered state correlations for a pre-specified set of correlations). The proposed method optimizes the shape of the supercell jointly with the occupation of the atomic sites, thus ensuring that the configurational space searched is exhaustive and not biased by a pre-specified supercell shape. The method has been implemented in the “mcsqs” code of the Alloy Theoretic Automated Toolkit (ATAT) in the most general framework of multicomponent multisublattice systems and in a way that minimizes the amount of input information the user needs to specify and that allows for efficient parallelization.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The accurate and efficient modeling of disordered crystalline alloys has a wide range of applications in materials science, as many technologically important materials with tunable properties take the form of disordered alloys. One popular, simple and transparent approach is to merely take a large supercell and randomly occupy each lattice site. However, this method is not the most efficient, as randomly generated structures in a finite supercell have a relatively large likelihood of deviating from “perfect” randomness in terms of local correlations between the chemical species occupation of nearby lattice sites. The concept of Special Quasirandom Structures (SQS) [1] provides a systematic improvement over this method. SQS represent the best periodic supercell approximation to the true disordered state for a given number of atoms per supercell. SQS are optimal according to the criterion that a specified set of correlations between neighboring site occupations in the SQS match the corresponding correlation of the true, fully disordered, state.

SQS have been very successfully used to obtain electronic and thermodynamic properties of disordered materials (see, for

example, [2–8]). SQS are traditionally generated by an exhaustive enumeration of all possible occupations of the sites in a supercell, an algorithm of exponential order of complexity. This is the only known method that is guaranteed to find optimal SQS, but it is only computationally tractable for small unit cells (currently up to about 25 atoms, even with the most efficient structure enumeration algorithms [9]). However, electronic structure methods can now routinely handle fairly large cells (e.g., over 50 atoms), thus making exhaustive enumeration impractical. The generation of SQS via stochastic approaches [5,10,11] thus becomes a desirable alternative.

Our contribution is four fold. First, we devise an SQS generation algorithm that seeks SQS that perfectly match the maximum number of correlation functions (as opposed to merely minimizing the distance between the SQS correlations and the disordered state correlations for a pre-specified set of correlations). We achieve this by generalizing the objective function traditionally used for stochastic SQS searches to include a term that “rewards” a perfect correlation match up to a given distance. Second, we implement the method in the most general framework of multicomponent multisublattice systems [12] in order to encompass large classes of technologically relevant alloys (many of which exhibit disorder on different sublattices [8]). Third, the proposed method optimizes the shape of the supercell jointly with the occupation of the atomic sites, thus ensuring that the

* Corresponding author. Tel.: +1 401 863 1476.

E-mail address: avdw@alum.mit.edu (A. van de Walle).

configurational space searched is exhaustive and not biased by a pre-specified supercell shape. Finally, we describe the implementation of this scheme in the Alloy Theoretic Automated Toolkit (ATAT) [12–14], which is designed to minimize the amount of input information the user needs to specify and exploits natural avenues for efficient parallelization.

This paper is organized as follows: we first recall the basic formalism needed to quantify spatial correlations in the disordered state, we then describe the algorithm and provide sample input and output from the ATAT implementation of the algorithm.

2. Cluster expansion formalism

Let us first briefly recall the cluster expansion formalism generalized for multicomponent multisublattice systems [12]. Let σ denote the vector of all the variables σ_i specifying the occupation of lattice site i . Typically, $\sigma_i = 0, \dots, M_i - 1$, if M_i distinct chemical species can occupy site i . We then need a way to quantify the correlation between the occupation of multiple sites i_1, i_2 , etc.

The list of sites considered in the calculation of a particular correlation is called a cluster, denoted α . In a binary system (where each site only has one compositional degree of freedom) it is sufficient to only specify which sites are part of the cluster ($\alpha_i = 1$ if site i belongs to cluster α and $\alpha_i = 0$ if not). In a multicomponent system, each site i of the cluster is assigned a numerical index α_i ranging from 0 to $M_i - 1$ that indicates which compositional degree of freedom is considered (with $\alpha_i = 0$ indicating that this site is not part of the cluster). The correlation associated with cluster α is then defined as

$$\rho_\alpha(\sigma) \equiv \langle \Gamma_\alpha(\sigma) \rangle_\alpha$$

where the average $\langle \dots \rangle_\alpha$ is taken over all clusters α' that are equivalent by symmetry to cluster α and where $\Gamma_{\alpha'}(\sigma)$ is a *cluster function*, defined as

$$\Gamma_\alpha(\sigma) = \prod_i \gamma_{\alpha_i, M_i}(\sigma_i) \quad (1)$$

where $\gamma_{\alpha_i, M_i}(\sigma_i)$ satisfies $\gamma_{0, M_i}(\sigma_i) = 1$ and the following orthogonality condition:

$$\frac{1}{M_i} \sum_{\sigma_i=0}^{M_i-1} \gamma_{\alpha_i, M_i}(\sigma_i) \gamma_{\beta_i, M_i}(\sigma_i) = \begin{cases} 1 & \text{if } \alpha_i = \beta_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

(In binary alloys, a common choice is $\gamma_{0,2}(0) = 1$, $\gamma_{0,2}(1) = 1$, $\gamma_{1,2}(0) = -1$, $\gamma_{1,2}(1) = +1$ and is directly related to the well-known Ising model.) The convention used in ATAT for multicomponent systems is described in [12]. Although the product (1) is, in principle, over all lattice sites, the choice $\gamma_{0, M_i}(\sigma_i) = 1$ ensures that it reduces to a product over sites within cluster α only.

3. Algorithm

Our algorithm can be described as follows. Let $\rho_\alpha(\sigma^{\text{rnd}})$ denote the correlations of the fully disordered state at some given composition while $\rho_\alpha(\sigma)$ denotes the correlations of a candidate SQS σ and let $\Delta\rho_\alpha(\sigma) = \rho_\alpha(\sigma) - \rho_\alpha(\sigma^{\text{rnd}})$. Note that $\rho_\alpha(\sigma^{\text{rnd}})$ can be easily calculated from the fact that, in the disordered state, site occupations are independent so that

$$\rho_\alpha(\sigma^{\text{rnd}}) = \left\langle \prod_i \gamma_{\alpha_i, M_i}(\sigma_i) \right\rangle_\alpha = \prod_i \langle \gamma_{\alpha_i, M_i}(\sigma_i) \rangle_\alpha$$

where $\langle \gamma_{\alpha_i, M_i}(\sigma_i) \rangle_\alpha$ can be directly computed from the average composition of site i .

Our objective function (to be minimized) is

$$Q = -\omega L + \sum_{\alpha \in A} |\Delta\rho_\alpha(\sigma)| \quad (3)$$

where

- L is the largest l such that $\Delta\rho_\alpha(\sigma) = 0$ for all clusters α with $\text{diam}(\alpha) \leq l$ (with $\text{diam}(\alpha)$ being the length of the largest pair contained in cluster α);
- A is a user-specified set of clusters (which is typically much bigger than the set for which we hope to match the correlations exactly);
- ω is a user-specified weight.

The $-\omega L$ in the objective function (3) is an important distinguishing feature of our approach. It reflects the fact that the quality of an SQS is traditionally measured in terms of the number of correlations of the fully disordered state it is able to match exactly. Typically, one attempts to preferably match shorter-range correlations while gradually enlarging the supercell to extend the range of matching correlations until convergence of the properties of interest. Traditionally, SQS are built based on matching the pair correlations, although for best accuracy, multibody correlations should be considered as well. The second term in (3) adds the absolute difference of the error in the remaining correlations and serves to guide the stochastic search in the right direction to extend the range of perfect match.

The objective function (3) is minimized using the following steps.

1. All supercells having a user-specified number n of atoms per cell are enumerated. (Since cell enumeration has polynomial complexity (in n), an exhaustive enumeration is computationally tractable.)
2. For each supercell, lattice sites are occupied randomly according to the user-specified composition on each sublattice. The algorithm proceeds by placing exactly $n_s x_{ts}$ atoms of type t on sublattice s , where n_s is the number of sites on sublattice s in the supercell and x_{ts} is a user-specified composition. This is accomplished by randomly drawing one of the $n_s!$ possible random permutations of the n_s sites on sublattice s and using it to shuffle the atomic occupations. (The code aborts if $n_s x_{ts}$ is not an integer, since this would mean that even the requested composition cannot be matched with the given supercell size.)
3. Among each supercell thus obtained, the best SQS, according to criterion (3), is identified. (All other supercells are also kept in memory for future use.)
4. The objective function (3) is then used in a simulated annealing loop in which configurations are sampled with a probability proportional to $\exp(-Q/T)$, where T is a user-specified fictitious temperature, using a Metropolis algorithm. At each step, the algorithm attempts to jump to a random different supercell shape and attempts to permute two atoms within the same sublattice of that supercell. Such moves are accepted or rejected based on the Metropolis algorithm.
5. The newly visited supercell and configuration is compared to the best SQS obtained so far and the latter is updated if needed. Every time a better SQS is found, it is output to disk.

The above algorithm can be easily parallelized: one simply needs to run multiple, independent, instances of the code (ensuring that the random number generators have a different seed), letting them search different portions of configuration spaces and keeping the best SQS generated among all of the processes.

Although it is unlikely, the SQS thus generated may happen to have a representation with smaller unit cell. As a result a search of SQS of size n implicitly also searches for SQS whose size is a factor of n . Another command of the ATAT package (`cellcvrt-s <bestsq.out>`) allows the user to check if a smaller unit cell exists.

4. Discussion

A few words of caution regarding the use of SQS are in order. First, the accuracy of an SQS for the calculation of a given physical property in a particular system is not only a function of the number of correlations that match the correlations of the fully disordered state. Different properties have different dependences on the correlations and this dependence is itself system-dependent, as previously noted in [1,10,15]. Hence, one cannot know a priori (based on purely geometrical grounds) how many (and which) correlations need to be matched to reach a given accuracy. It is therefore prudent to always investigate the convergence of the property of interest over a sequence of different SQS of increasing range of perfectly matched correlations.

This being said, there are fundamental theoretical reasons to expect such convergence to occur relatively quickly for properties such as the total energy. As is well-known in the field of so-called “order N” methods, numerous physical arguments lead to localized expansions of the total energy [16]. Such expansions naturally suggest a cluster hierarchy [13] in which smaller clusters (both in terms of spatial extent and number sites) tend to be more important. A rapid convergence is not automatic for any physical property, however. In extreme cases, properties such as the band gap or electrical conductivity may have a strong dependence on long-wavelength components that would result in SQS having poor predictive power.

Another potential problem with SQS lies in the prediction of tensor-valued properties such as elastic constants. An apparently good SQS for scalar properties may turn out to generate artificially anisotropic tensorial properties [17]. This problem arises because the coupling between correlations and properties follows different symmetry equivalence rules than in the scalar case [18]. Fortunately, this issue can easily be corrected by a symmetrization technique [19,17].

5. Code usage

5.1. Input files

In the description below, quantities in square brackets “[]” need to be replaced by the appropriate value, without brackets, while items not in bracket are literal. The `mcsqs` code requires 2 input files.

First, a file defining the random state (by default `rndstr.in`, which can be overridden with the `-l=[filename]` option) is needed. Its format is as follows (it is similar to ATAT’s `lat.in` file

that is needed for `maps` or `corrump` but with partial occupation of the sites):

1. First, the coordinate system \vec{a} , \vec{b} , \vec{c} is specified, either as

$$\begin{bmatrix} a & b & c \\ \alpha & \beta & \gamma \end{bmatrix} \quad (4)$$

or in terms of Cartesian coordinates, one axis per line:

$$\begin{bmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{bmatrix} \quad (5)$$

2. Then the lattice vectors \vec{u} , \vec{v} , \vec{w} are listed, one per line, expressed in the coordinate system just defined:

$$\begin{bmatrix} u_a & u_b & u_c \\ v_a & v_b & v_c \\ w_a & w_b & w_c \end{bmatrix} \quad (6)$$

3. Finally, the positions x_i of the crystallographic sites (expressed in the same coordinate system as the lattice vectors) are given, along with possible atom type(s) t_{1i}, t_{2i}, \dots occupying site i and the corresponding occupations (or concentration) c_{1i}, c_{2i}, \dots :

$$\begin{bmatrix} x_{a1} & x_{b1} & x_{c1} & [t_{11} = c_{11}, t_{21} = c_{21}, \dots] \\ x_{a2} & x_{b2} & x_{c2} & [t_{12} = c_{12}, t_{22} = c_{22}, \dots] \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Note that the concentrations must sum up to 1 for each site. An example of such file is given in Table 1. All symmetry equivalent sites must also have the same occupations. To override this requirement, “dummy” species (with zero occupation) that differ in the two sublattices must be included.

The second required input file (by default `clusters.out`, but this can be overridden with the `-cf=[filename]` option) indicates the clusters that the code must consider in attempting to match the disordered state’s correlations. This file can be generated with the `corrump` utility, for instance with the command line:

```
corrump -l=rndstr.in -ro -noe -nop -clus -2=...
-3=...
```

The (new) `-ro` option allows `corrump` to read the same lattice input file as `mcsqs` (here `rndstr.in`). The `-noe` and `-nop` skip the empty and point clusters that are not used by `mcsqs`. The `-clus` option instructs the code to generate clusters only. Finally, the options `-2=... -3=...` indicate the range of pairs, triplets, etc. More information can be found by typing `corrump -h`.

The following input files are optional.

- Creating a file called `sqsparm.in` (or as specified by the `-pf=[filename]` option) provides a way to set the two parameters controlling the optimization process: (i) the weight ω in Eq. (3) and (ii) the “temperature” T used in the simulated

Table 1

Sample input file (for an Al–Ti alloy with 25 atomic % Al and adopting the hcp crystal structure) defining the geometry of the lattice and the SQS composition desired.

```
3.1 3.1 5.062 90 90 120
1 0 0
0 1 0
0 0 1
0 0 0 Al=0.25, Ti=0.75
0.6666 0.3333 0.5 Al=0.25, Ti=0.75
```

(Coordinate system: $a b c \alpha \beta \gamma$ notation)
(Primitive unit cell: one vector per line
expressed in multiples of the above coordinate
system vectors)
(Atoms in the lattice)

annealing loop. This file just contains two numbers

$[\omega]$ [T] (7)

and can be changed while the code is running to fine-tune performances at run-time. This file is reread every *rt* Monte Carlo steps, as set by the `-rt=[rt]` option.

- Creating a file called `stopsqs` stops the code cleanly.
- If the `-rc` option is set, the file `sqs cell.out` is read in to specify which supercells are to be considered. Normally (without the `-rc` option), the `sqs cell.out` is generated automatically by the code via exhaustive enumeration. The file starts with the number *ns* of supercells and is followed by a list of the *ns* supercells in the format described by Eq. (6) above. This option is useful in two cases: (i) if the code needs to be rerun multiple times (or run in parallel), this saves the supercell generation time or (ii) if brute force enumeration takes too long, a few specific supercells can be given.

5.2. Command line options

The command is invoked as `mcsqs [options]`, where the options are one or more of the following.

- `-n=[int]` Specifies the desired number of atom/unit cell (must be a multiple of the number of sites in the `rndstr.in`). This is the only required command-line parameter.
- `-l=[string]` Input file defining the random structure (default: `rndstr.in`)
- `-cf=[string]` Input file defining the clusters (default: `clusters.out`)
- `-tol=[real]` Tolerance for matching correlations of the SQS and of the ideal random state (default: $1e-3$). Values of $|\Delta\rho_\alpha(\sigma)|$ below that threshold are considered to be 0, for the purpose of evaluating Eq. (3).
- `-wr=[real]` Weight (ω in Eq. (3)) assigned to the range of perfect correlation match in the objective function (default: 1).
- `-T=[real]` Temperature *T* used in the simulated annealing loop of the Monte Carlo (default: 1).
- `-pf=[string]` Specify the name of the input file defining the optimization parameters (default: `sqsparam.in`). The file (whose format is described in Eq. (7) above) is always attempted to be read (whether the `-pf` option is specified or not) and, if present, the settings found in this file override those given by the `-wr` and `-T` options.
- `-rc` Read supercells from file `sqs cell.out`. If not specified, the code generates the supercells internally and writes them to `sqs cell.out`.
- `-ip=[int]` Specifies the index *i* of current process (for parallel operation). One can independently run any number of instances of the code. This index is used as a suffix on the various output files, so that if the instances run in the same directory, no file clashes occur. It is the user's responsibility to find the best SQS among those generated by the different processes (this is straightforward, since the `bestcorr[i].out` files give the best value of the objective function found so far by process *i*). Just type `grep Objective_function bestcorr*.out|sort-n`.
- `-crf=[string]` Select correlation functions (default: `trigo`, a convention described in [12]). See file `corrskel.c++` for instructions regarding how to add user-defined correlation functions.
- `-sd=[int]` Seed for random number generation (default: use clock).
- `-rt=[int]` Read parameter file (`sqsparam.in`) every time these many Monte Carlo steps have been performed (default: 10 000).

`-sig=[int]` Number of significant digits to print in output files (default: 6).

`-h` Displays more help

5.3. Output files

The `bestsqs.out` file (or `bestsqs[i].out` in parallel mode, where *i* is the index specified by the `-ip` option) contains the best SQS found so far in standard ATAT structure file format (which is similar to that shown in Table 1, but with only one specie per site and without occupation variables). See Table 3 for an example.

The `bestcorr.out` file (or `bestcorr[i].out` in parallel mode) contains the correlations of the best SQS found so far (3rd column), along with the target disordered state correlation (4th column), and the difference between the two (5th column). Each correlation is prefixed by characteristics of the corresponding cluster: its number of point (1st column) and its diameter (2nd column).

A file called `rndstrgrp.out` contains the same information as the input file defining the random state (e.g., `rndstr.in`), but with symmetrically equivalent sites grouped together and separated by blank lines. This helps to determine which sites can have the same occupations. This file can be used as an input file in a later run.

A log file with miscellaneous self-explanatory information can be found in `mcsqs.log` (or `mcsqs[i].log` in parallel mode).

The code stops if a perfect match is found (all correlations requested match the disordered state), but this may never happen

Table 2

Sample input file (for a disordered Re–W alloy adopting a D8₈ crystal structure) defining the geometry of the lattice and the SQS composition desired. For clarity, the 5 symmetrically distinct sublattices are separated by blank lines.

```

10.759 10.759 5.656 90 90 90
1 0 0
0 1 0
0 0 1

0.000 0.000 0.500 Re
0.500 0.500 0.000 Re

0.561 0.235 0.000 Re
0.235 0.561 0.000 Re
-0.235 -0.561 0.000 Re
-0.561 -0.235 0.000 Re
1.061 0.265 -0.500 Re
0.265 1.061 -0.500 Re
0.735 -0.061 -0.500 Re
-0.061 0.735 -0.500 Re

0.103 0.103 0.000 W
-0.103 -0.103 0.000 W
0.603 0.397 -0.500 W
0.397 0.603 -0.500 W

0.318 0.318 0.730 Re=.5, W=.5
-0.318 -0.318 0.730 Re=.5, W=.5
0.318 0.318 0.270 Re=.5, W=.5
-0.318 -0.318 0.270 Re=.5, W=.5
0.818 0.182 0.230 Re=.5, W=.5
0.182 0.818 0.230 Re=.5, W=.5
0.818 0.182 -0.230 Re=.5, W=.5
0.182 0.818 -0.230 Re=.5, W=.5

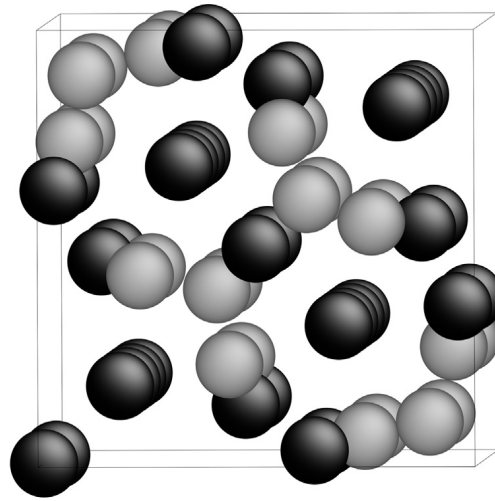
0.367 0.038 0.000 Re=.5, W=.5
0.038 0.367 0.000 Re=.5, W=.5
-0.038 -0.367 0.000 Re=.5, W=.5
-0.367 -0.038 0.000 Re=.5, W=.5
0.867 0.462 -0.500 Re=.5, W=.5
0.462 0.867 -0.500 Re=.5, W=.5
0.538 0.133 -0.500 Re=.5, W=.5
0.133 0.538 -0.500 Re=.5, W=.5

```

Table 3Sample SQS for a disordered Re–W alloy adopting a $D8_b$ crystal structure.

10.759	0.000	0.000
0.000	10.759	0.000
0.000	0.000	5.656
0.000	-1.000	0.000
1.000	0.000	0.000
0.000	0.000	2.000
<hr/>		
0.318	-0.682	0.730 W
0.318	-0.682	1.730 W
0.318	-0.682	1.270 Re
0.318	-0.682	0.270 W
0.682	-0.318	0.730 Re
0.682	-0.318	1.730 W
0.682	-0.318	1.270 W
0.682	-0.318	0.270 W
0.818	-0.818	0.230 Re
0.818	-0.818	1.230 Re
0.818	-0.818	0.770 Re
0.818	-0.818	1.770 W
0.182	-0.182	0.230 W
0.182	-0.182	1.230 Re
0.182	-0.182	0.770 Re
0.182	-0.182	1.770 Re
0.038	-0.633	2.000 W
0.038	-0.633	1.000 Re
0.367	-0.962	2.000 W
0.367	-0.962	1.000 W
0.633	-0.038	2.000 Re
0.633	-0.038	1.000 W
0.962	-0.367	2.000 Re
0.962	-0.367	1.000 Re
0.538	-0.867	1.500 Re
0.538	-0.867	0.500 W
0.867	-0.538	1.500 W
0.867	-0.538	0.500 W
0.133	-0.462	1.500 Re
0.133	-0.462	0.500 Re
0.462	-0.133	1.500 Re
0.462	-0.133	0.500 W
1.000	-1.000	0.500 Re
1.000	-1.000	1.500 Re
0.500	-0.500	2.000 Re
0.500	-0.500	1.000 Re
0.103	-0.897	2.000 W
0.103	-0.897	1.000 W
0.897	-0.103	2.000 W
0.897	-0.103	1.000 W
0.603	-0.603	1.500 W
0.603	-0.603	0.500 W
0.397	-0.397	1.500 W
0.397	-0.397	0.500 W
0.235	-0.439	2.000 Re
0.235	-0.439	1.000 Re
0.561	-0.765	2.000 Re
0.561	-0.765	1.000 Re
0.439	-0.235	2.000 Re
0.439	-0.235	1.000 Re
0.765	-0.561	2.000 Re
0.765	-0.561	1.000 Re
0.735	-0.061	1.500 Re
0.735	-0.061	0.500 Re
0.061	-0.735	1.500 Re
0.061	-0.735	0.500 Re
0.939	-0.265	1.500 Re
0.939	-0.265	0.500 Re
0.265	-0.939	1.500 Re
0.265	-0.939	0.500 Re

if there are too many clusters in clusters.out or -n (the number of atoms) is too small. In any case, the bestsq.out and bestcorr.out always contain the best solution found so far. Stopping the code prematurely (by typing `touch stopsqs`), if the solution is satisfactory, poses no problem.

**Fig. 1.** Graphical representation of the SQS given in Table 3.

6. Example

As an example of non-trivial SQS, we consider a disordered σ -Fe–Cr ($D8_b$) structure, which has a 30-atom unit cell and 5 symmetrically distinct sublattices [20,21]. This structure is relevant to the study of magnetic Fe–Cr alloys [22] as well as a number of high-temperature Re alloys [21], such as the Re–W alloy we consider below. We follow the prescription of Table 1 in [20] to determine likely occupations of the sites (they classify the 5 distinct sublattices into 3 sublattices, each sharing the same number of nearest neighbors). The two sublattices with the smallest coordination number (12) are fully occupied with Re while the sublattice with the largest coordination number (15) is fully occupied with W. The remaining two sublattices, with an intermediate number of nearest neighbors (14), are modeled as fully disordered with a composition of 50%. The appropriate `rndstr.in` input file is shown in Table 2 and the following command was used to generate the clusters:

```
corrdump -l=rndstr.in -ro -noe -nop -clus -2=6
-3=5.2 -4=5.2
```

This generated 15 pairs, 8 triplets and 3 quadruplets (in addition to the 2 point correlations). The SQS generation code was then simply run as

```
mcsqs -n=60
```

The resulting SQS is shown in Table 3 and Fig. 1. As seen in Table 4, its correlations exactly reproduce those of the disordered state over a range that corresponds to about 1.5 times the nearest neighbor shell radius (about 3.32–3.61 Å in this system).

7. Conclusion

The `mcsqs` code presented herein should prove especially useful in CALPHAD-type modeling since SQS provide a direct and an efficient way to obtain the enthalpy of mixing of a solid solution in the high-temperature limit. The current implementation is somewhat unique in terms of its ease-of-use and generality, which should greatly facilitate the use of ab initio data in CALPHAD modeling.

Table 4

Correlations of the SQS given in Table 3. For a given cluster α , let n_α be the number of sites it contains, let l_α be its diameter, let m_α be its multiplicity and let ρ_α be its associated correlation. (A perfectly random state would have correlations that are all zeros in this case.) Italicized correlations were not included in the objective function. Sublat. indicates which sublattices (in the order listed in Table 2) a given cluster overlaps.

n_α	l_α	m_α	Sublat.	ρ_α	n_α	l_α	m_α	Sublat.	ρ_α
2	2.60176	4	44	0	3	3.35509	8	445	0
2	2.97641	4	55	0	3	3.41838	8	445	0
2	3.05424	4	44	0	3	3.52521	16	555	0.25
2	3.35509	16	45	0	3	3.52521	16	545	0
2	3.41838	16	54	0	3	5.00591	8	545	0
2	3.52521	16	55	0	3	5.13818	16	445	0
2	5.00591	4	55	0	3	5.13818	16	455	0
2	5.13818	16	45	0	3	5.13818	8	554	0
2	5.25714	16	45	0	3	5.25714	8	445	0
2	5.33746	16	45	0	3	5.25714	16	455	0
2	5.34148	16	54	0	3	5.25714	16	554	−0.25
2	5.53845	4	44	0	3	5.25714	16	554	0.25
2	5.57954	16	44	0					
2	5.656	8	44	0	4	3.52521	4	5555	0
2	5.656	8	55	0	4	5.13818	16	4455	0
2	6.11912	4	44	0.5	4	5.13818	8	4455	0
2	6.16229	4	55	−0.5	4	5.25714	8	4455	0.5
					4	5.25714	16	4455	0.25
					4	5.25714	16	5554	0.25
					4	5.25714	16	5554	0

Acknowledgments

The authors acknowledge financial support from the US Office of Naval Research (ONR) under Grant nos. N00014-11-1-0886 and N00014-12-1-0557. This work made use of computational resources provided under the Extreme Science and Engineering Discovery Environment (XSEDE).

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.calphad.2013.06.006>.

References

- [1] A. Zunger, S.-H. Wei, L. Ferreira, J.E. Bernard, Special quasirandom structures, *Phys. Rev. Lett.* 65 (3) (1990) 353.
- [2] K.C. Hass, L.C. Davis, A. Zunger, Electronic structure of random $\text{Al}_{0.5}\text{Ga}_{0.5}\text{As}$ alloys: test of the ‘special-quasirandom-structure’ description, *Phys. Rev. B* 42 (6) (1990) 3757.
- [3] G. Ghosh, A. van de Walle, M. Asta, First-principles calculations of properties of bcc, fcc and hcp solid solutions in Al–TM (TM=Ti, Zr, Hf) systems: a comparison between cluster expansion and supercell methods, *Acta Mater.* 56 (2008) 3202.
- [4] D. Shin, R. Arroyave, Z.-K. Liu, A. van de Walle, Thermodynamic properties of binary hcp solution phases from special quasirandom structures, *Phys. Rev. B* 74 (2006) 024204.
- [5] D. Shin, A. van de Walle, Y. Wang, Z.-K. Liu, First-principles study of ternary fcc solution phases from special quasirandom structures, *Phys. Rev. B* 76 (2007) 144204.
- [6] A. van de Walle, G. Ceder, U.V. Waghmare, First-principles computation of the vibrational entropy of ordered and disordered Ni_3Al , *Phys. Rev. Lett.* 80 (1998) 4911.
- [7] D.W. Shin, Z.K. Liu, Enthalpy of mixing for ternary fcc solid solutions from special quasirandom structures, *CALPHAD J.* 32 (2008) 74.
- [8] C. Jiang, C.R. Stanek, K.E. Sickafus, B.P. Uberuaga, First-principles prediction of disordering tendencies in pyrochlore oxides, *Phys. Rev. B* 79 (2009) 104203.
- [9] G.L.W. Hart, R.W. Forcade, Algorithm for generating derivative structures, *Phys. Rev. B* 77 (2008) 224115.
- [10] I.A. Abrikosov, S.I. Simak, B. Johansson, A.V. Ruban, H.L. Skriver, Locally self-consistent green’s function approach to the electronic structure problem, *Phys. Rev. B* 56 (1997) 9319.
- [11] J. von Pezold, A. Dick, M. Friák, J. Neugebauer, Generation and performance of special quasirandom structures for studying the elastic properties of random alloys: application to Al–Ti, *Phys. Rev. B* 81 (2010) 094203.
- [12] A. van de Walle, Multicomponent multisublattice alloys, nonconfigurational entropy and other additions to the Alloy Theoretic Automated Toolkit, *CALPHAD J.* 33 (2009) 266.
- [13] A. van de Walle, G. Ceder, Automating first-principles phase diagram calculations, *J. Phase Equilibria* 23 (2002) 348.
- [14] A. van de Walle, M. Asta, G. Ceder, The alloy theoretic automated toolkit: a user guide, *CALPHAD J.* 26 (2002) 539.
- [15] A.V. Ruban, I.A. Abrikosov, Configurational thermodynamics of alloys from first principles: effective cluster interactions, *Rep. Prog. Phys.* 71 (2008) 046501.
- [16] S. Goedecker, Linear scaling electronic structure methods, *Rev. Mod. Phys.* 71 (1999) 1085.
- [17] F. Tasnádi, M. Odén, I.A. Abrikosov, Ab initio elastic tensor of cubic $\text{Ti}_{0.5}\text{Al}_{0.5}\text{N}$ alloys: dependence of elastic constants on size and shape of the supercell model and their convergence, *Phys. Rev. B* 85 (2012) 144112.
- [18] A. van de Walle, A complete representation of structure–property relationships in crystals, *Nat. Mater.* 7 (2008) 455.
- [19] J.Z. Liu, A. van de Walle, G. Ghosh, M. Asta, Structure, energetics, and mechanical stability of Fe–Cu bcc alloys from first-principles calculations, *Phys. Rev. B* 72 (2005) 144109.
- [20] J.-O. Andersson, A.F. Guillermet, M. Hillert, B. Jansson, B. Sundman, A compound-energy model of ordering in a phase with sites of different coordination numbers, *Acta Metall.* 34 (1986) 437.
- [21] Z.-K. Liu, Y.A. Chang, Evaluation of the thermodynamic properties of the Re–Ta and Re–W systems, *J. Alloys Compd.* 299 (2000) 153.
- [22] J. Cieslak, J. Tobola, S.M. Dubiel, W. Sikora, Magnetic properties of σ –FeCr alloys as calculated with the charge- and spin-self-consistent KKR(CPA) method, *Phys. Rev. B* 82 (2010) 224407.