

ポテンシャル合わせ込みソフトウェア マニュアル (ver1.00)

1. 実行方法

1.1 動作環境

Intel C/C++コンパイラ, もしくは g++コンパイラのインストールされた Linux で動作確認済み.

特に必要なライブラリもなく, 特殊な文法も使っていないので, 一般的な C++コンパイラがあれば, 動作するとは考えられる.

1.2 インストール

make を行う.

makefile をコンパイラによって変えて make を行う.

(1) icc の場合

```
% cp makefile.icc makefile
```

```
% make
```

(2) g++ の場合

```
% cp makefile.gnu makefile
```

```
% make
```

実行ファイル GA が作成される.

1.3 実行方法

./GA 作業ディレクトリ

作業ディレクトリには入力ファイル

- para.dat
- set.dat
- cond.dat
- element.dat

が必要となる.

条件によっては

- cont.dat (cond.dat において iscont=1 の場合)
- posi.dat (set.dat で構造数が 1 以上の場合)
- lattice.dat (set.dat で構造数が 1 以上の場合)
- energy.dat (set.dat で構造数が 1 以上の場合)
- ポテンシャル指定ファイル(element.dat で 2 元系, 3 元系を指定した場合)

が必要.

2. 入力ファイル

2.1 para.dat

ポテンシャルパラメータについての値範囲・合わせ込むかどうかの設定を行う。

書式

パラメータ数が n 個のとき、

```
0 パラメータ条件 パラメータ名 パラメータ下限 パラメータ上限
1 パラメータ条件 パラメータ名 パラメータ下限 パラメータ上限
...
n-1 パラメータ条件 パラメータ名 パラメータ下限 パラメータ上限
```

解説・注意事項など

- ・パラメータ条件は **FREE** (自由に変化させられる) と **FIX** (固定値) の 2 種類。
- ・**FREE** 条件の場合は下限値が上限値より小さくなるようにすること。
- ・行数を読み取ってパラメータ数としてカウントするので、最終行以降に余計な改行を加えないこと。

サンプル

0	FREE	A_AA	0.1000000000000000	10000.0000000000000000
1	FREE	B_AA	0.1000000000000000	10000.0000000000000000
2	FREE	L1_AA	0.1000000000000000	5.0000000000000000
3	FREE	L2_AA	0.1000000000000000	5.0000000000000000
4	FREE	ETB_AA	0.1000000000000000	30.0000000000000000
5	FIX	DT*ET_AA	0.5000000000000000	0.5000000000000000
6	FREE	p_AAA	0.1000000000000000	10.0000000000000000
7	FIX	q_AAA	3.0000000000000000	3.0000000000000000
8	FREE	c1_AAA	0.0000000000000000	1.0000000000000000
9	FREE	c2_AAA	0.0000000000000000	100.0000000000000000
10	FIX	c3_AAA	100.0000000000000000	100.0000000000000000
11	FREE	h_AAA	-1.0000000000000000	-0.7500000000000000
12	FIX	R1_AA	2.7000000000000000	2.7000000000000000
13	FIX	R2-R1_AA	0.3000000000000000	0.3000000000000000
14	FIX	Re_AA	0.0000000000000000	0.0000000000000000
15	FIX	Rcut	3.3000000000000000	3.3000000000000000

2.2 cond.dat

遺伝的アルゴリズム(GA)の条件ファイル。

書式

id: 1-4 のいずれかの整数値(ポテンシャル id の値)
iscont: 0 か 1 (0 は新規, 1 は継続)
nstep: 整数値 (GA のステップ数)
nint: nstep 以下の整数値(出力インターバルステップ数)

noi: 整数値 (GA における個体数)
nout: noi 以下の整数値 (出力する個体数)
nmut: パラメータ数(para.dat の行数)以下の整数値, もしくは 0 (突然変異させる遺伝子の数)
murate: 0-1 の実数値 (突然変位の範囲)
seed: 整数値 (乱数のシードの値)
scale: 0 か 1(0 はスケーリングを行わない, 1 はスケーリングを行う)

解説・注意事項など

id:1,2,3,4

ポテンシャル関数形の指定.

1:Tersoff ポテンシャル

2:GEAM ポテンシャル

3:LRFS(Long-range Finnis Sinclair)ポテンシャル

4:Finnis-Sinclair ポテンシャル

5:Morse ポテンシャル

6:Lennard-Jones ポテンシャル

7:拡張型 GEAM ポテンシャル

iscont: 0 か 1

0 なら新規スタート, 1 なら継続スタート. 継続スタートの場合は cont.dat が必要.

nstep: GA のステップ数 通常 20-100 くらい

nint:出力インターバルステップ数

出力インターバルステップごとに個体の値が gene.dat に出力される

noi:個体数, 通常 1000 以上

多いほど大きな探索範囲から探すことができるが, 余り個体数が多いと計算時間がかかる.

nout:出力個体数 noi より少ない値にする

ステップ数 nint ごとに nout-1 の個体に対応するポテンシャルパラメータが gene.dat に出力される. noi (個体数) より小さくしてはならない.

nmut:突然変位遺伝子数 ほぼ 0 で問題ない

para.dat で FREE に設定した遺伝子(ポテンシャルパラメータ)の内, ランダムに nmut だけ選んで突然変位させる. 0 にすれば, FREE に設定したすべての遺伝子を突然変異させる.

mutrate:突然変位範囲制御パラメータ 通常 0.01-0.1 くらい

突然変異時に値の $(1+mutrate)$ ×元の値を最大, $(1-mutate)$ ×元の値を最大, として変化させる

seed: 乱数発生シード, 0 なら時間に依存したランダムシード

scale: スケーリングを行うか 0 か 1

スケーリングは強制的に 1 構造のエネルギーと格子定数が合わせ込み値とおなじになるようにすること. 0 はスケーリングを行わない, 1 はスケーリングを行う. スケーリング可能な場合はしたほうが当然良いものの, スケーリング機能は一部のポテンシャルにしか用意されていないので注意が必要.

サンプル

```
id:      1
iscont:  0
nstep:   51
nint:    5
noi:     50000
nout:    1000
nmut:    0
mutrate: 0.1
seed:    1
scale:   1
```

2.3 element.dat

元素指定ファイル

2.3.1 単元系(元素 1 種類)の場合

1 行目に 1 と書くと単元系ポテンシャル合わせ込みになる.

書式

```
1
元素名
```

サンプル

```
1
Si
```

2.3.2 二元系(元素 2 種類)の場合

1 行目に 2 と書いた場合は 2 元系ポテンシャル合わせ込みになる. 1 種類目の元素を元素 1, 2 種類目の元素を元素 2 と表記する.

書式

```
2
元素 1 の元素名    元素 2 の元素名
元素 1 のポテンシャルファイル名    元素 2 のポテンシャルファイル名
```

解説・注意事項など

ポテンシャルファイルとは単元系合わせ込みで得られた出力されたポテンシャルファイル (poten.dat) のこと (後述). これをリネームして単元系ポテンシャルファイルとして指定する (poten.dat のままだと, 上書きされてしまうので厳禁.). 例えば, Si のポテンシャルファイルであったら potenSi.dat というようリネームにすると良い.

サンプル

2

Si O

potenSi.dat potenO.dat

2.3.3 三元系の場合

1 行目に 3 と書いた場合は 3 元系ポテンシャル合わせ込みになる. 1 種類目の元素を元素 1, 2 種類目の元素を元素 2, 3 種類目の元素を元素 3 と表記する.

書式

3

元素 1 の元素名 元素 2 の元素名 元素 3 の元素名

元素 1 のポテンシャルファイル名 元素 2 のポテンシャルファイル名 元素 3 のポテンシャルファイル名

元素 1 と元素 2 の間のポテンシャルファイル名 元素 1 と元素 3 の間のポテンシャルファイル名 元素 2 と元素 3 の間のポテンシャルファイル名

2.4 cont.dat

継続のための前回の合わせ込みによる個体ファイル

書式

コメント行

ポテンシャルパラメータ名 ポテンシャルパラメータの値

を para.dat の行数だけ繰り返す.

解説・注意事項など

cond.dat で iscont が 1 のとき必要になる. 前回合わせ込みの gene.dat を cont.dat とリネームすれば良い. また, 幾つかの gene.dat をつなげて cont.dat にすることも推奨される. ただし, 最大で頭から個体数(noi)までしか読まれないので, 多くの個体が出力され過ぎている場合は最終ステップの適応度の良い個体のみを cont.dat に含ませるように注意する. (例題 1 を参照)

2.5 ポテンシャルファイル

ポテンシャル指定ファイルで 1 行目が 2,3 (2 元系, 3 元系)の場合に必要な. 2 元系の場合, 単元系合わせ込みで得られた poten.dat, 3 元系の場合, 単元系, 2 元系合わせ込みで得られた poten.dat を指定する.

例えば, Si-O(シリコン・酸素) 2 元系の場合,

```
%cat tersoffSiO/step01/case001/element.dat
```

2

Si O

potenSi.dat potenO.dat

というように potenSi.dat と potenO.dat を指定している場合.

```
%cp tersoffSi/step02/case001/poten.dat tersoffSiO/step01/case001/potenSi.dat
```

```
%cp tersoffO/step02/case001/poten.dat tersoffSiO/step01/case001/potenO.dat
```

とする.

2.6 set.dat

合わせ込みに用いる完全結晶でない原子構造指定ファイルの指定を行う. 合わせ込みに用いる完全結晶については4章に詳細を述べる.

書式

N 個の原子構造について合わせ込みを行うとき

N 構造緩和開始ステップ 構造緩和ステップ間隔

0 構造名 エネルギー用重み 格子定数用重み 全原子数 欠陥数

1 構造名 エネルギー用重み 格子定数用重み 全原子数 欠陥数

...

N-1 構造名 エネルギー用重み 格子定数用重み 全原子数 欠陥数

解説・注意事項など

結晶のみで合わせ込む場合には0とだけ書いておけばよい. 合わせ込みに用いる結晶でない原子構造名・合わせ込み時の重みを列挙する. 合わせ込み中にそのときに最も良い原子間ポテンシャルを用いて, 構造緩和を行うことが可能であるため, 構造緩和開始ステップ, 構造緩和ステップ間隔も指定する. i番構造の構造名はグローバル変数 `char *name[i]`, エネルギー用重みはグローバル変数 `double weightForEnergyOfNonCrystallineStructure[i]`, 格子定数用重みはグローバル変数 `double weightForLengthOfNonCrystallineStructure[i]`, 全原子数はグローバル変数 `int natom[i]`, 欠陥数はグローバル変数 `int ndefect[i]`に格納されるので `fsingle.h`, `fbinary.h`, `fternary.h` などにおける適応度計算のために使うことができる.

サンプル

2

5 10

0 `diamont_Tint` 10.0 0.0 65 1

1 `diamond_vacancy` 10.0 0.0 63 1

2.7 posi.dat

合わせ込みに用いる完全結晶ではない原子構造, 原子間力を記述するファイル. 長さの単位は Å, 力の単位は eV/Å である.

書式

N 個の原子構造(i番目の構造における原子数が n_i)について合わせ込みを行うとき

0 構造名 原子数

1 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力

2 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力

...

n_0 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力

1 構造名 原子数

1 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力

```

2 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力
...
n1 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力
2 構造名 原子数
...
N-1 構造名 原子数
1 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力
2 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力
...
nN-1 番原子の原子番号 x座標 y座標 z座標 受けるx方向の力 受けるy方向の力 受けるz方向の力

```

解説・注意事項など

set.datに記述した構造の順番に、非完全結晶の原子種、原子座標、原子間力を記述する。構造名がset.datで入力した構造名と一致しない場合はエラーを返すようになっている。i番構造の原子間力はグローバル変数 `double *fpx[i], double *fpy[i], double *fpz[i]` に格納されるので `fsingle.h`, `fbinary.h`, `fternary.h` などにおける適応度計算のために使うことができる。set.datに記述した構造数が0の場合はこのファイルは不要である。

サンプル

0 diamond_Tint 65

```

14 2.715000000000000 2.715000000000000 2.715000000000000 0 0 0
14 0.000000000000000 0.000000000000000 0.000000000000000 0 0 0

```

～中略～

```

14 9.502500000000000 6.787500000000000 9.502500000000000 0 0 0

```

1 diamond_vacancy 63

```

14 2.715000000000000 2.715000000000000 0.000000000000000 0 0 0
14 2.715000000000000 0.000000000000000 2.715000000000000 0 0 0

```

～後略～

2.8 lattice.dat

合わせ込みに用いる完全結晶ではない構造の格子を記述するファイル。長さの単位は Å である。

書式

N個の原子構造 (i番構造の格子が $(h_{i,11}, h_{i,12}, h_{i,13}), (h_{i,21}, h_{i,22}, h_{i,23}), (h_{i,31}, h_{i,32}, h_{i,33})$ の3つのベクトルで表現できるときに)について合わせ込みを行うとき

0 構造名

```
h0,11 h0,12 h0,13
```

```
h0,21 h0,22 h0,23
```

```
h0,31 h0,32 h0,33
```

1 構造名

```
h1,11 h1,12 h1,13
```

```
h1,21 h1,22 h1,23
```

```
h1,31 h1,32 h1,33
```

...

N-1 構造名

hN-1,11 hN-1,12 hN-1,13

hN-1,21 hN-1,22 hN-1,23

hN-1,31 hN-1,32 hN-1,33

とする.

解説・注意事項など

set.dat に記述した構造の順番に, 非完全結晶構造の格子を記述する. 構造名が set.dat で入力した構造名と一致しない場合はエラーを返すようになっている. set.dat に記述した構造数が 0 の場合は不要である.

サンプル

0 diamond_Tint

10.860000000000000 0.000000000000000 0.000000000000000

0.000000000000000 10.860000000000000 0.000000000000000

0.000000000000000 0.000000000000000 10.860000000000000

1 diamond_vacancy

10.860000000000000 0.000000000000000 0.000000000000000

0.000000000000000 10.860000000000000 0.000000000000000

0.000000000000000 0.000000000000000 10.860000000000000

2.9 energy.dat

合わせ込みに用いる結晶でない構造のエネルギーを記述するファイル. エネルギーの単位は eV である.

書式

N 個の原子構造について合わせ込みを行うとき

0 構造名 エネルギー

1 構造名 エネルギー

...

N-1 構造名 エネルギー

解説・注意事項など

set.dat に記述した構造の順番に, 非完全結晶構造のエネルギーを記述する. 構造名が set.dat で入力した構造名と一致しない場合はエラーを返すようになっている. i 番構造のエネルギーはグローバル変数 `double fpNonCrystallineEnergy[i]` に格納されるので `fsingle.h`, `fbinary.h`, `fternary.h` などにおける適応度計算のために使うことができる. エネルギーの指定は絶対エネルギーでも相対エネルギーでも良い. どちらにするかは, 適応度記述関数(`fsingle.h`, `fbinary.h`, `fternary.h`)において各ユーザが計算手法を決める必要がある. set.dat に記述した構造数が 0 の場合はこのファイルは不要である.

サンプル

0 diamond_Tint 5.5

1 diamond_vacancy 3.5

3. 出力ファイル

3.1 poten.dat

最終的に出力されるポテンシャルパラメータのファイル. para.dat と同じ順番に出力される.

書式

パラメータ数が n のとき

1 番目のパラメータの名前	値
2 番目のパラメータの名前	値
...	
n 番目のパラメータの名前	値

サンプル

```
A_AA 1.83094224855087e+03
B_AA 4.71216609499780e+02
L1_AA 2.48108488777432e+00
L2_AA 1.73302763926072e+00
ETB_AA 7.87340000000000e-01
DT*ET_AA 5.00000000000000e-01
p_AAA 5.19749500000000e+00
q_AAA 3.00000000000000e+00
c1_AAA 1.09990000000000e-06
c2_AAA 4.21495630781239e+01
c3_AAA 2.62991089000000e+02
h_AAA -5.98250000000000e-01
R1_AA 2.70000000000000e+00
R2-R1_AA 3.00000000000000e-01
Re_AA 0.00000000000000e+00
Rcut 3.30000000000000e+00
```

3.2 stdout.dat

標準出力ファイル. ユーザが出力する項目を指定できる. 単元系なら `fsingle.h`, 2 元系なら `fbinary.h`, 3 元系なら `fternary.h` などにおいてグローバルなファイルハンドル `fstdout` に出力を指定する.

3.3 gene.dat

遺伝子ファイル. `cond.dat` における `nint` ステップごとに `nout` の個体数だけ個体の遺伝子の値 (=ポテンシャルパラメータ) が出力される. `cond.dat` における `iscont` が 1 の場合は `gene.dat` を `cont.dat` にリネームしたり, 一部を抜き出して組み合わせて `cont.dat` として用いる.

書式

パラメータ数が n のとき

コメント行

1 番目のパラメータの名前	値
---------------	---

```

2 番目のパラメータの名前      値
...
n 番目のパラメータの名前      値
コメント行
1 番目のパラメータの名前      値
2 番目のパラメータの名前      値
...
n 番目のパラメータの名前      値
コメント行
...

```

サンプル

```

step:0--fitnss[10727]=-8725552.914756--
      A_AA 2.76139007761836e+03
      B_AA 2.69967301939199e+02
~中略~
      Rcut 3.30000000000000e+00
step:0--fitnss[3633]=-8953471.187161--
      A_AA 5.47143656552648e+03
      B_AA 9.64607553322223e+01
~中略~
      Rcut 3.30000000000000e+00
step:0--fitnss[32009]=-8971211.373990--
      A_AA 3.62430925121724e+03
~後略~

```

3.4 condout.dat

主に計算条件などをユーザが指定した項目について、出力を行う。出力内容はグローバルなファイルハンドル `fcondout` に出力を指定した内容になる。

3.5 log.dat

GA における適応度が出力されるファイル。 `cond.dat` における `nint` ステップごとに `nout` の適応度のよいほうから個体数だけその適応度が出力される。

3.6 relax.dat

`set.dat` で結晶でない合わせ込み構造を 1 個以上指定している場合、プログラム中で構造緩和された後の原子構造が出力される。

3.7 allpoten.dat (GEAM ポテンシャル単元系の場合のみ)

従属ポテンシャルパラメータ (独立ポテンシャルパラメータの未から計算できるパラメータ) も含めた全ポテンシャルパラメータが出力される。

4. 合わせ込み結晶物性の指定, 重みの設定

setCondition.cpp に含まれるヘッダにおいて合わせ込む物性の指定, 重みの設定を行う. 例えば,

～setCondition.cpp 冒頭部ソースファイル前略～

```
#include"condZr.h"  
#include"condCu.h"  
#include"condPt.h"  
#include"condSi.h"  
#include"condO.h"  
#include"condC.h"
```

```
#include"condGaN.h"  
#include"condSiO.h"
```

～setCondition.cpp ソースファイル後略～

のように cond 元素名.h や cond 元素名 1 元素名 2.h のような形式で含まれているため, このヘッダ中のフラグを変更するのみである. ただし, ヘッダそのものを書き換えるので, 必ず再度コンパイルする必要がある. 凝集エネルギー, 格子定数, 体積弾性率, 弾性定数を合わせ込みに用いることができる.

4.1 合わせ込み結晶物性の指定

どの結晶構造のどの物性を合わせ込みに用いるかは, ヘッダにおいて定義されている関数 void setFlagsOfSingleCrystals...(…)や void setFlagsOfBinaryCrystals...(…)の中で

int isUsedForFitting[結晶名]: エネルギーと格子定数を計算に用いるか

int isCalcBulkModulus[結晶名]: 体積弾性率を計算に用いるか

int isCalcTensileElasticConstants[結晶名]: 引張りの弾性定数を計算に用いるか

int isCalcShearElasticConstants[結晶名]: せん断の弾性定数を計算に用いるか

の値を 0 か 1 かで指定する. (1: 用いる, 0: 用いない) デフォルトでは 0 になっているので, 使いたい物性の指定フラグのみ 1 とする. せん断の弾性定数は対応していない結晶も多いので注意.

例えば

```
void setFlagsOfSingleCrystalsSi(int *isUsedForFitting,int *isCalcBulkModulus,  
                                int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){  
    isUsedForFitting[DIMER] = 1;  
    isUsedForFitting[GRAPHITE] = 1;  
    isUsedForFitting[DIAMOND] = 1;  
    isUsedForFitting[SC] = 1;  
    isUsedForFitting[BCC] = 1;  
    isUsedForFitting[FCC] = 1;
```

～関数後略～

のようにしてあれば, DIMER, GRAPHITE, DIAMOND, SC, BCC, FCC 構造の格子定数とエネルギーを合わせ込みに使うことになる. 使える結晶構造については 6 章を参照していただきたい.

4.2 合わせ込みにおける重みの設定

合わせ込みの重みはヘッダにおいて定義されている関数 `void setWeightOfSingleCrystals...(...)` や `void seWeightOfBinaryCrystals...(...)` の中で

`double weightForCohesiveEnergy[結晶名]` : エネルギーの重み

`double weightForLatticeConstant[結晶名]` : 格子定数の重み

`double weightForBulkModulus[結晶名]` : 体積弾性率の重み

`double weightForTensileElasticConstants[結晶名]` : 引張りの弾性定数の重み

`double weightForShearElasticConstants[結晶名]` : せん断の弾性定数の重み

の値で設定する。値が大きいほど合わせ込みにおいて重視することになる。

例えば

```
void setWeightOfSingleCrystalsSi(引数は略){  
    weightForCohesiveEnergy[DIMER]      = 3.0;  
    weightForCohesiveEnergy[CHAIN]      = 0.0;  
    weightForCohesiveEnergy[GRAPHITE]   = 30.0;  
    weightForCohesiveEnergy[DIAMOND]    = 100.0;  
    weightForCohesiveEnergy[SC]         = 30.0;  
    weightForCohesiveEnergy[BCC]        = 10.0;  
}
```

～後略～

のようにしてあれば DIAMOND 構造の凝集エネルギーを重点的に合わせ込むことになる。

5. 合わせ込む完全結晶の物性入力方法

setRef.cpp に含まれるヘッダにおいて合わせ込む完全結晶の物性入力を行う。

～setRef.cpp 冒頭部ソースファイル前略～

```
#include"matpropN.h"  
#include"matpropNi.h"  
#include"matpropFe.h"  
#include"matpropAl.h"  
#include"matpropZr.h"  
#include"matpropCu.h"  
#include"matpropPt.h"
```

```
#include"matpropSiO.h"  
#include"matpropGaN.h"
```

～setRef.cpp ソースファイル後略～

のように単元系の場合 matprop 元素名.h, 2 元系の場合 matprop 元素名 1 元素名 2.h のような形式で含まれているため, このヘッダ中で物性値を入力する。ただし, ヘッダそのものを書き換えるので, 必ず再度コンパイルする必要がある。デフォルトでは凝集エネルギー, 格子定数, 体積弾性率, 弾性定数を入力することができる。

5.1 単元系の場合

ヘッダ中の関数 void setMaterialsPropertiesOf...(…)において

cohesiveEnergy[結晶名] : 格子の凝集エネルギー(eV)
latticeConstantA[結晶名] : a 軸の格子定数 (Å)
latticeConstantB[結晶名] : b 軸の格子定数 (Å)
latticeConstantC[結晶名] : c 軸の格子定数 (Å)
bulkModulus[結晶名] : 体積弾性率 (GPa)
C11[結晶名] : 弾性定数 C₁₁ (GPa)
C22[結晶名] : 弾性定数 C₂₂ (GPa)
C33[結晶名] : 弾性定数 C₃₃ (GPa)
C44[結晶名] : 弾性定数 C₄₄ (GPa)
C55[結晶名] : 弾性定数 C₅₅ (GPa)
C66[結晶名] : 弾性定数 C₆₆ (GPa)
C12[結晶名] : 弾性定数 C₁₁ (GPa)
C13[結晶名] : 弾性定数 C₁₃ (GPa)
C23[結晶名] : 弾性定数 C₂₃ (GPa)

の値を入力する。

例えば matpropSi.h において

```
void setMaterialsPropertiesOfSilicon(引数は略){  
    cohesiveEnergy[DIMER]    = (-4.63+3.01)*2.0;
```

```

cohesiveEnergy[CHAIN]    = (-4.63+2.260)*1.0;
cohesiveEnergy[GRAPHITE] = (-4.63+0.772)*4.0;
cohesiveEnergy[DIAMOND]  = (-4.63)*8.0;
cohesiveEnergy[SC]       = (-4.63+0.276)*1.0;
cohesiveEnergy[BCC]      = (-4.63+0.435)*2.0;
cohesiveEnergy[FCC]      = (-4.63+0.449)*4.0;

latticeConstantA[DIMER]  = 2.246;
latticeConstantA[CHAIN]  = 2.192;
latticeConstantA[GRAPHITE] = 2.241*sqrt(3.0);
latticeConstantA[DIAMOND] = 2.351/sqrt(3.0)*4.0;
latticeConstantA[SC]     = 2.513;
latticeConstantA[BCC]    = 2.650/sqrt(3.0)*2.0;
latticeConstantA[FCC]    = 2.712*sqrt(2.0);

bulkModulus[DIAMOND]    = 102.0;
C11[DIAMOND] = 167.0;
C12[DIAMOND] = 65.0;
C44[DIAMOND] = 81.0;
}

```

のようになる。

5.2 2 元系の場合

2 元系ポテンシャルの場合、絶対的な凝集エネルギーを合わせることはさほど意味はなく、生成エネルギーをあわせることが重要になる。このため、生成エネルギーを計算するための基準エネルギーの入力が必須となる。合わせ込みに用いる生成エネルギーを計算するための基準エネルギー、試行する原子間ポテンシャルの生成エネルギーを計算するための基準エネルギーの両方を入力しなくてはならない。

ヘッダ中の関数 `void setMaterialsPropertiesOf...(…)` において

```

cohesiveEnergy[結晶名] : 格子の凝集エネルギー(eV)
latticeConstantA[結晶名] : a 軸の格子定数 (Å)
latticeConstantB[結晶名] : b 軸の格子定数 (Å)
latticeConstantC[結晶名] : c 軸の格子定数 (Å)
bulkModulus[結晶名] : 体積弾性率 (GPa)
C11[結晶名] : 弾性定数 C11 (GPa)
C22[結晶名] : 弾性定数 C22 (GPa)
C33[結晶名] : 弾性定数 C33 (GPa)
C44[結晶名] : 弾性定数 C44 (GPa)
C55[結晶名] : 弾性定数 C55 (GPa)
C66[結晶名] : 弾性定数 C66 (GPa)
C12[結晶名] : 弾性定数 C11 (GPa)
C13[結晶名] : 弾性定数 C13 (GPa)

```

C23[結晶名] : 弾性定数 C_{23} (GPa)

fpBasisEnergyByAtom1[結晶名] : 合わせ込む基準エネルギー(eV/atom) (1 種類目)

fpBasisEnergyByAtom2[結晶名] : 合わせ込む基準エネルギー(eV/atom) (2 種類目)

basisEnergyByAtom1[結晶名] : 単元系ポテンシャルファイルによって計算された基準エネルギー (1 種目元素)

basisEnergyByAtom2[結晶名] : 単元系ポテンシャルファイルによって計算された基準エネルギー (2 種目元素)

の値を入力する.

ただし, 指定した単元系ポテンシャルによって計算される単元系結晶のエネルギー energyOfSymbol1[i], energyOfSymbol2[i]を使用することができる.

合わせ込む 2 元系結晶 i に 1 種類目の元素が N1[i]個, 2 種類目の元素が N2[i]個含まれる場合, 合わせ込むべき反応エネルギーであるグローバル変数 fpReactionEnergy[i]は

$$\text{fpReactionEnergy}[i] = (\text{cohesiveEnergy}[i] - N1[i] * \text{fpBasisEnergyByAtom1}[i] - N2[i] * \text{fpBasisEnergyByAtom2}[i]) / (\text{double})(N1[i] + N2[i]);$$

という式によって最終的に計算されることになる.

また fbinary.h 中の適応度関数 fbinary においては試行するポテンシャルによって計算されたエネルギー crystalEnergy[i]に対して, 反応エネルギーは

$$\text{reactionEnergy}[i] = (\text{crystalEnergy}[i] - N1[i] * \text{basisEnergyByAtom1}[i] - N2[i] * \text{basisEnergyByAtom2}[i]) / (\text{double})(N1[i] + N2[i]);$$

という式によって最終的に計算されることになる.

6. 適応度関数の定義

単元系の場合 `fsingle.h`, 2 元系の場合 `fbinary.h`, 3 元系の場合 `fternary.h` において定義される適応度関数の値が小さいほど良い原子間ポテンシャルとして評価され, GA において次世代の個体として残されるような仕組みになっている. このため, 適応度はデフォルトでは, 合わせ込みたい物性値と試行する原子間ポテンシャルによって計算された物性の差の合計として定義している. また, エネルギーの逆転禁止などのどうしても設定したいことに関しては大きなペナルティ値を設定するのも有効である.

7. 合わせ込みに使える結晶構造

プログラムにはあらかじめ使える結晶構造が入力してある。それぞれの結晶構造がマクロと結び付けてあり、マクロは `constant.h` において定義してある。

～`constant.h` ソースファイル冒頭前略～

```
#define DIMER      1
#define CHAIN     2
#define GRAPHITE  3
#define DIAMOND   4
#define SC        5
#define BCC       6
#define BCT       7
#define FCC       8
```

～`constant.h` ソースファイル後略～

といった具合に計数十あまりの結晶構造を使えるようになっている。構造の詳細は文献(熊谷知久, 東京大学博士論文 2007)を参照。

8. 省メモリモード

`structure.h` の先頭で

```
#define ENV 1
```

とあるが、これを以下のようにコメントアウトし、

```
##define ENV 1
```

コンパイルしなおすことにより

```
%make clean
```

```
%make
```

省メモリモードの合わせ込みコードを生成できる。

当然その分計算は遅くなるが、EAM ポテンシャルなど数GBのメモリを使用してしまうようなの長距離ポテンシャルについては有効に働く。

9. 例題 共有結合性結晶 Tersoff 型 Si 用単元系ポテンシャルの開発

Tersoff ポテンシャルは Si 用のポテンシャルとして有名であるが、弾性定数の再現性がやや悪いことが知られている。このため、弾性定数を改善した Si 用 Tersoff ポテンシャルの開発を例題として行う。なお本節で行われる例題はディレクトリ `tersoffSi` に含まれる。2 ステップに分けて合わせ込みを行うことにする。

8.1 Tersoff ポテンシャル関数形

本ソフトウェアで用いられる Tersoff ポテンシャル関数形は以下のようなものである。一見、オリジナルのものとは角度依存項 $g(\theta)$ が異なるように見えるが、簡単に下記の角度依存項に変形することができる。

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi_{ij}$$

$$\phi_{ij} = f_c(r_{ij}) [A \exp(-\lambda_1 r_{ij}) - b_{ij} B \exp(-\lambda_2 r_{ij})]$$

$$b_{ij} = (1 + \zeta_{ij}^\eta)^{-\delta}$$

$$\zeta_{ij} = \sum_{k \neq i, j} f_c(r_{ik}) g(r_{ik}) \exp[p \{ (r_{ij} - R_e) - (r_{ik} - R_e) \}^q]$$

$$g(\theta) = c_1 + \frac{c_2 (h - \cos \theta)^2}{c_3^2 + (h - \cos \theta)^2}$$

$$f_c(r) = \begin{cases} 1 & r < R_1 \\ \frac{1}{2} + \frac{1}{2} \cos\left(\pi \frac{r - R_1}{R_2 - R_1}\right) & R_1 \leq r \leq R_2 \\ 0 & r > R_2 \end{cases}$$

変形は

$$g(\theta) = a \left\{ 1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (h - \cos^2 \theta)} \right\}$$

$$= a + \frac{ac^2}{d^2} \left\{ 1 - \frac{d^2}{d^2 + (h - \cos^2 \theta)} \right\}$$

$$= a + \frac{ac^2}{d^2} \frac{(h - \cos^2 \theta)}{d^2 + (h - \cos^2 \theta)}$$

$$c_1 + \frac{c_2 (h - \cos \theta)^2}{c_3^2 + (h - \cos \theta)^2}$$

のようになるから、つまりオリジナルの a, c, d に対して

$$c_1 = a$$

$$c_2 = \frac{ac}{d^2}$$

$$c_3 = d^2$$

という関係になる。

オリジナルの角度依存項の形のままで c, d が大きくぶれてしまい合わせ込みが難しい。変形後の形式で c_3 を固定して合わせ込みするのが望ましい。

8.2 ヘッダの確認

8.2.1 合わせ込み条件ヘッダ condSi.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。一部抜粋になっているが、ダイマーと基本的な単元系結晶(グラファイト構造, ダイヤモンド構造, 正方格子, 体心立方格子, 面心立方格子)のエネルギーと格子定数, 安定構造のダイヤモンド構造の体積弾性率, 弾性定数を合わせ込みに用いる設定になっている。

```
void setFlagsOfSingleCrystalsSi(int *isUsedForFitting,int *isCalcBulkModulus,
                                int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
    isUsedForFitting[DIMER] = 1;
    isUsedForFitting[GRAPHITE] = 1;
    isUsedForFitting[DIAMOND] = 1;
    isUsedForFitting[SC] = 1;
    isUsedForFitting[BCC] = 1;
    isUsedForFitting[FCC] = 1;

    isCalcBulkModulus[DIAMOND] = 1;
    isCalcTensileElasticConstants[DIAMOND] = 1;
    isCalcShearElasticConstants[DIAMOND] = 1;
}
```

重みを設定する関数

```
void setWeightOfSingleCrystalsSi(引数略){
    weightForCohesiveEnergy[DIMER] = 3.0;
    weightForCohesiveEnergy[GRAPHITE] = 30.0;
    weightForCohesiveEnergy[DIAMOND] = 100.0;
    weightForCohesiveEnergy[SC] = 30.0;
    weightForCohesiveEnergy[BCC] = 10.0;
    weightForCohesiveEnergy[FCC] = 1.0;

    weightForLatticeConstant[DIMER] = 1.0;
    weightForLatticeConstant[GRAPHITE] = 10.0;
    weightForLatticeConstant[DIAMOND] = 10.0;
    weightForLatticeConstant[SC] = 10.0;
    weightForLatticeConstant[BCC] = 1.0;
    weightForLatticeConstant[FCC] = 1.0;

    weightForBulkModulus[DIAMOND] = 40.0;
    weightForTensileElasticConstants[DIAMOND] = 30.0;
    weightForShearElasticConstants[DIAMOND] = 20.0;
}
```

8.2.2 物性の入力されているヘッダ matpropSi.h の確認

合わせ込みに用いる物性が設定されているか確認する.

～matpropSi.h 冒頭部～

```
void setMaterialsPropertiesOfSilicon(double* latticeConstantA,
                                   double* latticeConstantB,
                                   double* latticeConstantC,
                                   double* internalParameterX1,
                                   double* internalParameterX2,
                                   double* internalParameterX3,
                                   double* cohesiveEnergy,
                                   double* bulkModulus,
                                   double* C11,double* C22,double* C33,
                                   double* C12,double* C13,double* C23,
                                   double* C44,double* C55,double* C66){
    cohesiveEnergy[DIMER]      = (-4.63+3.01)*2.0;
    cohesiveEnergy[CHAIN]     = (-4.63+2.260)*1.0;
    cohesiveEnergy[GRAPHITE]  = (-4.63+0.772)*4.0;
    cohesiveEnergy[DIAMOND]   = (-4.63)*8.0;
    cohesiveEnergy[SC]        = (-4.63+0.276)*1.0;
    cohesiveEnergy[BCC]       = (-4.63+0.435)*2.0;
    cohesiveEnergy[FCC]       = (-4.63+0.449)*4.0;

    latticeConstantA[DIMER]    = 2.246;
    latticeConstantA[CHAIN]    = 2.192;
    latticeConstantA[GRAPHITE] = 2.241*sqrt(3.0);
    latticeConstantA[DIAMOND]  = 2.351/sqrt(3.0)*4.0;
    latticeConstantA[SC]       = 2.513;
    latticeConstantA[BCC]      = 2.650/sqrt(3.0)*2.0;
    latticeConstantA[FCC]      = 2.712*sqrt(2.0);

    bulkModulus[DIAMOND]      = 102.0;
    C11[DIAMOND]              = 167.0;
    C12[DIAMOND]              = 65.0;
    C44[DIAMOND]              = 81.0;
}
```

凝集エネルギーは格子エネルギー(eV)の形で設定されており、原子一個当たりのエネルギーで設定されていないことに注意してほしい。同様に長さも格子定数(Å)で定義されており、結合長では設定されていない。体積弾性率、弾性定数の単位は GPa で設定してある。

8.3 第 1 ステップの入力ファイル

ディレクトリ tersoffSi/step01/case001/~case008/中の入力ファイルを設定する

8.3.1 element.dat

単元系 Si 用の原子間ポテンシャルなので case00*の element.dat を以下のように設定する

~tersoffSi/step01/case001/element.dat~

1

Si

8.3.2 cond.dat

case001 の cond.dat を以下のように設定する

~tersoffSi/step01/case001/set.dat~

id: 1 : Tersoff ポテンシャル
iscont: 0 : 新規スタート
nstep: 31 : ステップ数 31
nint: 5 : 5 ステップごとに出力
noi: 100000 : 100000 個体
nout: 1000 : トップ 999 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 1 : スケーリングを行う

第 1 ステップは多くの個体, 短いステップ数の方が望ましい. この個体数(100000)の場合, それなりに早い計算機でも数時間かかる. テスト計算を行いたい場合は noi=10000 個体程度に減らしても問題ない. case002~case008 の乱数の seed は 2~8 とした.

8.3.3 para.dat

既存のポテンシャルパラメータを参考にしておおよそのパラメータ範囲を決める. 以下に例を示す.

0	FREE	A_AA	0.0	10000.0	: A
1	FREE	B_AA	0.0	10000.0	: B
2	FREE	L1_AA	0.0	5.0	: λ_1
3	FREE	L2_AA	0.0	5.0	: λ_2
4	FREE	ETB_AA	0.0	30.0	: η
5	FIX	DT*ET_AA	0.5	0.5	: $\eta \times \delta$
6	FREE	p_AAA	0.00	0.00	: p
7	FIX	q_AAA	1.00	1.00	: q
8	FREE	c1_AAA	0.0	1.0	: c_1
9	FREE	c2_AAA	0.0	100.0	: c_2
10	FIX	c3_AAA	100.0	100.0	: c_3
11	FREE	h_AAA	-1.0	0.0	: h
12	FIX	R1_AA	2.7	2.7	: R_1
13	FIX	R2-R1_AA	0.3	0.3	: $R_2 - R_1$
14	FIX	Re_AA	0.0	0.0	: Re

小分けにして色々なパラメータで合わせ込みを行う。筆者の手元にあるPCはQuadCore×2のマシンであるため、8ジョブを投げることができるので、8ケースを実行することになる。振るパラメータはhとqにする。以下の8ケースに分ける。

表 パラメータ範囲の振り分け

	h 下限	h 上限	q
case001	-1	-0.75	3
case002	-0.75	-0.5	3
case003	-0.5	-0.25	3
case004	-0.25	0	3
case005	-1	-0.75	1
case006	-0.75	-0.5	1
case007	-0.5	-0.25	1
case008	-0.25	0	1

実際のサンプルに含まれる para.dat を grep すると

```
[kumagai@sst105 ver0.37]%grep h_AAA tersoffSi/step01/case00*/para.dat
```

```
tersoffSi/step01/case001/para.dat: 11 FREE h_AAA -1.0000000000000000 -0.7500000000000000
tersoffSi/step01/case002/para.dat: 11 FREE h_AAA -0.7500000000000000 -0.5000000000000000
tersoffSi/step01/case003/para.dat: 11 FREE h_AAA -0.5000000000000000 -0.2500000000000000
tersoffSi/step01/case004/para.dat: 11 FREE h_AAA -0.2500000000000000 0.0000000000000000
tersoffSi/step01/case005/para.dat: 11 FREE h_AAA -1.0000000000000000 -0.7500000000000000
tersoffSi/step01/case006/para.dat: 11 FREE h_AAA -0.7500000000000000 -0.5000000000000000
tersoffSi/step01/case007/para.dat: 11 FREE h_AAA -0.5000000000000000 -0.2500000000000000
tersoffSi/step01/case008/para.dat: 11 FREE h_AAA -0.2500000000000000 0.0000000000000000
```

```
[kumagai@sst105 ver0.37]%grep q_AAA tersoffSi/step01/case00*/para.dat
```

```
tersoffSi/step01/case001/para.dat: 7 FIX q_AAA 3.0000000000000000 3.0000000000000000
tersoffSi/step01/case002/para.dat: 7 FIX q_AAA 3.0000000000000000 3.0000000000000000
tersoffSi/step01/case003/para.dat: 7 FIX q_AAA 3.0000000000000000 3.0000000000000000
tersoffSi/step01/case004/para.dat: 7 FIX q_AAA 3.0000000000000000 3.0000000000000000
tersoffSi/step01/case005/para.dat: 7 FIX q_AAA 1.0000000000000000 1.0000000000000000
tersoffSi/step01/case006/para.dat: 7 FIX q_AAA 1.0000000000000000 1.0000000000000000
tersoffSi/step01/case007/para.dat: 7 FIX q_AAA 1.0000000000000000 1.0000000000000000
tersoffSi/step01/case008/para.dat: 7 FIX q_AAA 1.0000000000000000 1.0000000000000000
```

上記の設定のようになっていることが確認できる。

8.3.4 set.dat

空孔の入ったダイヤモンド構造, 挿入型欠陥の入ったダイヤモンド構造の欠陥生成エネルギーも合わせ込みに用いる. 最初のステップではポテンシャルがばらつきやすいので構造緩和は行わず, エネルギーもウォッチするだけにするので, 重みはゼロ, 緩和開始ステップは GA のステップ数(30)より多い 50 に設定した.

~tersoffSi/step01/case00*/set.dat~

2

50 10

0 diamond_Tint 0.0 0.0 65 1

1 diamond_vacancy 0.0 0.0 63 1

8.3.5 energy.dat

文献値から適当な値を設定する. 絶対エネルギーではなく相的な欠陥生成エネルギーの合わせ込みにしたいので, 欠陥生成エネルギーで定義した.

~tersoffSi/step01/case00*/enrgy.dat~

0 diamond_Tint 5.5

1 diamond_vacancy 3.5

8.3.6 lattice.dat, posi.dat

第一原理計算から得られた緩和後の原子構造を用いるのが理想的なだが, 格子の緩和による欠陥生成エネルギーの変化は小さいので, 通常の完全結晶構造に原子を挿入した構造 (挿入型欠陥), 原子を抜き取った構造 (空孔型欠陥) を初期原子構造とする. lattice.dat は以下のような格子定数 5.43 Å の 2×2×2 のダイヤモンド構造とした.

~tersoffSi/step01/case00*/set.dat~

0 diamond_Tint

10.860000000000000 0.000000000000000 0.000000000000000

0.000000000000000 10.860000000000000 0.000000000000000

0.000000000000000 0.000000000000000 10.860000000000000

1 diamond_vacancy

10.860000000000000 0.000000000000000 0.000000000000000

0.000000000000000 10.860000000000000 0.000000000000000

0.000000000000000 0.000000000000000 10.860000000000000

posi.dat も一部を示す. 挿入型欠陥構造では原子を 1 個加え, 空孔型欠陥構造では原子を 1 個抜いている.

~tersoffSi/step01/case00*/posi.dat~

0 diamond_Tint 65

14 2.715000000000000 2.715000000000000 2.715000000000000 0 0 0

14 0.000000000000000 0.000000000000000 0.000000000000000 0 0 0

~中略~

14 9.502500000000000 6.787500000000000 9.502500000000000 0 0 0

1 diamond_vacancy 63

14 2.715000000000000 2.715000000000000 0.000000000000000 0 0 0

14 2.715000000000000 0.000000000000000 2.715000000000000 0 0 0

～後略～

8.4 適応度関数の設定

ダイヤモンド構造が再安定構造になることを最優先にしたいので、そうでない場合には大きなペナルティをかけることにする。合わせ込みの原子番号が 14(シリコン)の場合にはもし、グラファイト、正方格子、体心立方格子、面心立方格子のエネルギーがダイヤモンド構造のエネルギーを下回ったら、`fsingle.h` 中の関数 `fsingle` がペナルティとして、非常に大きな値を返すという設定を行う。また、欠陥生成エネルギーの計算方法についても確認を行う。欠陥を含む構造のエネルギー、原子数、欠陥数をそれぞれ E, n, N とすると、ダイヤモンド構造のエネルギーを E^{ref} とした場合、欠陥生成エネルギーが $\Delta E = (E - nE^{\text{ref}})/N = E - nE^{\text{ref}}$ となっていることが確認できる(`set.dat` で $N=1$ に設定してあることに注意)。

～`fsingle.h` 前略～

```
if(fatomn1==14){
    if(crystalEnergy[GRAPHITE]<crystalEnergy[DIAMOND]){
        diff = (crystalEnergy[GRAPHITE]-crystalEnergy[DIAMOND])*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[SC]<crystalEnergy[DIAMOND]){
        diff = (crystalEnergy[DIAMOND]-crystalEnergy[SC])*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[BCC]<crystalEnergy[DIAMOND]){
        diff = (crystalEnergy[DIAMOND]-crystalEnergy[BCC])*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[FCC]<crystalEnergy[DIAMOND]){
        diff = (crystalEnergy[DIAMOND]-crystalEnergy[FCC])*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[FCC]<crystalEnergy[BCC]){
        diff = (crystalEnergy[BCC]-crystalEnergy[FCC])*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
}
```

～`fsingle.h` 中略～

```
for(int iset=0;iset<nset;iset++){
    if(ndefect[iset]!=0){
        defectFormationEnergy[iset]=
            (noncrystallineEnergy[iset]-(crystalEnergy[stable]*(double)(natom[iset])))/((double)ndefect[iset]);
    }else{
        defectFormationEnergy[iset] =
            (noncrystallineEnergy[iset]-(crystalEnergy[stable]*(double)(natom[iset])));
    }
}
```

～`fsingle.h` 後略～

8.5 プログラムのチェック

ディレクトリ `tersoffSi/check` を用いて動作確認を行う。このディレクトリの `para.dat` の上限値, 下限値は `Tersoff` ポテンシャルパラメータの値となっているため, そのときの物性値を確認することができる。前節まで変更を加えたので, 以下に `make` をしてから実行する。

```
%make
```

```
%/GA tersoffSi/check
```

下に行った結果の一部を示す。(`fsingle.h` でデフォルトで指定しているように) カッコ内が合わせ込む値 (実験値・第一原理計算), カッコ外が `Tersoff` ポテンシャルパラメータを用いて計算した結果である。エネルギーや格子定数の再現性は比較的良いが, 弾性定数の再現性がやや悪いことが確認できる。

～tersoffSi/check/stdout.dat 前略～

```
CRYSTAL:      DIMER  ENE=-1.3330 (-1.6200)   LT=2.2952 (2.2460)
CRYSTAL:      GRAPHITE  ENE=-3.9114 (-3.8580)   LT=3.9908 (3.8815)
CRYSTAL:      DIAMOND  ENE=-4.6296 (-4.6300)   LT=5.4320 (5.4294)
CRYSTAL:      SC      ENE=-4.3112 (-4.3540)   LT=2.5444 (2.5130)
CRYSTAL:      BCC     ENE=-4.1971 (-4.1950)   LT=3.0844 (3.0600)
CRYSTAL:      FCC     ENE=-3.8680 (-4.1810)   LT=3.8975 (3.8353)
BM[ DIAMOND]=97.767(102.000)
C11=142.536(167.000) C12=75.379(65.000)
C44=69.022(81.000)
NONCRYST:diamond_Tint  ENE=3.5851 (5.5000)   MAX_F=0.0051 ( 9[14])
NONCRYST:diamond_vacancy  ENE=3.7235 (3.5000)   MAX_F=0.0237 (29[14])
～後略～
```

8.6 プログラムの実行(1回目)

8 ケースについて, プログラムを実行する。

```
%/GA tersoffSi/step01/case001
%/GA tersoffSi/step01/case002
%/GA tersoffSi/step01/case003
%/GA tersoffSi/step01/case004
%/GA tersoffSi/step01/case005
%/GA tersoffSi/step01/case006
%/GA tersoffSi/step01/case007
%/GA tersoffSi/step01/case008
```

筆者の手元にある計算機では 2 時間で計算が終わった。

例えば `case003` の場合, 弾性定数が改善していることがわかる。

～tersoffSi/step01/case003/stdout.dat 終端部～

```
step=[31/31]
```

```
CRYSTAL:      DIMER  ENE=-1.2707 (-1.6200)   LT=2.3184 (2.2460)
CRYSTAL:      GRAPHITE  ENE=-3.6437 (-3.8580)   LT=4.0430 (3.8815)
```

```
CRYSTAL:    DIAMOND  ENE=-4.6300 (-4.6300)  LT=5.4294 (5.4294)
CRYSTAL:          SC  ENE=-4.2709 (-4.3540)  LT=2.5207 (2.5130)
CRYSTAL:          BCC ENE=-4.2449 (-4.1950)  LT=3.0290 (3.0600)
CRYSTAL:          FCC ENE=-4.2288 (-4.1810)  LT=3.8405 (3.8353)
BM[    DIAMOND]=102.905(102.000)
C11=165.016(167.000) C12=71.844(65.000)
C44=78.061(81.000)
NONCRYST:diamond_Tint  ENE=6.5242 (5.5000)  MAX_F=2.4569 (19[14])
NONCRYST:diamond_vacancy  ENE=4.2721 (3.5000)  MAX_F=0.5635 (29[14])
~tersoffSi/step01/case003/stdout.dat 後略~
```

8.7 第2ステップの入力ファイル

第2ステップではパラメータの収束を目指す.

8.7.1 cond.dat

cond.dat を以下のように設定する

```
id:      1      : Tersoff ポテンシャル
iscont:  1      : 新規スタート
nstep:   31     : ステップ数 31
nint:    10     : 10 ステップごとに出力
noi:     10000  : 10000 個体
nout:    1000  : トップ 999 個体の情報を出力
nmut:    0      : すべての FREE パラメータを突然変異
mutrate: 0.03   : 突然変異幅は 3%
seed:    1      : 乱数の seed は 1
scale:   1      : スケーリングを行う
```

第2ステップは変動幅を小さくした方が望ましい. 余り個体を増やす必要はない.

8.7.2 cont.dat

第2ステップの個体数(10000)よりも第一ステップで gene.dat に出力した個体数が多い場合には cont.dat のために個体を抜き出さなくてはならないが, 今回は第1ステップでトータルで 999 個体(nout-1)×7 回(nint/nstep+1=7 だから)=6993 個体しか出力されていないので, 単純に gene.dat を cont.dat にリネームすればよい.

```
%cp tersoffSi/step01/case001/gene.dat tersoffSi/step02/case001/cont.dat
%cp tersoffSi/step01/case002/gene.dat tersoffSi/step02/case002/cont.dat
%cp tersoffSi/step01/case003/gene.dat tersoffSi/step02/case003/cont.dat
%cp tersoffSi/step01/case004/gene.dat tersoffSi/step02/case004/cont.dat
%cp tersoffSi/step01/case005/gene.dat tersoffSi/step02/case005/cont.dat
%cp tersoffSi/step01/case006/gene.dat tersoffSi/step02/case006/cont.dat
%cp tersoffSi/step01/case007/gene.dat tersoffSi/step02/case007/cont.dat
%cp tersoffSi/step01/case008/gene.dat tersoffSi/step02/case008/cont.dat
```

8.7.3 set.dat

第1ステップではウォッチするだけであった欠陥生成エネルギーも軽く合わせ込みに用いる。構造緩和も行うため、5ステップ目で緩和し、以降10ステップごとに再緩和する。重みも設定する。

2

5 10 : 5ステップ目で緩和し、以降10ステップごとに再緩和

0 diamond_Tint 10.0 0.0 65 1 : 重み 10.0

1 diamond_vacancy 10.0 0.0 63 1 : 重み 10.0

8.8 プログラムの実行(2回目)

8 ケースについて、プログラムを実行する。

./GA tersoffSi/step02/case001

./GA tersoffSi/step02/case002

./GA tersoffSi/step02/case003

./GA tersoffSi/step02/case004

./GA tersoffSi/step02/case005

./GA tersoffSi/step02/case006

./GA tersoffSi/step02/case007

./GA tersoffSi/step02/case008

今度は途中で構造緩和されていることがわかる。標準出力でGAの5ステップ目が終わった後、構造緩和されると、最大の原子間力MAX_Fがほぼゼロになり、欠陥生成エネルギーも大幅に低下したことがわかる。

～前略～

NONCRYST:diamond_Tint ENE=5.6148 (5.5000) MAX_F=2.2247 (36[14])

NONCRYST:diamond_vacancy ENE=3.6638 (3.5000) MAX_F=1.0305 (52[14])

～中略～

end step=[5/31]

start set parameter for relax

Non-crystallines 0/2 (diamond_Tint) is relaxed at 1.000000 K step=0/1000

Non-crystallines 1/2 (diamond_vacancy) is relaxed at 1.000000 K step=0/1000

Non-crystallines 0/2 (diamond_Tint) is relaxed at 0.100000 K step=0/1000

Non-crystallines 1/2 (diamond_vacancy) is relaxed at 0.100000 K step=0/1000

Non-crystallines 0/2 (diamond_Tint) is relaxed at 0.000000 K step=0/1000

Non-crystallines 1/2 (diamond_vacancy) is relaxed at 0.000000 K step=0/1000

step=[6/31]

～中略～

NONCRYST:diamond_Tint ENE=3.1933 (5.5000) MAX_F=0.0058 (57[14])

NONCRYST:diamond_vacancy ENE=3.1562 (3.5000) MAX_F=0.0125 (29[14])

～後略～

例えばcase003の場合、挿入型欠陥エネルギーが1eV程度小さいだけで、エネルギー、格子定数、弾性定数を再現できていることがわかる。

～tersoffSi/step02/case003/stdout.dat 終端部～

step=[31/31]

CRYSTAL: DIMER ENE=-1.2539 (-1.6200) LT=2.3218 (2.2460)

CRYSTAL: GRAPHITE ENE=-3.7485 (-3.8580) LT=4.0236 (3.8815)

CRYSTAL: DIAMOND ENE=-4.6300 (-4.6300) LT=5.4294 (5.4294)

CRYSTAL: SC ENE=-4.2466 (-4.3540) LT=2.5308 (2.5130)

CRYSTAL: BCC ENE=-4.2078 (-4.1950) LT=3.0476 (3.0600)

CRYSTAL: FCC ENE=-4.1215 (-4.1810) LT=3.8534 (3.8353)

BM[DIAMOND]=101.960(102.000)

C11=165.360(167.000) C12=70.256(65.000)

C44=80.228(81.000)

NONCRYST:diamond_Tint ENE=4.3133 (5.5000) MAX_F=0.5552 (19[14])

NONCRYST:diamond_vacancy ENE=4.0977 (3.5000) MAX_F=0.0264 (29[14])

～tersoffSi/step02/case003/stdout.dat 後略～

得られたポテンシャルパラメータは poten.dat に出力されている.

～tersoffSi/step02/case003/poten.dat～

A_AA 2.14175451315679e+03

B_AA 2.78874605501992e+02

L1_AA 2.73591405501372e+00

L2_AA 1.63653587935928e+00

ETB_AA 7.94542999006257e-01

DT*ET_AA 5.00000000000000e-01

p_AAA 1.53047162126561e+00

q_AAA 3.00000000000000e+00

c1_AAA 2.29825436072382e-04

c2_AAA 2.85082168599121e+01

c3_AAA 1.00000000000000e+02

h_AAA -5.00000000000000e-01

R1_AA 2.70000000000000e+00

R2-R1_AA 3.00000000000000e-01

Re_AA 0.00000000000000e+00

Rcut 3.30000000000000e+00

9. 例題 Finnis-Sinclair 型 Fe 用単元系ポテンシャルパラメータの再開発

Finnis-Sinclair (FS) 型ポテンシャル関数は体心立方格子を安定構造にもつ金属用ポテンシャル関数として広く使われているポテンシャル関数である。練習として Fe 用ポテンシャルパラメータを再開発する。なお、本章で行われる例題はディレクトリ FSFe に含まれる。トータル 2 ステップで合わせ込みを行う。

9.1 FS ポテンシャル関数形

本ソフトウェアで用いられる FS ポテンシャル関数形は以下のようである。一般的には $N=1/2$ である。

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) - \sum_i A \rho_i^N$$

$$\phi(r) = \begin{cases} (r-c)^2 (c_0 + c_1 r + c_2 r^2) & r \leq c \\ 0 & c < r \end{cases}$$

$$\rho_i = \sum_{j \neq i} f(r_{ij})$$

$$f(r) = \begin{cases} (r-d)^2 + \beta(r-d)^3 / d & r \leq d \\ 0 & d < r \end{cases}$$

9.2 ヘッダの確認

9.2.1 合わせ込み条件ヘッダ condFe.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。一部抜粋になっているが、基本的なバルク単元系結晶(ダイヤモンド構造, 正方格子, 体心立方格子, 面心立方格子, 六方最密格子)のエネルギーと格子定数, 安定構造の体心立方格子と準安定構造の面心立方格子の体積弾性率, 弾性定数を合わせ込みに用いる設定になっている。

```
void setFlagsOfSingleCrystalsFe(int *isUsedForFitting,int *isCalcBulkModulus,
                                int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
    isUsedForFitting[DIAMOND] = 1;
    isUsedForFitting[SC] = 1;
    isUsedForFitting[BCC] = 1;
    isUsedForFitting[FCC] = 1;
    isUsedForFitting[IDEALHCP] = 1;
    isUsedForFitting[HCP] = 1;

    isCalcBulkModulus[BCC] = 1;
    isCalcTensileElasticConstants[BCC] = 1;
    isCalcShearElasticConstants[BCC] = 1;

    isCalcBulkModulus[FCC] = 1;
    isCalcTensileElasticConstants[FCC] = 1;
    isCalcShearElasticConstants[FCC] = 1;
}
```

重みを設定する関数

```
void setWeightOfSingleCrystalsFe(引数略){
    weightForCohesiveEnergy[DIAMOND] = 10.0;
    weightForCohesiveEnergy[SC]      = 20.0;
    weightForCohesiveEnergy[IDEALHCP] = 0.0;
    weightForCohesiveEnergy[HCP]     = 10.0;
    weightForCohesiveEnergy[FCC]     = 30.0;

    weightForLatticeConstant[DIAMOND] = 0.0;
    weightForLatticeConstant[SC]      = 0.0;
    weightForLatticeConstant[BCC]     = 50.0;
    weightForLatticeConstant[BCT]     = 0.0;
    weightForLatticeConstant[IDEALHCP] = 0.0;
    weightForLatticeConstant[HCP]     = 0.0;
    weightForLatticeConstant[FCC]     = 10.0;

    weightForBulkModulus[BCC] = 50.0;
    weightForBulkModulus[FCC] = 40.0;
    weightForTensileElasticConstants[BCC] = 25.0;
    weightForTensileElasticConstants[FCC] = 25.0;
    weightForShearElasticConstants[BCC] = 25.0;
    weightForShearElasticConstants[FCC] = 25.0;
}
```

9.2.2 物性の入力されているヘッダ matpropFe.h の確認

合わせ込みに用いる物性が設定されているか確認する。物性は Y. Mishin et al., *Acta Materialia* 53 (2005) 4029–4041, Michael Müller et al., *J. Phys.: Condens. Matter* 19 (2007) 326220 and references therein. から採った。

～matpropFe.h 冒頭部～

```
void setMaterialsPropertiesOfIron(double* latticeConstantA,
                                double* latticeConstantB,
                                double* latticeConstantC,
                                double* internalParameterX1,
                                double* internalParameterX2,
                                double* internalParameterX3,
                                double* cohesiveEnergy,
                                double* bulkModulus,
                                double* C11,double* C22,double* C33,
                                double* C12,double* C13,double* C23,
                                double* C44,double* C55,double* C66){
    //Material Properties are from J.Phys. 19 (2007) 326220 and references therein
```

//Material Properties are from Acta.Mat. 53 (2005) 4029 and references therein

cohesiveEnergy[DIMER] = -1.04*2.0;
cohesiveEnergy[CHAIN] = 0.0;
cohesiveEnergy[GRAPHITE] = 0.0 ;
cohesiveEnergy[DIAMOND] = (-4.28+1.17)*8.0;
cohesiveEnergy[SC] = (-4.28+0.75)*1.0;
cohesiveEnergy[BCC] = (-4.28+0.00)*2.0;
cohesiveEnergy[FCC] = (-4.28+0.07)*4.0;
cohesiveEnergy[IDEALHCP] = (-4.28+0.06)*4.0;
cohesiveEnergy[HCP] = (-4.28+0.06)*4.0;

latticeConstantA[DIMER] = 2.02;
latticeConstantA[CHAIN] = 0.0;
latticeConstantA[GRAPHITE] = 0.0;
latticeConstantA[DIAMOND] = 4.904;
latticeConstantA[SC] = 2.390;
latticeConstantA[BCC] = 2.860;
latticeConstantA[FCC] = 3.516;
latticeConstantA[IDEALHCP] = 2.484;
latticeConstantA[HCP] = 2.484;
latticeConstantC[HCP] = latticeConstantA[HCP]*1.58;

bulkModulus[BCC] = 172.0;

C11[BCC] = 242.0;
C12[BCC] = 146.5;
C44[BCC] = 112.0;

bulkModulus[FCC] = 133.0;

C11[FCC] = 154.0;
C12[FCC] = 122.5;
C44[FCC] = 77.0;

}

9.3 第1ステップの入力ファイル

ディレクトリ FSFe/step01/case001/~case008/中の入力ファイルを設定する.

9.3.1 element.dat

単元系 Fe 用の原子間ポテンシャルなので case00*の element.dat を以下のように設定する

~FSFe/step01/case00*/element.dat~

1

Fe

9.3.2 cond.dat

パラメータが余り多くないため, パラメータ範囲は振らず, 乱数の seed を振るにとどめる. case001 の cond.dat を以下のように設定する

~FSFe/step01/case001/cond.dat~

id: 4 : FS ポテンシャル
iscont: 0 : 新規スタート
nstep: 31 : ステップ数 31
nint: 5 : 5 ステップごとに出力
noi: 10000 : 10000 個体
nout: 1001 : トップ 1000 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 1 : スケーリングを行う

case002~case008 の乱数の seed は 2~8 とした.

9.3.3 para.dat

既存のポテンシャルパラメータを参考にしておおよそのパラメータ範囲を決める.

~FSFe/step01/case00*/para.dat~

0	FREE	d	0.1	5.0	: d
1	FREE	A	0.1	5.0	: A
2	FREE	beta	0.1	5.0	: β
3	FREE	c	0.00000001	5.0	: c
4	FREE	c0	0.00000001	5.0	: c0
5	FREE	c1	-1.00000000	-0.0	: c1
6	FREE	c2	-1.00000000	-0.0	: c2
7	FIX	N	0.5	0.5	: N=1/2
8	FIX	Rcut	5.0	5.0	: カットオフ距離 (c,d より長めに設定)

9.3.4 set.dat

空孔の入った体心立方格子の欠陥生成エネルギーも合わせ込みに用いる. 最初のステップではポテンシャルがばらつきやすいので構造緩和は行わず, エネルギーもウォッチするだけにするので, 重みはゼロ, 緩和開始ステップは

GA のステップ数(31)より多い 100 に設定した (つまり, 緩和は始まらない) .

~FSFe/step01/case00*/set.dat~

1

100 10

0 BCC_vacancy 0.0 0.0 127 1

カットオフが比較的長い (5 Å) ポテンシャルであることから体心立方格子の基本格子 (2 原子) を 4×4×4 積み重ね, 1 原子抜いた (=空孔) 原子構造であるため, 原子数が 127 個となっている.

9.3.5 energy.dat

文献値 (Mishin et al., Acta Materialia 53 (2005) 4029–4041 and references therein) から適当な値を設定する. 絶対エネルギーではなく相対的な欠陥生成エネルギーを合わせ込みたいので, 欠陥生成エネルギーで定義した.

~FSFe/step01/case00*/energy.dat~

0 BCC_vacancy 2.0

9.3.6 lattice.dat, posi.dat

第一原理計算から得られた緩和後の原子構造を用いるのが理想的なだが, 格子の緩和による欠陥生成エネルギーの変化は小さいので, 体心立方格子の完全結晶構造から原子を抜いた構造 (空孔型欠陥) を初期原子構造とする. lattice.dat は以下のような格子定数 2.86 Å の 4×4×4 の体心立方格子とした.

~FSFe/step01/case00*/lattice.dat~

0 BCC_vacancy

11.4400000000	0.0000000000	0.0000000000
0.0000000000	11.4400000000	0.0000000000
0.0000000000	0.0000000000	11.4400000000

posi.dat も一部を示す. 空孔型欠陥構造では体心立方格子から原子を 1 個抜いている.

~FSFe/step01/case00*/posi.dat~

0 BCC_vacancy 127

26	1.4300000000	1.4300000000	1.4300000000	0	0	0
26	0.0000000000	0.0000000000	2.8600000000	0	0	0
26	1.4300000000	1.4300000000	4.2900000000	0	0	0

~後略~

9.4 適応度関数の設定

体心立方格子が再安定構造になることを最優先にしたいので、そうでない場合には大きなペナルティをかけることにする。合わせ込みの原子番号が26(鉄)の場合に、もし、正方格子、面心立方格子、六方最密構造のエネルギーが体心立方格子のエネルギーを下回ったら、`fsingle.h`中の関数`fsingle`がペナルティとして、非常に大きな値を返すという設定を行う。さらに、面心立方格子と体心立方格子の間のエネルギー差はある程度(実験値は 0.07eV/atom)確保したいので、体心立方格子の原子あたりのエネルギー $+0.07\text{eV/atom}$ よりも面心立方格子のエネルギーが低くなった場合にペナルティを与えることにする。

～`fsingle.h` 前略～

```
}else if(fatomn1==26){
    if(crystalEnergy[FCC]<crystalEnergy[BCC]+0.07){
        diff = (fabs(crystalEnergy[BCC]+0.07-crystalEnergy[FCC]))*10000.0;
        dv_penalty += diff*diff;
    }
    if(crystalEnergy[HCP]<crystalEnergy[BCC]){
        diff = (fabs(crystalEnergy[HCP]-crystalEnergy[BCC]))*1000.0;
        dv_penalty += diff*diff;
    }
    if(crystalEnergy[SC]<crystalEnergy[BCC]){
        diff = (fabs(crystalEnergy[SC]-crystalEnergy[BCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[DIAMOND]<crystalEnergy[BCC]){
        diff = (fabs(crystalEnergy[DIAMOND]-crystalEnergy[BCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[SC]<crystalEnergy[FCC]){
        diff = (fabs(crystalEnergy[SC]-crystalEnergy[FCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[DIAMOND]<crystalEnergy[FCC]){
        diff = (fabs(crystalEnergy[DIAMOND]-crystalEnergy[FCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[SC]<crystalEnergy[HCP]){
        diff = (fabs(crystalEnergy[SC]-crystalEnergy[FCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
    if(crystalEnergy[DIAMOND]<crystalEnergy[HCP]){
        diff = (fabs(crystalEnergy[DIAMOND]-crystalEnergy[FCC]))*1000.0;
        dv_penalty += diff*diff+1000000000.0;
    }
}
```

}~fsingle.h 後略~

9.5 プログラムのチェック

ディレクトリ `FSFe/check` を用いて動作確認を行う。このディレクトリの `para.dat` の上限値, 下限値は FS ポテンシャルパラメータの値となっているため, そのときの物性値を確認することができる。前節まで変更を加えたので, 以下に `make` をしてから実行する。

```
%make
```

```
%/GA FSFe/check
```

下に行った結果の一部を示す。(`fsingle.h` でデフォルトで指定しているように) カッコ内が合わせ込む値 (実験値・第一原理計算), カッコ外が FS ポテンシャルパラメータを用いて計算した結果である。体心立方格子の格子定数, エネルギー, 弾性定数, 面心立方格子のエネルギーは再現できているが, 面心立方格子の格子定数や弾性定数の再現性はいまひとつである (ただし, 実験値も色々あるので一概には言えないが)。

～FSFe/check/stdout.dat 前略～

```
CRYSTAL:    DIAMOND  ENE=-2.4998 (-3.1100)  LT=5.8397 (4.9040)
CRYSTAL:           SC  ENE=-3.2674 (-3.5300)  LT=2.1720 (2.3900)
CRYSTAL:           BCC  ENE=-4.2800 (-4.2800)  LT=2.8665 (2.8600)
CRYSTAL:           FCC  ENE=-4.2300 (-4.2100)  LT=3.6938 (3.5160)
CRYSTAL:  IDEALHCP  ENE=-4.2300 (-4.2200)  LT=2.6119 (2.4840)
CRYSTAL:           HCP  ENE=-4.2300 (-4.2200)  LTA=2.6119 (2.4840) LTC=4.2652 (3.9247)
ACRATIO=1.6330 (1.5800)
BM[           BCC]=173.115(172.000)
C11=243.117(242.000) C12=138.110(146.500)
C44=121.909(112.000)
BM[           FCC]=156.408(133.000)
C11=228.099(154.000) C12=120.560(122.500)
C44=107.540(77.000)
NONCRYST: BCC_vacancy  ENE=1.8521 (2.0000)  MAX_F=0.0030 ( 0[26])
～後略～
```

9.6 プログラムの実行(1回目)

8 ケースについて, プログラムを実行する。

```
%/GA FSFe/step01/case001
%/GA FSFe/step01/case002
%/GA FSFe/step01/case003
%/GA FSFe/step01/case004
%/GA FSFe/step01/case005
%/GA FSFe/step01/case006
%/GA FSFe/step01/case007
%/GA FSFe/step01/case008
```

筆者の手元にある計算機では 1 時間半程度で計算が終わった。

例えば case002 の場合, 面心立方格子の弾性定数が改善していることがわかる。

～FSFe/step01/case003/stdout.dat 終端部～

step=[31/31]

CRYSTAL: DIAMOND ENE=-2.4983 (-3.1100) LT=5.6731 (4.9040)
CRYSTAL: SC ENE=-3.4672 (-3.5300) LT=2.1588 (2.3900)
CRYSTAL: BCC ENE=-4.2800 (-4.2800) LT=2.8600 (2.8600)
CRYSTAL: FCC ENE=-4.2063 (-4.2100) LT=3.6555 (3.5160)
CRYSTAL: IDEALHCP ENE=-4.2063 (-4.2200) LT=2.5848 (2.4840)
CRYSTAL: HCP ENE=-4.2063 (-4.2200) LTA=2.5848 (2.4840) LTC=4.2210 (3.9247)

ACRATIO=1.6330 (1.5800)

BM[BCC]=171.469(172.000)

C11=241.307(242.000) C12=136.550(146.500)

C44=104.350(112.000)

BM[FCC]=133.546(133.000)

C11=138.754(154.000) C12=130.942(122.500)

C44=98.647(77.000)

NONCRYST: BCC_vacancy ENE=2.0780 (2.0000) MAX_F=0.3606 (5[26])

～FSFe/step01/case003/stdout.dat 後略～

ただ、面心立方格子の格子定数は再現できなかった。単純な関数形(合わせ込みに使えるパラメータ少ない)なので、余り多くの物性を再現することは難しい。

9.7 第2ステップの入力ファイル

第2ステップではパラメータの収束を目指す。ただ、第1ステップでおおよそ合わせ込みは終わっているので、調整程度の収束を行うくらいである。

9.7.1 cond.dat

～FSFe/step02/case00*/cond.dat～

cond.dat を以下のように設定する

id: 4 : FS ポテンシャル
iscont: 1 : 継続スタート
nstep: 51 : ステップ数 51
nint: 10 : 10 ステップごとに出力
noi: 1000 : 10000 個体
nout: 101 : トップ 999 個体の情報を出力
nmur: 0 : すべての FREE パラメータを突然変異
mutrate: 0.03 : 突然変異幅は 3%
seed: 1 : 乱数の seed は 1

9.7.2 cont.dat

第2ステップの個体数(1000)よりも第1ステップで gene.dat に出力した個体数が多い場合には cont.dat のために個体を抜き出さなくてはならない。今回は第1ステップでトータルで 1000 個体(nout-1)×7回(nint/nstep+1=7だから)=7000 個体出力されているはずなので、gene.dat から GA の最終ステップの個体を抜き出して cont.dat とする。パラメータ数が9でコメント行が1行あるので、1個体について gene.dat に10行記述されることになる。1回で1000個体出力されるから、1回10000行(10行×1000個体)の出力ということになる。例えば以下のように確認できる。gene.dat 中の最後10000行の先頭は GA の30ステップ目だが、

```
%tail -n10000 FSFe/step01/case001/gene.dat | head
```

```
step:30--fitnss[0]=-983130.776267--  
    d 3.79836727215631e+00  
    A 1.03498761971597e+00  
beta 1.07482019224549e-01  
    C 3.42593550850355e+00  
    c0 1.39775781032531e+00  
    c1 -2.29102985571413e-01  
    c2 -1.17558191841542e-01  
    N 5.00000000000000e-01  
Rcut 5.00000000000000e+00
```

gene.dat 中の最後 10010 行の先頭は GA の 25 ステップ目である.

```
%tail -n10010 FSFe/step01/case001/gene.dat | head
```

```
step:25--fitnss[857]=-1720467.689128--  
    d 3.81196633899678e+00  
    A 1.13753423818228e+00  
beta 1.00000000000000e-01  
    C 3.26340700164472e+00  
    c0 1.75478122021051e+00  
    c1 -3.34652223887140e-01  
    c2 -1.04000869005231e-01  
    N 5.00000000000000e-01  
Rcut 5.00000000000000e+00
```

よって

```
%tail -n 10000 FSFe/step01/case001/gene.dat >FSFe/step02/case001/cont.dat  
%tail -n 10000 FSFe/step01/case002/gene.dat > FSFe/step02/case002/cont.dat  
%tail -n 10000 FSFe/step01/case003/gene.dat > FSFe/step02/case003/cont.dat  
%tail -n 10000 FSFe/step01/case004/gene.dat > FSFe/step02/case004/cont.dat  
%tail -n 10000 FSFe/step01/case005/gene.dat > FSFe/step02/case005/cont.dat  
%tail -n 10000 FSFe/step01/case006/gene.dat > FSFe/step02/case006/cont.dat  
%tail -n 10000 FSFe/step01/case007/gene.dat > FSFe/step02/case007/cont.dat  
%tail -n 10000 FSFe/step01/case008/gene.dat > FSFe/step02/case008/cont.dat  
のように cont.dat を作成する.
```

9.7.3 set.dat

第 1 ステップではウォッチするだけであった欠陥生成エネルギーも軽く合わせ込みに用いる. 構造緩和も行うため, 5 ステップ目で緩和し, 以降 10 ステップごとに再緩和する. 重みも設定する.

```
~FSFe/step02/case00*/set.dat~
```

```
1
```

```
5 10
```

```
0 BCC_vacancy 10.0 0.0 127 1
```

9.8 プログラムの実行(2回目)

8 ケースについて、プログラムを実行する。

```
%/GA FSFe/step02/case001
%/GA FSFe/step02/case002
%/GA FSFe/step02/case003
%/GA FSFe/step02/case004
%/GA FSFe/step02/case005
%/GA FSFe/step02/case006
%/GA FSFe/step02/case007
%/GA FSFe/step02/case008
```

例えば case003 の場合、step01(9.6 節参照)に比べ、step02 における調整によって大きな変化は見られないが、空孔構造を構造緩和しても第一原理計算の値と一致できていることを確認できた。

～FSFe/step02/case003/stdout.dat 終端部～

```
step=[51/51]
CRYSTAL:    DIAMOND  ENE=-2.5023 (-3.1100)  LT=5.6731 (4.9040)
CRYSTAL:         SC  ENE=-3.4631 (-3.5300)  LT=2.1606 (2.3900)
CRYSTAL:         BCC  ENE=-4.2800 (-4.2800)  LT=2.8600 (2.8600)
CRYSTAL:         FCC  ENE=-4.2083 (-4.2100)  LT=3.6545 (3.5160)
CRYSTAL:    IDEALHCP  ENE=-4.2083 (-4.2200)  LT=2.5841 (2.4840)
CRYSTAL:         HCP  ENE=-4.2083 (-4.2200)  LTA=2.5841 (2.4840) LTC=4.2199 (3.9247)
ACRATIO=1.6330 (1.5800)
BM[         BCC]=171.529(172.000)
C11=240.884(242.000) C12=136.851(146.500)
C44=104.714(112.000)
BM[         FCC]=134.138(133.000)
C11=139.697(154.000) C12=131.358(122.500)
C44=99.025(77.000)
NONCRYST: BCC_vacancy  ENE=1.8847 (2.0000)  MAX_F=0.0045 (96[26])
```

～FSFe/step02/case003/stdout.dat 後略～

得られたポテンシャルパラメータは poten.dat に出力されている。

```
～FSFe/step02/case003/poten.dat～
      d 3.79830513349216e+00
      A 1.04483936603585e+00
beta  1.00000000000000e-01
      C 3.41927381303210e+00
      c0 1.47449844489063e+00
      c1 -2.85085963589849e-01
      c2 -1.05796469492773e-01
      N 5.00000000000000e-01
      Rcut 5.00000000000000e+00
```

10. 例題 Long-Range Finnis-Sinclair 型 Cu 用単元系ポテンシャルの再開発

Long-Range Finnis-Sinclair (LRFS) 型ポテンシャル関数は面心立方格子を安定構造にもつ金属用ポテンシャル関数として使われているポテンシャル関数である。練習として Cu 用ポテンシャルパラメータを再開発する。a 文献値を用いずに、第一原理計算を行い、入力用物性を自分で決定する手法も解説する。本章で行われる例題はディレクトリ, LRFSCu に含まれる。

10.1 LRFS ポテンシャル関数形

本ソフトウェアで用いられる LRFS ポテンシャル関数形は以下のようなものである。一般的には $\varepsilon_i = \varepsilon_m$, $a_i = a_m$ であるが最終的には c を用いて必ず, $\varepsilon_i = \varepsilon_m$, $a_i = a_m$ と換算できるので, 異なっても特に問題はない。

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) - \sum_i \varepsilon_m c \sqrt{\rho_i}$$

$$\phi(r) = \varepsilon_i \left(\frac{a_i}{r} \right)^n$$

$$\rho_i = \sum_{j \neq i} f(r_{ij})$$

$$f(r) = \left(\frac{a_m}{r} \right)^m$$

10.2 ヘッダの確認

10.2.1 合わせ込み条件ヘッダ condCu.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。一部抜粋になっているが、基本的なバルク単元系結晶(ダイヤモンド構造, 正方格子, 体心立方格子, 面心立方格子, 六方最密格子)のエネルギーと格子定数, 安定構造の面心立方格子の体積弾性率, 弾性定数を合わせ込みに用いる設定になっている。

```
void setFlagsOfSingleCrystalsFe(int *isUsedForFitting,int *isCalcBulkModulus,
                                int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
    isUsedForFitting[DIAMOND] = 1;
    isUsedForFitting[SC] = 1;
    isUsedForFitting[BCC] = 1;
    isUsedForFitting[FCC] = 1;
    isUsedForFitting[IDEALHCP] = 1;
    isUsedForFitting[HCP] = 1;

    isCalcBulkModulus[FCC] = 1;
    isCalcTensileElasticConstants[FCC] = 1;
    isCalcShearElasticConstants[FCC] = 1;
}
```

重みを設定する関数

```
void setWeightOfSingleCrystalsCu(引数略){
    weightForCohesiveEnergy[DIAMOND] = 10.0;
```

```

weightForCohesiveEnergy[SC]          = 20.0;
weightForCohesiveEnergy[IDEALHCP] = 0.0;
weightForCohesiveEnergy[HCP]         = 10.0;
weightForCohesiveEnergy[FCC]         = 30.0;

weightForLatticeConstant[DIAMOND]    = 0.0;
weightForLatticeConstant[SC]         = 0.0;
weightForLatticeConstant[BCC]        = 0.0;
weightForLatticeConstant[BCT]        = 0.0;
weightForLatticeConstant[IDEALHCP]   = 0.0;
weightForLatticeConstant[HCP]        = 0.0;
weightForLatticeConstant[FCC]        = 10.0;

weightForBulkModulus[FCC] = 40.0;
weightForTensileElasticConstants[FCC] = 25.0;
weightForShearElasticConstants[FCC] = 25.0;
}

```

10.2.2 物性の入力されているヘッダ matpropCu.h の入力

合わせ込みに用いる物性を計算し入力する。第一原理計算の結果は実験とは少し異なるので、第一原理計算による安定構造（面心立方格子）に対する相対エネルギーと安定構造の格子定数が実験と合うようにスケーリングした格子定数を用いる。結果的に用いるエネルギー、格子定数は、以下の表のようになる。

表 Cu 合わせ込みに使うエネルギー

	原子数	DFT による格子エネルギー[eV]	DFT による 1 原子あたりエネルギー[eV/atom]	DFT による 1 原子あたり相対エネルギー[eV/atom]	合わせ込みに使うエネルギー[eV/atom]
diamond	8	-21.571	-2.696	1.027	-2.513
sc	1	-3.261	-3.261	0.462	-3.078
bcc	2	-7.376	-3.688	0.035	-3.505
fcc	4	-14.892	-3.723	0.000	-3.540 (実験値)
hcp	2	-7.428	-3.714	0.009	-3.531

原子 1 個あたりのエネルギーは差し引いていないので注意!!

表 Cu 合わせ込みに使う格子定数

	格子定数(DFT)[Å]	格子定数(スケーリング後)[Å]
diamond	5.368	5.343
sc	2.407	2.396
bcc	2.888	2.874
fcc	3.632	3.615(実験値)
hcp(a 軸,c 軸)	2.568, 4.210	2.556, 4.190

なお第一原理計算は密度汎関数 (DFT) 理論に基づく平面波擬ポテンシャル法ソフトウェア `vasp` により行った。交換相関項には Perdew, Burke, Ernzerhof らによって提案された一般化密度近似を用いている。実験値は, Y. Mishin et al., Phys. Rev. B, 63, 224106 and references therein による。

上記計算結果を `matpropCu.h` に入力すると以下のようなになる。

～`matpropCu.h` 冒頭部～

```
void setMaterialsPropertiesOfCopper(double* latticeConstantA,
                                   double* latticeConstantB,
                                   double* latticeConstantC,
                                   double* internalParameterX1,
                                   double* internalParameterX2,
                                   double* internalParameterX3,
                                   double* cohesiveEnergy,
                                   double* bulkModulus,
                                   double* C11,double* C22,double* C33,
                                   double* C12,double* C13,double* C23,
                                   double* C44,double* C55,double* C66){
    cohesiveEnergy[DIAMOND]=-2.51342200*8.0 ;
    cohesiveEnergy[SC]=-3.07792350 ;
    cohesiveEnergy[BCC]=-3.50520450*2.0 ;
    cohesiveEnergy[FCC]=-3.54000000*4.0 ;
    cohesiveEnergy[IDEALHCP]=-3.53120900*4.0 ;
    cohesiveEnergy[HCP]=-3.53120900*4.0 ;

    latticeConstantA[DIAMOND]=5.34257893 ;
    latticeConstantA[SC]=2.39581959 ;
    latticeConstantA[BCC]=2.87447384 ;
    latticeConstantA[FCC]=3.61500000 ;
    latticeConstantA[IDEALHCP]=2.55572325 ;
    latticeConstantA[HCP]=2.55572325 ;
    latticeConstantC[HCP]=4.18998439 ;

    //elastic properties are from Y. Mishin et al., PHYSICAL REVIEW B, VOLUME 63, 224106
    bulkModulus[FCC] = 138.3;

    C11[FCC] = 170.0;
    C12[FCC] = 122.5;
    C44[FCC] = 75.8;
}
```

格子のエネルギー (1 原子あたりのエネルギー×基本格子中の原子数) として入力していることに注意が必要である。合わせ込みプログラムの中では六方最密格子の基本格子は最小単位(2 原子)を 2 つ組み合わせて、4 原子含むものとしている。

10.3 第1ステップの入力ファイル

10.3.1 element.dat

単元系 Cu 用の原子間ポテンシャルなので case00* の element.dat を以下のように設定する

～LEFSCu/step01/case00*/element.dat～

1

Cu

10.3.2 cond.dat

カットオフ距離が長距離に及ぶため、それほど多くの個体を取り扱うことができない。case001 の cond.dat を以下のように設定する。

～CuLRFS/step01/case001/cond.dat～

id: 3 : LRFS ポテンシャル
iscont: 0 : 新規スタート
nstep: 31 : ステップ数 31
nint: 5 : 5 ステップごとに出力
noi: 5000 : 5000 個体
nout: 501 : トップ 500 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 1 : スケーリングを行う

case002～case008 の乱数の seed は 2～8 とした。

10.3.3 para.dat

既存のポテンシャルパラメータを参考にしておおよそのパラメータ範囲を決める。以下の表のように、パラメータ m,n を振ることにする。

表 パラメータ範囲の振り分け

	m	n
case001	6	9
case002	6	10
case003	6	11
case004	6	12
case005	8	9
case006	8	10
case007	8	11
case008	8	12

case001 の para.dat は以下のようなになる。

～LRFSCu/step01/case001/para.dat～

0 FREE eps2 0.000000 0.100000 : ε t

1	FREE	A2	0.0000000	5.0000000	: a _t
2	FIX	N2	9.0000000	9.0000000	: n
3	FREE	epsm	0.0000000	0.1000000	: ε _m
4	FREE	Cm	0.0000000	500.00000	: c
5	FREE	Am	0.0000000	5.0000000	: a _m
6	FIX	Mm	6.0000000	6.0000000	: m
7	FIX	Rcut	7.3000000	7.3000000	: カットオフ距離

面心立方格子を最安定とする EAM ポテンシャルの場合は平衡結合長の 3 倍以上程度のカットオフ距離が必要であることに注意が必要である。

grep すればパラメータの設定を確認することができる。

```
%grep N2 LRFSCu/step01/case00*/para.dat
```

```
LRFSCu/step01/case001/para.dat:2      FIX      N2      9.0000000e+00  9.0000000e+00
LRFSCu/step01/case002/para.dat:2      FIX      N2     10.0000000e+00 10.0000000e+00
LRFSCu/step01/case003/para.dat:2      FIX      N2     11.0000000e+00 11.0000000e+00
LRFSCu/step01/case004/para.dat:2      FIX      N2     12.0000000e+00 12.0000000e+00
LRFSCu/step01/case005/para.dat:2      FIX      N2      9.0000000e+00  9.0000000e+00
LRFSCu/step01/case006/para.dat:2      FIX      N2     10.0000000e+00 10.0000000e+00
LRFSCu/step01/case007/para.dat:2      FIX      N2     11.0000000e+00 11.0000000e+00
LRFSCu/step01/case008/para.dat:2      FIX      N2     12.0000000e+00 12.0000000e+00
```

```
%grep Mm LRFSCu/step01/case00*/para.dat
```

```
LRFSCu/step01/case001/para.dat:6      FIX      Mm      6.0000000e+00  6.0000000e+00
LRFSCu/step01/case002/para.dat:6      FIX      Mm      6.0000000e+00  6.0000000e+00
LRFSCu/step01/case003/para.dat:6      FIX      Mm      6.0000000e+00  6.0000000e+00
LRFSCu/step01/case004/para.dat:6      FIX      Mm      6.0000000e+00  6.0000000e+00
LRFSCu/step01/case005/para.dat:6      FIX      Mm      8.0000000e+00  8.0000000e+00
LRFSCu/step01/case006/para.dat:6      FIX      Mm      8.0000000e+00  8.0000000e+00
LRFSCu/step01/case007/para.dat:6      FIX      Mm      8.0000000e+00  8.0000000e+00
LRFSCu/step01/case008/para.dat:6      FIX      Mm      8.0000000e+00  8.0000000e+00
```

10.3.4 set.dat

空孔の入った面心立方格子の欠陥生成エネルギーも合わせ込みに用いる。最初のステップではポテンシャルがばらつきやすいので構造緩和は行わないが、ある程度非緩和構造のエネルギーでも合わせ込みをすることができるので、重みは 30.0、緩和開始ステップは GA のステップ数(31)より多い 100 に設定した（つまり、緩和は始まらない）。

```
~LRFSCu/step01/case00*/set.dat~
```

```
0
```

```
100 10
```

```
0 FCC_vacancy 30.0 0.0 499 1
```

カットオフが非常に長い (7.3Å) ポテンシャルであることから面心立方格子の基本格子 (4 原子) を 5×5×5 積み重ね、1 原子抜いた (=空孔) 原子構造であるため、原子数が 499 個となっている。

10.3.5 energy.dat

文献値 (Mishin et al., Phys. Rev. B, 63, 224106 and references therein) から適当な値を設定する. 絶対エネルギーではなく相対的な欠陥生成エネルギーを合わせ込みたいので, 欠陥生成エネルギーで定義した.

~LRFSCu/step01/case00*/energy.dat~

0 FCC_vacancy 1.27

10.3.6 lattice.dat, posi.dat

第一原理計算から得られた緩和後の原子構造を用いるのが理想的なだが, 格子の緩和による欠陥生成エネルギーの変化は小さいので, 面心立方格子の完全結晶から原子を抜いた構造 (空孔型欠陥) を初期原子構造とする. lattice.dat は以下のような格子定数 3.615 \AA の $5 \times 5 \times 5$ の面心立方格子とした.

~LRFSCu/step01/case00*/lattice.dat~

0 FCC_vacancy

18.0750000000	0.0000000000	0.0000000000
0.0000000000	18.0750000000	0.0000000000
0.0000000000	0.0000000000	18.0750000000

posi.dat も一部を示す. 空孔型欠陥構造では面心立方格子から原子を 1 個抜いている.

~LRFSCu/step01/case00*/posi.dat~

0 FCC_vacancy 499

29	1.8075000000	1.8075000000	0.0000000000	0	0	0
29	1.8075000000	0.0000000000	1.8075000000	0	0	0
29	0.0000000000	1.8075000000	1.8075000000	0	0	0
29	0.0000000000	0.0000000000	3.6150000000	0	0	0
29	1.8075000000	1.8075000000	3.6150000000	0	0	0
29	1.8075000000	0.0000000000	5.4225000000	0	0	0
29	0.0000000000	1.8075000000	5.4225000000	0	0	0
29	0.0000000000	0.0000000000	7.2300000000	0	0	0
29	1.8075000000	1.8075000000	7.2300000000	0	0	0

~後略~

10.4 適応度関数の設定

面心立方格子が再安定構造になることを最優先にしたいが, 元来, LRFSP ポテンシャル関数は面心立方格子・六方最密格子が安定しやすく, 面心立方格子のエネルギーと六方最密格子のエネルギーに差をつけることが難しい. このため, 特別な設定は行わない.

10.5 プログラムのチェック

ディレクトリ LRFSCu/check を用いて動作確認を行う。このディレクトリの para.dat の上限値、下限値は LRFS ポテンシャルパラメータの値となっているため、そのときの物性値を確認することができる。前節まで変更を加えたので、以下に make をしてから実行する。

```
%make
```

```
%/GA LRFSCu/check
```

カットオフ距離が非常に長いので、前処理に時間がかかっていることが体感できると思う。下に実行した結果の一部を示す。(fsingle.h でデフォルトで指定しているように) カッコ内が合わせ込む値(実験値・第一原理計算)、カッコ外が LRFS ポテンシャルパラメータを用いて計算した結果である。ダイヤモンド構造、正方格子、体心立方格子、面心立方格子の格子定数、エネルギー、面心立方格子の弾性定数は再現できているが、六方最密格子と面心立方格子のエネルギー順(六方最密格子のエネルギー > 面心立方格子のエネルギー)の再現はできていない。

～LRFSCu/check/stdout.dat 前略～

```
step=[5/5]
```

```
CRYSTAL:    DIAMOND  ENE=-2.7741 (-2.5134)  LT=5.3579 (5.3426)
```

```
CRYSTAL:         SC  ENE=-3.1981 (-3.0779)  LT=2.4031 (2.3958)
```

```
CRYSTAL:         BCC ENE=-3.4729 (-3.5052)  LT=2.8746 (2.8745)
```

```
CRYSTAL:         FCC  ENE=-3.4976 (-3.5400)  LT=3.6102 (3.6150)
```

```
CRYSTAL:    IDEALHCP ENE=-3.4980 (-3.5312)  LT=2.5528 (2.5557)
```

```
CRYSTAL:         HCP  ENE=-3.4980 (-3.5312)  LTA=2.5526 (2.5557) LTC=4.1695 (4.1900)
```

```
ACRATIO=1.6334 (1.6395)
```

```
BM[         FCC]=142.914(138.300)
```

```
C11=169.971(170.000) C12=129.374(122.500)
```

```
C44=57.921(75.800)
```

```
NONCRYST: FCC_vacancy  ENE=0.8699 (1.2700)  MAX_F=0.0172 ( 1[29])
```

～後略～

10.6 プログラムの実行(1回目)

8 ケースについて、プログラムを実行する。

```
%/GA LRCu/step01/case001
```

```
%/GA LRCu/step01/case002
```

```
%/GA LRCu/step01/case003
```

```
%/GA LRCu/step01/case004
```

```
%/GA LRCu/step01/case005
```

```
%/GA LRCu/step01/case006
```

```
%/GA LRCu/step01/case007
```

```
%/GA LRCu/step01/case008
```

筆者の手元にある計算機では 1 時間半程度で計算が終わった。デフォルトの m,n(つまり, case001)以外は弾性定数を再現することができなかった。ただし、何も設定しなかったにも関わらず運よく六方最密格子と面心立方格子のエネルギー順(六方最密格子のエネルギー > 面心立方格子のエネルギー)の再現ができています。

～LRFSCu/step01/case001/stdout.dat 終端部(成功例)～

step=[31/31]

CRYSTAL: DIAMOND ENE=-2.8061 (-2.5134) LT=5.3653 (5.3426)
CRYSTAL: SC ENE=-3.2350 (-3.0779) LT=2.4064 (2.3958)
CRYSTAL: BCC ENE=-3.5129 (-3.5052) LT=2.8786 (2.8745)
CRYSTAL: FCC ENE=-3.5400 (-3.5400) LT=3.6150 (3.6150)
CRYSTAL: IDEALHCP ENE=-3.5383 (-3.5312) LT=2.5563 (2.5557)
CRYSTAL: HCP ENE=-3.5383 (-3.5312) LTA=2.5561 (2.5557) LTC=4.1752 (4.1900)
ACRATIO=1.6334 (1.6395)
BM[FCC]=144.078(138.300)
C11=171.148(170.000) C12=130.531(122.500)
C44=58.496(75.800)
NONCRYST: FCC_vacancy ENE=0.9263 (1.2700) MAX_F=0.1392 (98[29])
~LRFSCu/step01/case001/stdout.dat 後略~

~LRFSCu/step01/case008/stdout.dat 終端部(弾性定数過剰見積もり)~

step=[31/31]

CRYSTAL: DIAMOND ENE=-2.7224 (-2.5134) LT=5.5054 (5.3426)
CRYSTAL: SC ENE=-3.1301 (-3.0779) LT=2.4421 (2.3958)
CRYSTAL: BCC ENE=-3.4834 (-3.5052) LT=2.8860 (2.8745)
CRYSTAL: FCC ENE=-3.5400 (-3.5400) LT=3.6150 (3.6150)
CRYSTAL: IDEALHCP ENE=-3.5399 (-3.5312) LT=2.5562 (2.5557)
CRYSTAL: HCP ENE=-3.5399 (-3.5312) LTA=2.5563 (2.5557) LTC=4.1740 (4.1900)
ACRATIO=1.6328 (1.6395)
BM[FCC]=256.144(138.300)
C11=310.417(170.000) C12=228.980(122.500)
C44=100.917(75.800)
NONCRYST: FCC_vacancy ENE=0.9359 (1.2700) MAX_F=0.0986 (480[29])
~LRFSCu/step01/case008/stdout.dat 後略~

10.7 第2ステップの入力ファイル

唯一上手く行った case001 に関して第2ステップではパラメータの収束を目指す。ただ、第1ステップでおおよそ合わせ込みは終わっているので、緩和後の空孔生成エネルギーを確認しながら、調整程度の収束を行うくらいである。

10.7.1 cond.dat

~LRFSCu/step02/case001/cond.dat~

cond.dat を以下のように設定する

id: 4 : FS ポテンシャル
iscont: 1 : 継続スタート
nstep: 51 : ステップ数 51
nint: 10 : 10 ステップごとに出力
noi: 500 : 500 個体
nout: 101 : トップ 100 個体の情報を出力

nmut: 0 :すべての FREE パラメータを突然変異
mutrate: 0.01 : 突然変異幅は 1%
seed: 1 : 乱数の seed は 1

10.7.2 cont.dat

GA の最終ステップで出力された個体(500)のみ, 次のステップに用いる(少ない分には問題ない). パラメータ数は 8 なのでコメント行と合わせて 1 個体当たり 9 行となるから, gene.dat の最後の 4500 行(9 行×500 個体)を第 2 ステップの cont.dat に用いる. よって

```
%tail -n4500 LRFSCu/step01/case001/gene.dat > LRFSCu/step02/case001/cont.dat
```

のように cont.dat を作成する.

10.7.3 set.dat

構造緩和を行うため, 最初に 1 ステップ目で緩和し, 以降 10 ステップごとに再緩和する.

～ LRFSCu/step02/case001/set.dat～

```
1  
1 10  
0 FCC_vacancy 30.0 0.0 499 1
```

10.8 プログラムの実行(2 回目)

プログラムを実行する.

```
%./GA LRFSCu/step02/case001
```

step01 と比べてほとんど変化はなかった.

～LRFSCu/step02/case001/stdout.dat 終端部～

```
step=[51/51]
```

```
CRYSTAL: DIAMOND ENE=-2.8061 (-2.5134) LT=5.3653 (5.3426)  
CRYSTAL: SC ENE=-3.2350 (-3.0779) LT=2.4064 (2.3958)  
CRYSTAL: BCC ENE=-3.5129 (-3.5052) LT=2.8786 (2.8745)  
CRYSTAL: FCC ENE=-3.5400 (-3.5400) LT=3.6150 (3.6150)  
CRYSTAL: IDEALHCP ENE=-3.5383 (-3.5312) LT=2.5563 (2.5557)  
CRYSTAL: HCP ENE=-3.5383 (-3.5312) LTA=2.5561 (2.5557) LTC=4.1752 (4.1900)  
ACRATIO=1.6334 (1.6395)  
BM[ FCC]=144.078(138.300)  
C11=171.148(170.000) C12=130.531(122.500)  
C44=58.496(75.800)  
NONCRYST: FCC_vacancy ENE=0.9027 (1.2700) MAX_F=0.0169 (495[29])
```

～LRFSCu/step02/case001/stdout.dat 後略～

得られたポテンシャルパラメータは poten.dat に出力されている.

～LRFSCu/step02/case001/poten.dat～

```
eps2 2.12916255623907e-01  
A2 2.63880176087711e+00  
N2 9.00000000000000e+00
```

epsm 2.12916255623907e-01

Cm 4.82248815224088e+01

Am 1.31840035375178e+00

Mm 6.00000000000000e+00

Rcut 7.30000000000000e+00

11. 例題 AI 用 Generalized Embedded Atom Method(GEAM)ポテンシャルの開発

前述の FS ポテンシャルを含むような EAM ポテンシャルは一般的に、体心立方格子、面心立方格子、六方最密格子のいずれかを最安定とするような金属にしか使えないことが多い。これに対し、GEAM ポテンシャルはこれら 3 つの構造を最安定とする金属について単一のポテンシャル関数で対応することができる。また、異元素間のポテンシャルパラメータも簡単に作成することができる。このため近年注目を集めているがアルミニウムのポテンシャルだけが実験値と大きく異なる物性を示す（という話である）。このため本章では AI 用 GEAM ポテンシャルパラメータを開発する。本章における例題はディレクトリ GEAMAI に含まれる。

11.1 GEAM ポテンシャル関数

GEAM ポテンシャル関数は以下に示すとおりである。

$$E = \frac{1}{2} \sum_{i \neq j} \phi(r_{ij}) + \sum_i F(\rho_i)$$

$$\phi(r) = \frac{A \cdot \exp\left[-\alpha\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \kappa\right)^{20}} - \frac{B \cdot \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^{20}}, \quad F(\rho) = \begin{cases} \sum_{i=0}^3 F_{ni} \left(\frac{\rho}{\rho_n} - 1\right)^i, & \rho < \rho_n \\ \sum_{i=0}^3 F_i \left(\frac{\rho}{\rho_e} - 1\right)^i, & \rho_n \leq \rho < \rho_o \\ F_e \left[1 - \ln\left(\frac{\rho}{\rho_s}\right)\right]^\eta \cdot \left(\frac{\rho}{\rho_s}\right)^\eta, & \rho_o \leq \rho \end{cases}$$

$$\rho_i = \sum_{j, j \neq i} f_j(r_{ij}), \quad f(r) = \frac{f_e \cdot \exp\left[-\beta\left(\frac{r}{r_e} - 1\right)\right]}{1 + \left(\frac{r}{r_e} - \lambda\right)^{20}}$$

$F(\rho)$ に関して、 ρ の値によって3つの関数を繋ぎ合わせているため、その連続性を考慮する必要がある。具体的には $\rho = 0.85 \rho_e$ 、 $\rho = 1.15 \rho_e$ で値や微分値が連続になるようにしなくてはならない。オリジナルの文献では $\rho = 0.85 \rho_e$ において値、1階微分、2階微分が連続であり、 $\rho = 1.15 \rho_e$ において値、1階微分が連続である。しかし、本例題では $\rho_n = 0.85 \rho_e$ 、 $\rho_o = 1.15 \rho_e$ 両方において値、1階微分、2階微分が連続とする。よって、 $F(\rho)$ に関わる9個のパラメータのうち、 $F_{n0}, F_{n1}, F_{n2}, F_{n3}, F_0, F_1, F_2, F_3, F_e, \eta$ のうち独立なパラメータは3つしか設定できない。さらに、 F_0, F_1, η を一応独立なパラメータとするものの $\rho = \rho_e$ のときの埋め込み関数の傾きをゼロとするため $F_1 = 0$ であるから実質的な独立パラメータは2つのみである。また、 f_e の値も不定で問題ない。結局は ρ_e, ρ_s で割るので、 $\rho_e / f_e, \rho_s / f_e$ だけわかっているだけで問題ないことになる。

11.2 ヘッダの確認

11.2.1 合わせ込み条件ヘッダ condAl.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。一部抜粋になっているが、基本的な単元系結晶(ダイヤモンド構造、正方格子、体心立方格子、面心立方格子)のエネルギーと格子定数、安定構造の面心立方格子の体積弾性率、弾性定数を合わせ込みに用いる設定になっている。

```
void setFlagsOfSingleCrystalsAl(int *isUsedForFitting, int *isCalcBulkModulus,
                                int *isCalcTensileElasticConstants, int *isCalcShearElasticConstants){
    isUsedForFitting[GRAPHITE] = 1;
```

```

isUsedForFitting[DIAMOND] = 1;
isUsedForFitting[SC] = 1;
isUsedForFitting[BCC] = 1;
isUsedForFitting[IDEALHCP] = 1;
isUsedForFitting[HCP] = 1;
if(isUsedForFitting[HCP]==1){
    isUsedForFitting[IDEALHCP] = 1;
}
isUsedForFitting[FCC] = 1;

isCalcBulkModulus[FCC] = 1;
isCalcTensileElasticConstants[FCC] = 1;
isCalcShearElasticConstants[FCC] = 1;
}

```

重みを設定する関数

```

void setWeightOfSingleCrystalsAl(引数略){
    weightForCohesiveEnergy[GRAPHITE] =0.0;
    weightForCohesiveEnergy[DIAMOND] =20.0;
    weightForCohesiveEnergy[SC] = 30.0;
    weightForCohesiveEnergy[BCC] = 50.0;
    weightForCohesiveEnergy[IDEALHCP] = 50.0;
    weightForCohesiveEnergy[HCP] = 60.0;
    weightForCohesiveEnergy[BCT] = 0.0;
    weightForCohesiveEnergy[FCC] = 100.0;

    weightForLatticeConstant[GRAPHITE] = 0.0;
    weightForLatticeConstant[DIAMOND] = 10.0;
    weightForLatticeConstant[SC] = 10.0;
    weightForLatticeConstant[BCC] = 30.0;
    weightForLatticeConstant[IDEALHCP] = 0.0;
    weightForLatticeConstant[HCP] = 30.0;
    weightForLatticeConstant[FCC] = 30.0;
}

```

11.2.2 物性の入力されているヘッダ `matpropAl.h` の確認

合わせ込みに用いる物性が設定されているか確認する. `matpropAl.h` には 2 シリーズの物性が入力してあるのだが, //for GEAM-Al potential とコメントしてある方が新しい計算の結果であるので, そちらを用いてほしい. また, 体心立方格子の凝集エネルギー, 体積弾性率, 弾性定数については Y. Mishin / Acta Materialia 52 (2004) 1451–1467 から採った.

～`matpropAl.h` 冒頭部～

```

void setMaterialsPropertiesOfAluminum(double* latticeConstantA,

```

```
double* latticeConstantB,  
double* latticeConstantC,  
double* internalParameterX1,  
double* internalParameterX2,  
double* internalParameterX3,  
double* cohesiveEnergy,  
double* bulkModulus,  
double* C11,double* C22,double* C33,  
double* C12,double* C13,double* C23,  
double* C44,double* C55,double* C66){
```

～中略～

```
//for GEAM-Al potential
```

```
//material properties of FCC are from Y. Mihin, Acata Materializa 52 (2004), 1451-1467
```

```
//other properties are calculated by T. Kumagai
```

```
cohesiveEnergy[GRAPHITE] = -2.17210150*4.0 ;  
cohesiveEnergy[DIAMOND]  = -2.61349350*8.0 ;  
cohesiveEnergy[SC]       = -2.98798900*1.0 ;  
cohesiveEnergy[BCC]      = -3.26314100*2.0 ;  
cohesiveEnergy[IDEALHCP] = -3.32687500*4.0 ;  
cohesiveEnergy[HCP]      = -3.32687500*4.0 ;  
cohesiveEnergy[FCC]      = -3.36000000*4.0 ;
```

```
latticeConstantA[GRAPHITE] = 4.488260115;  
latticeConstantA[DIAMOND]  = 6.060146768;  
latticeConstantA[SC]       = 2.72726785;  
latticeConstantA[BCC]      = 3.242846044;  
latticeConstantA[IDEALHCP] = 2.864261517;  
latticeConstantA[HCP]      = 2.864261517;  
latticeConstantC[HCP]      = 4.716576903;  
latticeConstantA[FCC]      = 4.05;
```

```
bulkModulus[FCC] = 79.0;
```

```
C11[FCC] = 114.0;  
C12[FCC] = 61.9;  
C44[FCC] = 31.6;
```

```
}
```

11.3 第1ステップの入力ファイル

ディレクトリ GEAMAl/step01/case001/～case008/中の入力ファイルを設定する

11.3.1 element.dat

単元系 Al 用の原子間ポテンシャルなので case00* の element.dat を以下のように設定する

～GEAMAI/step01/case001/element.dat～

1

Al

11.3.2 cond.dat

case001 の cond.dat を以下のように設定する

～GEAMAI/step01/case001/set.dat～

id: 1 : Tersoff ポテンシャル
iscont: 0 : 新規スタート
nstep: 31 : ステップ数 31
nint: 5 : 5 ステップごとに出力
noi: 20000 : 20000 個体
nout: 1001 : トップ 1000 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 1 : スケーリングを行う

case002～case008 の乱数の seed は 2～8 とした。

11.3.3 para.dat

パラメータ F_0 を以下の表のように case001-case008 で振ることにする。

表：パラメータ振り分け

	F_0 (FIX で上下限)
case001	-2.43
case002	-2.53
case003	-2.63
case004	-2.73
case005	-2.83
case006	-2.93
case007	-3.03
case008	-3.13

case001 の para.dat を示す。いくつか余計なフリーパラメータ ($\beta_2, N_\kappa, N_\lambda$) などが設定されているが、今回は利用しない。カットオフ距離も本来もう少し長いほうが望ましいものの(平衡結合長×3)計算時間短縮のため、少し短めに設定してある。

0	FIX	re	2.863924	2.863924	: re
1	FREE	rhoe/fe	1.000000	100.0000	: ρ_e / f_e
2	FIX	rhos/e	1.000000	1.000000	: ρ_s / ρ_e
3	FREE	alpha	0.000000	10.00000	: α
4	FREE	beta1	0.000000	10.00000	: $\beta_1 = \beta$

5	FIX	beta2/beta1	1.000000	1.000000	: $\beta_2/\beta_1=1$
6	FIX	A	0.000000	1.000000	: A
7	FIX	B	0.000000	1.000000	: B
8	FREE	lambda	0.000000	1.000000	: λ
9	FREE	kappa/lambda	0.200000	0.800000	: κ/λ
10	FIX	nkappa	20.000000	20.000000	: $N_\kappa=20$
11	FIX	nlambda1	20.000000	20.000000	: $N_\lambda=20$
12	FIX	f0	-2.43	-2.43	: F_0
13	FIX	f1	0.0	0.0	: F_1
14	FREE	eta	0.0	1.5	: η
15	FIX	rhon	0.85	0.85	: $\rho_n/\rho_e=0.85$
16	FIX	rhoo	1.15	1.15	: $\rho_o/\rho_e=1.15$
17	FIX	Rcut	7.5	7.5	: カットオフ距離

grep を行うと F_0 の設定が出来ていることがわかる.

```
>grep f0 GEAMAI/step01/case00*/para.dat
```

```
GEAMAI/step01/case001/para.dat: 12      FIX      f0      -2.43      -2.43
GEAMAI/step01/case002/para.dat: 12      FIX      f0      -2.53      -2.53
GEAMAI/step01/case003/para.dat: 12      FIX      f0      -2.63      -2.63
GEAMAI/step01/case004/para.dat: 12      FIX      f0      -2.73      -2.73
GEAMAI/step01/case005/para.dat: 12      FIX      f0      -2.83      -2.83
GEAMAI/step01/case006/para.dat: 12      FIX      f0      -2.93      -2.93
GEAMAI/step01/case007/para.dat: 12      FIX      f0      -3.03      -3.03
GEAMAI/step01/case008/para.dat: 12      FIX      f0      -3.13      -3.13
```

11.3.4 set.dat

空孔の入った面心立方格子の欠陥生成エネルギーも合わせ込みに用いる. 最初のステップではポテンシャルがばらつきやすいので構造緩和は行わないが, ある程度非緩和構造のエネルギーでも合わせ込みをすることができるので, 重みは 30.0, 緩和開始ステップは GA のステップ数(31)より多い 50 に設定した (つまり, 緩和は始まらない) .

```
~GEAMAI/step01/case00*/set.dat~
```

```
0
```

```
100 10
```

```
0 FCC_vacancy 30.0 0.0 499 1
```

カットオフが非常に長い (7.5 Å) ポテンシャルであることから面心立方格子の基本格子 (4 原子) を 5×5×5 積み重ね, 1 原子抜いた (=空孔) 原子構造であるため, 原子数が 499 個となっている.

11.3.5 energy.dat

文献値 (Y. Mihm, *Acata Materializa* 52 (2004), 1451-1467 and references therein) から適当な値を設定する. 絶対エネルギーではなく相対的な欠陥生成エネルギーを合わせ込みたいので, 欠陥生成エネルギーで定義した.

```
~GEAMAI/step01/case00*/energy.dat~
```

```
0 FCC_vacancy 0.68
```

11.3.6 lattice.dat, posi.dat

第一原理計算から得られた緩和後の原子構造を用いるのが理想的のだが、格子の緩和による欠陥生成エネルギーの変化は小さいので、面心立方格子の完全結晶から原子を抜いた構造（空孔型欠陥）を初期原子構造とする。lattice.dat は以下のような格子定数 4.05 \AA の $5 \times 5 \times 5$ の面心立方格子とした。

～GEAMAI/step01/case00*/lattice.dat～

0 FCC_vacancy

```
20.2500000000    0.0000000000    0.0000000000
 0.0000000000   20.2500000000    0.0000000000
 0.0000000000    0.0000000000   20.2500000000
```

posi.dat も一部を示す。空孔型欠陥構造では面心立方格子から原子を 1 個抜いている。

～GEAMAI/step01/case00*/posi.dat～

0 FCC_vacancy 499

```
13  2.0250000000    2.0250000000    0.0000000000 0 0 0
13  2.0250000000    0.0000000000    2.0250000000 0 0 0
13  0.0000000000    2.0250000000    2.0250000000 0 0 0
13  0.0000000000    0.0000000000    4.0500000000 0 0 0
13  2.0250000000    2.0250000000    4.0500000000 0 0 0
13  2.0250000000    0.0000000000    6.0750000000 0 0 0
13  0.0000000000    2.0250000000    6.0750000000 0 0 0
13  0.0000000000    0.0000000000    8.1000000000 0 0 0
13  2.0250000000    2.0250000000    8.1000000000 0 0 0
```

～後略～

11.4 適応度関数の設定

面心立方格子が再安定構造になることを最優先にしたい。六方最密格子の凝集エネルギー、体心立方格子の凝集エネルギーにくらべ、面心立方格子の凝集エネルギーが十分に低くなるようなペナルティ設定を行う。六方最密格子の凝集エネルギーと面心立方格子の凝集エネルギーの差が 0.025 eV/atom 以内（第一原理計算ではおよそ 0.033 eV/atom 差）、体心立方格子の凝集エネルギーと面心立方格子の凝集エネルギーの差が 0.050 eV/atom （第一原理計算ではおよそ 0.1 eV/atom 差）以内の場合大きなペナルティを与えた。

～fsingle.h 前略～

```
if(fatomn1==13){
  if(crystalEnergy[HCP]<(crystalEnergy[FCC]+0.025)){
    diff = (fabs(crystalEnergy[FCC]+0.025-crystalEnergy[HCP]))*3000.0;
    dv_penalty += diff*diff+1000000000.0;
  }
  if(crystalEnergy[BCC]<(crystalEnergy[FCC]+0.050)){
    diff = (fabs(crystalEnergy[FCC]+0.05-crystalEnergy[BCC]))*3000.0;
    dv_penalty += diff*diff+1000000000.0;
  }
  if(crystalEnergy[SC]<crystalEnergy[BCC]){
```

```

diff = (fabs(crystalEnergy[SC]-crystalEnergy[BCC]))*1000.0;
dv_penalty += diff*diff+1000000000.0;
}
if(crystalEnergy[DIAMOND]<crystalEnergy[SC]){
diff = (fabs(crystalEnergy[DIAMOND]-crystalEnergy[SC]))*1000.0;
dv_penalty += diff*diff+1000000000.0;
}

```

～fsingle.h 後略～

11.5 プログラムのチェック

ディレクトリ GEAMAI/check/Cu を用いて動作確認を行う。AI はオリジナルポテンシャルの値がおかしなことになっているため、チェックには使えない。このディレクトリの para.dat の上限値、下限値はオリジナルの GEAM ポテンシャルパラメータの Cu の値となっているため、そのときの物性値を確認することができる。しかし、関数形の説明でも述べたように本例題においてはオリジナルと異なり、埋め込み関数 $F(\rho)$ の連続性を 2 次まで確保している（オリジナルは下側接続点 $\rho = 0.85 \rho_e$ では 2 次まで確保しているのだが、上側接続点 $\rho = 1.15 \rho_e$ で 1 次までしか確保していない）関数形であるため、2 階微分値がオリジナルの GEAM ポテンシャルからは変化し、弾性的性質が再現できないことが予測される。前節まで変更を加えたので、以下に make をしてから実行する。

%make

%/GA GEAMAI/check/Cu

下に実行した結果の一部を示す。（fsingle.h でデフォルトで指定しているように）カッコ内が合わせ込む値（実験値・第一原理計算）、カッコ外が GEAM ポテンシャルパラメータを用いて計算した結果である。ダイヤモンド構造、正方格子（イマイチか）、体心立方格子、面心立方格子の格子定数、エネルギーは再現できているが、弾性的性質は再現できていない。（つまり、2 次までの連続性を上側接続点で確保すると F_2 などのポテンシャルパラメータの値がオリジナルのポテンシャルパラメータからは変わってしまうということである。）

～GEAMAI/check/Cu/stdout.dat 前略～

step=[10/10]

```

CRYSTAL:    DIAMOND  ENE=-2.5375 (-2.5134)  LT=5.2924 (5.3426)
CRYSTAL:         SC  ENE=-3.2588 (-3.0779)  LT=2.4459 (2.3958)
CRYSTAL:         BCC  ENE=-3.5135 (-3.5052)  LT=2.9659 (2.8745)
CRYSTAL:         FCC  ENE=-3.5400 (-3.5400)  LT=3.6149 (3.6150)
CRYSTAL:    IDEALHCP  ENE=-3.5347 (-3.5312)  LT=2.5754 (2.5557)
CRYSTAL:         HCP  ENE=-3.5377 (-3.5312)  LTA=2.5504 (2.5557) LTC=4.2760 (4.1900)
ACRATIO=1.6766 (1.6395)
BM[         FCC]=100.120(138.300)
C11=131.773(170.000) C12=84.282(122.500)
C44=75.746(75.800)

```

～後略～

そこでオリジナルのポテンシャルパラメータの有効性を確認する為、fsingle.h において、

```
ci->myPotentialParameter()->setAuthorizedParameter(fatomn1);
```

のコメントを外して実行する。fatomn1 は Cu の場合原子番号である 29 である。これによってプログラム中でオリジナルの Cu（原子番号 29）のポテンシャルパラメータそのものがセットされる。setAuthorizedParameter は文献値のポテンシャルパラメータをセットするメソッドであり、他の元素、関数にも用意されているため確認

してほしい.

```
%make
```

```
%/GA GEAMAI/check/Cu
```

のように実行すると

```
step=[10/10]
```

```
CRYSTAL:    DIAMOND  ENE=-2.5992 (-2.5134)  LT=5.2248 (5.3426)
```

```
CRYSTAL:      SC  ENE=-3.1580 (-3.0779)  LT=2.3994 (2.3958)
```

```
CRYSTAL:      BCC  ENE=-3.4876 (-3.5052)  LT=2.8280 (2.8745)
```

```
CRYSTAL:      FCC  ENE=-3.5398 (-3.5400)  LT=3.6148 (3.6150)
```

```
CRYSTAL:  IDEALHCP  ENE=-3.5246 (-3.5312)  LT=2.5499 (2.5557)
```

```
CRYSTAL:      HCP  ENE=-3.5249 (-3.5312)  LTA=2.5582 (2.5557) LTC=4.1240 (4.1900)
```

```
ACRATIO=1.6121 (1.6395)
```

```
BM[      FCC]=138.216(138.300)
```

```
C11=180.414(170.000) C12=117.109(122.500)
```

```
C44=70.657(75.800)
```

のようになり、弾性的性質も十分再現できていることがわかる。このままではプログラムが合わせ込みに使えなくなってしまうので、このコメントアウトは元に戻しておく必要がある。

また、Ni でも確認を行う。この Ni パラメータは筆者らが開発したパラメータ (T. Kumagai et al., *Materials Transactions*, Vol.48 No.06 (2007) pp.1313-1321) であり、元々埋め込み関数 F の 2 階微分が上側接続点、下側接続点における連続性を確保するよう開発したパラメータである。

```
%/GA GEAMAI/check/Ni
```

下に実行した結果の一部を示す。(fsingle.h でデフォルトで指定しているように) カッコ内が合わせ込む値 (実験値・第一原理計算)、カッコ外が筆者らの開発した GEAM ポテンシャルパラメータを用いて計算した結果である。ダイヤモンド構造、正方格子、体心立方格子、面心立方格子の格子定数、エネルギー、面心立方格子の弾性定数は再現できている。

```
~GEAMAI/check/Ni/stdout.dat 前略~
```

```
step=[10/10]
```

```
CRYSTAL:    DIAMOND  ENE=-3.1869 (-3.2432)  LT=5.1710 (5.1005)
```

```
CRYSTAL:      SC  ENE=-3.9420 (-3.7476)  LT=2.3657 (2.3289)
```

```
CRYSTAL:      BCC  ENE=-4.3507 (-4.3449)  LT=2.8052 (2.8020)
```

```
CRYSTAL:      FCC  ENE=-4.4400 (-4.4400)  LT=3.5200 (3.5200)
```

```
CRYSTAL:  IDEALHCP  ENE=-4.4208 (-4.4171)  LT=2.4919 (2.4876)
```

```
CRYSTAL:      HCP  ENE=-4.4215 (-4.4171)  LTA=2.4837 (2.4876) LTC=4.1068 (4.0820)
```

```
ACRATIO=1.6535 (1.6409)
```

```
BM[      FCC]=179.397(180.400)
```

```
C11=243.716(246.500) C12=147.235(147.300)
```

```
C44=127.997(124.700)
```

```
~GEAMAI/check/Ni/stdout.dat 後略~
```

11.6 プログラムの実行(1回目)

8 ケースについて、プログラムを実行する。

```
./GA GEAMAI/step01/case001
./GA GEAMAI/step01/case002
./GA GEAMAI/step01/case003
./GA GEAMAI/step01/case004
./GA GEAMAI/step01/case005
./GA GEAMAI/step01/case006
./GA GEAMAI/step01/case007
./GA GEAMAI/step01/case008
```

筆者の手元にある計算機では12時間程度で計算が終わった。パラメータが多く(18)、制約条件も多い合わせ込みなので、多目の20000個体を設定したためである。

case001~case008の場合で、合わせ込みに用いた物性はある程度再現できた。しかし体心立方格子、六法最密格子の格子定数の再現性はいまひとつであった。(色々試したところ、弾性定数の再現性を少し犠牲にすれば、体心立方格子、六法最密格子の再現性を高めることができるようである)例えば中間程度の F_0 を持つcase004の場合、標準出力は以下のようなものである。面心立方格子の凝集エネルギー、格子定数が完全には合っていないが、これはGEAMポテンシャルの場合はポテンシャルパラメータ r_e が固定されているため、一般的なスケーリング手法を用いることができず、特殊なスケーリング方法を用いているからである。

~GEAMAI/step01/case004/stdout.dat 終端部~

step=[31/31]

```
CRYSTAL:      DIMER  ENE=-0.4463 (-0.7750)  LT=2.4251 (2.5600)
CRYSTAL:      CHAIN  ENE=-1.0171 (-1.5070)  LT=2.3201 (2.4230)
CRYSTAL:      GRAPHITE  ENE=-2.0330 (-2.1721)  LT=4.0188 (4.4883)
CRYSTAL:      DIAMOND  ENE=-2.9698 (-2.6135)  LT=5.7018 (6.0601)
CRYSTAL:      SC  ENE=-3.1589 (-2.9880)  LT=2.6908 (2.7273)
CRYSTAL:      BCC  ENE=-3.3032 (-3.2631)  LT=3.0895 (3.2428)
CRYSTAL:      FCC  ENE=-3.3569 (-3.3600)  LT=4.0433 (4.0500)
CRYSTAL:      IDEALHCP  ENE=-3.3295 (-3.3269)  LT=2.8470 (2.8643)
CRYSTAL:      HCP  ENE=-3.3300 (-3.3269)  LTA=2.8249 (2.8643) LTC=4.8487 (4.7166)
```

ACRATIO=1.7164 (1.6467)

BM[FCC]=80.022(79.000)

C11=108.893(114.000) C12=65.592(61.900)

C44=41.324(31.600)

NONCRYST: FCC_vacancy ENE=0.7071 (0.6800) MAX_F=0.1863 (1[13])

~GEAMAI/step01/case004/stdout.dat 後略~

11.7 第2ステップの入力ファイル

第1ステップでおおよそ合わせ込みは終わっているので、緩和後の空孔生成エネルギーを確認しながら、調整程度の収束を行う。

11.7.1 cont.dat

第2ステップの個体数(1000)よりも第1ステップでgene.datに出力した個体数が多いのでcont.datのために個

体を抜き出さなくてはならない。今回は第1ステップでトータルで1000個体($n_{out}-1$) \times 7回($n_{int}/n_{step}+1=7$ だから)=7000個体出力されているはずなので、gene.dat から GA の最後の1000個体を抜き出して cont.dat とする。パラメータ数が18でコメント行が1行あるので、1個体について gene.dat に19行記述されることになる。1回で1000個体出力されるから、1回19000行(19行 \times 1000個体)の出力ということになる。

よって

```
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case001/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case002/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case003/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case004/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case005/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case006/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case007/cont.dat
%tail -n 19000 GEAMAI/step01/case001/gene.dat >GEAMAI/step02/case008/cont.dat
```

のように cont.dat を作成する。

11.8 プログラムの実行(2回目)

8 ケースについて、プログラムを実行する。

```
./GA GEAMAI/step02/case001
./GA GEAMAI/step02/case002
./GA GEAMAI/step02/case003
./GA GEAMAI/step02/case004
./GA GEAMAI/step02/case005
./GA GEAMAI/step02/case006
./GA GEAMAI/step02/case007
./GA GEAMAI/step02/case008
```

空孔生成エネルギー以外の物性はすべてのケース(case001-case008)でおおよそ再現することができたが、空孔生成エネルギーは F_0 に強く依存するようである。空孔生成エネルギーの再現性に最も優れていたのは case004 であった。

～GEAMAI/step02/case004/stdout.dat 終端部～

```
step=[51/51]
CRYSTAL:    DIMER  ENE=-0.4652 (-0.7750)  LT=2.3906 (2.5600)
CRYSTAL:    CHAIN  ENE=-1.0509 (-1.5070)  LT=2.2982 (2.4230)
CRYSTAL:    GRAPHITE  ENE=-2.0706 (-2.1721)  LT=4.0033 (4.4883)
CRYSTAL:    DIAMOND  ENE=-2.9922 (-2.6135)  LT=5.6995 (6.0601)
CRYSTAL:    SC  ENE=-3.1721 (-2.9880)  LT=2.7233 (2.7273)
CRYSTAL:    BCC  ENE=-3.3008 (-3.2631)  LT=3.0794 (3.2428)
CRYSTAL:    FCC  ENE=-3.3567 (-3.3600)  LT=4.0432 (4.0500)
CRYSTAL:    IDEALHCP  ENE=-3.3290 (-3.3269)  LT=2.8476 (2.8643)
CRYSTAL:    HCP  ENE=-3.3297 (-3.3269)  LTA=2.8240 (2.8643) LTC=4.8609 (4.7166)
ACRATIO=1.7213 (1.6467)
BM[    FCC]=79.031(79.000)
```

C11=112.341(114.000) C12=62.382(61.900)
C44=40.937(31.600)
NONCRYST: FCC_vacancy ENE=0.6335 (0.6800) MAX_F=0.0055 (1[13])
～GEAMAI/step02/case004/stdout.dat 後略～

得られたポテンシャルパラメータは poten.dat に出力されている.

～GEAMAI/step02/case004/poten.dat～

```
re 2.86392400000000e+00
rhoe/fe 1.96779168366825e+01
rhos/e 1.00000000000000e+00
alpha 6.66410425426956e+00
beta1 1.96738924452616e+00
beta2/beta1 1.00000000000000e+00
A 1.18085207213078e-01
B 1.36699284153982e-01
lambda 8.23279475886340e-01
kappa/lambda 4.85973520954148e-01
nkappa 2.00000000000000e+01
nlambda1 2.00000000000000e+01
f0 -2.73000000000000e+00
f1 0.00000000000000e+00
eta 7.25427767701295e-01
rhon 8.50000000000000e-01
rhoo 1.15000000000000e+00
Rcut 7.50000000000000e+00
```

しかし, GEAM ポテンシャルを古典分子動力学計算に使うにあたっては従属ポテンシャルパラメータ ($F_{n0}, F_{n1}, F_{n2}, F_{n3}, F_2, F_3, Fe$) を入力する必要がある. GEAM ポテンシャルの場合に限りこれらを含む全ポテンシャルパラメータ allpoten.dat が出力される.

～GEAMAI/step02/case004/allpoten.dat～

```
re:          2.86392400000000
fe:          1.00000000000000
rho_e:       19.67791683668250
rho_s:       19.67791683668250
alpha:       6.66410425426956
beta1:       1.96738924452616
beta2:       1.96738924452616
A:           0.11808520721308
B:           0.13669928415398
kappa:       0.40009202562577
lambda1:     0.82327947588634
lambda2:     0.82327947588634
```

nkappa:	20.000000000000000
nlambda1:	20.000000000000000
nlambda2:	20.000000000000000
Fn0:	-2.71303062834796
Fn1:	-0.19793404669991
Fn2:	0.60853639345276
Fn3:	-1.90656018819529
F0:	-2.730000000000000
F1:	0.000000000000000
F2:	0.71015899125270
F3:	-0.29356869595595
eta:	0.72542776770130
Fe:	-2.73002916034213
rhon:	0.850000000000000
rhoo:	1.150000000000000

12. 例題 Ni 単元系のための Morse ポテンシャルの開発

Ni 単元系用 Morse ポテンシャルの開発を例題として行う。なお本節で行われる例題はディレクトリ MorseNi に含まれる。

12.1 ポテンシャル関数形

関数形は以下のようである (F. Milstein, J. Appl. Phys. 44 (1973), 3825) .

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi_{ij}$$

$$\phi_{ij} = f_c(r_{ij}) \frac{D}{m-1} \left[\exp\{-mA(r_{ij} - r_0)\} - m \exp\{-A(r_{ij} - r_0)\} \right]$$

$$f_c(r) = \begin{cases} 1 & r \leq R \\ 0 & r > R \end{cases}$$

$m=2$ であることが多い(例えば L.A. Girifalco, and V. G. Weizer, Phys. Rev., 114 (1959), 687), 合わせ込みに用いることのできるパラメータは m, D, A, r_0, R の 5 つである。

12.2 ヘッダの確認

12.2.1 合わせ込み条件ヘッダ condNi.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。実質的なポテンシャルパラメータが少ないため、面心立方格子のエネルギー、格子定数、体積弾性率のみを合わせ込みに用いるが、2 体ポテンシャルであるため、計算時間はかからないので、確認用にそれ以外の基本的な結晶構造の物性もウオッチすることにする(重みはゼロにしながらか合わせ込みに用いる)。以下は一部抜粋である。

```
void setFlagsOfSingleCrystalsNi(int *isUsedForFitting,int *isCalcBulkModulus,
                                int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
//In setting
isUsedForFitting[DIMER] = 0;
isUsedForFitting[CHAIN] = 0;
isUsedForFitting[GRAPHITE] = 0;
isUsedForFitting[DIAMOND] = 1;
isUsedForFitting[SC] = 1;
isUsedForFitting[BCC] = 1;
isUsedForFitting[BCT] = 0;
isUsedForFitting[IDEALHCP] = 0;
isUsedForFitting[HCP] = 1;
if(isUsedForFitting[HCP]==1){
    isUsedForFitting[IDEALHCP] = 1;
}
isUsedForFitting[FCC] = 1;
~中略~
isCalcBulkModulus[FCC] = 1;
isCalcTensileElasticConstants[FCC] = 1;
```

```
isCalcShearElasticConstants[FCC] = 1;
}
```

重みを設定する関数

```
void setWeightOfSingleCrystalsNi(引数略){
  ~略~
  weightForCohesiveEnergy[FCC] = 100.0;
  ~略~
  weightForLatticeConstant[FCC] = 100.0;
  ~略~
  weightForBulkModulus[FCC] = 100.0;
}
```

12.2.2 物性の入力されているヘッダ matpropNi.h の確認

合わせ込みに用いる物性が設定されているか確認する.

～matpropNi.h 冒頭部～

```
void setMaterialsPropertiesOfNickel(double* latticeConstantA,
                                   double* latticeConstantB,
                                   double* latticeConstantC,
                                   double* internalParameterX1,
                                   double* internalParameterX2,
                                   double* internalParameterX3,
                                   double* cohesiveEnergy,
                                   double* bulkModulus,
                                   double* C11,double* C22,double* C33,
                                   double* C12,double* C13,double* C23,
                                   double* C44,double* C55,double* C66){
```

```
cohesiveEnergy[DIMER]=0.0;
cohesiveEnergy[CHAIN]=0.0;
cohesiveEnergy[GRAPHITE]=0.0 ;
cohesiveEnergy[DIAMOND]=-3.2432493884*8.0;
cohesiveEnergy[SC]=-3.7475982500*1.0;
cohesiveEnergy[BCC]=-4.3448922500*2.0;
cohesiveEnergy[FCC]=-4.4400000000*4.0;
cohesiveEnergy[IDEALHCP]=-4.4171177500*4.0;
cohesiveEnergy[HCP]=-4.4171177500*4.0;
```

```
latticeConstantA[DIMER]    = 2.5 ;
latticeConstantA[CHAIN]    = 2.5 ;
latticeConstantA[GRAPHITE] = 4.4882601146 ;
latticeConstantA[DIAMOND]=5.100465827;
```

```
latticeConstantA[SC]=2.328891894;
latticeConstantA[BCC]=2.802010579;
latticeConstantA[FCC]=3.52;
latticeConstantA[IDEALHCP]=2.487622728;
latticeConstantA[HCP]=2.487622728;
double acratio=1.640923292;
latticeConstantC[HCP]=latticeConstantA[HCP]*acratio;
```

～中略～

```
bulkModulus[FCC] = 180.4;
```

```
C11[FCC] = 246.5;
```

```
C12[FCC] = 147.3;
```

```
C44[FCC] = 124.7;
```

```
}
```

12.3 入力ファイル

ディレクトリ MorseNi/fit/中の入力ファイルを設定する

12.3.1 element.dat

単元系 Ni 用の原子間ポテンシャルなので element.dat を以下のように設定する

～MorseNi/fit/element.dat～

```
1
```

```
Ni
```

12.3.2 cond.dat

cond.dat を以下のように設定する

～MorseNi/fit/cond.dat～

```
id:      5      : Morse ポテンシャル
iscont:  0      : 新規スタート
nstep:   31     : ステップ数 31
nint:    10     : 10 ステップごとに出力
noi:     1000   : 1000 個体
nout:    101   : トップ 100 個体の情報を出力
nmut:    0      : すべての FREE パラメータを突然変異
mutrate: 0.1    : 突然変異幅は 10%
seed:    1      : 乱数の seed は 1
scale:   1      : スケーリングを行う
```

12.3.3 para.dat

para.dat を示す. カットオフ距離は十分に長くして(平衡結合長×3 以上), 一般的な値である $m=2$ と固定した. このため自由なパラメータは D, A, r_0 の 3 つだけであり, 格子定数と凝集エネルギーについてスケーリングも行うから, 実質的な自由度は 1 である.

～MorseNi/fit/para.dat～

0	FREE	D_AA	0.0100	1.0000	: D
1	FREE	A_AA	1.0000	2.0000	: A
2	FREE	R0_AA	1.0000	4.0000	: r0
3	FIX	M_AA	2.0000	2.0000	: m
4	FIX	CUT_AA	7.5000	7.5000	: R
5	FIX	RCUT	7.7000	7.7000	

12.3.4 set.dat

実質的な合わせ込みの自由度が1であり、非結晶系の合わせ込みはほとんど不可能であることから、合わせ込みに用いない。

～MorseNi/fit/set.dat～

0

12.4 適応度関数の設定

実質的な自由度が1であり、通常の適応度を用いた合わせ込み以上のことができるとは考えられないので、特に設定を行わない。

12.5 プログラムのチェック

ディレクトリ Morse/check/Ni/を用いて動作確認を行う。このディレクトリの para.dat の上限値、下限値は文献にあるポテンシャルパラメータの Ni の値となっているため、そのときの物性値を確認することができる。(L.A. Girifalco, and V. G. Weizer, Phys. Rev., 114 (1959), 687).

%/GA MorseNi/

下に実行した結果の一部を示す。カッコ内が合わせ込む値(実験値・第一原理計算)、カッコ外が Morse ポテンシャルパラメータを用いて計算した結果である。正方格子、体心立方格子、面心立方格子の格子定数、凝集エネルギー、最安定構造である面心立方格子の体積弾性率、弾性定数についてそれなりには再現できている。厳密には第一原理計算や実験値とは差が見られるが、これはパラメータの開発時(1959)の合わせこんだ物性が現在とは異なるためであろうと思われる。実際市販のソフトウェア(富士通 ME4)を使って確認したところ同一の物性がえら得られた。

～MorseNi/check/Ni/stdout.dat 前略～

step=[11/11]

CRYSTAL: DIAMOND ENE=-1.8383 (-3.2432) LT=5.4151 (5.1005)

CRYSTAL: SC ENE=-3.3247 (-3.7476) LT=2.3362 (2.3289)

CRYSTAL: BCC ENE=-4.2005 (-4.3449) LT=2.8032 (2.8020)

CRYSTAL: FCC ENE=-4.2289 (-4.4400) LT=3.5298 (3.5200)

CRYSTAL: IDEALHCP ENE=-4.2215 (-4.4171) LT=2.4978 (2.4876)

CRYSTAL: HCP ENE=-4.2215 (-4.4171) LTA=2.4979 (2.4876) LTC=4.0788 (4.0820)

ACRATIO=1.6329 (1.6409)

BM[FCC]=186.644(180.400)

C11=234.696(246.500) C12=162.618(147.300)

C44=162.618(124.700)

12.6 プログラムの実行

プログラムを実行する.

```
%/GA MorseNi/fit/
```

おおよそ合わせ込みに用いた物性はある程度再現できた.

～MorseNi/fit/stdout.dat 終端部～

```
step=[31/31]
```

```
CRYSTAL:      DIMER  ENE=-0.4463 (-0.7750)  LT=2.4251 (2.5600)
```

```
CRYSTAL:      CHAIN  ENE=-1.0171 (-1.5070)  LT=2.3201 (2.4230)
```

```
CRYSTAL:      GRAPHITE  ENE=-2.0330 (-2.1721)  LT=4.0188 (4.4883)
```

```
CRYSTAL:      DIAMOND  ENE=-2.9698 (-2.6135)  LT=5.7018 (6.0601)
```

```
CRYSTAL:      SC  ENE=-3.1589 (-2.9880)  LT=2.6908 (2.7273)
```

```
CRYSTAL:      BCC  ENE=-3.3032 (-3.2631)  LT=3.0895 (3.2428)
```

```
CRYSTAL:      FCC  ENE=-3.3569 (-3.3600)  LT=4.0433 (4.0500)
```

```
CRYSTAL:      IDEALHCP  ENE=-3.3295 (-3.3269)  LT=2.8470 (2.8643)
```

```
CRYSTAL:      HCP  ENE=-3.3300 (-3.3269)  LTA=2.8249 (2.8643) LTC=4.8487 (4.7166)
```

```
ACRATIO=1.7164 (1.6467)
```

```
BM[      FCC]=80.022(79.000)
```

```
C11=108.893(114.000) C12=65.592(61.900)
```

```
C44=41.324(31.600)
```

```
NONCRYST: FCC_vacancy  ENE=0.7071 (0.6800)  MAX_F=0.1863 ( 1[13])
```

～MorseNi/fit/stdout.dat/stdout.dat 後略～

13. 例題 仮想 FCC 単元系のための Lennard-Jones ポテンシャルの開発

仮想 FCC 単元系用 Lennard-Jones ポテンシャルの開発を例題として行う。なお本節で行われる例題はディレクトリ LJIF に含まれる。ここで、仮想 FCC 単元系とは平衡原子間距離が 1Å, そのときの凝集エネルギーが -1eV/atom の系を意味する

13.1 ポテンシャル関数形

関数形は以下のようなものである。

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi_{ij}$$

$$\phi_{ij} = f_c(r_{ij}) A \left[\left(\frac{B}{r_{ij}} \right)^{N_R} - \left(\frac{B}{r_{ij}} \right)^{N_A} \right]$$

$$f_c(r) = \begin{cases} 1 & r \leq R \\ 0 & r > R \end{cases}$$

気体であれば $N_R=12, N_A=6$, 固体であれば $N_R=8, N_A=4$ であることが多い。

13.2 ヘッダの確認

13.2.1 合わせ込み条件ヘッダ condIF.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。実質的なポテンシャルパラメータが少ないため、面心立方格子のエネルギー、格子定数、体積弾性率のみを合わせ込みに用いる。以下は一部抜粋である。

```
void setFlagsOfSingleCrystalsIdealFCC(int *isUsedForFitting,int *isCalcBulkModulus,
                                     int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
    //In setting
    ~前略~
    isUsedForFitting[FCC] = 1;
    ~中略~
    isCalcBulkModulus[FCC] = 1;
    isCalcTensileElasticConstants[FCC] = 1;
    isCalcShearElasticConstants[FCC] = 1;
}
```

次節で示す設定では実質的な合わせ込みパラメータ数はゼロになるので重みはゼロで問題ない。

重みを設定する関数

```
void setWeightOfSingleCrystalsIdealFCC(引数略){
    ~略~
    weightForCohesiveEnergy[FCC] = 0.0;
    ~略~
    weightForLatticeConstant[FCC] = 0.0;
    ~略~
    weightForBulkModulus[FCC] = 0.0;
```

```
}
```

13.2.2 物性の入力されているヘッダ `matpropIF.h` の確認

合わせ込みに用いる物性が設定されているか確認する。

～`matpropIF.h` 冒頭部～

```
void setMaterialsPropertiesOfIdealFCC(double* latticeConstantA,  
                                     double* latticeConstantB,  
                                     double* latticeConstantC,  
                                     double* internalParameterX1,  
                                     double* internalParameterX2,  
                                     double* internalParameterX3,  
                                     double* cohesiveEnergy,  
                                     double* bulkModulus,  
                                     double* C11,double* C22,double* C33,  
                                     double* C12,double* C13,double* C23,  
                                     double* C44,double* C55,double* C66){
```

```
cohesiveEnergy[FCC]=-4.0;  
latticeConstantA[FCC]=sqrt(2.0);
```

```
bulkModulus[FCC] = 1.0;
```

```
C11[FCC] = 1.0;  
C12[FCC] = 1.0;  
C44[FCC] = 1.0;}
```

13.3 入力ファイル

ディレクトリ `LJIF/fit/` 中の入力ファイルを設定する

13.3.1 `element.dat`

仮想 FCC 単元系用の原子間ポテンシャルなので `element.dat` を以下のように設定する

～`LJIF/fit/element.dat`～

```
1  
IF
```

13.3.2 `cond.dat`

`cond.dat` を以下のように設定する

～`LJIF/fit/cond.dat`～

```
id:      6      : Lennard-Jones ポテンシャル  
iscont:  0      : 新規スタート  
nstep:   3      : ステップ数 3  
nint:    1      : 1 ステップごとに出力  
noi:     10     : 10 個体  
nout:    9      : トップ 10 個体の情報を出力
```

nmut: 0 :すべての FREE パラメータを突然変異
mutrate: 0.1 :突然変異幅は 10%
seed: 1 :乱数の seed は 1
scale: 1 :スケーリングを行う

13.3.3 para.dat

para.dat を示す. 固体であるので $N_R=8$, $N_A=4$ と固定した. このため自由なパラメータは A,B の 2 つだけであるが, 格子定数と凝集エネルギーについてスケーリングも行うから, 実質的な自由度は 0 である.

~LJIF/fit/para.dat~

0	FIX	A_AA	0.1000000000000000	0.3000000000000000	:A
1	FIX	B_AA	0.8000000000000000	1.0000000000000000	:B
2	FIX	NR_AA	8.0000000000000000	8.0000000000000000	:N _R
3	FIX	NA_AA	4.0000000000000000	4.0000000000000000	:N _A
3	FIX	CUT_AA	3.0550000000000000	3.0550000000000000	:R
4	FIX	RCUT	3.1000000000000000	3.1000000000000000	

13.3.4 set.dat

実質的な合わせ込みの自由度が 0 であり, 合わせ込みに用いない.

~LJIF/fit/set.dat~

0

13.4 適応度関数の設定

実質的な自由度が 0 であるため, 特に設定を行わない.

13.5 プログラムの実行

プログラムを実行する.

%.GA LJIF/fit/

スケーリングしているため, 当然格子定数と凝集エネルギーは再現できている. 弾性率を変えたい場合は, N_R , N_A を変更すると良い.

~LJIF/fit/stdout.dat 終端部~

step=[3/3]

CRYSTAL: FCC ENE=-1.0000 (-1.0000) LT=1.4142 (1.4142)

BM[FCC]=805.625(1.000)

C11=1089.773(1.000) C12=663.551(1.000)

C44=663.551(1.000)

~LJIF/fit/stdout.dat/stdout.dat 後略~

14. 例題 Al-N 2 元系のための Tersoff ポテンシャルの開発

2 元系 (2 種類の元素を含む) ポテンシャルの例として化合物半導体である Al-N 用 Tersoff ポテンシャルの開発を例題として行う. なお本節で行われる例題はディレクトリ `tersoffAlN` に含まれる. 単元系ポテンシャル (Al 単元系, N 単元系) パラメータの決定手法は Si の場合とほぼ同じなので, 既に開発したものをを用いることにする.

14.1 ポテンシャル関数形

基本的には単元系ポテンシャル関数と同じ定式化であるが, ポテンシャルパラメータが元素によって異なる. このため, 合わせ込むポテンシャルパラメータの数が多くなることと一部のポテンシャルパラメータは対象性から同じ値をとること (para.dat 解説参照) に注意したい. ここで a_i, a_j, a_k はそれぞれ原子 i, j, k の原子種を示している.

$$\Phi = \frac{1}{2} \sum_{i \neq j} \phi_{ij}$$

$$\phi_{ij} = f_{c,a_i,a_j}(r_{ij}) \left[A_{a_i,a_j} \exp(-\lambda_{1,a_i,a_j} r_{ij}) - b_{ij} B \exp(-\lambda_{2,a_i,a_j} r_{ij}) \right]$$

$$b_{ij} = \left(1 + \zeta_{ij}^{\eta_{a_i,a_j}} \right)^{-\delta_{a_i,a_j}}$$

$$\zeta_{ij} = \sum_{k \neq i,j} f_c(r_{ik}) g_{a_i,a_j,a_k}(\theta_{ijk}) \exp \left[p_{a_i,a_j,a_k} \left\{ (r_{ij} - \text{Re}_{a_i,a_j}) - (r_{ik} - \text{Re}_{a_i,a_k}) \right\}^{q_{a_i,a_j,a_k}} \right]$$

$$g_{abc}(\theta) = c_{1,abc} + \frac{c_{2,abc} (h_{abc} - \cos \theta)^2}{c_{3,abc}^2 + (h_{abc} - \cos \theta)^2}$$

$$f_{c,ab}(r) = \begin{cases} 1 & r < R_{1,ab} \\ \frac{1}{2} + \frac{1}{2} \cos \left(\pi \frac{r - R_{1,ab}}{R_{2,ab} - R_{1,ab}} \right) & R_{1,ab} \leq r \leq R_{2,ab} \\ 0 & r > R_{2,ab} \end{cases}$$

14.2 単元系ポテンシャルを用いて計算される物性の確認と単元系ポテンシャルファイルの生成

既に開発済みのポテンシャルパラメータを用いて計算される物性の確認と単元系ポテンシャルファイルの生成を行う. 生成用ディレクトリの `para.dat` におけるパラメータの上限値, 下限値は文献値(熊谷ら, 日本機械学会年次大会講演論文集 2006(1) pp.915-916)に使用したのと同じものである.

Al の場合

```
./GA tersoffAlN/check/Al
```

N の場合

```
./GA tersoffAlN/check/N
```

として実行する.

結果, 合わせ込みに用いた値がほぼ再現できていることがわかる.

~tersoffAlN/check/Al/stdout.dat 前略~

```
step=[11/11]
```

```
CRYSTAL:      DIMER  ENE=-0.7919 (-0.7750)  LT=2.5264 (2.5600)
CRYSTAL:      CHAIN  ENE=-1.5066 (-1.5070)  LT=2.5427 (2.4230)
CRYSTAL:      GRAPHITE  ENE=-2.1618 (-2.1620)  LT=4.4293 (4.4912)
CRYSTAL:      DIAMOND  ENE=-2.6602 (-2.6600)  LT=5.9664 (6.0645)
```

CRYSTAL: SC ENE=-3.0665 (-3.0670) LT=2.6697 (2.7310)
CRYSTAL: BCC ENE=-3.2343 (-3.2340) LT=3.1713 (3.2355)
CRYSTAL: FCC ENE=-3.3900 (-3.3900) LT=4.0500 (4.0500)
BM[FCC]=81.481(79.000)
C11=108.250(114.000) C12=68.093(61.900)
C44=38.347(31.600)

～後略～

～tersoffAlN/check/N/stdout.dat 前略～

step=[11/11]

CRYSTAL: DIMER ENE=-4.8868 (-4.9500) LT=1.4216 (1.1100)
CRYSTAL: CHAIN ENE=-2.7323 (-2.6000) LT=1.6279 (1.3010)
CRYSTAL: GRAPHITE ENE=-1.9445 (-2.2520) LT=3.0287 (2.6154)
CRYSTAL: DIAMOND ENE=-1.7935 (-1.7940) LT=4.1760 (4.1754)
CRYSTAL: SC ENE=-1.6942 (-1.7400) LT=1.8831 (1.8550)
CRYSTAL: BCC ENE=-0.7297 (-0.6320) LT=2.4955 (2.4353)
CRYSTAL: FCC ENE=-0.7316 (-0.3980) LT=3.4804 (3.1084)
BM[DIAMOND]=156.994(157.600)

～後略～

また、以下のように poten.dat も生成された。

～tersoffAlN/check/Al/poten.dat～

A_AA 1.47567283465541e+03
B_AA 2.14838444381231e+01
L1_AA 3.05505224005580e+00
L2_AA 8.94910022312490e-01
ETA_AA 1.83373209955786e+00
DT*ET_AA 6.00000000000000e-01
p_AAA 7.73757438773590e-01
q_AAA 1.00000000000000e+00
c1_AAA 2.17685805762020e-01
c2_AAA 2.10434488983874e+05
c3_AAA 1.00000000000000e+06
h_AAA -3.53100147677060e-01
R1_AA 2.88000000000000e+00
R2-R1_AA 2.00000000000000e-01
Re_AA 2.35100000000000e+00
Rcut 3.30000000000000e+00

～tersoffAlN/check/N/poten.dat～

A_AA 3.55979843143666e+04
B_AA 3.55744587448859e+02
L1_AA 6.17711607665977e+00
L2_AA 2.21600874463167e+00
ETA_AA 1.76423382293285e+01

```

DT*ET_AA 1.29465189293814e+00
  p_AAA 0.000000000000000e+00
  q_AAA 1.000000000000000e+00
  c1_AAA 4.78431085162600e-01
  c2_AAA 1.92968248046579e+00
  c3_AAA 4.92961944075370e-01
  h_AAA 1.43879676296430e-01
  R1_AA 2.300000000000000e+00
R2-R1_AA 2.000000000000000e-01
  Re_AA 2.351000000000000e+00
  Rcut 2.800000000000000e+00

```

これらを element.dat で指定した名前(potenAl.dat, potenN.dat)にリネームしてコピーする.

```

%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step01/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step02/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step03/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step04/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step05/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step06/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step07/potenAl.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step08/potenAl.dat

```

```

%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step01/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step02/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step03/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step04/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step05/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step06/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step07/potenN.dat
%cp tersoffAlN/check/Al/poten.dat tersoffAlN/check/AlN/case001/step08/potenN.dat

```

14.3 ヘッダの確認

14.3.1 合わせ込み条件ヘッダ condAlN.h の確認

合わせ込みに使う物性の設定をしている関数を確認する. 一部抜粋になっているが, 一部の基本的なバルク 2 元系結晶のエネルギーと格子定数, 安定構造の閃亜鉛鉱構造(ZB)の体積弾性率, 弾性定数を合わせ込みに用いる設定になっている. これらの 2 元系バルク結晶の詳細については文献(熊谷知久, 東京大学博士論文, 2007)参照.

```

void setFlagsOfBinaryCrystalsAluminumNitride(int *isUsedForFitting,int *isCalcBulkModulus, int
*isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
  isUsedForFitting[ABGRAPHITE] = 1;
  isUsedForFitting[ABDIAMOND] = 1;
  isUsedForFitting[ABSC] = 1;

```

```

isUsedForFitting[ABBCC]      = 1;

isUsedForFitting[HETERODIMER] = 1;
isUsedForFitting[HETEROCHAIN] = 1;
isUsedForFitting[HETEROGRA]   = 1;
isUsedForFitting[ZB]          = 1;

isUsedForFitting[SC_A7B1] = 1;
isUsedForFitting[SC_A6B2] = 1;
isUsedForFitting[NACL]    = 1;
isUsedForFitting[SC_A2B6] = 1;
isUsedForFitting[SC_A1B7] = 1;

isUsedForFitting[BCC_A14B2] = 1;
isUsedForFitting[BCC_A12B4] = 1;
isUsedForFitting[BCC_A10B6] = 1;
isUsedForFitting[BCC_A6B10] = 1;
isUsedForFitting[BCC_A4B12] = 1;
isUsedForFitting[BCC_A2B14] = 1;

isCalcTensileElasticConstants[ZB] = 1;
}

重みを設定する関数は以下のものである。
void setWeightOfBinaryCrystalsAIN(double *weightForCohesiveEnergy,double *weightForLatticeConstant,
double *weightForBulkModulus,double *weightForTensileElasticConstants,double
*weightForShearElasticConstants){

weightForCohesiveEnergy[ABGRAPHITE] = 10.0;
weightForCohesiveEnergy[ABDIAMOND]  = 10.0;
weightForCohesiveEnergy[ABSC]       = 10.0;
weightForCohesiveEnergy[ABBCC]      = 10.0;

weightForCohesiveEnergy[HETERODIMER] = 5.0;
weightForCohesiveEnergy[HETEROCHAIN] = 5.0;
weightForCohesiveEnergy[HETEROGRA]   = 20.0;
weightForCohesiveEnergy[ZB]          = 100.0;

weightForCohesiveEnergy[SC_A7B1] = 20.0;
weightForCohesiveEnergy[SC_A6B2] = 20.0;
weightForCohesiveEnergy[NACL]    = 20.0;
weightForCohesiveEnergy[SC_A2B6] = 20.0;

```

```
weightForCohesiveEnergy[SC_A1B7] = 20.0;
```

```
weightForCohesiveEnergy[BCC_A14B2] = 15.0;  
weightForCohesiveEnergy[BCC_A12B4] = 15.0;  
weightForCohesiveEnergy[BCC_A10B6] = 15.0;  
weightForCohesiveEnergy[BCC_A6B10] = 15.0;  
weightForCohesiveEnergy[BCC_A4B12] = 15.0;  
weightForCohesiveEnergy[BCC_A2B14] = 15.0;
```

```
weightForLatticeConstant[ABGRAPHITE] = 0.0;  
weightForLatticeConstant[ABDIAMOND] = 0.0;  
weightForLatticeConstant[ABSC] = 0.0;  
weightForLatticeConstant[ABBCC] = 0.0;  
weightForLatticeConstant[ABFCC] = 0.0;
```

```
weightForLatticeConstant[HETERODIMER] = 0.0;  
weightForLatticeConstant[HETEROCHAIN] = 0.0;  
weightForLatticeConstant[HETEROGRA] = 30.0;  
weightForLatticeConstant[ZB] = 150.0;
```

```
weightForLatticeConstant[SC_A7B1] = 0.1;  
weightForLatticeConstant[SC_A6B2] = 0.1;  
weightForLatticeConstant[NACL] = 0.1;  
weightForLatticeConstant[SC_A2B6] = 0.1;  
weightForLatticeConstant[SC_A1B7] = 0.1;
```

```
weightForLatticeConstant[BCC_A14B2] = 3.0;  
weightForLatticeConstant[BCC_A12B4] = 3.0;  
weightForLatticeConstant[BCC_A10B6] = 3.0;  
weightForLatticeConstant[BCC_A6B10] = 1.0;  
weightForLatticeConstant[BCC_A4B12] = 3.0;  
weightForLatticeConstant[BCC_A2B14] = 10.0;
```

```
weightForBulkModulus[ZB] = 50.0;  
weightForTensileElasticConstants[ZB] = 30.0;  
weightForShearElasticConstants[ZB] = 30.0;
```

```
}
```

14.3.2 物性の入力されているヘッダ `matpropAlN.h` の確認

2 元系結晶の場合は絶対的なエネルギーを合わせ込むと、ずれが大きくなってしまいますので、生成エネルギーを合わせ込むほうが良い。このためには、生成エネルギーを計算するための基底エネルギーとしての単元系結晶エネルギーの入力が必要になる。第一原理計算に対する基底エネルギーは元素 1 種類目については `fpBasisEnergyByAtom1[i]`,

元素 2 種類目については `fpBasisEnergyByAtom2[i]`, 試行ポテンシャルに関する基底エネルギーは元素 1 種類目に関しては `basisEnergyByAtom1[i]`, 元素 2 種類目に関しては `basisEnergyByAtom2[i]` である. 試行ポテンシャルに関する基底エネルギーを入力するために, あらかじめ計算して引数としている `energyOfSymbol1[i]`, `energyOfSymbol2[i]` を使うことができる. 第一原理計算の基底エネルギーは自分で入力する必要がある.

～matpropAlN.h～

```
void setMaterialsPropertiesOfAluminumNitride
```

```
(double* latticeConstantA,double* latticeConstantB,double* latticeConstantC,  
double* internalParameterX1,double* internalParameterX2,double* internalParameterX3,  
double* cohesiveEnergy,double* bulkModulus,  
double* C11,double* C22,double* C33,  
double* C12,double* C13,double* C23,  
double* C44,double* C55,double* C66,  
double* energyOfSymbol1,double* energyOfSymbol2,  
double* basisEnergyByAtom1,double* basisEnergyByAtom2,  
double* fpBasisEnergyByAtom1,double* fpBasisEnergyByAtom2){
```

```
int i;
```

```
int fatomn1,fatomn2;
```

```
fatomn1 = 13;
```

```
fatomn2 = 7;
```

```
double fpEnergy1[MAXTYPE],fpEnergy2[MAXTYPE];
```

```
fpEnergy1[DIAMOND] = -2.660;
```

```
fpEnergy2[DIAMOND] = -1.794;
```

```
cohesiveEnergy[HETERODIMER]=-1.54*2.0000 ;
```

```
cohesiveEnergy[HETEROCHAIN]=-4.033*2.0000 ;
```

```
cohesiveEnergy[HETEROGRA]=-5.344*4.0000 ;
```

```
cohesiveEnergy[ZB]=-5.812*8.0000 ;
```

```
cohesiveEnergy[ABGRAPHITE]=-3.823*10.0000 ;
```

```
cohesiveEnergy[ABDIAMOND]=-3.478*24.0000 ;
```

```
cohesiveEnergy[ABSC]=-2.805*4.0000 ;
```

```
cohesiveEnergy[ABBCC]=-2.697*10.0000 ;
```

```
cohesiveEnergy[ABFCC]=-1.929*28.0000 ;
```

```
cohesiveEnergy[SC_A7B1]=-3.012*8.0000 ;
```

```
cohesiveEnergy[SC_A6B2]=-2.993*8.0000 ;
```

```
cohesiveEnergy[NAACL]=-5.632*8.0000 ;
```

```
cohesiveEnergy[SC_A2B6]=-3.652*8.0000 ;
```

```
cohesiveEnergy[SC_A1B7]=-2.608*8.0000 ;
```

```
cohesiveEnergy[BCC_A14B2]=-3.145*16.0000 ;
```

```

cohesiveEnergy[BCC_A12B4]=-2.895*16.0000 ;
cohesiveEnergy[BCC_A10B6]= -3.24*16.0000 ;
cohesiveEnergy[BCC_A6B10]=-3.209*16.0000 ;
cohesiveEnergy[BCC_A4B12]=-2.255*16.0000 ;
cohesiveEnergy[BCC_A2B14]=-1.558*16.0000 ;

latticeConstantA[HETERODIMER]=1.692;
latticeConstantA[HETEROCHAIN]=1.672*2.0000 ;
latticeConstantA[HETEROGRA]=1.791*sqrt(3.0);
latticeConstantA[ZB]=1.894*4.0/sqrt(3.0);
latticeConstantA[ABGRAPHITE]=1.757*sqrt(3.0)*2.0;
latticeConstantA[ABDIAMOND]=1.806*8.0/sqrt(3.0);
latticeConstantA[ABSC]=1.869*2.0000 ;
latticeConstantA[ABBCC]=1.955*4.0/sqrt(3.0);
latticeConstantA[ABFCC]=2.181*2.0*sqrt(2.0);
latticeConstantA[SC_A7B1]=2.598*2.0000 ;
latticeConstantA[SC_A6B2]=2.494*2.0000 ;
latticeConstantA[NACL]=2.025*2.0000 ;
latticeConstantA[SC_A2B6]=1.987*2.0000 ;
latticeConstantA[SC_A1B7]=1.943*2.0000 ;
latticeConstantA[BCC_A14B2]=2.711*4.0/sqrt(3.0);
latticeConstantA[BCC_A12B4]=2.662*4.0/sqrt(3.0);
latticeConstantA[BCC_A10B6]=2.466*4.0/sqrt(3.0);
latticeConstantA[BCC_A6B10]=2.156*4.0/sqrt(3.0);
latticeConstantA[BCC_A4B12]=2.147*4.0/sqrt(3.0);
latticeConstantA[BCC_A2B14]=2.099*4.0/sqrt(3.0);

for(i=STARTBINARY;i<=ENDBINARY;i++){
    fpBasisEnergyByAtom1[i]=fpEnergy1[DIAMOND];
    fpBasisEnergyByAtom2[i]=fpEnergy2[DIAMOND];
    basisEnergyByAtom1[i]=energyOfSymbol1[DIAMOND];
    basisEnergyByAtom2[i]=energyOfSymbol2[DIAMOND];
}

bulkModulus[ZB] = 196.0;

C11[ZB] = 306.7;
C12[ZB] = 155.7;
C44[ZB] = 201.1;
}

```

14.4 入力ファイル

14.4.1 element.dat

2 元系 Al-N 用の原子間ポテンシャルなので case00* の element.dat を以下のように設定する。単元系と異なり、ポテンシャルファイルの指定が必須である。

~TersoffAlN/step01/case00*/element.dat~

2

Al N

potenAl.dat potenN.dat

14.4.2 cond.dat

case001 の cond.dat を以下のように設定する。

~TersoffAlN/step01/case001/cond.dat~

id: 1 : Tersoff ポテンシャル
iscont: 0 : 新規スタート
nstep: 31 : ステップ数 31
nint: 5 : 5 ステップごとに出力
noi: 10000 : 5000 個体
nout: 1001 : トップ 1000 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 1 : スケーリングを行う

case002~case008 の乱数の seed は 2~8 とした。2 元系ポテンシャルであっても、Tersoff ポテンシャルはスケーリングが可能である。

14.4.3 para.dat

2 元系用の Tersoff ポテンシャルの para.dat は以下ようになる。対称性があることに注意が必要である。

0	FREE	A_AB	1.0000000000000000	10000.0000000000000000	: A _{ab} =A _{ba}
1	FREE	B_AB	1.0000000000000000	1000.0000000000000000	: B _{ab} =B _{ba}
2	FREE	L1_AB	1.0000000000000000	10.0000000000000000	: $\lambda_{1,ab} = \lambda_{1,ba}$
3	FREE	L2_AB	1.0000000000000000	10.0000000000000000	: $\lambda_{2,ab} = \lambda_{2,ba}$
4	FREE	ETB_AB	0.0000000000000000	20.0000000000000000	: η_{ab}
5	FREE	DT*ET_AB	0.0100000000000000	1.0000000000000000	: $\eta_{ab} \times \delta_{ab}$
6	FREE	ETB_BA	0.0000000000000000	20.0000000000000000	: η_{ba}
7	FREE	DT*ET_BA	0.0100000000000000	1.0000000000000000	: $\eta_{ba} \times \delta_{ba}$
8	FIX	p_AAB	0.0000000000000000	0.0000000000000000	: p _{aab} =p _{aba}
9	FIX	p_ABB	0.0000000000000000	0.0000000000000000	: p _{abb}}
10	FIX	p_BAA	0.0000000000000000	0.0000000000000000	: p _{baa}}
11	FIX	p_BBA	0.0000000000000000	0.0000000000000000	: p _{bba} =p _{bab}}
12	FIX	q_AAB	1.0000000000000000	1.0000000000000000	: q _{aab} =q _{aba}

13	FIX	q_ABB	1.0000000000000000	1.0000000000000000 : q _{abb}
14	FIX	q_BAA	1.0000000000000000	1.0000000000000000 : q _{baa}
15	FIX	q_BBA	1.0000000000000000	1.0000000000000000 : q _{bba} =q _{bab}
16	FREE	c1_AAB	0.0000000000000000	1.0000000000000000 : c _{1,aab} =c _{1,aba}
17	FREE	c1_ABB	0.0000000000000000	1.0000000000000000 : c _{1,abb}
18	FREE	c1_BAA	0.0000000000000000	1.0000000000000000 : c _{1,baa}
19	FREE	c1_BBA	0.0000000000000000	1.0000000000000000 : c _{1,bba} =c _{1,bab}
20	FREE	c2_AAB	0.0000000000000000	100.00000000000000 : c _{2,aab} =c _{2,aba}
21	FREE	c2_ABB	0.0000000000000000	100.00000000000000 : c _{2,abb}
22	FREE	c2_BAA	0.0000000000000000	100.00000000000000 : c _{2,baa}
23	FREE	c2_BBA	0.0000000000000000	100.00000000000000 : c _{2,bba} =c _{2,bab}
24	FIX	c3_AAB	100.00000000000000	100.00000000000000 : c _{3,aab} =c _{3,aba}
25	FIX	c3_ABB	100.00000000000000	100.00000000000000 : c _{3,abb}
26	FIX	c3_BAA	100.00000000000000	100.00000000000000 : c _{3,baa}
27	FIX	c3_BBA	100.00000000000000	100.00000000000000 : c _{3,bba} =c _{3,bab}
28	FREE	h_AAB	-1.000000	-0.000000 : h _{,aab} =h _{,aba}
29	FREE	h_ABB	-1.000000	-0.000000 : h _{,abb}
30	FREE	h_BAA	-1.000000	-0.000000 : h _{,baa}
31	FREE	h_BBA	-1.000000	-0.000000 : h _{,bba} =h _{,bab}
32	FIX	R1_AB	2.7000000000000000	2.7000000000000000 : R _{1,ab} =R _{1,ba}
33	FIX	R2-R1_AB	0.2000000000000000	0.2000000000000000 : R _{2,ab} =R _{2,ab}
34	FIX	Re_AB	2.3510000000000000	2.3510000000000000 : R _{e, ab} =R _{e, ba}
35	FIX	Rcut	3.5000000000000000	3.5000000000000000 : カットオフ距離

パラメータ振り分けは以下の表のように行う。

表 パラメータ範囲の振り分け

	h_AAB	h_ABB	h_BAA	h_BBA
case001	-1 ~-0.5	-1 ~-0.5	-1 ~-0.5	-1 ~-0.5
case002	-1 ~-0.5	-1 ~-0.5	-1 ~-0.5	-0.5~-0.0
case003	-1 ~-0.5	-1 ~-0.5	-0.5~-0.0	-1 ~-0.5
case004	-1 ~-0.5	-1 ~-0.5	-0.5~-0.0	-0.5~-0.0
case005	-1 ~-0.5	-0.5~-0.0	-1 ~-0.5	-1 ~-0.5
case006	-1 ~-0.5	-0.5~-0.0	-1 ~-0.5	-0.5~-0.0
case007	-1 ~-0.5	-0.5~-0.0	-0.5~-0.0	-1 ~-0.5
case008	-1 ~-0.5	-0.5~-0.0	-0.5~-0.0	-0.5~-0.0
case009	-0.5~-0.0	-1 ~-0.5	-1 ~-0.5	-1 ~-0.5
case010	-0.5~-0.0	-1 ~-0.5	-1 ~-0.5	-0.5~-0.0
case011	-0.5~-0.0	-1 ~-0.5	-0.5~-0.0	-1 ~-0.5
case012	-0.5~-0.0	-1 ~-0.5	-0.5~-0.0	-0.5~-0.0
case013	-0.5~-0.0	-0.5~-0.0	-1 ~-0.5	-1 ~-0.5
case014	-0.5~-0.0	-0.5~-0.0	-1 ~-0.5	-0.5~-0.0
case015	-0.5~-0.0	-0.5~-0.0	-0.5~-0.0	-1 ~-0.5

case016	-0.5~-0.0	-0.5~-0.0	-0.5~-0.0	-0.5~-0.0
---------	-----------	-----------	-----------	-----------

grep で上記の表の設定になっていることを確認する.

```
%grep h_AAB step01/*/para.dat | sed s/" "/" /g
```

```
step01/case001/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case002/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case003/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case004/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case005/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case006/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case007/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case008/para.dat: 28      FREE      h_AAB    -1.000000      -0.500000
step01/case009/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case010/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case011/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case012/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case013/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case014/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case015/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
step01/case016/para.dat: 28      FREE      h_AAB    -0.500000      -0.000000
```

```
%grep h_ABB step01/*/para.dat | sed s/" "/" /g
```

```
step01/case001/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case002/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case003/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case004/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case005/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case006/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case007/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case008/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case009/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case010/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case011/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case012/para.dat: 29      FREE      h_ABB    -1.000000      -0.500000
step01/case013/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case014/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case015/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
step01/case016/para.dat: 29      FREE      h_ABB    -0.500000      -0.000000
```

```
%grep h_BAA step01/*/para.dat | sed s/" "/" /g
```

```
step01/case001/para.dat: 30      FREE      h_BAA    -1.000000      -0.500000
step01/case002/para.dat: 30      FREE      h_BAA    -1.000000      -0.500000
step01/case003/para.dat: 30      FREE      h_BAA    -0.500000      -0.000000
step01/case004/para.dat: 30      FREE      h_BAA    -0.500000      -0.000000
```

```

step01/case005/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case006/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case007/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
step01/case008/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
step01/case009/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case010/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case011/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
step01/case012/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
step01/case013/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case014/para.dat: 30    FREE    h_BAA  -1.000000    -0.500000
step01/case015/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
step01/case016/para.dat: 30    FREE    h_BAA  -0.500000    -0.000000
%grep h_BBA step01/*/para.dat | sed s/" "/" /g
step01/case001/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case002/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case003/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case004/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case005/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case006/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case007/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case008/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case009/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case010/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case011/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case012/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case013/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case014/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000
step01/case015/para.dat: 31    FREE    h_BBA  -1.000000    -0.500000
step01/case016/para.dat: 31    FREE    h_BBA  -0.500000    -0.000000

```

14.4.4 set.dat

空孔の入った閃亜鉛鉱構造の欠陥生成エネルギーも合わせ込みに用いる。2元系なので、Al原子を抜いた場合とN原子を抜いた場合の2種類の空孔構造が存在する。少しだけ合わせ込めるかを試してみたいので、重みは小さめに10.0としている。最初のステップではポテンシャルがばらつきやすいので構造緩和は行わないため、緩和開始ステップはGAのステップ数(31)より多い200に設定した(つまり、緩和は始まらない)。

~TersoffAlN/step01/case00*/set.dat~

```

2
200 100
0 AlN_ZB_Al_defect    10.0 0.0 63 1
1 AlN_ZB_N_defect     10.0 0.0 63 1

```

14.4.5 energy.dat

文献値から適当な値を設定する。絶対エネルギーではなく相対的な欠陥生成エネルギーを合わせ込みたいので、欠陥生成エネルギーで定義した。

～TersoffAlN/step01/case00*/energy.dat～

0 AlN_ZB_Al_defect 11.5

1 AlN_ZB_N_defect 8.26

14.4.6 lattice.dat, posi.dat

第一原理計算から得られた緩和後の原子構造を用いるのが理想的なだが、格子の緩和による欠陥生成エネルギーの変化は小さいので、完全結晶から原子を抜いただけで緩和をしていない構造（空孔型欠陥）を初期原子構造とする。lattice.dat は以下のような格子定数 3.615 \AA の $5 \times 5 \times 5$ の面心立方格子とした。

～TersoffAlN/step01/case00*/lattice.dat～

0 AlN_ZB_Al_defect

8.748 0.000 0.000

0.000 8.748 0.000

0.000 0.000 8.748

1 AlN_ZB_N_defect

8.748 0.000 0.000

0.000 8.748 0.000

0.000 0.000 8.748

posi.dat も一部を示す。

～TersoffAlN/step01/case00*/posi.dat～

0 AlN_ZB_Al_defect 63

13 2.1870000000 2.1870000000 0.0000000000 0 0 0

13 2.1870000000 0.0000000000 2.1870000000 0 0 0

13 0.0000000000 2.1870000000 2.1870000000 0 0 0

7 1.0935000000 1.0935000000 1.0935000000 0 0 0

7 3.2805000000 3.2805000000 1.0935000000 0 0 0

7 1.0935000000 3.2805000000 3.2805000000 0 0 0

7 3.2805000000 1.0935000000 3.2805000000 0 0 0

13 0.0000000000 0.0000000000 4.3740000000 0 0 0

～中略～

7 7.6545000000 5.4675000000 7.6545000000 0 0 0

1 AlN_ZB_N_defect 63

13 0.0000000000 0.0000000000 0.0000000000 0 0 0

13 2.1870000000 2.1870000000 0.0000000000 0 0 0

13 2.1870000000 0.0000000000 2.1870000000 0 0 0

13 0.0000000000 2.1870000000 2.1870000000 0 0 0

7 3.2805000000 3.2805000000 1.0935000000 0 0 0

7 1.0935000000 3.2805000000 3.2805000000 0 0 0

7 3.2805000000 1.0935000000 3.2805000000 0 0 0

～後略～

14.5 適応度関数の設定

2 元系ポテンシャルの場合は、非完全結晶の欠陥生成エネルギーの計算方法に一般的な方法があるわけではないので、第一原理計算で計算した方法と同じ方法で計算するように指定する必要がある。

$$N_{Al} > N_N \quad \Delta E = E_{tot} + (N_{Al} - N_N) E_{Al}^{diamond} - N_{Al} E_{AlN}^{ZnS}$$

$$N_N > N_{Al} \quad \Delta E = E_{tot} + (N_N - N_{Al}) E_N^{diamond} - N_N E_{AlN}^{ZnS}$$

crystalEnergy[i]は fbinary.h 中で原子あたりのエネルギーとして計算(つまり/=CrystalNatom[i])しているので、

$N_{Al} E_{AlN}^{ZnS}$, $N_N E_{AlN}^{ZnS}$ は crystalEnergy[ZB]*(double)(natom1*2), crystalEnergy[ZB]*(double)(natom2*2)などと

計算されていることに注意. ndefect[i]は set.dat で設定したように 1 である。

～fbinary.h 前略～

```
for(i=STARTBINARY;i<=ENDBINARY;i++){
    if(isUsedForFitting[i]==1){
        if((i>=DIAMOND_A7B1) && (i<=DIAMOND_A1B7)){
            lta[i] = lta[i-5];
            //start from frozen structures
        }

        double latticeEnergy=ci->equivalentCrystalEnergy
            (i,lta[i],ltb[i],ltc[i],intx1[i],intx2[i],intx3[i],convdiff,cgitmax,linitmax);
        reactionEnergy[i]=ci->getCrystalStructure(i->getReactionEnergyPerAtom(latticeEnergy);
        crystalEnergy[i]=latticeEnergy/(double)crystalNatom[i];
    }
}
```

～fbinary.h 中略～

```
for(int iset=0;iset<nset;iset++){
    if(fatomn1==13&&fatomn2==7){
        int natom1 = (nci->getNumberOfAtom1(iset));
        int natom2 = (nci->getNumberOfAtom2(iset));
        if(ndefect[iset]!=0){
            if(natom1<natom2){
                defectFormationEnergy[iset] =
                    (noncrystallineEnergy[iset]+(double)(natom2-natom1)*crystalEnergy_1[DIAMOND]-
                    crystalEnergy[ZB]*(double)(natom2*2))/(double)(ndefect[iset]);
            }else{
                defectFormationEnergy[iset] =
                    (noncrystallineEnergy[iset]+(double)(natom1-natom2)*crystalEnergy_2[DIAMOND]-
                    crystalEnergy[ZB]*(double)(natom1*2))/(double)(ndefect[iset]);
            }
        }
    }
}
```

～後略～

14.6 プログラムのチェック

ディレクトリ `tersoffAlN/check/AlN` を用いて動作確認を行う。このディレクトリの `para.dat` の上限値、下限値は文献値の値となっているため、そのときの物性値を確認することができる。前節まで変更を加えたので、以下に `make` をしてから実行する。

```
%make
```

```
%.GA tersoffAlN/check/AlN
```

完全結晶の生成エネルギー・閃亜鉛鉱構造の弾性定数はおおよそ再現できているが、完全結晶の格子定数や欠陥生成エネルギーの再現性はやや悪い。

```
~tersoffAlN/check/AlN/stdout.dat 前略~
```

```
step=[11/11]
```

```
end maxf iset=0/2
```

```
end maxf iset=1/2
```

```
===== total =====
```

```
CRYSTAL: ABGRAPHITE  ENE=-3.9427 (-3.8230)  REE=-1.8026 (-1.6826)
CRYSTAL: ABDIAMOND   ENE=-3.4879 (-3.4780)  REE=-1.4055 (-1.3953)
CRYSTAL:      ABSC    ENE=-2.8689 (-2.8050)  REE=-0.8587 (-0.7945)
CRYSTAL:      ABBCC   ENE=-2.5894 (-2.6970)  REE=-0.6226 (-0.7298)
CRYSTAL: HETERODIMER ENE=-1.9940 (-1.5400)  REE=0.2328 (0.6870)
CRYSTAL: HETEROCHAIN ENE=-3.6303 (-4.0330)  REE=-1.4034 (-1.8060)
CRYSTAL: HETEROGRA   ENE=-5.2019 (-5.3440)  REE=-2.9751 (-3.1170)
CRYSTAL:      ZB     ENE=-5.8119 (-5.8120)  REE=-3.5850 (-3.5850)
CRYSTAL:      SC_A7B1 ENE=-2.9819 (-3.0120)  REE=-0.4301 (-0.4602)
CRYSTAL:      SC_A6B2 ENE=-3.1070 (-2.9930)  REE=-0.6635 (-0.5495)
CRYSTAL:      NAACL   ENE=-5.2551 (-5.6320)  REE=-3.0283 (-3.4050)
CRYSTAL:      SC_A2B6 ENE=-3.5660 (-3.6520)  REE=-1.1225 (-1.2085)
CRYSTAL:      SC_A1B7 ENE=-2.5261 (-2.6080)  REE=0.0257 (-0.0562)
CRYSTAL: BCC_A14B2   ENE=-2.9097 (-3.1450)  REE=-0.3579 (-0.5932)
CRYSTAL: BCC_A12B4   ENE=-2.9217 (-2.8950)  REE=-0.4782 (-0.4515)
CRYSTAL: BCC_A10B6   ENE=-3.4938 (-3.2400)  REE=-1.1586 (-0.9048)
CRYSTAL: BCC_A6B10   ENE=-3.2215 (-3.2090)  REE=-0.8863 (-0.8738)
CRYSTAL: BCC_A4B12   ENE=-2.7027 (-2.2550)  REE=-0.2592 (0.1885)
CRYSTAL: BCC_A2B14   ENE=-1.2738 (-1.5580)  REE=1.2781 (0.9938)
CRYSTAL: ABGRAPHITE  LT=6.4318 (6.0864)  BL=0.0000 (0.0000)
CRYSTAL: ABDIAMOND   LT=8.8992 (8.3416)  BL=0.0000 (0.0000)
CRYSTAL:      ABSC    LT=4.0459 (3.7380)  BL=0.0000 (0.0000)
CRYSTAL:      ABBCC   LT=4.9994 (4.5149)  BL=0.0000 (0.0000)
CRYSTAL: HETERODIMER LT=1.7972 (1.6920)  BL=0.0000 (1.6920)
CRYSTAL: HETEROCHAIN LT=3.6521 (3.3440)  BL=0.0000 (0.0000)
CRYSTAL: HETEROGRA   LT=3.1872 (3.1021)  BL=0.0000 (0.0000)
CRYSTAL:      ZB     LT=4.3750 (4.3740)  BL=0.0000 (0.0000)
```

CRYSTAL: SC_A7B1 LT=5.0795 (5.1960) BL=0.0000 (0.0000)
CRYSTAL: SC_A6B2 LT=4.7657 (4.9880) BL=0.0000 (0.0000)
CRYSTAL: NACL LT=4.3507 (4.0500) BL=0.0000 (0.0000)
CRYSTAL: SC_A2B6 LT=4.1011 (3.9740) BL=0.0000 (0.0000)
CRYSTAL: SC_A1B7 LT=4.1009 (3.8860) BL=0.0000 (0.0000)
CRYSTAL: BCC_A14B2 LT=6.1306 (6.2608) BL=0.0000 (0.0000)
CRYSTAL: BCC_A12B4 LT=5.8566 (6.1476) BL=0.0000 (0.0000)
CRYSTAL: BCC_A10B6 LT=5.7863 (5.6950) BL=0.0000 (0.0000)
CRYSTAL: BCC_A6B10 LT=5.4148 (4.9791) BL=0.0000 (0.0000)
CRYSTAL: BCC_A4B12 LT=5.1073 (4.9583) BL=0.0000 (0.0000)
CRYSTAL: BCC_A2B14 LT=5.2213 (4.8474) BL=0.0000 (0.0000)
BM[ZB]=205.359(196.000)
C11=305.540(306.700) C12=155.259(155.700)
C44=199.000(201.100)
NONCRYST:AlN_ZB_Al_defect ENE=8.8449 (11.5000) TOT=-360.4558 MAX_F=0.0013 (18[13])
NONCRYST:AlN_ZB_N_defect ENE=3.2214 (8.2600) TOT=-366.9459 MAX_F=0.0011 (24[13])
～後略～

14.7 プログラムの実行(1回目)

16 ケースについて、プログラムを実行する.

```
%/GA tersoffAlN/step01/case001  
%/GA tersoffAlN/step01/case002  
%/GA tersoffAlN/step01/case003  
%/GA tersoffAlN/step01/case004  
%/GA tersoffAlN/step01/case005  
%/GA tersoffAlN/step01/case006  
%/GA tersoffAlN/step01/case007  
%/GA tersoffAlN/step01/case008  
%/GA tersoffAlN/step01/case009  
%/GA tersoffAlN/step01/case010  
%/GA tersoffAlN/step01/case011  
%/GA tersoffAlN/step01/case012  
%/GA tersoffAlN/step01/case013  
%/GA tersoffAlN/step01/case014  
%/GA tersoffAlN/step01/case015  
%/GA tersoffAlN/step01/case016
```

筆者の手元にある計算機では1時間半程度で計算が終わった。同じプログラムを使って開発したので当然ではあるが、文献値同様、完全結晶の生成エネルギー・閃亜鉛鉱構造の弾性定数はおおよそ再現できているが、完全結晶の格子定数や欠陥生成エネルギーの再現性はやや悪い。例えば case003 は以下のような結果となった。

～tersoffAlN/step01/case003/stdout.dat 後略～

step=[31/31]

CRYSTAL: ABGRAPHITE ENE=-3.7066 (-3.8230) REE=-1.5664 (-1.6826)
CRYSTAL: ABDIAMOND ENE=-3.8748 (-3.4780) REE=-1.7924 (-1.3953)
CRYSTAL: ABSC ENE=-2.5153 (-2.8050) REE=-0.5051 (-0.7945)
CRYSTAL: ABBCC ENE=-2.8344 (-2.6970) REE=-0.8675 (-0.7298)
CRYSTAL: HETERODIMER ENE=-1.5444 (-1.5400) REE=0.6824 (0.6870)
CRYSTAL: HETEROCHAIN ENE=-3.0888 (-4.0330) REE=-0.8620 (-1.8060)
CRYSTAL: HETEROGRA ENE=-4.6332 (-5.3440) REE=-2.4063 (-3.1170)
CRYSTAL: ZB ENE=-5.8120 (-5.8120) REE=-3.5852 (-3.5850)
CRYSTAL: SC_A7B1 ENE=-3.5153 (-3.0120) REE=-0.9635 (-0.4602)
CRYSTAL: SC_A6B2 ENE=-4.3917 (-2.9930) REE=-1.9482 (-0.5495)
CRYSTAL: NAACL ENE=-5.4179 (-5.6320) REE=-3.1910 (-3.4050)
CRYSTAL: SC_A2B6 ENE=-2.9627 (-3.6520) REE=-0.5192 (-1.2085)
CRYSTAL: SC_A1B7 ENE=-2.2272 (-2.6080) REE=0.3246 (-0.0562)
CRYSTAL: BCC_A14B2 ENE=-3.5374 (-3.1450) REE=-0.9855 (-0.5932)
CRYSTAL: BCC_A12B4 ENE=-3.8454 (-2.8950) REE=-1.4019 (-0.4515)
CRYSTAL: BCC_A10B6 ENE=-3.8520 (-3.2400) REE=-1.5168 (-0.9048)
CRYSTAL: BCC_A6B10 ENE=-3.7685 (-3.2090) REE=-1.4334 (-0.8738)
CRYSTAL: BCC_A4B12 ENE=-2.7971 (-2.2550) REE=-0.3536 (0.1885)
CRYSTAL: BCC_A2B14 ENE=-1.8517 (-1.5580) REE=0.7001 (0.9938)
CRYSTAL: ABGRAPHITE LT=6.5093 (6.0864) BL=0.0000 (0.0000)
CRYSTAL: ABDIAMOND LT=8.7480 (8.3416) BL=0.0000 (0.0000)
CRYSTAL: ABSC LT=4.0569 (3.7380) BL=0.0000 (0.0000)
CRYSTAL: ABBCC LT=4.8390 (4.5149) BL=0.0000 (0.0000)
CRYSTAL: HETERODIMER LT=1.8791 (1.6920) BL=0.0000 (1.6920)
CRYSTAL: HETEROCHAIN LT=3.7582 (3.3440) BL=0.0000 (0.0000)
CRYSTAL: HETEROGRA LT=3.2547 (3.1021) BL=0.0000 (0.0000)
CRYSTAL: ZB LT=4.3740 (4.3740) BL=0.0000 (0.0000)
CRYSTAL: SC_A7B1 LT=4.9808 (5.1960) BL=0.0000 (0.0000)
CRYSTAL: SC_A6B2 LT=4.5748 (4.9880) BL=0.0000 (0.0000)
CRYSTAL: NAACL LT=4.1168 (4.0500) BL=0.0000 (0.0000)
CRYSTAL: SC_A2B6 LT=4.0992 (3.9740) BL=0.0000 (0.0000)
CRYSTAL: SC_A1B7 LT=4.0928 (3.8860) BL=0.0000 (0.0000)
CRYSTAL: BCC_A14B2 LT=6.1223 (6.2608) BL=0.0000 (0.0000)
CRYSTAL: BCC_A12B4 LT=5.7974 (6.1476) BL=0.0000 (0.0000)
CRYSTAL: BCC_A10B6 LT=5.7342 (5.6950) BL=0.0000 (0.0000)
CRYSTAL: BCC_A6B10 LT=5.6369 (4.9791) BL=0.0000 (0.0000)
CRYSTAL: BCC_A4B12 LT=5.6121 (4.9583) BL=0.0000 (0.0000)
CRYSTAL: BCC_A2B14 LT=5.5747 (4.8474) BL=0.0000 (0.0000)
BM[ZB]=206.463(196.000)
C11=308.839(306.700) C12=155.265(155.700)
C44=206.604(201.100)

NONCRYST:AIN_ZB_AI_defect ENE=9.8294 (0.3000) TOT=-360.3451 MAX_F=0.0001 (32[13])
NONCRYST:AIN_ZB_N_defect ENE=6.8014 (0.3000) TOT=-362.5064 MAX_F=0.2800 (1[13])
～tersoffAIN/step01/case003/stdout.dat 後略～

14.8 第2ステップの入力ファイル

第2ステップではパラメータの収束を目指す。ただ、第1ステップでおおよそ合わせ込みは終わっているので、緩和後の空孔生成エネルギーを確認しながら、調整程度の収束を行うくらいである。

14.8.1 cond.dat

～tersoffAIN/step02/case001/cond.dat～

cond.dat を以下のように設定する

id: 1 : Tersoff ポテンシャル
iscont: 1 : 継続スタート
nstep: 51 : ステップ数 31
nint: 10 : 5 ステップごとに出力
noi: 3000 : 3000 個体
nout: 1001 : トップ 3000 個体の情報を出力
nmut: 0 : すべての FREE パラメータを突然変異
mutrate: 0.03 : 突然変異幅は 3%
seed: 1 : 乱数の seed は 1

14.8.2 cont.dat

GA の最終ステップで出力された個体のうち、最後の 5000 個体を第2ステップのスタートに用いる。1 個体あたり 1 コメント行+36 パラメータ=37 行だから、3000 個体は 11100 行となるので

```
%tail -n11100 tersoffAIN/step01/case001/gene.dat > tersoffAoN/step02/case001/cont.dat  
%tail -n11100 tersoffAIN/step01/case002/gene.dat > tersoffAoN/step02/case002/cont.dat  
%tail -n11100 tersoffAIN/step01/case003/gene.dat > tersoffAoN/step02/case003/cont.dat  
%tail -n11100 tersoffAIN/step01/case004/gene.dat > tersoffAoN/step02/case004/cont.dat  
%tail -n11100 tersoffAIN/step01/case005/gene.dat > tersoffAoN/step02/case005/cont.dat  
%tail -n11100 tersoffAIN/step01/case006/gene.dat > tersoffAoN/step02/case006/cont.dat  
%tail -n11100 tersoffAIN/step01/case007/gene.dat > tersoffAoN/step02/case007/cont.dat  
%tail -n11100 tersoffAIN/step01/case008/gene.dat > tersoffAoN/step02/case008/cont.dat  
%tail -n11100 tersoffAIN/step01/case001/gene.dat > tersoffAoN/step02/case001/cont.dat  
%tail -n11100 tersoffAIN/step01/case002/gene.dat > tersoffAoN/step02/case002/cont.dat  
%tail -n11100 tersoffAIN/step01/case003/gene.dat > tersoffAoN/step02/case003/cont.dat  
%tail -n11100 tersoffAIN/step01/case004/gene.dat > tersoffAoN/step02/case004/cont.dat  
%tail -n11100 tersoffAIN/step01/case005/gene.dat > tersoffAoN/step02/case005/cont.dat  
%tail -n11100 tersoffAIN/step01/case006/gene.dat > tersoffAoN/step02/case006/cont.dat  
%tail -n11100 tersoffAIN/step01/case007/gene.dat > tersoffAoN/step02/case007/cont.dat  
%tail -n11100 tersoffAIN/step01/case008/gene.dat > tersoffAoN/step02/case008/cont.dat
```

のように cont.dat を作成する。

14.8.3 set.dat

構造緩和を行うため、最初に1ステップ目で緩和し、以降10ステップごとに再緩和する。caseごとに欠陥生成エネルギーもそれなりに異なるので、重みをcase001より少しだけ重たくする(10→20)。

～tersoffAlN/ step02/case001/set.dat

2

5 10

0 AlN_ZB_Al_defect 20.0 0.0 63 1

1 AlN_ZB_N_defect 20.0 0.0 63 1

14.9 プログラムの実行(2回目)

プログラムを実行する。

%/GA tersoffAlN/step02/case001

%/GA tersoffAlN/step02/case002

%/GA tersoffAlN/step02/case003

%/GA tersoffAlN/step02/case004

%/GA tersoffAlN/step02/case005

%/GA tersoffAlN/step02/case006

%/GA tersoffAlN/step02/case007

%/GA tersoffAlN/step02/case008

%/GA tersoffAlN/step02/case009

%/GA tersoffAlN/step02/case010

%/GA tersoffAlN/step02/case011

%/GA tersoffAlN/step02/case012

%/GA tersoffAlN/step02/case013

%/GA tersoffAlN/step02/case014

%/GA tersoffAlN/step02/case015

%/GA tersoffAlN/step02/case016

結果は例えば以下のものであり、文献値と同程度の再現性であった。

～tersoffAlN/step02/case003/stdout.dat 終端部～

step=[51/51]

CRYSTAL: DIAMOND ENE=-2.8061 (-2.5134) LT=5.3653 (5.3426)

CRYSTAL: SC ENE=-3.2350 (-3.0779) LT=2.4064 (2.3958)

CRYSTAL: BCC ENE=-3.5129 (-3.5052) LT=2.8786 (2.8745)

CRYSTAL: FCC ENE=-3.5400 (-3.5400) LT=3.6150 (3.6150)

CRYSTAL: IDEALHCP ENE=-3.5383 (-3.5312) LT=2.5563 (2.5557)

CRYSTAL: HCP ENE=-3.5383 (-3.5312) LTA=2.5561 (2.5557) LTC=4.1752 (4.1900)

ACRATIO=1.6334 (1.6395)

BM[FCC]=144.078(138.300)

C11=171.148(170.000) C12=130.531(122.500)

C44=58.496(75.800)

NONCRYST: FCC_vacancy ENE=0.9027 (1.2700) MAX_F=0.0169 (495[29])

～tersoffAlN/step02/case003/stdout.dat 後略～

得られたポテンシャルパラメータは poten.dat に出力されている.

～tersoffAlN/step02/case003/poten.dat～

eps2 2.12916255623907e-01

A2 2.63880176087711e+00

N2 9.00000000000000e+00

epsm 2.12916255623907e-01

Cm 4.82248815224088e+01

Am 1.31840035375178e+00

Mm 6.00000000000000e+00

Rcut 7.30000000000000e+00

15. 例題 CuNi-2 元系用 Generalized Embedded Atom Method(GEAM)仮パラメータの開発

本章では CuNi-2 元系用 GEAM ポテンシャルパラメータを開発する。一見、2 元系用パラメータが存在しないように見える GEAM ポテンシャルであるが、単元系の f_e/ρ_e , f_e/ρ_s さえ同じ値ならば f_e がどのような値でも物性値は変わらない。この f_e は 2 元系のときのみ有効になるパラメータである。今回は例題として仮に 1 原子置換型 CuNiFCC 構造の生成エネルギーが -0.25eV/atom となるようにする。この値は第一原理計算値・実験値ではなく、例題のための仮想的な値なので注意が必要である。本章で用いられる例題は GEAMCuNi に含まれる。

15.1 2 元系 GEAM ポテンシャル関数

GEAM ポテンシャルを 2 元系の合わせ込みに使いやすいよう以下のように変形する

$$E = \frac{1}{2} \sum_{i \neq j} \phi_{a_i a_j}(r_{ij}) + \sum_i F_{a_i}(\rho_i)$$

$$\phi_{ab}(r) = \frac{1}{2} \left(f_{e,ba} \frac{f_b(r)}{f_a(r)} \phi_{aa}(r) + f_{e,ab} \frac{f_a(r)}{f_b(r)} \phi_{ba}(r) \right),$$

$$\phi_{aa}(r) = \frac{A_a \cdot \exp \left[-\alpha_a \left(\frac{r}{r_{e,a}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e,a}} - \kappa_a \right)^{20}} - \frac{B_a \cdot \exp \left[-\beta_a \left(\frac{r}{r_{e,a}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e,a}} - \lambda_a \right)^{20}},$$

$$F_a(\rho) = \begin{cases} \sum_{i=0}^3 F_{ni,a} \left(\frac{\rho}{\rho_{n,a}} - 1 \right)^i, & \rho < \rho_{n,a}' \\ \sum_{i=0}^3 F_{i,a} \left(\frac{\rho}{\rho_{e,a}} - 1 \right)^i, & \rho_{n,a}' \leq \rho < \rho_{o,a}' \\ F_{e,a} \left[1 - \ln \left(\frac{\rho}{\rho_{s,a}} \right)^{\eta_a} \right] \cdot \left(\frac{\rho}{\rho_{s,a}} \right)^{\eta}, & \rho_{o,a}' \leq \rho \end{cases}$$

$$\rho_i = \sum_{j, j \neq i} f_{e,a_j a_i} f_{a_j}(r_{ij}), \quad f_a(r) = \frac{\exp \left[-\beta_a \left(\frac{r}{r_{e,a}} - 1 \right) \right]}{1 + \left(\frac{r}{r_{e,a}} - \lambda_a \right)^{20}}$$

ここで、 a_i は i 番原子の原子種を示す。ただし、

$$f_{e,ab} = \frac{f_{e,b}}{f_{e,a}}, \quad \rho_{n,a}' = 0.85 \frac{\rho_{e,a}}{f_{e,a}}, \quad \rho_{e,a}' = \frac{\rho_{e,a}}{f_{e,a}}, \quad \rho_{n,a}'' = 1.15 \frac{\rho_{e,a}}{f_{e,a}}$$

15.2 ヘッダの確認

15.2.1 合わせ込み条件ヘッダ condCuNi.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。2 元系パラメータは f_{eCuNi1} しかないため、合わせ込む物性も 1 つだけにする。

～condCuNi.h 冒頭部～

```
void setFlagsOfBinaryCrystalsCopperNickel
(int *isUsedForFitting,int *isCalcBulkModulus,
 int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
～前略～
isUsedForFitting[FCC_A3B1] = 1;
～後略～
}
```

置換型 FCC 構造しか合わせ込みに使わないので、適当な重みをセットする。
重みを設定する関数

```
void setWeightOfBinaryCrystalsCuNi (引数略){
～前略～
 weightForCohesiveEnergy[FCC_A3B1] = 1.0;
～後略～
}
```

15.2.2 物性の入力されているヘッダ matpropCuNi.h の確認

合わせ込みに用いる物性が設定されているか確認する。今回は仮に 1 原子置換型 CuNiFCC 構造の生成エネルギーが -0.25eV/atom となるようにする。この値は第一原理計算などの値ではなく、例題のための仮想的な値なので注意が必要である。4 をかけてあるのは基本格子中の原子数が 4 だからである。

～matpropCuNi.h 冒頭部～

```
void setMaterialsPropertiesOfCopperNickel
(double* latticeConstantA,
 double* latticeConstantB,
 double* latticeConstantC,
 double* internalParameterX1,
 double* internalParameterX2,
 double* internalParameterX3,
 double* cohesiveEnergy,double* bulkModulus,
 double* C11,double* C22,double* C33,
 double* C12,double* C13,double* C23,
 double* C44,double* C55,double* C66,
 double* energyOfSymbol1,double* energyOfSymbol2,
 double* basisEnergyByAtom1,double* basisEnergyByAtom2,
 double* fpBasisEnergyByAtom1,double* fpBasisEnergyByAtom2,
 int &scaleType){

 int i;
 int fatomn1,fatomn2;
```

```

fatomn1 = 29;
fatomn2 = 28;

double fpEnergy1[MAXTYPE],fpEnergy2[MAXTYPE];
//this is dummy
fpEnergy1[FCC] = 0.0;
fpEnergy2[FCC] = 0.0;

for(i=STARTBINARY;i<=ENDBINARY;i++){
    fpBasisEnergyByAtom1[i]=fpEnergy1[FCC];
    fpBasisEnergyByAtom2[i]=fpEnergy2[FCC];
    basisEnergyByAtom1[i]=energyOfSymbol1[FCC];
    basisEnergyByAtom2[i]=energyOfSymbol2[FCC];
}

latticeConstantA[FCC_A3B1]    = 3.6;
cohesiveEnergy[FCC_A3B1]      = fpEnergy1[FCC]*3.0+fpEnergy2[FCC]*1.0-0.25*4.0;
}

```

15.3 入力ファイル

ディレクトリ GEAMCuNi/fit/中の入力ファイルを設定する

15.3.1 element.dat

CuNi₂ 元系用の原子間ポテンシャルなので element.dat を以下のように設定する. 今回は Cu と Ni の単元系ポテンシャルにはオリジナルのものを使うため, 単元系ポテンシャルファイルはダミーのものを用意する. Al ポテンシャルのところで述べたように, 本ソフトウェアでは 2 回微分まで連続となる埋め込み関数形を採用しており, 1 回微分までしか連続でないオリジナルの GEAM ポテンシャルを入力に用いることは出来ない仕様になっている.

～GEAMCuNi/fit/element.dat～

2

Cu Ni

potenCu.dat potenNi.dat

15.3.2 cond.dat

cond.dat を以下のように設定する

～GEAMCuNi/fit/set.dat～

```

id:      2      : GEAM ポテンシャル
iscont:  0      : 新規スタート
nstep:   51     : ステップ数 5
nint:    10     : 10 ステップごとに出力
noi:     1000   : 1000 個体
nout:    101    : トップ 100 個体の情報を出力
nmult:   0      : すべての FREE パラメータを突然変異

```

mutrate: 0.1 : 突然変異幅は 10%
seed: 1 : 乱数の seed は 1
scale: 0 : スケーリングは行わない.

15.3.3 para.dat

$f_{e,CuNi}$ のみを合わせ込みに使う.

～GEAMCuNi/fit/para.dat～

0	FREE	fe_ij	0.0000000000000000	10.0000000000000000	: f_{eCuNi}
1	FIX	cut	7.5000000000000000	7.5000000000000000	

15.3.4 set.dat

合わせ込みパラメータの数が 1 つだから非結晶系は合わせ込みに用いない.

～GEAMCuNi/fit/set.dat～

0

15.4 適応度関数の設定

合わせ込みパラメータの数が 1 つだから, 特にペナルティは設けない.

15.5 fbinary.h における GEAM オリジナル Cu,Ni 単元系パラメータの設定

AI ポテンシャルのところでも述べたように, 本ソフトウェアでは 2 回微分まで連続となる埋め込み関数形を採用しており, 1 回微分までしか連続でないオリジナルの GEAM ポテンシャルを外部入力に用いることは出来ない仕様になっている. このため, fbinary.h でベタ書きすることによって GEAM オリジナル単元系 Cu,Ni パラメータを入力することにする. 具体的には fbinary.h の赤字部分のコメントをはずすことによって, setAuthorizedParameter を有効にする.

～fbinary.h～

```
double fbinary(CrystalInterface *ci, NonCrystallineInterface *nci, int stype,  
               int fatomn1, int fatomn2, double* g, double *lowerBound, double *upperBound){
```

～前略～

```
stable = ci->getStable();  
stable_1 = ci->getStable1();  
stable_2 = ci->getStable2();
```

```
ci->myPotentialParameter0->setAuthorizedParameter(fatomn1);  
ci->myPotentialParameter0->setAuthorizedParameter(fatomn2);
```

```
ci->myPotentialParameter0->setParameter(fatomn1, fatomn2, g);  
if(isScale==1){
```

～後略～

15.6 プログラムのチェック

ディレクトリ GEAMCuNi/check/Cu, GEAMCuNi/check/Ni を用いて動作確認を行う. すでに述べたように, 本

ソフトウェアでは 2 回微分まで連続となる埋め込み関数形を採用しており、1 回微分までしか連続でないオリジナルの GEAM ポテンシャルを外部入力に用いることは出来ない仕様になっている。このため、fsingle.h でベタ書きすることによって GEAM オリジナル単元系 Cu,Ni パラメータを入力することにする。具体的に以下のように fsingle.h の赤字部分のコメントをはずすことによって、setAuthorizedParameter を有効にする。

～fsingle.h～

```
double fsingle(CrystalInterface *ci,NonCrystallineInterface *nci,
               int fatomn1,double* g,double *lowerBound,double *upperBound){
～前略～
    if(isScale==1){
        scaleParameter(ci,stype,crystalNatom[stype],fpCrystalEnergy[stype],
                       fplta[stype],fpltb[stype],fpltc[stype],g,lowerBound,upperBound);
    }

    ci->myPotentialParameter()->setParameter(fatomn1,g);
    ci->myPotentialParameter()->setAuthorizedParameter(fatomn1);
```

～後略～

%make

%/GA GEAMCuNi/check/Cu

%/GA GEAMCuNi/check/Ni

下に実行した結果の一部を示す。(fsingle.h でデフォルトで指定しているように) カッコ内が合わせ込む値(実験値・第一原理計算), カッコ外が GEAM ポテンシャルパラメータを用いて計算した結果である。おおよそ実験値第一原理計算値を再現できていることがわかる。また, fbinary.h にベタ書きした setAuthorizedParameter に上書きされるので何でもよいのだが, 一応ここで生成されたポテンシャルファイルをダミーとして用いる。

%cp GEAMCuNi/check/Cu/poten.dat GEAMCuNi/fit/potenCu.dat

%cp GEAMCuNi/check/Ni/poten.dat GEAMCuNi/fit/potenNi.dat

～GEAMCuNi/check/Cu/stdout.dat 前略～

step=[10/10]

CRYSTAL: DIAMOND ENE=-2.5992 (-2.5134) LT=5.2248 (5.3426)

CRYSTAL: SC ENE=-3.1580 (-3.0779) LT=2.3994 (2.3958)

CRYSTAL: BCC ENE=-3.4876 (-3.5052) LT=2.8280 (2.8745)

CRYSTAL: FCC ENE=-3.5398 (-3.5400) LT=3.6148 (3.6150)

CRYSTAL: IDEALHCP ENE=-3.5246 (-3.5312) LT=2.5499 (2.5557)

CRYSTAL: HCP ENE=-3.5249 (-3.5312) LTA=2.5582 (2.5557) LTC=4.1240 (4.1900)

ACRATIO=1.6121 (1.6395)

BM[FCC]=138.211(138.300)

C11=180.417(170.000) C12=117.109(122.500)

C44=70.658(75.800)

～後略～

～GEAMCuNi/check/Ni/stdout.dat 前略～

step=[10/10]

CRYSTAL: DIAMOND ENE=-3.2156 (-3.2432) LT=5.1453 (5.1005)
CRYSTAL: SC ENE=-3.9672 (-3.7476) LT=2.3765 (2.3289)
CRYSTAL: BCC ENE=-4.3555 (-4.3449) LT=2.7947 (2.8020)
CRYSTAL: FCC ENE=-4.4500 (-4.4400) LT=3.5196 (3.5200)
CRYSTAL: IDEALHCP ENE=-4.4297 (-4.4171) LT=2.4926 (2.4876)
CRYSTAL: HCP ENE=-4.4307 (-4.4171) LTA=2.4820 (2.4876) LTC=4.1183 (4.0820)
ACRATIO=1.6592 (1.6409)
BM[FCC]=180.501(180.400)
C11=254.580(246.500) C12=143.462(147.300)
C44=120.973(124.700)
～後略～

15.7 プログラムの実行

プログラムを実行する.

```
%./GA GEAMCuNi/fit
```

1パラメータで1物性を再現するだけであるから, 生成エネルギーの再現も問題なくできた. 1パラメータで生成エネルギーを再現するだけなので, 非常に使い手が良い.

～GEAMCuNi/fit/stdout.dat 終端部～

```
step=[51/51]  
CRYSTAL: FCC_A3B1 ENE=-4.0150 (-0.2500) REE=-0.2500 (-0.2500)  
CRYSTAL: FCC_A3B1 LT=3.5974 (3.6000) BL=0.0000 (0.0000)  
dv=8.31714e-22:  
dp=0.00000e+00:  
lt=0.00000e+00:  
df=0.00000e+00:  
pl=0.00000e+00:  
tot=8.31714e-22  
～GEAMCuNi/fit/stdout.dat 後略～
```

16. 例題 ZrNi-2 元系用拡張型 Generalized Embedded Atom Method(GEAM)パラメータの開発

本章では ZrNi-2 元系用拡張型 GEAM ポテンシャルパラメータを開発する。前章で述べたオリジナルの GEAM ポテンシャルでは、1 パラメータのみを用いて、簡易的に生成エネルギーを合わせこんだ。しかし、精密な計算を行いたい場合には多くの 2 元系の物性値を合わせ込む必要がある。そこで、本章で 2 元系の 2 体間ポテンシャル部が単元系ポテンシャルの 2 体間ポテンシャルとは独立であるポテンシャル関数形を用いる(T.Kumagai et al., Materials transactions, vol. 48(2007), pp. 1313-1321). 本章で用いられる例題は GEAMZrNi に含まれる。

16.1 2 元系拡張型 GEAM ポテンシャル関数

関数形は以下に示す通りである。2 体ポテンシャルパラメータを 2 種類の元素に依存としている。

$$E = \frac{1}{2} \sum_{i \neq j} \phi_{a_i a_j}(r_{ij}) + \sum_i F_{a_i}(\rho_i)$$

$$\phi_{ab}(r) = \frac{A_{ab} \cdot \exp\left[-\alpha_{ab} \left(\frac{r}{r_{e,ab}} - 1\right)\right]}{1 + \left(\frac{r}{r_{e,ab}} - \kappa_{ab}\right)^{N_{\kappa,aa}}} - \frac{B_{ab} \cdot \exp\left[-\beta_{ab} \left(\frac{r}{r_{e,ab}} - 1\right)\right]}{1 + \left(\frac{r}{r_{e,ab}} - \lambda_{ab}\right)^{N_{\lambda,aa}}},$$

$$F_a(\rho) = \begin{cases} \sum_{i=0}^3 F_{ni,a} \left(\frac{\rho}{\rho_{n,a}} - 1\right)^i, & \rho < \rho'_{n,a} \\ \sum_{i=0}^3 F_{i,a} \left(\frac{\rho}{\rho_{e,a}} - 1\right)^i, & \rho'_{n,a} \leq \rho < \rho'_{o,a} \\ F_{e,a} \left[1 - \ln\left(\frac{\rho}{\rho_{s,a}}\right)^{\eta_a}\right] \cdot \left(\frac{\rho}{\rho_{s,a}}\right)^{\eta}, & \rho'_{o,a} \leq \rho \end{cases}$$

$$\rho_i = \sum_{j, j \neq i} f_{e,a_j a_i} f_{a_j}(r_{ij}), \quad f_a(r) = \frac{\exp\left[-\beta_{aa} \left(\frac{r}{r_{e,aa}} - 1\right)\right]}{1 + \left(\frac{r}{r_{e,aa}} - \lambda_{aa}\right)^{N_{\lambda,aa}}}$$

ここで、 a_i は i 番原子の原子種を示す。

16.2 ヘッダの確認

16.2.1 合わせ込み条件ヘッダ condZrNi.h の確認

合わせ込みに使う物性の設定をしている関数を確認する。Cu₂Mg や Fe₂P 構造など ZrNi 特有の合金構造を合わせ込みに用いる。

～condZrNi.h 冒頭部～

```
[kumagai@sstl01 ver0.415]%cat condZrNi.h
void setFlagsOfBinaryCrystalsZirconiumNickel
(int *isUsedForFitting,int *isCalcBulkModulus,
 int *isCalcTensileElasticConstants,int *isCalcShearElasticConstants){
```

～前略～

```
isUsedForFitting[SC_A7B1] = 1;  
isUsedForFitting[SC_A6B2] = 1;  
isUsedForFitting[NACL]    = 1;  
isUsedForFitting[SC_A2B6] = 1;  
isUsedForFitting[SC_A1B7] = 1;
```

```
isUsedForFitting[BCC_A14B2] = 0;  
isUsedForFitting[BCC_A12B4] = 1;  
isUsedForFitting[BCC_A10B6] = 1;  
isUsedForFitting[CSCL]      = 1;  
isUsedForFitting[BCC_A6B10] = 1;  
isUsedForFitting[BCC_A4B12] = 1;  
isUsedForFitting[BCC_A2B14] = 1;
```

```
isUsedForFitting[FCC_A3B1] = 1;  
isUsedForFitting[FCC_A2B2] = 0;  
isUsedForFitting[FCC_A1B3] = 1;
```

```
isUsedForFitting[IDEALWZ] = 0;  
isUsedForFitting[WZ]      = 0;
```

```
isUsedForFitting[CRB]     = 1;
```

```
isUsedForFitting[AL2CU_A2B1] = 1;  
isUsedForFitting[AL2CU_A1B2] = 1;
```

```
isUsedForFitting[TI2NI_A2B1] = 1;  
isUsedForFitting[TI2NI_A1B2] = 0;
```

```
isUsedForFitting[CU2MG_A2B1] = 1;  
isUsedForFitting[CU2MG_A1B2] = 1;
```

```
isUsedForFitting[FE2P_A2B1] = 1;  
isUsedForFitting[FE2P_A1B2] = 1;
```

～中略～

```
isCalcBulkModulus[CU2MG_A2B1] = 1;  
isCalcBulkModulus[CU2MG_A1B2] = 1;
```

～後略～

}

重みを設定する関数は以下に示すとおりである.

```
isCalcShearElasticConstants[WZ] = 0;
}

void setWeightOfBinaryCrystalsZrNi
(double *weightForCohesiveEnergy,double *weightForLatticeConstant,
 double *weightForBulkModulus,double *weightForTensileElasticConstants,double
 *weightForShearElasticConstants){

    weightForCohesiveEnergy[SC_A7B1] = 1.0;
    weightForCohesiveEnergy[SC_A6B2] = 1.0;
    weightForCohesiveEnergy[NACL]    = 1.0;
    weightForCohesiveEnergy[SC_A2B6] = 1.0;
    weightForCohesiveEnergy[SC_A1B7] = 1.0;

    weightForCohesiveEnergy[BCC_A14B2] = 5.0;
    weightForCohesiveEnergy[BCC_A12B4] = 5.0;
    weightForCohesiveEnergy[BCC_A10B6] = 5.0;
    weightForCohesiveEnergy[CSCL]      = 5.0;
    weightForCohesiveEnergy[BCC_A6B10] = 5.0;
    weightForCohesiveEnergy[BCC_A4B12] = 5.0;
    weightForCohesiveEnergy[BCC_A2B14] = 5.0;

    weightForCohesiveEnergy[FCC_A3B1] = 10.0;
    weightForCohesiveEnergy[FCC_A2B2] = 10.0;
    weightForCohesiveEnergy[FCC_A1B3] = 10.0;

    weightForCohesiveEnergy[CU2MG_A2B1] = 10.0;
    weightForCohesiveEnergy[CU2MG_A1B2] = 10.0;

    weightForBulkModulus[FCC_A3B1] = 0.0;
    weightForBulkModulus[FCC_A2B2] = 0.0;
    weightForBulkModulus[FCC_A1B3] = 0.0;

    weightForBulkModulus[CU2MG_A2B1] = 30.0;
    weightForBulkModulus[CU2MG_A1B2] = 30.0;
}
```

16.3 物性の入力されているヘッダ `matpropZrNi.h` の確認
合わせ込みに用いる物性が設定されているか確認する.

～matpropZrNi.h～

```
void setMaterialsPropertiesOfCopperNickel
(double* latticeConstantA,
 double* latticeConstantB,
 double* latticeConstantC,
 double* internalParameterX1,
 double* internalParameterX2,
 double* internalParameterX3,
 double* cohesiveEnergy,double* bulkModulus,
 double* C11,double* C22,double* C33,
 double* C12,double* C13,double* C23,
 double* C44,double* C55,double* C66,
 double* energyOfSymbol1,double* energyOfSymbol2,
 double* basisEnergyByAtom1,double* basisEnergyByAtom2,
 double* fpBasisEnergyByAtom1,double* fpBasisEnergyByAtom2,
 int &scaleType){

    int i;
    int fatomn1,fatomn2;

    fatomn1 = 29;
    fatomn2 = 28;

    double fpEnergy1[MAXTYPE],fpEnergy2[MAXTYPE];
    //this is dummy
    fpEnergy1[FCC] = 0.0;
    fpEnergy2[FCC] = 0.0;

    for(i=STARTBINARY;i<=ENDBINARY;i++){
        fpBasisEnergyByAtom1[i]=fpEnergy1[FCC];
        fpBasisEnergyByAtom2[i]=fpEnergy2[FCC];
        basisEnergyByAtom1[i]=energyOfSymbol1[FCC];
        basisEnergyByAtom2[i]=energyOfSymbol2[FCC];
    }

    latticeConstantA[FCC_A3B1]    = 3.6;
    cohesiveEnergy[FCC_A3B1]      = fpEnergy1[FCC]*3.0+fpEnergy2[FCC]*1.0-0.25*4.0;
}
```

16.4 入力ファイル

ディレクトリ GEAMCuNi/fit/中に入力ファイルを設定する

16.4.1 element.dat

ZrNi2 元系用の原子間ポテンシャルなので element.dat を以下のように設定する.

~GEAMZrNi/fit/element.dat~

2

Zr Ni

potenZr.dat potenNi.dat

16.4.2 cond.dat

cond.dat を以下のように設定する. case001 から case008 までの seed は 1 から 8 と設定する.

~GEAMZrNi/fit/case001/set.dat~

id: 7 : 拡張 GEAM ポテンシャル
 iscont: 0 : 新規スタート
 nstep: 26 : ステップ数 26
 nint: 5 : 5 ステップごとに出力
 noi: 1000 : 1000 個体
 nout: 101 : トップ 100 個体の情報を出力
 nmut: 0 : すべての FREE パラメータを突然変異
 mutrate: 0.1 : 突然変異幅は 10%
 seed: 1 : 乱数の seed は 1
 scale: 0 : スケーリングは行わない.

16.4.3 para.dat

重要な $f_{e,ZrNi}$ を以下の表のように振って合わせ込みに使う.

表: パラメータ振り分け

	f_e 下限	f_e 上限
case001	0.1	0.2
case002	0.2	0.3
case003	0.3	0.4
case004	0.4	0.5
case005	0.5	0.6
case006	0.6	0.7
case007	0.7	0.8
case008	0.8	0.9

例えば case001 の場合 f_e の下限が 0.1, 上限が 0.2 となる.

~GEAMZrNi/fit/case001/para.dat~

0	FREE	f_{e_ij}	0.1000000000000000	0.2000000000000000	: $f_{e,ZrNi}$
1	FREE	r_{e_ij}	2.4000000000000000	3.6000000000000000	: $r_{e,ZrNi}$
2	FREE	α_{ij}	0.0000000000000000	20.0000000000000000	: α_{ZrNi}
3	FREE	β_{ij}	0.0000000000000000	10.0000000000000000	: β_{ZrNi}
4	FREE	A_{ij}	0.0000000000000000	1.0000000000000000	: A_{ZrNi}
5	FREE	B_{ij}	0.0000000000000000	1.0000000000000000	: B_{ZrNi}

6	FREE	kappa_ij	0.1000000000000000	0.5000000000000000	: κ_{ZrNi}
7	FREE	(1+lambda)*re	5.8000000000000000	6.1000000000000000	:(1+ λ_{ZrNi}) $\times r_{\text{e,ZrNi}}$
8	FREE	nkappa_ij	12.0000000000000000	36.0000000000000000	: $N_{\kappa,\text{ZrNi}}$
9	FREE	nlambda_ij	12.0000000000000000	36.0000000000000000	: $N_{\lambda,\text{ZrNi}}$
10	FREE	cutoff_ij	8.0000000000000000	8.0000000000000000	

16.4.4 set.dat

本来多くの非結晶系構造を合わせ込みに用いるべきだが、例題として簡単に計算できるようにする為、今回は結晶構造は使わずに合わせ込みを行う。なお、筆者が論文中で使用した非結晶系（欠陥）構造は、ディレクトリ GEAMZrNi/check/ZrNi2/中の set.dat, lattice.dat, posi.dat, energy.dat であるので、利用したい場合は参考にしたい。

～GEAMZrNi/fit/set.dat～

0

16.5 適応度関数の設定

今回は重みだけで合わせ込みするものとし、別段のセッティングは行わない。

16.6 省メモリモードの設定

合わせ込む物性値が多いため、そのままではメモリ使用量が 5GB を超える。このため、省メモリモードとしてコンパイルしなおす必要がある。structure.h の冒頭部でを以下のようにコメントアウトし、

```
#include<cmath>
#include<cfloat>
#define MARGIN 0.2
```

```
##define ENV 1
```

コンパイルしなおす。

```
%make clean
```

```
%make
```

16.7 プログラムのチェック

ディレクトリ GEAMZrNi/check/Zr, GEAMZrNi/check/Ni, GEAMZrNi/check/ZrNi を用いて動作確認を行う。上限値, 下限値ともに(T.Kumagai et al., Materials transactions, vol. 48(2007), pp. 1313-1321) のパラメータの値に設定してあるため、物性値を確認することができる。

```
./GA GEAMAI/check/Zr
```

```
./GA GEAMAI/check/Ni
```

下に実行した結果の一部を示す。カッコ内が合わせ込む値（実験値・第一原理計算）、カッコ外がポテンシャルを用いて計算した結果である。おおよそ実験値, 第一原理計算値を再現できていることがわかる

～GEAMZrNi/check/ZrNi/stdout.dat 終端部～

```
step=[10/10]
```

```
CRYSTAL: SC_A7B1 ENE=-5.5905 (-7.3954) REE=-0.0332 (-0.1044)
```

```
CRYSTAL: SC_A6B2 ENE=-5.5641 (-7.2698) REE=-0.2375 (-0.3234)
```

CRYSTAL: NACL ENE=-5.5227 (-7.1382) REE=-0.6576 (-0.8813)
 CRYSTAL: SC_A2B6 ENE=-4.8950 (-6.1735) REE=-0.4915 (-0.6060)
 CRYSTAL: SC_A1B7 ENE=-4.4739 (-5.4691) REE=-0.3011 (-0.2462)
 CRYSTAL: BCC_A12B4 ENE=-5.9189 (-7.7200) REE=-0.1892 (-0.0521)
 CRYSTAL: BCC_A10B6 ENE=-5.7837 (-7.5386) REE=-0.2839 (-0.2362)
 CRYSTAL: CSCL ENE=-5.7346 (-7.3830) REE=-0.4646 (-0.4459)
 CRYSTAL: BCC_A6B10 ENE=-5.4502 (-6.9149) REE=-0.4101 (-0.3432)
 CRYSTAL: BCC_A4B12 ENE=-5.1811 (-6.5875) REE=-0.3707 (-0.3813)
 CRYSTAL: BCC_A2B14 ENE=-4.8239 (-6.0042) REE=-0.2434 (-0.1634)
 CRYSTAL: FCC_A3B1 ENE=-5.8759 (-7.7419) REE=-0.1039 (-0.0214)
 CRYSTAL: FCC_A1B3 ENE=-5.2417 (-6.7263) REE=-0.3577 (-0.4398)
 CRYSTAL: AL2CU_A2B1 ENE=-5.9461 (-7.8353) REE=-0.2994 (-0.3275)
 CRYSTAL: AL2CU_A1B2 ENE=-5.2547 (-6.7299) REE=-0.2114 (-0.1912)
 CRYSTAL: TI2NI_A2B1 ENE=-5.9285 (-7.7469) REE=-0.2819 (-0.2390)
 CRYSTAL: CRB ENE=-5.6958 (-7.4853) REE=-0.3509 (-0.4620)
 CRYSTAL: FE2P_A2B1 ENE=-5.8475 (-7.7212) REE=-0.2008 (-0.2134)
 CRYSTAL: FE2P_A1B2 ENE=-5.3204 (-6.6905) REE=-0.2771 (-0.1518)
 CRYSTAL: CU2MG_A2B1 ENE=-5.2621 (-6.9118) REE=0.3845 (0.5961)
 CRYSTAL: CU2MG_A1B2 ENE=-5.4627 (-6.9540) REE=-0.4193 (-0.4153)
 CRYSTAL: SC_A7B1 LT=5.6202 (5.6057) BL=0.0000 (0.0000)
 CRYSTAL: SC_A6B2 LT=5.4312 (5.4137) BL=0.0000 (0.0000)
 CRYSTAL: NACL LT=5.0213 (5.1877) BL=0.0000 (0.0000)
 CRYSTAL: SC_A2B6 LT=4.7882 (4.8331) BL=0.0000 (0.0000)
 CRYSTAL: SC_A1B7 LT=4.7331 (4.7344) BL=0.0000 (0.0000)
 CRYSTAL: BCC_A12B4 LT=6.7290 (6.6723) BL=0.0000 (0.0000)
 CRYSTAL: BCC_A10B6 LT=6.5126 (6.5271) BL=0.0000 (0.0000)
 CRYSTAL: CSCL LT=3.1570 (3.1792) BL=0.0000 (0.0000)
 CRYSTAL: BCC_A6B10 LT=6.1423 (6.1626) BL=0.0000 (0.0000)
 CRYSTAL: BCC_A4B12 LT=5.9918 (5.9401) BL=0.0000 (0.0000)
 CRYSTAL: BCC_A2B14 LT=5.7349 (5.7406) BL=0.0000 (0.0000)
 CRYSTAL: FCC_A3B1 LT=4.3343 (4.2387) BL=0.0000 (0.0000)
 CRYSTAL: FCC_A1B3 LT=3.6418 (3.7209) BL=0.0000 (0.0000)
 CRYSTAL: AL2CU_A2B1 LTA=6.2800 (6.4473) LTC=5.2423 (5.1791) INT=0.165347
 CRYSTAL: AL2CU_A1B2 LTA=4.8355 (4.9777) LTC=7.5689 (7.2701) INT=0.178619
 CRYSTAL: TI2NI_A2B1 LT=11.9689 (12.0823) BL=0.0000 (0.0000)
 CRYSTAL: CRB LTA=2.9984 (3.2670) LTB=9.5747 (9.9030) LTC=4.3013 (4.1070) INT=0.077129 2.998387
 CRYSTAL: FE2P_A2B1 LTA=7.1127 (6.8269) LTC=3.6024 (4.1561) INT=0.590284 7.112709
 CRYSTAL: FE2P_A1B2 LTA=7.3927 (7.3559) LTC=2.6808 (2.7191) INT=0.645070 7.392745
 CRYSTAL: CU2MG_A2B1 LT=7.7007 (7.6926) BL=0.0000 (0.0000)
 CRYSTAL: CU2MG_A1B2 LT=6.9499 (6.9160) BL=0.0000 (0.0000)
 BM[FCC_A3B1]=95.517(103.000)
 BM[FCC_A1B3]=154.935(157.000)

BM[CU2MG_A2B1]=79.628(81.000)

BM[CU2MG_A1B2]=143.754(148.000)

～後略～

～GEAMZrNi/check/Zr/stdout.dat 終端部～

step=[10/10]

step=[10/10]

CRYSTAL: DIAMOND ENE=-5.2306 (-3.9400) LT=6.2454 (6.1139)

CRYSTAL: SC ENE=-5.7881 (-5.4088) LT=3.0657 (2.8926)

CRYSTAL: BCC ENE=-6.1894 (-6.1717) LT=3.5863 (3.5655)

CRYSTAL: FCC ENE=-6.2159 (-6.2104) LT=4.5424 (4.5190)

CRYSTAL: IDEALHCP ENE=-6.2478 (-6.2500) LT=3.2047 (3.2047)

CRYSTAL: HCP ENE=-6.2500 (-6.2500) LTA=3.2370 (3.2276) LTC=5.1480 (5.1592)

ACRATIO=1.5904 (1.5985)

BM[HCP]=102.494(96.600)

C11=128.160(144.000) C12=68.006(74.000) C13=88.767(67.000) C33=175.045(166.000)

～後略～

～GEAMZrNi/check/Ni/stdout.dat 終端部～

step=[10/10]

CRYSTAL: DIAMOND ENE=-3.1869 (-3.2432) LT=5.1710 (5.1005)

CRYSTAL: SC ENE=-3.9420 (-3.7476) LT=2.3657 (2.3289)

CRYSTAL: BCC ENE=-4.3507 (-4.3449) LT=2.8052 (2.8020)

CRYSTAL: FCC ENE=-4.4400 (-4.4400) LT=3.5200 (3.5200)

CRYSTAL: IDEALHCP ENE=-4.4208 (-4.4171) LT=2.4919 (2.4876)

CRYSTAL: HCP ENE=-4.4215 (-4.4171) LTA=2.4837 (2.4876) LTC=4.1068 (4.0820)

ACRATIO=1.6535 (1.6409)

BM[FCC]=179.396(180.400)

C11=243.720(246.500) C12=147.235(147.300)

C44=127.998(124.700)

～後略～

16.8 プログラムの実行

プログラムを実行する.

%./GA GEAMZrNi/fit/case001 &

%./GA GEAMZrNi/fit/case002 &

%./GA GEAMZrNi/fit/case003 &

%./GA GEAMZrNi/fit/case004 &

%./GA GEAMZrNi/fit/case005 &

%./GA GEAMZrNi/fit/case006 &

%./GA GEAMZrNi/fit/case007 &

%./GA GEAMZrNi/fit/case008 &

case001 から case008 までどの場合でも、およその物性は再現できた。これでは合わせ込みに用いた物性が不足しているということで、ポテンシャルを選ぶことができない。筆者らが過去に研究を行った際には、開発したポテンシャルを用いてアモルファス構造を作成し、実験における動径分布関数の再現性の高いものを選んだ。

～GEAMZrNi/fit/case004/stdout.dat 終端部～

step=[26/26]

CRYSTAL:	SC_A7B1	ENE=-5.5400 (-7.3954)	REE=0.0173 (-0.1044)
CRYSTAL:	SC_A6B2	ENE=-5.4846 (-7.2698)	REE=-0.1580 (-0.3234)
CRYSTAL:	NACL	ENE=-5.3112 (-7.1382)	REE=-0.4461 (-0.8813)
CRYSTAL:	SC_A2B6	ENE=-4.6332 (-6.1735)	REE=-0.2297 (-0.6060)
CRYSTAL:	SC_A1B7	ENE=-4.2284 (-5.4691)	REE=-0.0557 (-0.2462)
CRYSTAL:	BCC_A12B4	ENE=-5.9372 (-7.7200)	REE=-0.2075 (-0.0521)
CRYSTAL:	BCC_A10B6	ENE=-5.8434 (-7.5386)	REE=-0.3435 (-0.2362)
CRYSTAL:	CSCL	ENE=-5.8460 (-7.3830)	REE=-0.5759 (-0.4459)
CRYSTAL:	BCC_A6B10	ENE=-5.4737 (-6.9149)	REE=-0.4335 (-0.3432)
CRYSTAL:	BCC_A4B12	ENE=-5.1950 (-6.5875)	REE=-0.3847 (-0.3813)
CRYSTAL:	BCC_A2B14	ENE=-4.7160 (-6.0042)	REE=-0.1355 (-0.1634)
CRYSTAL:	FCC_A3B1	ENE=-5.9234 (-7.7419)	REE=-0.1514 (-0.0214)
CRYSTAL:	FCC_A1B3	ENE=-5.1555 (-6.7263)	REE=-0.2715 (-0.4398)
CRYSTAL:	AL2CU_A2B1	ENE=-5.8841 (-7.8353)	REE=-0.2374 (-0.3275)
CRYSTAL:	AL2CU_A1B2	ENE=-5.1316 (-6.7299)	REE=-0.0883 (-0.1912)
CRYSTAL:	TI2NI_A2B1	ENE=-5.9154 (-7.7469)	REE=-0.2687 (-0.2390)
CRYSTAL:	CRB	ENE=-5.6776 (-7.4853)	REE=-0.3326 (-0.4620)
CRYSTAL:	FE2P_A2B1	ENE=-5.8362 (-7.7212)	REE=-0.1895 (-0.2134)
CRYSTAL:	FE2P_A1B2	ENE=-5.2793 (-6.6905)	REE=-0.2360 (-0.1518)
CRYSTAL:	CU2MG_A2B1	ENE=-5.2881 (-6.9118)	REE=0.3585 (0.5961)
CRYSTAL:	CU2MG_A1B2	ENE=-5.4356 (-6.9540)	REE=-0.3923 (-0.4153)
CRYSTAL:	SC_A7B1	LT=5.6415 (5.6057)	BL=0.0000 (0.0000)
CRYSTAL:	SC_A6B2	LT=5.4296 (5.4137)	BL=0.0000 (0.0000)
CRYSTAL:	NACL	LT=5.1616 (5.1877)	BL=0.0000 (0.0000)
CRYSTAL:	SC_A2B6	LT=5.0120 (4.8331)	BL=0.0000 (0.0000)
CRYSTAL:	SC_A1B7	LT=4.9015 (4.7344)	BL=0.0000 (0.0000)
CRYSTAL:	BCC_A12B4	LT=6.6441 (6.6723)	BL=0.0000 (0.0000)
CRYSTAL:	BCC_A10B6	LT=6.4676 (6.5271)	BL=0.0000 (0.0000)
CRYSTAL:	CSCL	LT=3.1603 (3.1792)	BL=0.0000 (0.0000)
CRYSTAL:	BCC_A6B10	LT=6.1885 (6.1626)	BL=0.0000 (0.0000)
CRYSTAL:	BCC_A4B12	LT=6.0493 (5.9401)	BL=0.0000 (0.0000)
CRYSTAL:	BCC_A2B14	LT=5.8610 (5.7406)	BL=0.0000 (0.0000)
CRYSTAL:	FCC_A3B1	LT=4.3020 (4.2387)	BL=0.0000 (0.0000)
CRYSTAL:	FCC_A1B3	LT=3.7260 (3.7209)	BL=0.0000 (0.0000)
CRYSTAL:	AL2CU_A2B1	LTA=6.3329 (6.4473)	LTC=5.3216 (5.1791) INT=0.165558
CRYSTAL:	AL2CU_A1B2	LTA=4.6568 (4.9777)	LTC=8.2761 (7.2701) INT=0.177732

CRYSTAL: TI2NI_A2B1 LT=12.0270 (12.0823) BL=0.0000 (0.0000)

CRYSTAL: CRB LTA=3.0051 (3.2670) LTB=9.7926 (9.9030) LTC=4.3606 (4.1070) INT=0.078256
3.005114

CRYSTAL: FE2P_A2B1 LTA=7.1124 (6.8269) LTC=3.6837 (4.1561) INT=0.599669 7.112438

CRYSTAL: FE2P_A1B2 LTA=7.4607 (7.3559) LTC=2.6514 (2.7191) INT=0.643343 7.460729

CRYSTAL: CU2MG_A2B1 LT=7.6156 (7.6926) BL=0.0000 (0.0000)

CRYSTAL: CU2MG_A1B2 LT=6.8798 (6.9160) BL=0.0000 (0.0000)

BM[FCC_A3B1]=100.673(103.000)

BM[FCC_A1B3]=158.347(157.000)

BM[CU2MG_A2B1]=79.917(81.000)

BM[CU2MG_A1B2]=147.412(148.000)

~GEAMZrNi/fit/case004/stdout.dat 後略~

