

卒業論文
粒界の歪み場の視覚化

関西学院大学 理工学部
情報科学科 西谷研究室
27014576 白木 智也

平成31年2月21日

目次

第1章	序論	2
1.1	過去の研究	2
1.2	問題点	2
1.3	改善案, 目的	3
第2章	基本的事項および研究背景	4
2.1	EAM	4
2.2	Ruby	4
2.3	numo/gnuplot	5
2.4	cairo	5
第3章	結果と議論	6
3.1	配位数による ratio 値の確認	6
3.2	cu の配位数の変化によるエネルギー値の変化	10
3.3	Cu の eam view の改良	10
3.4	view の 2 色透明度表示	12
第4章	結論	15
付録A	eam と eam_cu の違い	18
付録B	eam_plot.rb のソース	21

第1章 序論

1.1 過去の研究

昨年度の菊岡が歪み場を EAM で解析する手法を提案した [1] . 図 1.1 に示した通り, 粒界近傍の圧縮・膨張ひずみの視覚化に適用している. 視覚化するにあたってよく使われる三色の原色を用いた色環表示を行なっている. 図 1.1 の左下パネルの粒界部拡大色環表示に示した通り, 赤は膨張領域, 緑はひずみのない完全結晶部, 青は圧縮領域である. 結晶粒どうしの境界である粒界や境界線に生じる原子配列の乱れである転位が読み取れる.

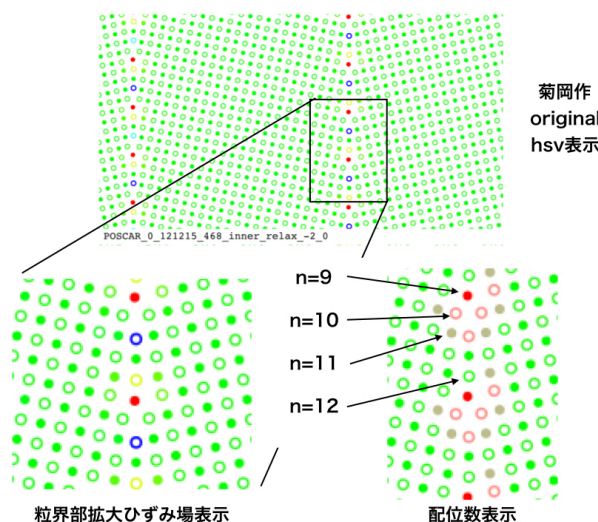


図 1.1: 菊岡による色環表示による粒界近傍の圧縮, 膨張ひずみ領域と, 配位数表示の比較.

1.2 問題点

しかし, いくつかの課題が残っていた.

fcc 構造での $\text{ratio} = E_b/E_r$ の依存性を元に作っている

ratio は膨張エネルギーと圧縮エネルギーの比から計算された値である. 配位数が違う構造で, 適切に圧縮・膨張が表示できるかは不明である.

図 1.1 の右下パネルには、西谷が開発した配位数表示を示した。完全結晶部では Al の fcc 構造により 12 配位であるが、転位近傍では隣接する原子の乱れに伴って、9 から 12 配位まで変化している。ところが、ひずみ場表示と配位数表示は対応しているわけではない。

つまり、配位数が変わった状態で、9 配位と 12 配位と違う時では、平衡距離が変わるため、ひずみが適切に評価されているかは不明であった。

EAM 歪み解析の正当性を確認するため、異なる配位数の格子構造において、ratio がどのような振る舞いをするかを確認する必要がある。

歪み場の表示が見にくい

表示は、GPA という解析手法 (GATAN 製) に組み込まれている標準ソフトに倣って作成した。そこでは、色環に基づいて赤、緑、青と変化する表示が採用されている。ところが、問題点として

- ほとんど歪みを含んでない領域を表示する緑の領域が広い。
- 慣れないと圧縮、膨張ひずみの領域を判別するのが難しい。

という 2 つが挙げられる。そこで、より直感的と思われる色の透明度を表すアルファ値を使った表示を試みた。

Cu の EAM が新たに必要

昨年度までは、Al の粒界を対象としていたが、今年度本格的に Cu の粒界計算を実行している。その結果を手軽に視覚化するために、Cu の EAM を作成する必要性が出てきた。

1.3 改善案, 目的

以上の課題を解決することを念頭に、本研究の目的は次の通りとした。

- 配位数の違う種々の構造で E-V 曲線をかく。
- 色環を使う表示だと、領域の区別がつきにくい。圧縮、膨張の 2 色の強弱に変える。
- Cu の EAM 解析を可能にする。

である。

第2章 基本的事項および研究背景

2.1 EAM

原子挿入法 (Embedding Atom Method:EAM) は, 80 年代終わりから金属系の物性をよく再現する標準ポテンシャルとして多くの研究で用いられている. EAM は経験的ポテンシャルで, 原子間のエネルギーを近似する方法を用いて, 金属性物質の物性を大雑把には再現するが, 信頼性はそれほど高くない. ここでは西谷 [2] に従って EAM を解説する.

単純な EAM の表式は, i -原子の周りの j -原子との相互作用から,

$$\begin{aligned} E_i &= E_{\text{repulsive}} + E_{\text{binding}} \\ &= \sum_j \phi(r_{ij}) - \sqrt{\rho} \\ \rho &= \sum_j h(r_{ij})^2 \end{aligned} \tag{2.1}$$

で求められる. ここで 1 行目の式が EAM の一般形を示している. 2 行目の式は埋め込み関数を理論的に意味のある最も単純な形をとった場合の基本形である. ϕ は単純な二体間ポテンシャルであり, ρ は電子密度とよばれる. ϕ, h それぞれの相互作用は, $i-j$ 間の原子間距離 r_{ij} を用いて,

$$\begin{aligned} \phi(r_{ij}) &= A0 \exp(-pr_{ij}) \\ h(r_{ij}) &= B0 \exp(-qr_{ij}) \end{aligned} \tag{2.2}$$

で求められる. それぞれのパラメータ ($A0, B0, p, q$) は物質の物性値に合わせてフィッティングされる. p/q の決め方には任意性があり, 原子空孔の生成エネルギーに合わせて作成した.

2.2 Ruby

西谷研で開発された視覚化ソフト (eam) は ruby が使用されており, 本研究のコード開発には Ruby を用いた. Ruby はまつもとゆきひろ氏により開発されたオブジェクト指向スクリプト言語. プログラムを実行するためにはコンパイルという手続きを介して初めて実行する C 言語や Java とは異なり, そのまま実行することができるため, コンパイラ言語に比べてプログラミングを手軽に行うことができるのが特徴である. また, オブジェクト指向言語であり, すべてのデータがオブジェクトとして表現されている [3].

2.3 numo/gnuplot

ruby の library として用意されている numo/gnuplot は科学的データグラフ化ツールとして広く利用されている。

gnuplot は 2 次元や 3 次元のグラフを描画するためのコマンドラインツール gnuplot を Ruby 言語で使えるようにラップしたものである。

numo/gnuplot とは、gnuplot を ruby 言語から使うためのラッパー (wrapper) である。

2.4 cairo

cairo は 2 次元画像を描画するためのアプリケーションである。cairo の特徴としては、cairo はベクトルベースの描画モデルを採用している。この描画モデルでは拡大や縮小しても画質が落ちないため、高品質の出力をおこなうことができる。描画コンテキストを用いた API であるため、描画するためのコードを短く簡潔に作成することができる。また複数出力のサポートがあるため、出力先のフォーマットに影響されることなく、描画処理をおこなうことが可能である [4]。

ruby には cairo のコマンド群を ruby 言語から利用するラッパーライブラリ rcairo が存在する。

rcairo が用意しているカラー表示には、rgb(red green blue) の三要素だけではなく、それに a(alpha)、つまり不透明度を加えた rgba がある。2 色透明度表示はこれをもとに作成した。

使い方は、<http://rcairo.github.io/doc/ja/cairo-context.html>にある通り、

```
set_source_rgb(red, green, blue)
set_source_rgba(red, green, blue, alpha=1.0)
```

コンテキストのソースパターンの色を指定する。
alpha を指定することによって半透明にすることも可能である。
この色は新しいソースパターンが設定されるまで、
この後の全ての描画操作に使われる。

色とアルファ値は 0 から 1 までの浮動小数点である。

もし、この範囲におさまらなかった場合は強制的にこの範囲におさめる。

より簡単に色を指定したい場合は

Cairo::Context#set_source_color を使う。

red: 色の赤の部分

green: 色の緑の部分

blue: 色の青の部分

alpha: 色のアルファチャンネルの部分

である。これを使う。

第3章 結果と議論

3.1 配位数による ratio 値の確認

ratio が配位数の異なった構造においてどのように振る舞うかを確認する．面心立方格子 (face-centered cubic: fcc) 構造に対する各エネルギー値の距離 (distance) 依存性を，表 3.1 に記入した．repulsive と binding は (2.1) 式の 1 行目の各エネルギー項を示している．total はこの repulsive と binding の和である．ratio は repulsive と binding のエネルギー値の比である．表 3.2 では単純立方格子 (simple cubic: sc) 構造に対する距離依存性を表示した [5] ．

表 3.1: fcc の EAM の距離依存性．

distance	repulsive	binding	total	ratio
0.500	38.53674	-14.96128	23.57547	2.57577
0.600	20.85790	-12.10155	8.75636	1.72357
0.700	11.28928	-9.78843	1.50085	1.15333
0.800	6.11029	-7.91745	-1.80716	0.77175
0.900	3.30718	-6.40409	-3.09691	0.51642
1.000	1.79000	-5.18000	-3.39000	0.34556
1.100	0.96883	-4.18988	-3.22105	0.23123
1.200	0.52438	-3.38902	-2.86464	0.15473
1.300	0.28382	-2.74124	-2.45742	0.10354
1.400	0.15362	-2.21727	-2.06365	0.06928
1.500	0.08314	-1.79346	-1.71031	0.04636

これをグラフに plot したのが，図 3.1 である．fcc 構造において，repulsive は単調に減少し，binding は単調に増加している．最安定位置は distance = 1.0，ratio = 0.346 である．距離が 1.0 未満の時，トータルエネルギー (total) は減少しており，圧縮状態であることを意味している．一方，1.0 を超えた値の時は緩やかに増加し，膨張状態である．

sc 構造においても最安定位置は距離が 0.6-0.7 の間にある．その時，ratio = 0.449, 0.254 の間である ratio = 0.346 に近い値である．fcc 構造と同様に repulsive は単調に減少し，binding は単調に増加する．距離 (distance) が安定位置未満の時，トータルエネルギー (total) は減少しており，圧縮状態にある．一方，安定位置を超えた値の時は緩やかに増加し，膨張状態にある．

つまり，fcc の最安定位置での ratio と sc の最安定位置での ratio はほぼ一致していることがわかった．一方，sc の最安定の格子定数 $a=0.65$ は fcc の最近接距離と比較すること

表 3.2: sc の EAM の距離依存性.

distance	repulsive	binding	total	ratio
0.500	5.40368	-6.81790	-1.41423	0.79257
0.600	2.26806	-5.05083	-2.78277	0.44905
0.700	0.95196	-3.74174	-2.78979	0.25442
0.800	0.39956	-2.77195	-2.37239	0.14414
0.900	0.16771	-2.05351	-1.88581	0.08167
1.000	0.07039	-1.52128	-1.45089	0.04627
1.100	0.02954	-1.12699	-1.09745	0.02622
1.200	0.01240	-0.83490	-0.82250	0.01485
1.300	0.00520	-0.61851	-0.61330	0.00842
1.400	0.00218	-0.45820	-0.45602	0.00477
1.500	0.00092	-0.33944	-0.33853	0.00270

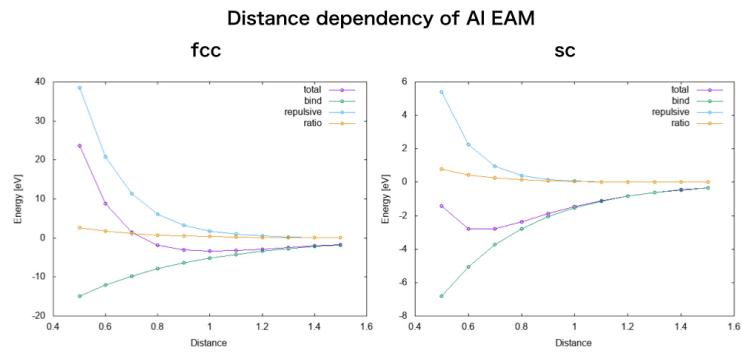


図 3.1: fcc 構造, sc 構造の各エネルギー値の距離依存性.

が困難であった。

想的な格子での計算当初，このような配位数の変化は三次元的な構造を再現する fcc, simple cubic, body centered cubic, diamond 構造をとることが不可欠と思われた。

ところが，EAM のエネルギー (2.1) 式より，配位数を n として，隣接原子だけの相互作用を考えた最隣接近似をとる．全ての隣接サイトが同じ距離にあると考える．すると

$$E_{\text{total}} = nA_0 \exp(-pr_0) - \sqrt{nB_0^2 \exp(-2qr_0)} \quad (3.1)$$

となる．

$$\begin{aligned}
 E_i &= E_{\text{repulsive}} + E_{\text{binding}} \\
 &= \sum_j \phi(r_{ij}) - \sqrt{\sum_j h(r_{ij})^2} \\
 &\quad \begin{array}{l} \phi(r_{ij}) = A_0 \exp(-pr_{ij}) \\ h(r_{ij}) = B_0 \exp(-qr_{ij}) \end{array} \\
 E_{\text{total}} &= n A_0 \exp(-pr_0) - \sqrt{n B_0^2 \exp(-2qr_0)}
 \end{aligned}$$

図 3.2: 配位数 (n) に依存する EAM の式の変形．

式 (3.1) をよく見ると隣接距離と配位数のみに依存した簡単な関係であることがわかった。

したがって，配位数に依存した EAM の振る舞いは，fcc の原子列から幾つの原子を取り出すかで見積もることが可能であることが判明した

そうして得られた total energy の距離依存性から，安定位置を出すことが可能である．この様子を図 3.3 に模式的に示した．fcc の場合と sc の場合で配位数の違いによって，トータルエネルギーの距離依存性は違う．したがって，それらの安定位置は異なっている．現実の Al の物性値にフィットして得られたパラメータ

$$A_0 = 69.1378255$$

$$B_0 = 12.47431958$$

$$P = 2.148157653$$

$$PQ = 2.893854749$$

$$Q = 0.7423170267$$

を用いる．配位数が減少するにつれて，最安定エネルギーは上昇し，距離は短くなっていく．

こうして得られた Al の配位数変化による安定位置と，各エネルギー値の変化の関係を表にまとめると次の通りとなる．表 3.3 からわかる通り，安定位置では ratio は同じ値をとることがわかる．

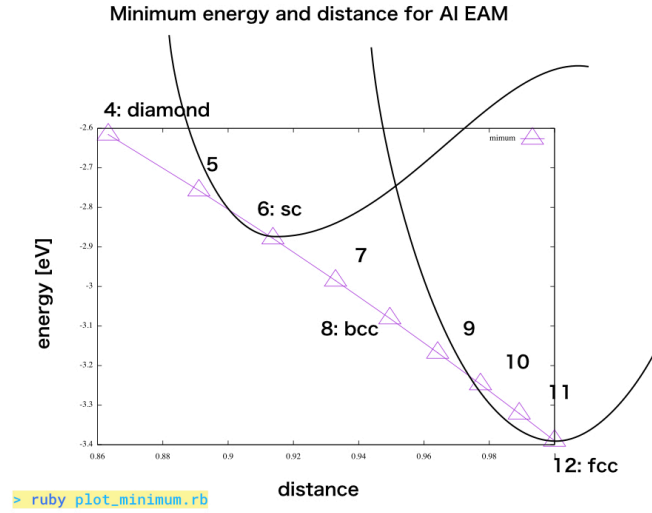


図 3.3: Al の配位数変化による最安定位置の変化.

表 3.3: Al の配位数変化による安定位置と, 各エネルギー値の変化.

coordination num	distance	repulsive	binding	total	ratio
12	1.000000	1.79000	-5.18000	-3.39000000	0.34555984
11	0.989170	1.75363	-5.07473	-3.32110070	0.34556107
10	0.977305	1.71471	-4.96195	-3.24723604	0.34557223
9	0.964195	1.67257	-4.84007	-3.16749276	0.34556837
8	0.949535	1.62674	-4.70740	-3.08066333	0.34556997
7	0.932900	1.57638	-4.56148	-2.98509958	0.34558547
6	0.913735	1.51993	-4.39839	-2.87845982	0.34556461
5	0.891045	1.45591	-4.21315	-2.75723849	0.34556311
4	0.863270	1.38121	-3.99701	-2.61580107	0.34556084

3.2 cu の配位数の変化によるエネルギー値の変化

次に cu の配位数の変化によるエネルギー値の変化を求めた.

Cu のパラメータセットは

A0_Cu = 0.1733752415

B0_Cu = 1.676027468

POQ_Cu = 2.89

Q_Cu = 2.959

である. VASP で求めた Cu の E-V 曲線にフィッティングしている [6]

表 3.4: 配位数の変化によるエネルギー値の変化.

coordination num	distance	repulsive	binding	total	ratio
12	1.006260	1.97206	-5.69937	-3.72731712	0.34601291
11	0.998480	1.93208	-5.58381	-3.65173318	0.34601431
10	0.990000	1.88854	-5.45924	-3.57069808	0.34593453
9	0.980540	1.84290	-5.32611	-3.48320914	0.34601233
8	0.970010	1.79249	-5.18043	-3.38793995	0.34601150
7	0.958065	1.73718	-5.02026	-3.28308038	0.34603393
6	0.944285	1.67523	-4.84129	-3.16605833	0.34603028
5	0.927985	1.60483	-4.63786	-3.03302303	0.34602928
4	0.908030	1.52269	-4.40048	-2.87778451	0.34602887

表 3.4の結果より, 配位数が変わると最安定位置は変化することがわかる. これを Al と同様に式 (3.1) で求めていく.

Al と Cu の最安定位置の比較を図 3.4に示した.

- Al と Cu の同じ配位数の時, 最安定位置を比べると Al の方がエネルギー値が大きい.
- Al と Cu ではエネルギーの値が配位数を変えてもほとんど変化しない.
- Al と Cu はエネルギーの値の差が約 0.0005 ある.
- 配位数が小さくするほど distance の 1 配位あたりの値の変化が大きくなる, 12 配位から 11 配位の場合は差が 0.007780, 5 配位から 4 配位の場合は差が 0.019955.

3.3 Cu の eam view の改良

eam view は EAMCalc を start させるだけで, その下請けとして, class EAM 内の atom_energy を呼び出している. そこへ, Cu の parameter set を持ち込むだけにしたかったが, 距離の規格化が必要であった. それを option 引数として指定する方法もあるが, プログラムコードとそれをコマンドとしてマニュアル化するのが煩雑である.

そこで, 別のコマンドとして作成する方法を取った.

Minimum energy and distance for Al and Cu EAM

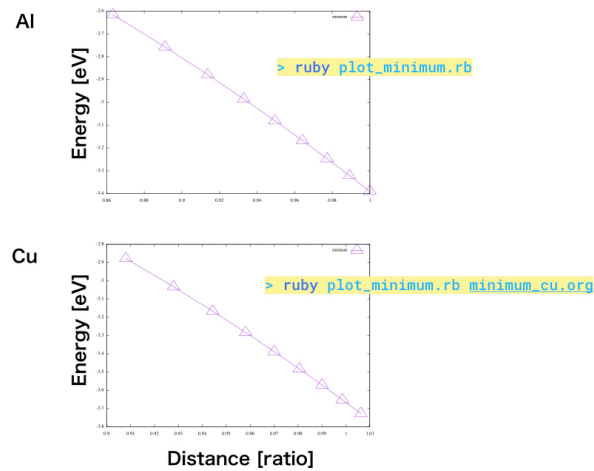


図 3.4: Al と Cu の配位数の変化による最安定位置の変化.

eam_cu

というコマンドを作った. その中身は eam と同じく, EAMCalc を呼び出す

```
EAMCalc::CLI.start(ARGV)
```

と, atom_energy の関数と parameter set を class EAM に上書きする.

Cu の EAM は西谷によって新たに開発された手法を持ちいて, 求めたパラメータを使っている.

```
#!/usr/bin/env ruby
require_relative "../lib/eam_calc"
```

```
class EAM
  A0_Cu = 0.1733752415
  B0_Cu = 1.676027468
  POQ_Cu = 2.89
  Q_Cu = 2.959
  P_Cu = POQ_Cu * Q_Cu
  CUT_OFF = 3.2
  include Math

  def atom_energy(i)
    rho = 0.0
    rep = 0.0
    ai = @atoms[i]
    ai.ni.each do |j|
```

```

        r = distance(ai.pos, @atoms[j].pos)
        r0 = 3.61496/sqrt(2.0)*1.0064
#        next if r > r0*1.2
#        r0 = 2.857701345
        del = (r-r0)/r0
        rep += A0_Cu * exp(-P_Cu * del)
        h = B0_Cu * exp(-Q_Cu * del)
        rho += h * h
    end
    bind = - sqrt(rho)
    [rep+bind, rep, bind]
end
end

EAMCalc::CLI.start(ARGV)

```

3.4 view の 2 色透明度表示

class TwoDView の draw_atom_by_val method を書き換える.

```

    val = case opts[:val]
when :ene then @normalized_ene[i]
when :nl then @normalized_nl[i]
when :ratio then @normalized_ratio[i]
end
    #rgb = hsv_to_rgb(val, 100, 100)
    rgba = val_to_rgba(@ratio[i])
end
    #@context.set_source_rgb(*rgb)
@context.set_source_rgba(*rgba)

```

元々は, val として 360 で normalize した値を hsv_to_rgb で変換していた. これを, 次のような val_to_rgba で変換するように書き換えた.

```

def val_to_rgba(val)
    v_c = 0.34556 # for Al
    v_c = 0.3460  # for Cu
    contrast = 10
    if val > v_c
        a = (val-v_c)/v_c*contrast
        return [0,0,1,a]
    else

```

```

    a = (v_c-val)/v_c*contrast
    return [1,0,0,a]
end
end

```

contrast とは対比のことであり contrast の値を大きくする程、各原子のエネルギー変化を誇張して表示する． v_c は EAM の fitting で明らかになった通り、平衡状態を示す基準値である．一方、contrast の調整は微妙である．

以下に original の原色の組み合わせである色環表示、contrast=10、contrast=100 での 2 色透明度表示の結果を示した．

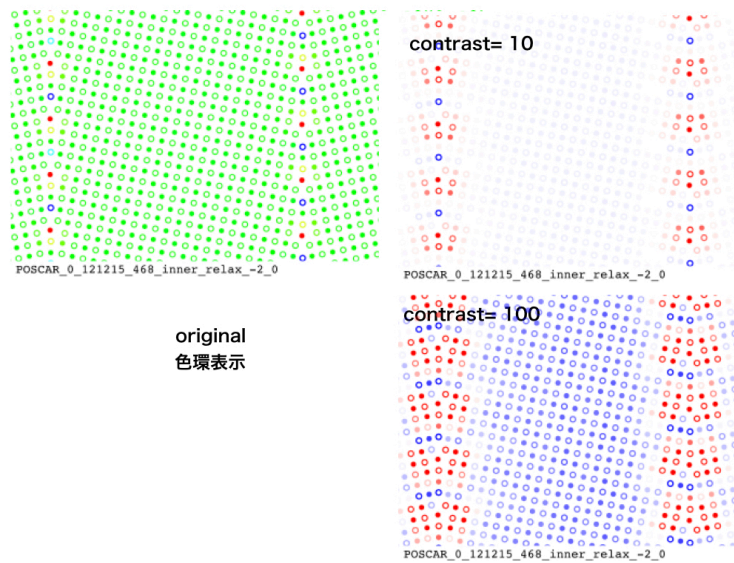


図 3.5: view の変更による表示の比較,Red:膨張,Green:緩衝領域,Blue:圧縮,Alpha:透明度.

2 色透明度表示を contrast = 10 で表示させた view では、粒界近傍に圧縮領域と膨張領域が集中している様子を綺麗に読み取ることができる．contrast=100 にするとその領域がより広がっていることがわかる．また、粒界の間の完全結晶と思われた領域も圧縮状態であることが読み取れる．また、圧縮ひずみ領域と膨張ひずみ領域とのあいだにほぼ中和するような領域が帯状に存在することがわかる．

図 3.5 で色環表示と 2 色透明度表示を比較すると、膨張状態が最安定位置である 1.0 より大きい値をとる時であり、小さい値をとるときは圧縮状態である．これを contrast で色の範囲を指定をする．contrast の範囲内では透明度によって色の強弱をつけることで、膨張エネルギーと圧縮エネルギーの大きさを視覚的にわかりやすくした．contrast の範囲外では全て原色の色で表現される．contrast の値を大きくするほど、微小なエネルギーも表示することができるため、色環表示と比較すると、contrast=100 の時に緑の完全結晶だと思われていた部分も膨張状態、圧縮状態であることがわかる．

新たな 2 色透明度表示は、original の色環表示と比べた場合、圧縮・膨張ひずみが粒界領域からどこまで広がっているかを確認するには適当である．一方で、完全結晶の領域がどの程度であるかを確認するには、original の緑の領域表示が比較的適切であることが

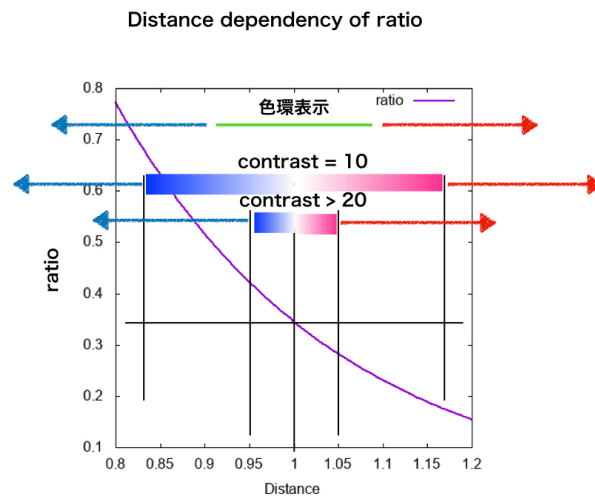


図 3.6: 色環表示と 2 色透明度表示の比較

判明した。これらの結果を踏まえて、2 色透明度表示および色環表示の切り替え、さらに contrast の調整などより interactive な viewer の開発が望まれる。

第4章 結論

配位数の異なる構造での最安定位置での ratio はほぼ一致していることがわかり，EAM を用いた歪み解析正当性が確認された．

また歪み解析において，EAM のエネルギー表記において，配位数と隣接原子だけの相互作用を考えた最隣接近似をとり，全ての隣接原子が同じ距離にあると仮定した結果，隣接距離と配位数のみに依存した関係であった．これは配位数の変化は三次元的な fcc や sc のような構造が不可欠ではなく，隣接距離と配位数のみに依存した関係であることがわかる．一方，sc の最安定の格子定数 $a=0.65$ は fcc の最近接距離と比較することが困難でした．Cu の EAM 表示を可能としたことにより，配位数が異なる Cu は最安定位置が変化することが判明した．

2 色透明度表示では original の色環表示と比べた場合，original の色環表示と比べた場合，圧縮・膨張ひずみが粒界領域からどこまで広がっているかを確認するには適当である．一方で，完全結晶の領域がどの程度であるかを確認するには，original の緑の領域表示が比較的適切であることが判明した．これらの結果を踏まえて，2 色透明度表示および色環表示の切り替え，さらに contrast の調整などより interactive な viewer の開発が望まれる．

関連図書

- [1] "結晶の圧縮・膨張のスペクトル表示", 菊岡 達也, 関西学院大学理工学部卒業論文, 2017.
- [2] "固体物理の基礎", 西谷 滋人, (森北, 2006), pp.43-7.
- [3] "Ruby とは", <https://www.ruby-lang.org/ja/about/> , (saccessed 2019/2/15).
- [4] "cairo:2次元画像描画ライブラリ-るびま", <https://magazine.rubyist.net/articles/0019/0019-cairo.html> (saccessed 2019/2/15).
- [5] 金属材料と結晶構造, <http://solid4.mech.okayama-u.ac.jp/金属材料と結晶構造.pdf> (saccessed 2019/2/15).
- [6] "Cu の粒界エネルギーの第一原理計算", 山田 智子, 関西学院大学 理工学部卒業論文, 2018.

謝辞

本研究を進めるにあたり、終始多大なるご指導、御鞭撻をいただいた西谷滋人教授に対し、深くご御礼申し上げます。また、本研究の進行に伴い、様々な助力、知識の供給をいただきました西谷研究室の同輩に心から感謝の意を示します。

付 録 A eam と eam_cu の違い

新たに開発した eam_cu の使い方と結果を残しておく.

```
> bundle exec exe/eam_cu show_atom CONTCAR_0_2213_16_relax_1 1 ratio 4
exe/eam_cu:10: warning: already initialized constant EAM::CUT_OFF
/Users/bob/materials_science/boundary/lib/pseudo_vasp/eam.rb:137:
warning: previous definition of CUT_OFF was here
```

no	x	y	z	nei	total	diff	rep	bind	ratio
0	6.56494	0.53722	1.80748	12	-3.65684	-0.267	2.02802	-5.68486	0.36
1	6.15537	2.21984	0.00000	10	-3.53771	-0.148	2.16383	-5.70155	0.38
2	-0.05954	1.72152	1.80748	10	-3.55031	-0.160	2.20892	-5.75923	0.38
3	8.43274	1.15509	0.00000	12	-3.66137	-0.271	2.05590	-5.71727	0.36
4	7.80638	2.80118	1.80748	12	-3.64611	-0.256	2.10561	-5.75172	0.37
5	9.98511	3.50240	-0.00000	14	-3.69923	-0.309	1.41474	-5.11397	0.28
6	7.21220	4.41552	-0.00000	12	-3.63873	-0.249	2.11106	-5.74978	0.37
7	8.84631	5.00770	1.80748	10	-3.53619	-0.146	2.20799	-5.74418	0.38
8	4.98630	-0.06301	-0.00000	10	-3.55031	-0.160	2.20892	-5.75923	0.38
9	3.38691	0.50341	1.80748	12	-3.66137	-0.271	2.05590	-5.71727	0.36
10	3.80047	2.17915	0.00000	10	-3.53619	-0.146	2.20799	-5.74418	0.38
11	4.93928	3.68445	1.80748	14	-3.69923	-0.309	1.41474	-5.11397	0.28
12	1.51911	1.12129	0.00000	12	-3.65684	-0.267	2.02802	-5.68486	0.36
13	2.16637	2.77133	1.80748	12	-3.63873	-0.249	2.11106	-5.74978	0.37
14	2.76055	4.38567	-0.00000	12	-3.64611	-0.256	2.10561	-5.75172	0.37
15	1.10954	4.96701	1.80748	10	-3.53771	-0.148	2.16383	-5.70155	0.38

data for atom 0

0	6.56494	0.53722	1.80748	
1	6.15537	2.21984	0.00000	2.50318
3	8.43274	1.15509	0.00000	2.67160
4	7.80638	2.80118	1.80748	2.58199
6	7.21220	4.41552	-0.00000	2.53152
7	8.84631	5.00770	1.80748	2.51470
8	4.98630	-0.06301	-0.00000	2.47374
9	3.38691	0.50341	1.80748	3.17821

11	4.93928	3.68445	1.80748	2.88314
19	6.15537	2.21984	3.61496	2.50318
23	8.43274	1.15509	3.61496	2.67160
29	7.21220	4.41552	3.61496	2.53152
33	4.98630	-0.06301	3.61496	2.47374

> bundle exec exe/eam show_atom POSCAR_Al_2213_-2_-2 1 ratio 4

no	x	y	z	nei	total	diff	rep	bind	ratio
0	7.33779	0.64500	2.02070	11	-3.29930	0.091	1.58724	-4.88654	0.32
1	6.88802	2.56684	0.00000	10	-3.23603	0.154	1.62923	-4.86525	0.33
2	-0.06193	1.86214	2.02070	10	-3.24398	0.146	1.67549	-4.91947	0.34
3	9.42131	1.29113	0.00000	11	-3.30568	0.084	1.61448	-4.92016	0.33
4	8.73049	3.17307	2.02070	11	-3.29326	0.097	1.61176	-4.90502	0.33
5	11.13565	3.91114	0.00000	11	-3.20086	0.189	1.11181	-4.31267	0.26
6	8.05361	5.01567	0.00000	10	-3.22473	0.165	1.55311	-4.77784	0.33
7	9.86476	5.64335	2.02070	10	-3.23834	0.152	1.67024	-4.90858	0.34
8	5.57406	0.01653	0.00000	10	-3.24398	0.146	1.67549	-4.91947	0.34
9	3.78532	0.58753	2.02070	11	-3.30568	0.084	1.61448	-4.92016	0.33
10	4.22877	2.49753	-0.00000	10	-3.23834	0.152	1.67024	-4.90858	0.34
11	5.49966	4.22974	2.02070	11	-3.20086	0.189	1.11181	-4.31267	0.26
12	1.70179	1.23366	0.00000	11	-3.29930	0.091	1.58724	-4.88654	0.32
13	2.41762	3.12521	2.02070	10	-3.22473	0.165	1.55311	-4.77784	0.33
14	3.09449	4.96781	-0.00000	11	-3.29326	0.097	1.61176	-4.90502	0.33
15	1.25203	5.57404	2.02070	10	-3.23603	0.154	1.62923	-4.86525	0.33

data for atom 0

0	7.33779	0.64500	2.02070	
1	6.88802	2.56684	0.00000	2.82470
3	9.42131	1.29113	0.00000	2.97352
4	8.73049	3.17307	2.02070	2.88630
6	8.05361	5.01567	0.00000	2.85895
7	9.86476	5.64335	2.02070	2.82541
8	5.57406	0.01653	0.00000	2.75480
11	5.49966	4.22974	2.02070	3.24771
19	6.88802	2.56684	4.04140	2.82470
23	9.42131	1.29113	4.04140	2.97352
29	8.05361	5.01567	4.04140	2.85895
33	5.57406	0.01653	4.04140	2.75480

eam_cu では, Cu の POSCAR を読み込んで位置とエネルギーを出力する. これらのデー

タをもとに，view で ratio あるいは energy とサイトを視覚化している．元のコード eam と違い，原子間距離や total energy が異なっている．

付 録B eam_plot.rbのソース

```
require 'numo/gnuplot'

fcc_atoms = [[0.0, 0.5, 0.5], [0.5, 0.0, 0.5], [0.5, 0.5, 0.0],
             [0.0, -0.5, 0.5], [-0.5, 0.0, 0.5], [-0.5, 0.5, 0.0],
             [0.0, 0.5, -0.5], [0.5, 0.0, -0.5], [0.5, -0.5, 0.0],
             [0.0, -0.5, -0.5], [-0.5, 0.0, -0.5], [-0.5, -0.5, 0.0]]

sc_atoms = [[0.0, 0.0, 1.0], [0.0, 1.0, 0.0], [1.0, 0.0, 0.0],
            [-1.0, 0.0, 0.0], [0.0, -1.0, 0.0], [0.0, 0.0, -1.0]]

include Math
A_i = [0.0, 0.0, 0.0].freeze

A0 = 69.1378255
B0 = 12.47431958
P = 2.148157653
POQ = 2.893854749
Q = 0.7423170267
p $p0 = POQ*Q

def atom_energy(atoms, lat_const)
  rho = 0.0
  rep = 0.0
  atoms.each do |aj|
    r = distance(A_i, aj, lat_const)
    rep += A0 * exp(-$p0 * r)
    h = B0 * exp(-Q * r)
    rho += h * h
  end
  bind = - sqrt(rho)
  [rep, bind, rep + bind]
end
```

```

def distance(i_atom, j_atom, lat_const)
  dist = 0.0
  3.times do |i|
    tmp = i_atom[i] - j_atom[i]
    dist += tmp * tmp
  end
  Math.sqrt(dist) * lat_const
end

def input_parameters
  puts "Usage: ruby eam_plot.rb LOW HIGH NN COORD_NUM"
  p ARGV[0], ARGV[1], ARGV[2], ARGV[3]
  low = ARGV[0] || 0.5
  high = ARGV[1] || 1.50
  n = ARGV[2] || 10
  coord_num = ARGV[3].to_i || 12
  return [low, high, n, coord_num]
end

def print_table(params, atoms)
  low, high, n = params
  printf("|%10s |%10s |%10s |%10s |%10s\n",
'distance', 'repulsive', 'binding', 'total', 'ratio')
  5.times { printf('|-----:|') }
  puts('')

  data = [[], [], [], [], []]

  dx = (high.to_f - low.to_f) / n.to_i
  distance = low.to_f - dx
  (n.to_i + 1).times do
    distance += dx
    lat_const = 4.0414 * distance
    rep, bind, ene = atom_energy(atoms, lat_const)
    ratio = - rep / bind
    printf("|%10.3f |%10.5f |%10.5f |%10.5f |%10.5f\n",
distance, rep, bind, ene, ratio)
    [distance, rep, bind, ene, ratio].each_with_index { |val, i| data[i] << val }
  end
  return data
end

```

```

end

def data_print(data)
  data.each_with_index do |ele, i|
    head = %w[distance e_rep e_bind e_total ratio][i]
    line = ele[0..-2].inject("#{head}=") { |s, e| s << format('%7.3f,', e) }
    puts line << format("%7.3f\\n", ele[-1])
  end
end

params = input_parameters
p ['coordination number:', params[-1]]
p atoms = fcc_atoms[0..params[-1]-1]
#p atoms = sc_atoms[0..params[-1]-1]
data = print_table(params, atoms)

def plot(data)
  distance, e_rep, e_bind, e_total, ratio = data
  require 'numo/gnuplot'
  file = 'fcc_plot.png'
  Numo.gnuplot do
    set output: file
    set term: 'png'
    set ylabel: 'Energy [eV]'
    set xlabel: 'Distance'
    plot [distance, e_total, title: 'total', w: :lp, pt: 6],
        [distance, e_bind, title: 'bind', w: :lp, pt: 6],
        [distance, e_rep, title: 'repulsive', w: :lp, pt: 6],
        [distance, ratio, title: 'ratio', w: :lp, pt: 6]
    system "open ./#{file}"
  end
end

plot(data)

```

eam_plot では ARGV[0], ARGV[1], ARGV[2], ARGV[3] で数値を読み込んでいる。 ARGV[0], ARGV[1] では distance の範囲を受け取り, ARGV[2] で distance の個数を指定している。 ARGV[3] で配位数の指定をできるようにした。 実行例を下に示した。

```

>ruby eam_plot.rb 0.5 1.5 10 4
2.148157652979355
Usage: ruby eam_plot.rb LOW HIGH NN COORD_NUM
"0.5"

```


"1.5"

"10"

"4"

["coordination number:", 4]

[[0.0, 0.5, 0.5], [0.5, 0.0, 0.5], [0.5, 0.5, 0.0], [0.0, -0.5, 0.5]]

distance	repulsive	binding	total	ratio
-----:	-----:	-----:	-----:	-----:
0.500	12.84558	-8.63790	4.20768	1.48712
0.600	6.95263	-6.98683	-0.03420	0.99511
0.700	3.76309	-5.65135	-1.88826	0.66587
0.800	2.03676	-4.57114	-2.53438	0.44557
0.900	1.10239	-3.69740	-2.59501	0.29815
1.000	0.59667	-2.99067	-2.39401	0.19951
1.100	0.32294	-2.41903	-2.09609	0.13350
1.200	0.17479	-1.95665	-1.78186	0.08933
1.300	0.09461	-1.58265	-1.48805	0.05978
1.400	0.05121	-1.28014	-1.22894	0.04000
1.500	0.02771	-1.03545	-1.00774	0.02677