

卒業論文

VASPにおける構造緩和計算法

関西学院大学 理工学部 情報科学科
2689 中田 悠介

2006年3月

指導教員 西谷 滋人 教授

概要

本研究においては、VASP という第一原理計算パッケージに用意されている構造緩和手法の計算時間の比較を行った。対象系として $\text{Fe}_{53}\text{Cu}_1$ 系については特に詳しく検討した。また、比較のために Ti の層状構造についても調べた。

構造緩和は、多次元空間で最小値を求める一般的な手法である、最急降下法、共役勾配法、準ニュートン法を用いて実行した。

$\text{Fe}_{53}\text{Cu}_1$ 系では、計算時間が最も早かったのは準ニュートン法であった。そして、準ニュートンは、収束性も他の手法に比べてきれいに閾値に収束をしていた。したがって、準ニュートン法がこの系での構造緩和において最良の手法であると結論できる。

しかし、比較の対象系として行った Ti の層状構造の緩和計算では、計算時間は共役勾配法が一番いい結果が得られた。従って、共役勾配法と準ニュートン法とでは一概に優劣はつけがたく、問題に依存する。少なくとも、このような構造緩和に対しては最急降下法は妥当ではない事が判明した。

目次

第1章	緒言	5
第2章	手法	7
2.1	構造緩和における計算手法	7
2.1.1	Schrodinger 方程式	9
2.2	最急降下法	9
2.3	共役勾配法	10
2.3.1	共役な方向	10
2.4	準ニュートン法	12
2.4.1	Hesse 行列	14
2.5	maple での各手法の模式図の出力方法	14
第3章	結果	17
3.1	最急降下法における構造緩和	19
3.2	共役勾配法における構造緩和	22
3.3	準ニュートン法における構造緩和	23
3.4	各手法における鉄原子のポジション	23
3.4.1	最急降下法における鉄原子のポジション	24
3.4.2	共役勾配法における鉄原子のポジション	24
3.4.3	準ニュートン法における鉄原子のポジション	24
3.5	各手法における計算時間の比較	31
3.6	Ti の層状構造の緩和における計算時間の比較	31
3.7	検討結果	33
第4章	まとめ	34
付録A		35
A.1	vasp の使用方法	35
A.2	構造緩和のグラフの描写方法	35
A.2.1	INCAR ファイルの変更	36
A.2.2	POSCAR ファイルの変更	37
A.2.3	KPOINTS ファイルの変更	39
A.2.4	OUTCAR ファイルの編集	39

A.2.5 gnuplot での描画	40
A.3 各手法における鉄原子のポジションの作成方法	41
A.3.1 OUTCAR ファイルの編集	42
A.4 DOS の描写方法	42
A.4.1 INCAR ファイルの変更	43
A.4.2 POSCAR ファイルの変更	44
A.4.3 KPOINTS ファイルの変更	44
A.4.4 DOSCAR ファイルのデータ編集	45
A.4.5 maple での描画	45
A.5 VASP マニュアルの和訳	46

目 次

2.1	構造緩和のフローチャート.	8
2.2	最急降下法の模式図.	10
2.3	共役勾配法の模式図.	11
2.4	準ニュートン法の模式図.	13
3.1	全原子の位置と対称性を考えるための模式図 (正面から見た場合). . .	17
3.2	1回目の構造緩和において各原子に働いている力.	19
3.3	構造緩和後の原子位置.	20
3.4	最急降下法での構造緩和.	22
3.5	共役勾配法での構造緩和.	22
3.6	準ニュートン法での構造緩和.	23
3.7	最急降下法における鉄原子の z 軸方向へのポジションの移り変わり. . .	24
3.8	共役勾配法における鉄原子の z 軸方向へのポジションの移り変わり. . .	25
3.9	準ニュートン法における鉄原子の z 軸方向へのポジションの移り変 わり.	25
A.1	DOS のグラフ.	46

表 目 次

3.1	pos1 から pos9 の座標.	18
3.2	1 回目の構造緩和において各原子に働いている力.	18
3.3	構造緩和によって動く原子の移動距離.	20
3.4	最急降下法を用いた場合に, 構造緩和の各回において各原子にかかっ ている z 軸方向の力 ($\text{eV}/\text{\AA}$).	21
3.5	構造緩和 11 回目以降のセルフコンシステントループの回数.	21
3.6	共役勾配法を用いた場合に, 構造緩和の各回において各原子にかかっ ている z 軸方向の力 ($\text{eV}/\text{\AA}$).	26
3.7	準ニュートン法を用いた場合に, 構造緩和の各回において各原子に かかっている z 軸方向の力 ($\text{eV}/\text{\AA}$).	27
3.8	最急降下法での各回の鉄原子の z 軸方向のポジションの移動距離. . .	28
3.9	共役勾配法での各回の鉄原子の z 軸方向のポジションの移動距離. . .	29
3.10	準ニュートン法での各回の鉄原子の z 軸方向のポジションの移動距離.	30
3.11	各手法の計算時間.	31
3.12	Ti の層状構造の緩和における各手法の計算時間.	32

第1章 緒言

第一原理による電子状態計算手法によって扱える原子の数は、100-1000個程度までであり、それはアボガドロ定数に遠く及ばない。1000原子のオーダーでようやく最も簡単な構造のたんぱく質(或いはアミノ酸)が扱えるかもしれないというレベルである。また、実際に計算で扱う時間の問題も存在する。第一原理分子動力学法で扱える時間は、最大でも数ピコから数十ピコ秒程度の分子動力学しか扱えない。実時間で1秒間を実際に計算の上で再現させることは現実問題として不可能に近い。更に、電子状態を解くために用いる近似手法(密度汎関数法、局所密度近似、一電子近似、断熱近似等)は、現実の化学反応を正確には記述できるとは言い難く、ましてや生体内の代謝反応やDNAの複製過程、植物の光合成のような大規模で複雑な反応を第一原理計算だけで再現することは著しく困難と言わざるを得ない。これを乗り越えようとする試みも存在はする。オーダー N 法や、ハイブリッド法は、1000原子より一桁以上大きなサイズの系を扱えることが目標であり可能となりつつある。ただ、方法論として未だ発展途上で、精度に関しての十分な検証が必要である。一方、現実の化学反応等をより精度良く記述するために、TDDFTのように時間依存性を考慮したり、断熱近似を越えるような試み、局所密度近似を越える試みなどがなされている。この第一原理計算のプログラムであるVASPで構造緩和を行う[2]。

構造緩和(relaxation)は、電子構造計算において最も計算量が要求されるルーチンである。この構造緩和というのは、通常、ユニットセル内の原子に働く力によって当該原子をより安定な位置、方向に動かすことによって遂行される。このユニットセルは原子が十数個詰まった箱と考えてもらってよい。構造最適化計算が終了したかどうかは、通常最適化すべき原子に働く力の値がどのくらい小さくなったかによって判定する[3]。

計算上、完全にエネルギーをゼロにするのは不可能(結晶構造の持つ対称性から力の働かない原子が存在する場合もある)である。しかし、実効上ある値より小さくなれば、もう原子はほとんど動かなくなる。そこで、電子構造計算パッケージVASPを用いて精度よくかつ素早く構造緩和させる最善の方法を考えることを目的としている。このVASPというのは、密度汎関数法を用いた平面波-擬ポテンシャル法電子構造計算プログラムで、その特徴は精度の高い構造最適化計算と、第一原理分子動力学計算にある。このVASPの基本となるアルゴリズムは行列-ベクトル演算、行列-行列演算、高速フーリエ変換と同様である。

本研究で対象とするのはFe-Cu系である。極低炭素鋼(Interstitial Free Steel:IF

鋼)に1~2%のCuを添加し、Cuの析出硬化を利用したCu添加極低炭素鋼板は高強度と高 \bar{r} 値を併せもつ高張力銅板として近年、注目されている。一方では、中性子照射をうける原子炉の圧力容器材でFe中に固溶したCuが550K程度の運転温度でもクラスター化して脆化を引き起こすことが問題となっており、その微細組織が調べられてきた。 α -Fe中でのCuの析出は古くから調べられており平衡相 ϵ -Cuの出現前の準安定なbcc構造のCu粒子(以後bcc-Cuと呼ぶ)が存在するとの認識でほぼ一致している。bcc-Cuは臨界半径に達するとbccからfccに相変態して平衡相の ϵ -Cuになると考えられている [1]。

西谷は、この系でのbcc-Cuの析出現象を第一原理計算から求める新奇な計算手法を開発した。この計算においては、原子種の違いによる構造緩和が最終の計算結果に与える影響が大きい事が判明している。従って、大規模な系において精度の高い緩和計算を高速に実行する事が、研究遂行上、重要となっている。そこで本研究においては $\text{Fe}_{53}\text{Cu}_1$ で構成するbcc構造を対象とした。

第2章 手法

2.1 構造緩和における計算手法

構造緩和は図 2.1 のようなループを繰り返して計算を終了させる。セルフコンシステントループとはまず, Schrodinger 方程式 (小節 2.1.1 参照) の計算において各原子のエネルギー準位を求める。この Schrodinger 方程式は次のように表される。

$$H\psi = E\psi \quad (2.1)$$

次に, 波動関数 ψ を計算する。このときの波動関数の変更分を摂動としてエネルギーを更新する。そして, ψ とエネルギーをそのときのポテンシャルのもとでなるべく正確に解き直す。この作業を繰り返していくと, エネルギーの更新分は小さくなり, 十分にセルフコンシステントな解が得られるというループである。そうして得られた電子構造をもとに原子構造の構造緩和の計算を行う。構造緩和をフローチャートで示すと図 2.1 のように表せる。構造緩和は一般的な数値計算の問題である。「多次元の最小値を求める」問題と見なすことが可能である。そこで, vasp では最小値を求める一般的な手法である。

1. 最急降下法 (2.2 節参照)
2. 共役勾配法 (2.3 節参照)
3. 準ニュートン法 (2.4 節参照)

を使って構造緩和を実行することが可能である。

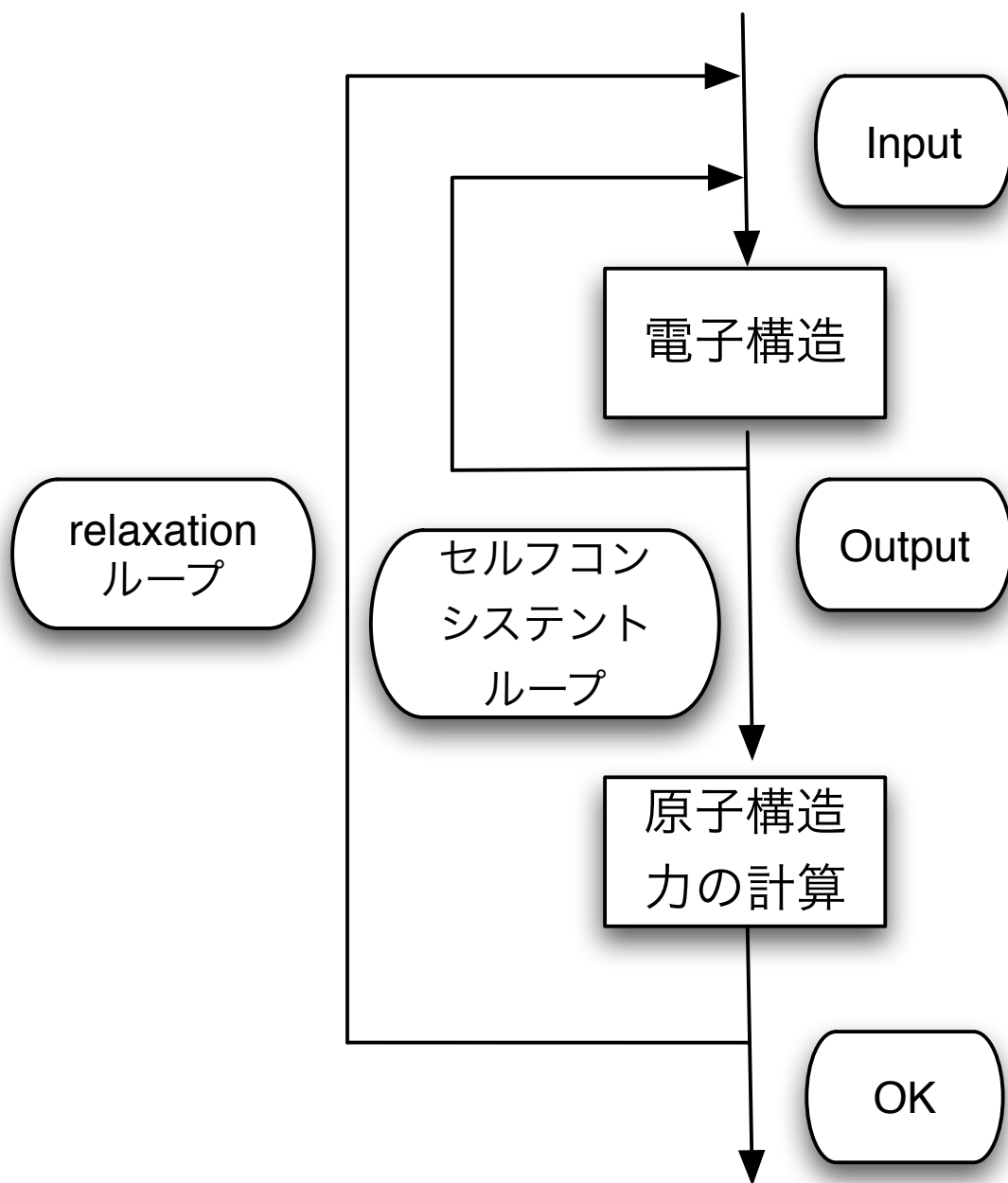


図 2.1: 構造緩和のフローチャート.

2.1.1 Schrodinger 方程式

Schrodinger 方程式は量子力学の一形式である波動力学の基礎方程式であり、状態ベクトル数の時間発展を記述する。

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = \hat{H} |\psi\rangle \quad (2.2)$$

以上は、時間 t を含む形式（時間を含む Schrodinger 方程式）である。ここで、 \hat{H} はハミルトニアン演算子（物理学におけるエネルギーを表す関数または演算子もしくはは行列のこと）というエルミート演算子であり、対応する古典系のハミルトニアンを正準量子化（古典力学的な理論から量子力学的な理論を推測する手法（量子化）の一種である。具体的には、古典的なハミルトン力学における正準変数を、正準交換関係をみたすようなエルミート演算子に置き換えること）する事により得られることが多い。

$$|\psi(\mathbf{r}, t)\rangle = e^{-\frac{iEt}{\hbar}} |\phi(\mathbf{r})\rangle \quad (2.3)$$

式 (2.3) はブラ - ケット表記（量子力学に関する洞察をそのまま記号形式に置き換え、ベクトル表記したもの）をした状態ベクトルで、適当なヒルベルト空間（完備な内積空間、すなわち、内積の定義された、ベクトル空間であって、その内積から導かれるノルム（平面あるいは空間における幾何学的ベクトルの”長さ”の概念の一般化であり、ベクトル空間に対して「距離」を与えるための数学の道具）によって距離を入れるとき、距離空間として完備となるような位相ベクトル空間のこと）の元である。式 (2.3) でハミルトニアン部分が時間に依存しない場合は、 $|\phi\rangle$ として、式 (2.2) に代入すると、時間を含まない Schrodinger 方程式

$$\hat{H} |\phi\rangle = E |\phi\rangle \quad (2.4)$$

が得られる。[2]

2.2 最急降下法

最急降下法は、点 P_0 から出発する。点 P_i から下り坂の方向 $-\Delta f(P_i)$ に移動し、その点を P_{i+1} とする。これを必要な回数だけ行う。すなわち、ある適当な初期パラメーターからはじめて、その値を繰り返し更新（修正）することにより最適なパラメータの値を求める方法。この方法は2次形式であっても多数の小ステップを取る。1ステップでその線に沿って極小まで一気に進むことを期待したいが、図 2.2 のように極小での新しい勾配の向きはもとの直線にほぼ垂直に進む傾向がある。この図 2.2 の作成方法に関しては 2.5 節に掲載する。[4]

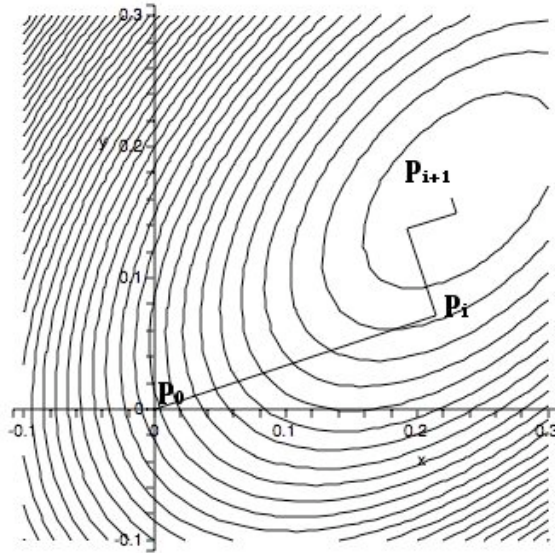


図 2.2: 最急降下法の模式図.

2.3 共役勾配法

一次元の最小化を利用した方法で, 勾配の計算を行う. 具体的には最初の方向に沿って最小を求め (一次元の最小化), その点から出発して共役な方向 (小節 2.3.1 参照) に沿って最小を求める. それを何度もくり返すことによって関数値の最小を求めるという手法というのが共役勾配法である. [5]

2.3.1 共役な方向

特定の点 P を座標系の原点とし, この点を基準とした座標を x とするとき, 関数 f は次の Taylor 級数 (小節 2.3.1 参照) で近似できる.

$$f(x) = c - bx + \frac{1}{2}xAx \quad (2.5)$$

ここで,

$$c = f(P) \quad (2.6)$$

$$b = -\Delta f|_p \quad (2.7)$$

$$[A]_{ij} = \frac{\delta^2 f}{\delta x_i \delta x_j} \Big|_p \quad (2.8)$$

である。行列 A は関数の点 P における Hesse 行列 (小節 2.4.1 参照) という。近似 (2.5) では、 f の勾配は次のように求められる。

$$\Delta f = Ax - b \quad (2.9)$$

ある方向に沿って進むと勾配 Δf は、

$$\delta(\Delta f) = A(\delta x) \quad (2.10)$$

式 (2.10) のように変化する。ある方向 u に沿って極小まで進み終え、次に新しい方向 v に沿って進むとすると、この v に沿って進むことが、 u に沿った最小化を壊さないための条件は、勾配が u に垂直であり続けることである。つまり、

$$u\delta(\Delta f) = uAv = 0 \quad (2.11)$$

この式 (2.11) が u, v について成り立つとき、この 2 個のベクトルは共役であるという。この関係がベクトルの集合のすべての要素の対について成り立つとき、このベクトルの集合は共役な集合であるという。この共役勾配法を表したのが、図 2.3 である。この図 2.3 の作成方法に関しても 2.5 節に掲載する。 [5]

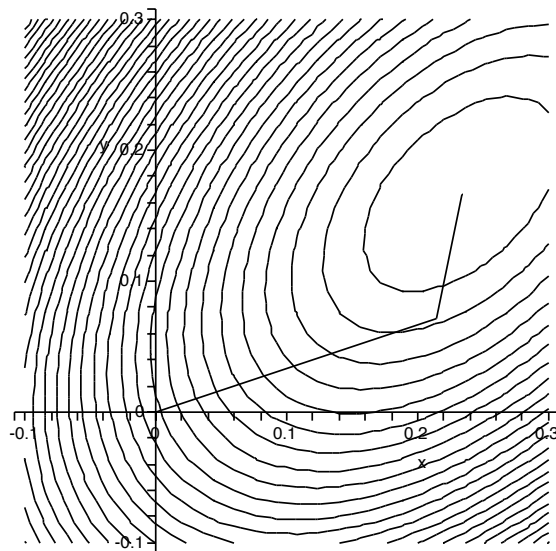


図 2.3: 共役勾配法の模式図.

Taylor 級数

数学において、テイラー展開 (Taylor expansion) とは、無限回微分可能な関数 $f(x)$ から、テイラー級数 (Taylor series) と呼ばれる、負冪の項を持たない冪級数

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \quad (2.12)$$

を得ることを言い、この式 (2.12) がもとの関数 $f(x)$ に一致するとき、 $f(x)$ はテイラー展開可能であるという。名称は数学者ブルック・テイラーに由来する。厳密にはこの展開は $x=a$ の近傍でのみ考えるものであり、 $x=a$ におけるテイラー展開とか、 $x=a$ のまわりでのテイラー展開などという。 $a=0$ のとき

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n \quad (2.13)$$

式 (2.13) を特にマクローリン展開と呼ぶ。テイラー展開がある大域的な領域の各点で可能な関数は、その領域において解析的である、またはその領域上の解析関数であるという。関数が無限回微分可能であっても、テイラー級数が元の関数とすべての x で一致するとは限らない。一致するかどうかは、テイラーの定理 (微積分学における定理の一つで、関数がある一点における高階の微分係数を用いて近似するものである) における剰余項 R_n が 0 に収束するかどうかによって判定できる、ここで R_n はある $c \in (a, x)$ が存在して、

$$R_n(x) = \frac{f^{(n)}(c)}{n!} (x-a)^n \quad (2.14)$$

と書ける。または積分を用いて、次のように表せる [2]

$$R_n(x) = \int_a^x \frac{(x-t)^{n-1}}{(n-1)!} f^{(n)}(t) \quad (2.15)$$

2.4 準ニュートン法

準ニュートン法は共役勾配法と同じく、直線上の最小化を繰り返して情報を集め N 回の直線上の最小化で N 次元の 2 次形式を厳密に最小化する手法である。関数 $f(x)$ が式 (2.5) のように 2 次形式で近似できると仮定する。2 次形式のパラメータ A, b についての情報は最初から持ち合わせているのではなく、関数値の計算や直線上の最小化によって拾い集めることができるだけである。基本的な考え方は反復によって Hesse 行列 (小節 2.4.1 参照) の逆行列 A^{-1} の近似を高めていくことである。つまり、

$$\lim_{i \rightarrow \infty} H_i = A^{-1} \quad (2.16)$$

この極限が無大ではなく、 N 回の反復によってできれば良い。このとき極小点 x_m は式 (2.9) より、

$$Ax_m = b \quad (2.17)$$

を満たす。また現在の点 x_i とすると、

$$Ax_i = \Delta f(x_i) + b \quad (2.18)$$

が成り立つ。これら2式の差に逆行列 A^{-1} を掛けると、

$$x_m - x_i = A^{-1}[-\Delta f(x_i)] \quad (2.19)$$

正確な A^{-1} を求めることによって極小点が求まる。この式 (2.19) の表式を用いて、

$$x_{i+1} - x_i = H_{i+1}(\Delta f_{i+1} - \Delta f_i) \quad (2.20)$$

として、 $H_{i+1} = H_i \times$ 補正項という形の式を用いて H_i の近似を高めていく。この準ニュートン法を表したのが、図 2.4 である。この図 2.4 の作成方法に関しても 2.5 節に掲載する。 [6]

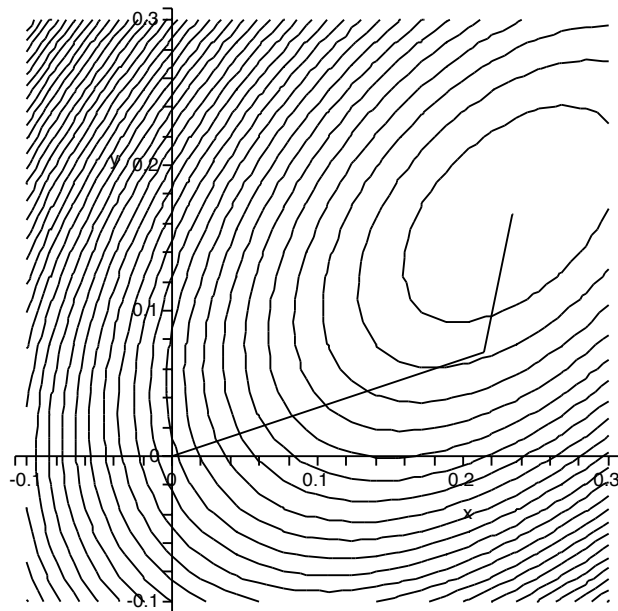


図 2.4: 準ニュートン法の模式図.

2.4.1 Hesse 行列

Hesse 行列は $f(x_1, x_2, \dots, x_n)$ を, $a = [a_1, a_2, \dots, a_n]$ の近傍で定義された 2 回連続偏微分可能な関数とし, $f_{x_i}(a) = 0$ ($i = 1, 2, \dots, n$) とする. このとき, 次の行列 H を Hesse 行列という.

$$H = \begin{pmatrix} f_{x_1x_1}(a) & f_{x_1x_2}(a) & \cdots & f_{x_1x_n}(a) \\ f_{x_2x_1}(a) & f_{x_2x_2}(a) & \cdots & f_{x_2x_n}(a) \\ \vdots & \vdots & \ddots & \vdots \\ f_{x_nx_1}(a) & f_{x_nx_2}(a) & \cdots & f_{x_nx_n}(a) \end{pmatrix} \quad (2.21)$$

この行列を使うと, $f(a)$ の極値は, 次のように判定できる.

1. H が正値 $\Rightarrow f(a)$ は極小値
2. H が負値 $\Rightarrow f(a)$ は極大値
3. H が正則であり, 正値でも負値でもない. $\Rightarrow f(x)$ は, $x = a$ で極値をとらない.

そして, 式 (2.22) というように Hesse 行列を定義することもある.

$$H = \begin{pmatrix} f_{x_1x_1} & f_{x_1x_2} & \cdots & f_{x_1x_n} \\ f_{x_2x_1} & f_{x_2x_2} & \cdots & f_{x_2x_n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{x_nx_1} & f_{x_nx_2} & \cdots & f_{x_nx_n} \end{pmatrix} \quad (2.22)$$

2.5 maple での各手法の模式図の出力方法

図 2.2, 図 2.3, 図 2.4 で使用したグラフのプログラムを載せておく. まず, 最初に出力するのに必要な定義のコードから,

```
E:=(dx1,dx2)->k1*(dx1-0.1)^2+k2*(dx1-dx2-0.2)^2+k3*(dx2-0.3)^2;
k1:=1:k2:=1:k3:=1:
with(LinearAlgebra):
p:=Vector([0,0]):
g:=Vector([1,1]):
%E を x で微分し関数の最小値を関数を出力する関数を定義%%
FindMinDim:=proc(p::Vector,g::Vector)
return solve(diff(E(op(convert(p+x*g,list))),x)=0,x);
end proc;
```


%%Eをxで微分したものと、yで微分したものに分けて2行1列の行列にし、それぞれにx=0,y=0を代入したものを出力する関数を定義%%

```
dFunc:=proc(p::Vector)
  local dx,dy;
  dx:=subs({x=p[1],y=p[2]},diff(E(x,y),x));
  dy:=subs({x=p[1],y=p[2]},diff(E(x,y),y));
  return Vector([dx,dy]);
end proc;
with(plots):
c1:=contourplot(E(x,y),x=-0.1..0.3,y=-0.1..0.3,contours=40,color=black):
```

ここまではすべての手法に用いる定義となるコードである。ここからは最急降下法についてのプログラムコードをのせる。

```
p_s:=[]:
p:=Vector([0,0]):
for i from 1 to 5 do
  p_s:=[op(p_s),convert(p,list)];
  g:=dFunc(p);
  xx:=FindMinDim(p,g);
  p:=p+xx*g;
end do:
pp1:=pointplot(p_s,connect=true):
display(pp1,c1);
```

次に共役勾配法のプログラムコードを載せる。

```
p_s:=[]:
p:=Vector([0,0]):
p_s:=[op(p_s),convert(p,list)]:
xi:=dFunc(p): %関数の最小値を求める
g:=-xi:
h:=g:
xi:=g:
for i from 1 to 3 do
  xx:=FindMinDim(p,xi);
  p:=p+xx*xi;
  p_s:=[op(p_s),convert(p,list)];
  xi:=dFunc(p): %関数の最小値を求める
  gg:=g.g;
  dgg:=(xi+g).xi;
```

```

gam:=dgg/gg;
g:=-xi;
h:=g+gam*h;
xi:=h;
end do:
pp2:=pointplot(p_s,connect=true);
display(pp2,c1);

```

次に準ニュートン法のプログラムコードを載せる.

```

p_s:=[]:
p:=Vector([0,0]):
hessian:=Matrix(2,2,[[1,0],[0,1]]): %%Hesse行列(小節2.5.1参照)
を作る
p_s:=op(p_s,convert(p,list)):
g:=dFunc(p):
xi:=-g:
for i from 1 to 3 do
xx:=FindMinDim(p,xi);
p:=p+xx*xi;
p_s:=op(p_s,convert(p,list));
dg:=g;
g:=dFunc(p);
dg:=g-dg;
hdg:=hessian.dg;
fac:=dg.xi;
fae:=dg.hdg;
fac:=1/fac;
fad:=1/fae;
dg:=fac*xi-fad*hdg;
hessian:=hessian+fac*OuterProductMatrix(xi,xi)
-fad*OuterProductMatrix(hdg,hdg)
-fae*OuterProductMatrix(dg,dg);
xi:=-hessian.g;
end do:
pp3:=pointplot(p_s,connect=true):
display(pp3,c1);

```

第3章 結果

図3.1は全原子を正面から見た図である。後ろに原子がそれぞれ3つずつ並んでいる。一点鎖線で示した原子は周期的境界条件によって配置される。いわば、影のような原子である。独立な原子は名目上54原子であるが、系の対称性のために左下に位置する本研究では pos1 の原子を置き換えるため、pos2~pos9 までの8原子となる。一点鎖線上の原子はこの対称性のために移動しない。さらに等価な方向を持っているため、実質的には独立な移動方向は6方向となる。その点線で囲った部分の原子についている矢印は構造緩和によってかかる力の方向を示したもので、従来その向きに働くとされている。そして、図3.1の黒く塗りつぶしている原子はすべて、深さ $1/6$ (座標で表すと y 軸方向に $1/6$) に位置している。そして、原子に付いている pos というものは今回、研究の対象となる原子を表している。その座標を表したものが表3.1である。

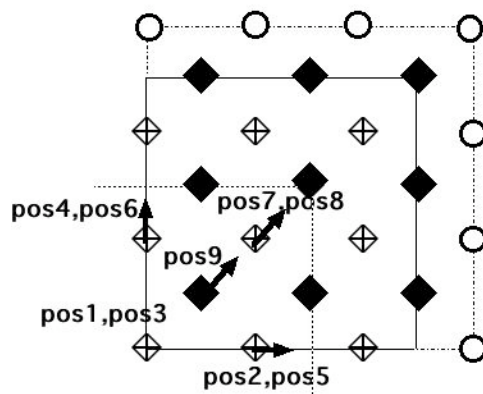


図 3.1: 全原子の位置と対称性を考えるための模式図 (正面から見た場合)。

図3.2は1回目の構造緩和において各原子に働いている力をベクトルで表記した図である。表3.1のそのままの値を用いるとFORCEのベクトルが小さくて見えないので、10倍に拡大した。そして、表3.2はそれぞれの原子のFORCEの x, y, z 軸方

表 3.1: pos1 から pos9 の座標.

	x 軸	y 軸	z 軸
pos1	0	0	0
pos2	1/3	0	0
pos3	0	1/3	0
pos4	0	0	1/3
pos5	1/3	1/3	0
pos6	0	1/3	1/3
pos7	1/3	0	1/3
pos8	1/3	1/3	1/3
pos9	1/6	1/6	1/6

向の値を表にしたものである. pos1 は周期的境界条件のために全く動いていなかった. そして, 表 3.2 から pos2, pos3, pos4 と pos5, pos6, pos7 のように boundary(境界)に接している原子の FORCE はそれぞれの組み合わせで同じ値を取っていた. そして, pos8, pos9 はそれぞれ他の原子に比べて FORCE が大きく, 向きが同じであった. 図 3.3 は 1 回目の構造緩和によって動く原子の移動方向を表した図である. こ

表 3.2: 1 回目の構造緩和において各原子に働いている力.

	x 軸	y 軸	z 軸
pos2	0.026427	0	0
pos3	0	0.026427	0
pos4	0	0	0.026427
pos5	-0.01449	-0.01449	0
pos6	0	-0.01449	-0.01449
pos7	-0.01449	0	-0.01449
pos8	0.046771	0.046771	0.046771
pos9	0.135281	0.135281	0.135281

の図 3.3 も FORCE と同じ理由のため 200 倍に拡大した. そして, 表 3.3 はそれぞれの原子の x, y, z 軸方向に動いた距離を表にしたものである. 図 3.2 から力と移動方向は全く同じ方向になっていた. そして移動する距離も力と同じく pos2, pos3, pos4 と pos5, pos6, pos7 のように boundary(境界)に接している原子の距離はそれぞれの

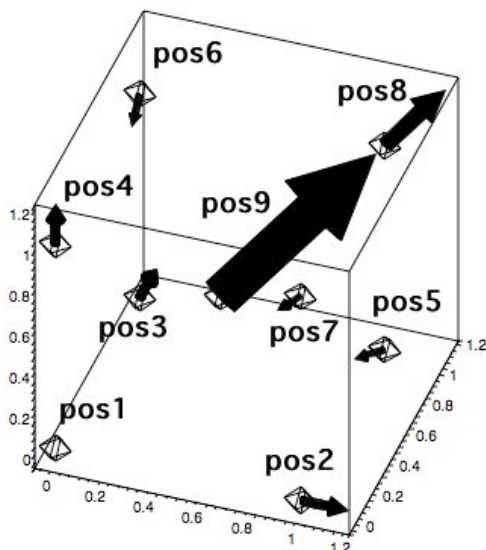


図 3.2: 1 回目の構造緩和において各原子に働いている力.

組み合わせで同じ値を取っていた. pos8 と pos9 に関しても同じく, 他の原子に比べて, 値が大きく移動方向も同じであった.

よって, 各手法に関して 6 個の z 軸方向の力を参照することで, 位置と方向の対称性から全原子の全方向 (x 軸方向, y 軸方向, z 軸方向) に対応できることが図 3.2 によって明らかになった. これからは, z 軸方向のみを研究対象とする.

3.1 最急降下法における構造緩和

構造緩和の精度に関しては, 各原子にかかる力に対して, ある閾値を設定した. これによって収束性のみが検討対象となる.

表 3.4 は各原子の FORCE を各回に分けて並べたものである (表 3.4 に載せたのは原子 12 個分 (原子 6 個の力とその原子と対象な方向にある原子があるので 12 個となる) で実際は, z 軸方向には 36 個の原子に FORCE が働いていたが, 残りの 24 個は全く表 3.4 に載せたものと FORCE が同じだったので割愛した.). FORCE は各原子でそれぞれ正負で対称になっていることがわかる. これは力の作用反作用の法則によるものだと考えられる. そして, 図 3.4 は表 3.4 の値をそれぞれ線で結んだ図である. そうすれば, 最終的に FORCE が 0 に落ち着いているかを見れるので, 各手法の構造緩和における収束性を観察できる. そして, 図 3.4 の中の数字はセルフコンシステントループ (図 2.1 参照) の回数を表している. 図 3.4 から計算回数 10 回で収束しているように考えられるが実際は計算は 20 回以上続いた. このことから最急降下法はうまく収束しないという結果が得られた. そして, 2 つの

表 3.3: 構造緩和によって動く原子の移動距離.

	x 軸	y 軸	z 軸
pos2	0.00127	0	0
pos3	0	0.00127	0
pos4	0	0	0.00127
pos5	-0.0007	-0.0007	0
pos6	0	-0.0007	-0.0007
pos7	-0.0007	0	-0.0007
pos8	0.00226	0.00226	0.00226
pos9	0.00653	0.00653	0.00653

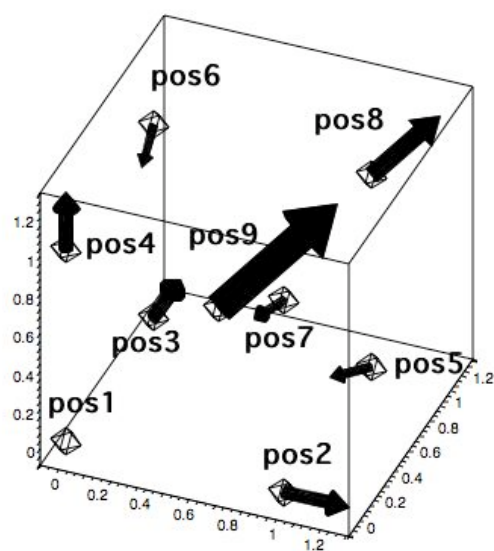


図 3.3: 構造緩和後の原子位置.

表 3.4: 最急降下法を用いた場合に、構造緩和の各回において各原子にかかっている z 軸方向の力 (eV/Å).

回数	z 軸方向の力					
1 回目	0.135281	-0.029488	-0.003605	0.026427	-0.014490	0.046771
2 回目	0.053459	-0.006218	-0.014550	-0.037115	0.014954	0.048119
3 回目	-0.037051	0.003945	0.004742	-0.031335	0.010507	-0.003983
4 回目	-0.027308	0.001653	0.004594	-0.016537	0.005419	-0.007980
5 回目	0.030045	-0.002865	0.006338	0.033399	-0.007902	-0.011545
6 回目	0.019850	-0.002041	0.004335	-0.002136	-0.003209	-0.006424
7 回目	-0.035526	-0.003728	0.003342	-0.013265	0.007172	0.010796
8 回目	-0.002314	-0.001829	-0.000084	0.019051	0.003620	0.004553
9 回目	0.012065	0.005482	0.002758	-0.018305	-0.001219	-0.002895
10 回目	-0.001464	0.001863	0.001958	-0.015311	0.000239	-0.000282
11 回目	0.001067	-0.002602	0.000750	0.022114	0.000889	0.000864
1 回目	-0.135281	0.029488	0.003605	-0.026427	0.014490	-0.046771
2 回目	-0.053459	0.006218	0.014550	0.037115	-0.014954	-0.048119
3 回目	0.037051	-0.003945	-0.004742	0.031335	-0.010507	0.003983
4 回目	0.027308	-0.001653	-0.004594	0.016537	-0.005419	0.007980
5 回目	-0.030045	0.002865	-0.006338	-0.033399	0.007902	0.011545
6 回目	-0.019850	0.002041	-0.004335	0.002136	0.003209	0.006424
7 回目	0.035526	0.003728	-0.003342	0.013265	-0.007172	-0.010796
8 回目	0.002314	0.001829	0.000084	-0.019051	-0.003620	-0.004553
9 回目	-0.012065	-0.005482	-0.002758	0.018305	0.001219	0.002895
10 回目	0.001464	-0.001863	-0.001958	0.015311	-0.000239	0.000282
11 回目	-0.001067	0.002602	-0.000750	-0.022114	-0.000889	-0.000864

表 3.5: 構造緩和 11 回目以降のセルフコンシステントループの回数.

relaxation ループ回数	12 回目	13 回目	14 回目
self-consistent ループ回数	6	4	6
relaxation ループ回数	15 回目	16 回目	17 回目
self-consistent ループ回数	4	6	6
relaxation ループ回数	18 回目	19 回目	20 回目
self-consistent ループ回数	4	6	4

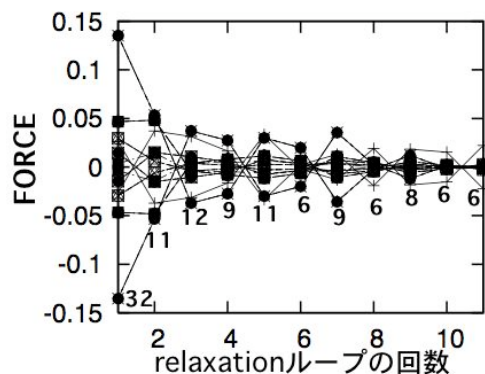


図 3.4: 最急降下法での構造緩和.

原子の FORCE が1回目の構造緩和のときに、他の原子に比べて、大きな値を取っていて、閾値から離れて収束していた。この原子の一つは図 3.2 で示した、pos4 の原子であった。そして、表 3.5 から構造緩和 11 回目以降にかかるセルフコンシステントループの回数が少ないことが理解できた。図 3.4 の作成方法に関しては、A.2 節に記載している。

3.2 共役勾配法における構造緩和

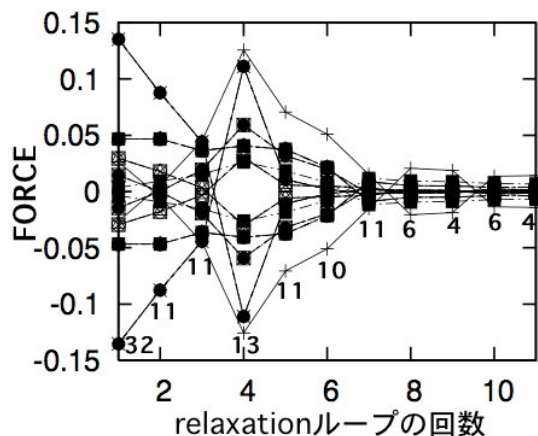


図 3.5: 共役勾配法での構造緩和.

表 3.6 から FORCE は最急降下法と同様、正負で対称となっている (これも最急降下法に同じく原子 12 個分を表に載せた (3.1 節参照)). 図 3.5 から、セルフコンシステントループで収束するまでには多数回が必要とされる。それは 4 回めの構造

緩和のときの FORCE が大きく開いており、ループ回数が多くなっていることがわかった。そこが原因で relaxation ループが少し多くなったためにセルフコンシステントループが多くなったと考えられる。そして、最急降下法と同様、2つの原子の FORCE が1回目の構造緩和のときに、他の原子に比べて、大きな値を取っていて、閾値から離れて収束していた。この原子の一つも最急降下法と同じく、図3.2で示した pos4 の原子であった。図3.5の作成方法に関しては、A.2節に記載している。

3.3 準ニュートン法における構造緩和

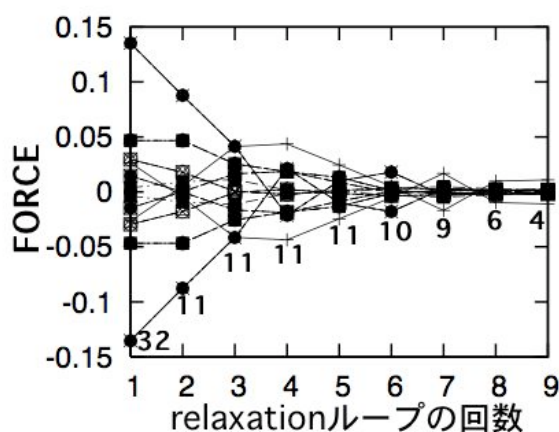


図 3.6: 準ニュートン法での構造緩和.

上の2つの手法と同様、表3.7から正負において、対称となっている(これも他の2つの手法に同じく原子12個分を表に載せた(3.1節参照))。図3.6から準ニュートン法については徐々に FORCE が0に近づいていることが理解できる。そして、他の手法と同様、2つの原子の FORCE が1回目の構造緩和のときに、他の原子に比べて、大きな値を取っていて、閾値から離れて収束していた。この原子の一つは他の手法と同じく図3.2で示した、pos4の原子であった。図3.6の作成方法に関しては、A.2節に記載している。

3.4 各手法における鉄原子のポジション

各手法における鉄原子のポジションをグラフにした。最初の構造緩和において移動した原子を0とし、次の構造緩和において移動した原子のz軸方向の距離の差分を取る、という作業を繰り返すことで可能にした。具体的なポジショングラフの書き方に関してはA.3節に記載している。

3.4.1 最急降下法における鉄原子のポジション

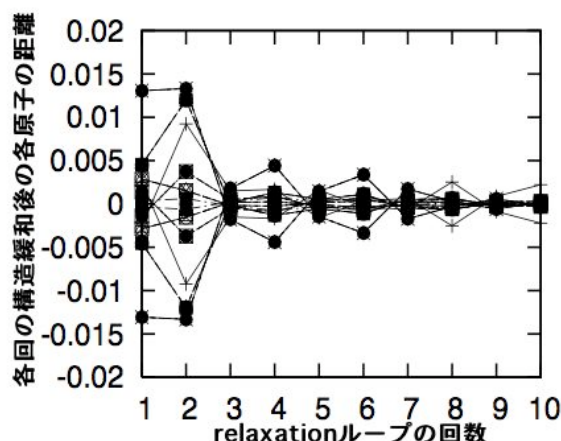


図 3.7: 最急降下法における鉄原子の z 軸方向へのポジションの移り変わり.

表 3.8 は最急降下法における各原子の z 軸方向への動きを表にまとめたものである (表 3.8 に載せたのは FORCE と同様, 原子 12 個分 (原子 6 個の力とその原子と対象な方向にある原子があるので 12 個となる) で, 実際は z 軸方向には 36 個の原子が動いていたが, 残りの 24 個は全く表 3.8 に載せたものと距離が同じだったので割愛した). 図 3.7 は表 3.8 の値をそれぞれ線に結んだ図である. この図 3.7 も FORCE と同様, 最終的に 0 に落ち着くかどうかで, 各手法の収束性を見る. 図 3.7 から, FORCE のときと同様, 2 つの原子が閾値から離れて収束していた. この原子の一つは FORCE と同様, 図 3.3 の pos4 の原子であった. そして, 図 3.4 から FORCE に開きがある構造緩和ほど距離が開いていた.

3.4.2 共役勾配法における鉄原子のポジション

表 3.9 は共役勾配法における各原子の動きを表にまとめたものである. 最急降下法と同じく原子 12 個分を載せた. 3.2 節で記した 4 回目の FORCE の大きな開きが原因で, 図 3.8 から 4 回目の構造緩和の時に距離が大きく開いていることがわかる. そして, 6 回目の構造緩和のときにも距離が開いていた. このために距離が 0 に落ち着くまでに, 構造緩和が数回, 他の手法に比べて多かった.

3.4.3 準ニュートン法における鉄原子のポジション

表 3.10 は準ニュートン法における各原子の動きを表にまとめたものである. 他の手法と同じく原子 12 個分を載せた. この図 3.9 から, 最急降下法のポジションと同様, 図 3.6 から FORCE に開きがある構造緩和ほど距離が開いていた.

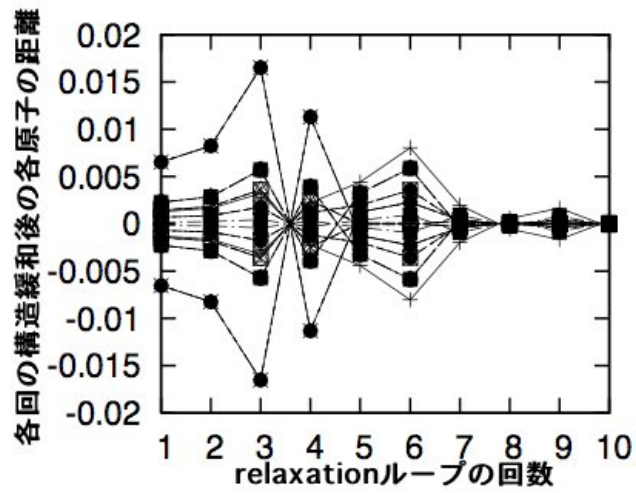


図 3.8: 共役勾配法における鉄原子の z 軸方向へのポジションの移り変わり.

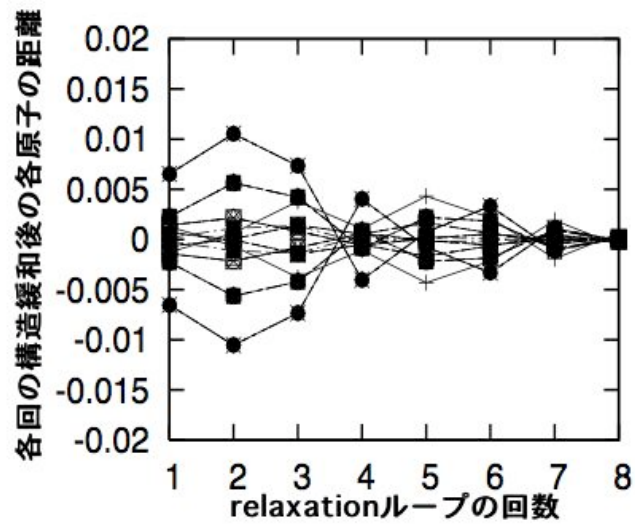


図 3.9: 準ニュートン法における鉄原子の z 軸方向へのポジションの移り変わり.

表 3.6: 共役勾配法を用いた場合に, 構造緩和の各回において各原子にかかっている z 軸方向の力 (eV/Å).

回数	z 軸方向の力					
1 回目	0.135281	-0.029488	-0.003605	0.026427	-0.014490	0.046771
2 回目	0.087534	-0.017755	-0.009229	-0.005145	0.000283	0.046712
3 回目	0.044124	-0.003290	-0.015394	-0.045035	0.019241	0.036285
4 回目	-0.111150	0.030264	-0.026186	-0.125607	0.059051	0.040542
5 回目	-0.010663	0.005835	-0.018367	-0.070562	0.031964	0.037431
6 回目	-0.002051	0.004748	-0.008086	-0.050804	0.019789	0.021351
7 回目	0.002246	0.003243	0.011635	-0.015093	-0.001825	-0.008853
8 回目	0.001005	-0.001405	0.009361	0.020744	0.000430	-0.005243
9 回目	0.001053	-0.000635	0.008843	0.018838	0.000364	-0.005026
10 回目	-0.001015	0.002258	0.007173	-0.013655	0.000988	-0.003615
11 回目	-0.001237	0.001371	0.006712	-0.014067	0.000277	-0.003801
1 回目	-0.135281	0.029488	0.003605	-0.026427	0.014490	-0.046771
2 回目	-0.087534	0.017755	0.009229	0.005145	-0.000283	-0.046712
3 回目	-0.044124	0.003290	0.015394	0.045035	-0.019241	-0.036285
4 回目	0.111150	-0.030264	0.026186	0.125607	-0.059051	-0.040542
5 回目	0.010663	-0.005835	0.018367	0.070562	-0.031964	-0.037431
6 回目	0.002051	-0.004748	0.008086	0.050804	-0.019789	-0.021351
7 回目	-0.002246	-0.003243	-0.011635	0.015093	0.001825	0.008853
8 回目	-0.001005	0.001405	-0.009361	-0.020744	-0.000430	0.005243
9 回目	-0.001053	0.000635	-0.008843	-0.018838	-0.000364	0.005026
10 回目	0.001015	-0.002258	-0.007173	0.013655	-0.000988	0.003615
11 回目	0.001237	-0.001371	-0.006712	0.014067	-0.000277	0.003801

表 3.7: 準ニュートン法を用いた場合に、構造緩和の各回において各原子にかかっている z 軸方向の力 (eV/Å).

回数	z 軸方向の力					
1 回目	0.135281	-0.029488	-0.003605	0.026427	-0.014490	0.046771
2 回目	0.087534	-0.017755	-0.009229	-0.005145	0.000283	0.046712
3 回目	0.041601	-0.000580	-0.011237	-0.041313	0.016365	0.025492
4 回目	-0.021253	0.002324	-0.001445	-0.043585	0.018666	0.018425
5 回目	0.009972	-0.001406	-0.000704	-0.024431	0.008697	0.013172
6 回目	0.017955	-0.003407	0.003417	-0.005073	-0.000587	0.001246
7 回目	-0.003054	-0.002589	0.004855	0.016877	0.000560	-0.001481
8 回目	-0.001616	0.000608	0.002683	-0.009841	0.002012	-0.001608
9 回目	-0.002224	-0.001848	0.001842	-0.010894	0.000650	-0.000651
1 回目	-0.135281	0.029488	0.003605	-0.026427	0.014490	-0.046771
2 回目	-0.087534	0.017755	0.009229	0.005145	-0.000283	-0.046712
3 回目	-0.041601	0.000580	0.011237	0.041313	-0.016365	-0.025492
4 回目	0.021253	-0.002324	0.001445	0.043585	-0.018666	-0.018425
5 回目	-0.009972	0.001406	0.000704	0.024431	-0.008697	-0.013172
6 回目	-0.017955	0.003407	-0.003417	0.005073	0.000587	-0.001246
7 回目	0.003054	0.002589	-0.004855	-0.016877	-0.000560	0.001481
8 回目	0.001616	-0.000608	-0.002683	0.009841	-0.002012	0.001608
9 回目	0.002224	0.001848	-0.001842	0.010894	-0.000650	0.000651

表 3.8: 最急降下法での各回の鉄原子の z 軸方向のポジションの移動距離.

回数	z 軸方向の移動距離					
1 回目	0.01305	-0.00285	-0.00035	0.00255	-0.00140	0.00451
2 回目	0.01331	-0.00154	-0.00062	-0.00924	0.00372	0.01198
3 回目	-0.00179	0.00019	0.00023	-0.00151	0.00051	-0.00019
4 回目	-0.00442	0.00026	0.00074	-0.00168	0.00088	-0.00129
5 回目	0.00145	-0.00013	0.00031	0.00161	-0.00038	-0.00056
6 回目	0.00337	-0.00035	0.00073	-0.00036	-0.00055	-0.00109
7 回目	-0.00172	-0.00018	0.00017	-0.00064	0.00035	0.00052
8 回目	-0.00030	-0.00024	-0.00002	0.00253	0.00048	0.00061
9 回目	0.00058	0.00026	0.00014	-0.00088	-0.00006	-0.00014
10 回目	-0.00021	0.00027	0.00028	-0.00221	0.00004	-0.00004
1 回目	-0.01305	0.00285	0.00035	-0.00255	0.00140	-0.00451
2 回目	-0.01331	0.00154	0.00062	0.00924	-0.00372	-0.01198
3 回目	0.00179	-0.00019	-0.00023	0.00151	-0.00051	0.00019
4 回目	0.00442	-0.00026	-0.00074	0.00168	-0.00088	0.00129
5 回目	-0.00145	0.00013	-0.00031	-0.00161	0.00038	0.00056
6 回目	-0.00337	0.00035	-0.00073	0.00036	0.00055	0.00109
7 回目	0.00172	0.00018	-0.00017	0.00064	-0.00035	-0.00052
8 回目	0.00030	0.00024	0.00002	-0.00253	-0.00048	-0.00061
9 回目	-0.00058	-0.00026	-0.00014	0.00088	0.00006	0.00014
10 回目	0.00021	-0.00027	-0.00028	0.00221	-0.00004	0.00004

表 3.9: 共役勾配法での各回の鉄原子の z 軸方向のポジションの移動距離.

回数	z 軸方向の移動距離					
1 回目	0.00653	-0.00142	-0.00017	0.00127	-0.00070	0.00226
2 回目	0.00825	-0.00180	-0.00022	0.00162	-0.00088	0.00285
3 回目	0.01651	-0.00360	-0.00044	0.00322	-0.00177	0.00571
4 回目	-0.01131	0.00247	0.00030	-0.00221	0.00121	-0.00391
5 回目	0.00125	-0.00004	-0.00037	-0.00442	0.00197	0.00323
6 回目	0.00228	-0.00006	-0.00095	-0.00803	0.00357	0.00588
7 回目	0.00037	0.00030	0.00092	-0.00197	0.00008	-0.00041
8 回目	-0.00010	-0.00009	-0.00026	0.00056	-0.00002	0.00012
9 回目	0.00009	-0.00005	0.00078	0.00165	0.00003	-0.00044
10 回目	-0.00002	0.00001	-0.00015	-0.00032	-0.00001	0.00008
1 回目	-0.00653	0.00142	0.00017	-0.00127	0.00070	-0.00226
2 回目	-0.00825	0.00180	0.00022	-0.00162	0.00088	-0.00285
3 回目	-0.01651	0.00360	0.00044	-0.00322	0.00177	-0.00571
4 回目	0.01131	-0.00247	-0.00030	0.00221	-0.00121	0.00391
5 回目	-0.00125	0.00004	0.00037	0.00442	-0.00197	-0.00323
6 回目	-0.00228	0.00006	0.00095	0.00803	-0.00357	-0.00588
7 回目	-0.00037	-0.00030	-0.00092	0.00197	-0.00008	0.00041
8 回目	0.00010	0.00009	0.00026	-0.00056	0.00002	-0.00012
9 回目	-0.00009	0.00005	-0.00078	-0.00165	-0.00003	0.00044
10 回目	0.00002	-0.00001	0.00015	0.00032	0.00001	-0.00008

表 3.10: 準ニュートン法での各回の鉄原子の z 軸方向のポジションの移動距離.

回数	z 軸方向の移動距離					
1回目	0.00653	-0.00142	-0.00017	0.00127	-0.00070	0.00226
2回目	0.01054	-0.00214	-0.00112	-0.00063	0.00004	0.00563
3回目	0.00734	-0.00081	-0.00137	0.00381	0.00143	0.00420
4回目	-0.00402	0.00059	0.00039	-0.00108	0.00054	-0.00086
5回目	0.00073	-0.00009	-0.00022	-0.00430	0.00165	0.00220
6回目	0.00331	-0.00056	-0.00013	-0.00216	0.00065	0.00183
7回目	-0.00109	0.00007	0.00025	0.00192	-0.00047	-0.00095
8回目	0.00011	-0.00001	0.00007	-0.00063	0.00020	0.00015
1回目	-0.00653	0.00142	0.00017	-0.00127	0.00070	-0.00226
2回目	-0.01054	0.00214	0.00112	0.00063	-0.00004	-0.00563
3回目	-0.00734	0.00081	0.00137	-0.00381	-0.00143	-0.00420
4回目	0.00402	-0.00059	-0.00039	0.00108	-0.00054	0.00086
5回目	-0.00073	0.00009	0.00022	0.00430	-0.00165	-0.00220
6回目	-0.00331	0.00056	0.00013	0.00216	-0.00065	-0.00183
7回目	0.00109	-0.00007	-0.00025	-0.00192	0.00047	0.00095
8回目	-0.00011	0.00001	-0.00007	0.00063	-0.00020	-0.00015

3.5 各手法における計算時間の比較

表3.11のtotal-timeは構造緩和の総合計算時間を表していて、self-consistent timeは図2.1のセルフコンシステントループ一回あたりの計算時間を表している。そして、relaxation timeは構造緩和一回あたりの計算時間を表している。表3.11から、最急降下法に関してはrelaxationループ20回での総合計算時間を示したが、relaxationループ11回での総合計算時間は23165秒であり、total-timeに関しては共役勾配法よりも早いという結果が得られた。その共役勾配法はやはり、4回目のFORCEの開きが原因(3.2節参照)で時間がかかったものと思われる。そして、最急降下法のrelaxation timeに関してはどの手法よりもtimeが小さいが、このことは、表3.5からrelaxationループ11回目以降のセルフコンシステントループの回数が原因で、20回目まで、6が5回、4が4回というように、1回の構造緩和におけるセルフコンシステントループの回数が少ないために、一番早かったと考えられる。準ニュートン法はself-consistent time,total-timeの両方において早いという結果が得られた。

表 3.11: 各手法の計算時間.

手法	total time [sec]	relaxation ループ回数	self-consistent time [sec]	relaxation time [sec]
最急降下法※	32360	20	199.7	1618.0
共役勾配法	24550	11	206.3	2231.8
準ニュートン法	19500	9	185.7	2166.7

※ relaxation ループ (図 2.1 参照)20 回での計算時間.

3.6 Tiの層状構造の緩和における計算時間の比較

Tiの層状構造でも計測したのでそれを載せる。表3.12はTiを層状に積んだときの構造緩和における計算時間を表にしたものである。Tiの場合はずべての手法において収束したが、最急降下法だけがrelaxationループ回数が多かった。そして、共役勾配法が準ニュートン法よりもすべてのtimeにおいて早いという結果が出た。

表 3.12: Ti の層状構造の緩和における各手法の計算時間.

手法	total	relaxation	self-consistent	relaxation
	time [sec]	ループ回数	time [sec]	time [sec]
最急降下法	941.4	10	11.8	94.1
共役勾配法	553.8	6	10.1	92.3
準ニュートン法	563.9	6	10.1	94.0

3.7 検討結果

第3章から図3.2, 図3.3から構造緩和によって加わる力とそれによって動く原子の移動方向が同じだった。そして, FORCE, 移動方向の両方について言えることで, pos9の原子は他の原子に比べて大きな値を取っていた。これは, pos1からの大きな力が働き, pos1から1番近いpos9の原子に働いたからだと思われる。そして, pos2,pos3,pos4とpos5,pos6,pos7の組み合わせのboundary(境界)に接している原子はそれぞれの組み合わせでFORCEと距離の両方において同じ値をとっていた。

次に各計算手法についての構造緩和の結果について述べる。最急降下法はrelaxationループが20回を超えても収束しなかった。図2.2のように, たとえ谷が完璧な2次形式であっても, この方法は多数の小ステップをとる。このことから, relaxationループが20回を超えても収束しなかったことが窺える。共役勾配法は図3.5から, 4回目の構造緩和のときのFORCEの大きな開きがなければ, もっと早く収束していたと考えられる。準ニュートン法に関しては, どの手法よりも収束が早かった。そして, 模式図による準ニュートン法(図2.4参照)と模式図による共役勾配法(図2.3参照)が全く同じ動きをしているが, 現実の系においては計算回数に差ができている(表3.11参照)。

このことは上で記した, FORCEの開きが原因だと考えられる。そして, 各手法において, 2つの原子のFORCEが1回目の構造緩和のときに, 他の原子に比べて, 大きな値を取っていて, 閾値から離れて収束していた。このうちの一つの原子は図3.2でいう, pos4の原子であった。実際, 一番FORCEの大きかったpos9の原子がその原子だと思っていたが, 違っていた。

次に各手法においての鉄原子のポジションについて, 最急降下法では, 図3.7から, FORCEのときと同様, 2つの原子が閾値から離れて収束していた。この原子の一つはFORCEと同様, 図3.3のpos4の原子であった。そして, 図3.4からFORCEに開きがある構造緩和ほど距離が開いていた。そして, 共役勾配法については, FORCEと同じく4回目の構造緩和で大きな力が働いたために, 図3.8から, 4回目あたりで大きく距離が開いていることが窺える。そしてFORCEに関係なく6回目の構造緩和で距離が大きく開いていた。準ニュートン法に関しては, 図3.9から, 最急降下法と同様, 図3.6でFORCEの大きい構造緩和ほど距離の開きが大きかった。原子のポジションについても各手法において0から離れて収束する原子が出てくると思ったが, その原子が見られたのは, 最急降下法だけで, 他の手法では見られなかった。

第4章 まとめ

本研究においては、VASP という第一原理計算パッケージに用意されている構造緩和手法の計算時間の比較を行った。対象系としては、 $\text{Fe}_{53}\text{Cu}_1$ を用いて詳しく調べた。このことに関しては第1章で記した。

計算手法に関して、まず、最急降下法は構造緩和が20回を超えても収束しなかった。2.2節で記した収束性の悪さがうかがえた。共役勾配法は構造緩和の FORCE の開きが原因で計算回数が増えていた。準ニュートン法は徐々に閾値に収束しており、計算時間も他の手法に比べて早かった。

結果として、表 3.11 から、計算時間が最も早かったのは準ニュートン法であった。そして、準ニュートン法は、収束性も他の手法に比べてきれいに閾値に収束をしていたことから準ニュートン法が構造緩和において、一番いい結果が得られると言える。

しかし、比較の対象系として行った Ti の層状構造の緩和では表 3.12 から、収束性については定かではないが、計算時間においては共役勾配法が一番いい結果が得られた。従って、共役勾配法と準ニュートン法とでは一概に優劣はつけがたく、問題に依存する。しかし、このような構造緩和に対しては最急降下法は妥当ではない事が判明した。

付 録 A

A.1 vasp の使用方法

vasp は第一原理計算を行うプログラムである。vasp を使用するまでの順序を記載すると、次のようになる。

1. 原子フォルダを，計算機の中に用意する。
2. linux から計算機の中に入り，原子フォルダの中に移動する。
3. そこで `/usr/local/mpich/bin/mpirun -np` 使用する CPU の数 `-machinefile machines /usr/local/vasp/vasp4628mpi` というコマンドを入力する。ただし，CPU 数が 1 の場合は，`/usr/local/vasp/vasp4628serial` というコマンドの方が計算時間は早い。

本研究では `/usr/local/vasp/vasp4628serial` というコマンドを用いて構造緩和を行った。

A.2 構造緩和のグラフの描写方法

図 3.4, 図 3.5, 図 3.6 を出力するに至るまでの過程を説明する。

1. 金属原子のフォルダを用意する (その中のいくつかのファイルのパラメータを変更する)。
2. INCAR ファイルの変更 (小節 A.2.1 参照)
3. POSCAR ファイルの変更 (小節 A.2.2 参照)
4. KPOINTS ファイルの変更 (小節 A.2.3 参照)
5. OUTCAR ファイルの編集 (小節 A.2.4 参照)
6. gnuplot での描画 (小節 A.2.5 参照)

上の要領でグラフを作成した。ただし，上の 2,3,4 の順番は入れ替わっても良い。

A.2.1 INCAR ファイルの変更

INCAR ファイルとは VASP の中心となる入力ファイルで、VASP をどのように何をさせるか決めるものである。INCAR ファイルを扱うときにはエラーや失敗の元となるので気をつけなければならない。INCAR ファイルを呼び出すと、以下のようなファイルが開く。

```
# SCF input for VASP
# Note that VASP uses the FIRST occurrence of a keyword
SYSTEM = Cu3.56 (VASP)

PREC = Medium
NBANDS = 300
NELM = 100
NELMIN = 2
EDIFF = 1.0e-05
ISPIN = 2
ICHARG = 2
LWAVE = .FALSE.
LCHARG = .TRUE.
ISMEAR = 0
SIGMA = 0.1
IALGO = 48
LREAL = Auto
IBRION = 3
ISIF = 2
NFREE = 2
NSW = 60
```

ここで NBANDS, ISMEAR, LREAL, IBRION, ISIF, NFREE, NSW の値をそれぞれ上に示した値に変更。その他の値についてはかえる必要はない。この変更の必要なパラメータについて説明をする。 [7]

- NBANDS はバンド計算をどのくらい行うのかを決めるパラメータである。(300 以上にしないとエラーが出る)
- ISMEAR は波動関数をどのような手法で表示するのかを設定するパラメータで 0 にすることでガウスモデルを用いている。

- LREAL は原子の配置に関するパラメータで Auto にすることで自動的に最適化してくれる。(Auto にしないとエラーが出る)
- IBRION はイオンがどのように置き換えられ動かされるのかを決めるパラメータである。IBRION=1 で準ニュートン法, IBRION=2 で共役勾配法, IBRION=3 で最急降下法を用いる。この IBRION を用いて手法を変える。
- ISIF は応力テンソルを計算するかどうかを決めるパラメータである。応力テンソルの計算は比較的時間がかかる。構造緩和するには、パラメータを 2 にする必要がある。
- NFREE はステップ数の履歴を残すためのパラメータで 2 に設定して計算を行った。
- NSW は計算のステップ数を決めるパラメータで構造緩和は計算量が多いので 60 ぐらいに設定した。

A.2.2 POSCAR ファイルの変更

POSCAR ファイルは原子のポジションをどのように設定するかを決めるパラメータである。POSCAR ファイルを呼び出すと以下のようなファイルが現れる。

```

Fe
  1.0
      8.60100000 0.00000000 0.00000000
      0.00000000 8.60100000 0.00000000
      0.00000000 0.00000000 8.60100000
  1 53
Selective dynamics %%ここで原子をそれぞれ移動できるように設定
Direct %%Tをつけることで原子が移動する。(Fに設定すると原子は移動しない)
      0.0000000000000000 0.0000000000000000 0.0000000000000000 T T T
      0.3333333333333333 0.0000000000000000 0.0000000000000000 T T T
      0.6666666666666667 0.0000000000000000 0.0000000000000000 T T T

      0.0000000000000000 0.3333333333333333 0.0000000000000000 T T T
      0.3333333333333333 0.3333333333333333 0.0000000000000000 T T T
      0.6666666666666667 0.3333333333333333 0.0000000000000000 T T T
      0.0000000000000000 0.6666666666666667 0.0000000000000000 T T T
      0.3333333333333333 0.6666666666666667 0.0000000000000000 T T T
      0.6666666666666667 0.6666666666666667 0.0000000000000000 T T T

```



```

0.6666666666666667 0.6666666666666667 0.6666666666666667 T T T
0.1666666666666667 0.1666666666666667 0.8333333333333333 T T T
0.5000000000000000 0.1666666666666667 0.8333333333333333 T T T
0.8333333333333333 0.1666666666666667 0.8333333333333333 T T T
0.1666666666666667 0.5000000000000000 0.8333333333333333 T T T
0.5000000000000000 0.5000000000000000 0.8333333333333333 T T T
0.8333333333333333 0.5000000000000000 0.8333333333333333 T T T
0.1666666666666667 0.8333333333333333 0.8333333333333333 T T T
0.5000000000000000 0.8333333333333333 0.8333333333333333 T T T
0.8333333333333333 0.8333333333333333 0.8333333333333333 T T T

```

上記のように POSCAR ファイルを変更する.

A.2.3 KPOINTS ファイルの変更

KPOINTS ファイルは K 点を設定するファイルのことである. KPOINTS ファイルがフォルダ内にあるので KPOINTS ファイルを呼び出す. 呼び出すと以下のようなファイルが現れる.

```

Automatic mesh
0
Monkhorst Pack
4 4 4
0. 0. 0.

```

4行目の値を上記のように 4 4 4 に変更 (KPOINT が少ないと精度に関して不安が残るので 4 4 4 で行う) [7]

A.2.4 OUTCAR ファイルの編集

OUTCAR ファイルはデータの出力をするファイルで, 10 万行ぐらいある膨大なファイル.

POSITION			TOTAL-FORCE (eV/Angst)		
0.00000	0.00000	0.00000	0.000000	0.000000	0.000000
2.86700	0.00000	0.00000	0.026427	0.000000	0.000000
.
.

```

1.43350  1.43350  1.43350    0.135281  0.135281  0.135281 ※1
4.30050  1.43350  1.43350    0.000000 -0.029488 -0.029488
.        .        .        .        .        .
.        .        .        .        .        .
2.86700  5.73400  2.86700    0.046771 -0.046771  0.046771
5.73400  5.73400  2.86700   -0.046771 -0.046771  0.046771 ※1
.        .        .        .        .        .
.        .        .        .        .        .
0.00000  0.00000  5.73400    0.000000  0.000000 -0.026427 ※2
2.86700  0.00000  5.73400   -0.014490  0.000000  0.014490
.        .        .        .        .        .
.        .        .        .        .        .
4.30050  7.16750  7.16750    0.000000  0.029488  0.029488
7.16750  7.16750  7.16750   -0.135281 -0.135281 -0.135281 ※2
-----
total drift:                0.000000  0.000000  0.000000

```

この右端の行の※1, ※2の部分をテキストにコピー (右端の0ではない部分, 行でいうと数字の行が54行あるその10行目から27行目, 37行目から54行目まで), このデータテーブルがrelaxationループの回数分存在する. コピーする時は,
1つめのデータテーブルの10行目の数字
2つめのデータテーブルの10行目の数字

.
.

というふうに表3.6のようなテキストを作る. ただし, そのときのテキストの形式はdatファイルにする必要がある (次に説明するgnuplotは標準テキストを読み込めない).

A.2.5 gnuplotでの描画

gnuplotという図を作成するLinuxの便利なソフトウェアがある. まずそれをダウンロードする. 最初にシェルを開いて, デスクトップに移動して以下のコマンドを入力

```

Yusuke@nakatayuukainokonpyuta[~/Desktop] sudo apt-get update
Yusuke@nakatayuukainokonpyuta[~/Desktop] apt-cache search gnuplot
Yusuke@nakatayuukainokonpyuta[~/Desktop] sudo apt-get install gnuplot

```

これをインストールすればすぐに使える。テキストのデータからグラフを書くにはまず、シェルを開きそのデータのあるフォルダに進む。そして以下のようにコマンドを入力する。

```
gnuplot> set style data linespoint  %% (データから読み込み図を書くための定義)
gnuplot> unset key  %% (凡例を消す)
gnuplot> plot "データ名.dat" using 1:2,
> "データ名.dat" using 1:3 linecolor rgb "black",
> "データ名.dat" using 1:4 linecolor rgb "black",
> "データ名.dat" using 1:5 linecolor rgb "black",
> "データ名.dat" using 1:6 linecolor rgb "black",
> "データ名.dat" using 1:7 linecolor rgb "black",
> "データ名.dat" using 1:8 linecolor rgb "black",
> "データ名.dat" using 1:9 linecolor rgb "black",
> "データ名.dat" using 1:10 linecolor rgb "black"
gnuplot>set size 0.4,0.4  %% (線を太くする)
gnuplot>set term postscript eps enhanced  %% (グラフを eps に変換)
gnuplot>set output "グラフ名.eps"
```

A.3 各手法における鉄原子のポジションの作成方法

作成方法は最初の構造緩和において移動した原子を 0 とし、次の構造緩和において移動した原子の差分を取る、という作業を繰り返すことで可能にした。以下の順序でグラフを作成した。

1. 金属原子のフォルダを用意する (その中のいくつかのファイルのパラメータを変更する).
2. INCAR ファイルの変更 (小節 A.2.1 参照)
3. POSCAR ファイルの変更 (小節 A.2.2 参照)
4. KPOINTS ファイルの変更 (小節 A.2.3 参照)
5. OUTCAR ファイルの編集
6. gnuplot での描画 (小節 A.2.5 参照)

上の 2,3,4,6 は A.2 節と全く同じなので割愛する。この節では 5 の OUTCAR ファイルの編集を説明する。

A.3.1 OUTCAR ファイルの編集

%%構造緩和 1 回目

POSITION			TOTAL-FORCE (eV/Angst)		
0.00000	0.00000	0.00000	0.000000	0.000000	0.000000
2.86700	0.00000	0.00000	0.026427	0.000000	0.000000
.
.
1.43350	1.43350	1.43350 ※ 1	0.135281	0.135281	0.135281
4.30050	1.43350	1.43350	0.000000	-0.029488	-0.029488

%%構造緩和 2 回目

POSITION			TOTAL-FORCE (eV/Angst)		
0.00000	0.00000	0.00000	0.000000	0.000000	0.000000
2.86955	0.00000	0.00000	-0.037115	0.000000	0.000000
.
.
1.44655	1.44655	1.44655 ※ 2	0.053459	0.053459	0.053459
4.30050	1.43065	1.43065	0.000000	-0.006218	-0.006218

この※ 1 の部分 10 行目から 18 行目までの値は、すべて同じ値になっている。これらの値を基準にしてデータテキストを作成する。そして、構造緩和 2 回目の※ 2 の部分の 10 行目から 18 行目までとの差を取ることで値は、0.01305, -0.00285 と表せる。このようにして、

構造緩和 1 回目と 2 回目の差

構造緩和 2 回目と 3 回目の差

.

.

というように原子 9 個分、表 3.8 のように作成する。

上で記した要領で、19 行目から 27 行目、37 行目から 45 行目、46 行目から 54 行目の合計 4 セット分 (原子 36 個) を作成する。

A.4 DOS の描写方法

DOS は固体におけるの電子の状態密度のことをいう。

1. Tibcc, Tihcp, Pt などのフォルダを用意する (その中のいくつかのファイルのパラメータを変更する)。

2. INCAR ファイルの変更 (小節 A.4.1 参照)
3. POSCAR ファイルの変更 (小節 A.4.2 参照)
4. KPOINTS ファイルの変更 (小節 A.4.3 参照)
5. DOSCAR ファイルのデータ編集 (小節 A.4.4 参照)
6. maple での描画 (小節 A.4.5 参照)

上のような順序で DOS を描く。ただし、2,3,4 の順序は関係なく、どれからはじめてもよい。

A.4.1 INCAR ファイルの変更

INCAR ファイルについては小節 A.2.1 を参考。ここでは変更の必要な値のみを説明する。

```
# SCF input for VASP
# Note that VASP uses the FIRST occurrence of a keyword
SYSTEM = Cu3.56 (VASP)

PREC = Medium
NELM = 50
NELMIN = 2
NEDOS = 600
EDIFF = 1.0e-05
ISPIN = 1
ICHARG = 0
LWAVE = .FALSE.
LCHARG = .TRUE.
ISMear = -5
SIGMA = 0.2
IALGO = 48
LREAL = .FALSE.
```

ここで ICHARG, NEDOS, ISMEAR, SIGMA, ISPIN の値をそれぞれ上に示した値に変更。その他の値についてはかえる必要はない。それぞれのパラメータについて説明をする。 [7]

- ICHARG は原子のどの電荷の構造を計算するかを決めるパラメーターでこの値を 0 にかえることで波動関数の最初の電荷の集まりを計算させている。

- NEDOS は状態密度のエネルギーの刻み幅を設定するパラメーター
- ISMEAR は波動関数をどのような手法で表示するのかを設定するパラメーターで値を-5 にすることで Blochl 補正のテトラヘドロン法を最初のループで使う (DOS をみるときは-5 といわれている).
- SIGMA に関しては磁性のある金属の場合は 0.2 にすればいい
- ISPIN は原子をスピンさせるかさせないかを行うパラメーターで 1 でスピンを起こさないように設定している.

A.4.2 POSCAR ファイルの変更

POSCAR ファイルに関しては小節 A.2.2 を参照. ここでは変更に必要なことだけ説明する. 今は一つのセルで計算をさせている (セルの数は DOS には影響ないので計算時間の早い一つのセルで計算をさせた). [7]

```

1.0
      2.55477500000    -1.475000000    0.000000000
      0.000000000    2.950000000    0.000000000
      0.000000000    0.000000000    4.68000000000

2
Direct
      0.33333000 0.66667000 0.25000
      0.66667000 0.33333000 0.75000

```

上記のように POSCAR ファイルを変更する.

A.4.3 KPOINTS ファイルの変更

KPOINTS ファイルに関しては小節 A.2.3 を参照. ここでは変更に必要なことだけ説明する. 高品位の DOS を描くには高い 16 程度の KPOINT が必要になってくる. [7]

```

Automatic mesh
0
Monkhorst Pack
  15  15  15
  0.  0.  0.

```

4 行目の値を上記のように 15 15 15 に変更する (16 16 16 に変更するとエラーが出るので 15 15 15 で行う)

A.4.4 DOSCAR ファイルのデータ編集

各ファイルでパラメータを変更した後、`vasp` で計算を走らせる。そのあとに DOSCAR ファイルを呼び出すと以下のようなファイルが表示される。

```
2 2 0 0
0.1763561E+02 0.2950000E-09 0.2950000E-09 0.4680000E-09 0.5000000E-15
1.0000000000000000E-004
CAR
Cu3.56 (VASP)
9.36854896 -5.07705959 600 2.96234692 1.00000000
-5.077 0.0000E+00 0.0000E+00
-5.053 0.0000E+00 0.0000E+00
.
.
.
9.344 0.0000E+00 0.1600E+02
9.369 0.0000E+00 0.1600E+02
```

この上記の 9.36854896 という値は E_{max} で最下部の左端の値を表していて、-5.07705959 という値は E_{min} で上から 7 行めの左端の値を表している。そして、2.96234692 という値は E_F (フェルミエネルギー) といって電子がつまっているエネルギー準位の最大値を表している。上記の上から 7 行めから最後までをコピーして、テキストエディットの標準テキストに保存する。

A.4.5 maple での描画

E_F のときの値を y 軸に設定して表示をさせた。

```
> with(Maplets[Examples]):
> file1:=GetFile();
      file1 := "/Users/Yusuke/Desktop/s.txt"
> T:=readdata(file1,3):
> T[1,1];
      -5.077
> n:=nops(T);
      n := 600
> Ef:=2.9623;
      Ef := 2.9623
> x1:=[];
> x2:=[];
```

```

> for i from 1 to n do
>   x1:=[op(x1), [T[i,1]-Ef,T[i,2]]];
>   x2:=[op(x2), [T[i,1], -T[i,3]]];
> end do:

                                x1 := []
                                x2 := []

> with(plots):
> p1:=pointplot(x1,connect=true):
> p2:=pointplot(x2,connect=true):
> display(p1);

```

file1 のところで作ったファイルを呼び込む. すると以下のような DOS のグラフができる.

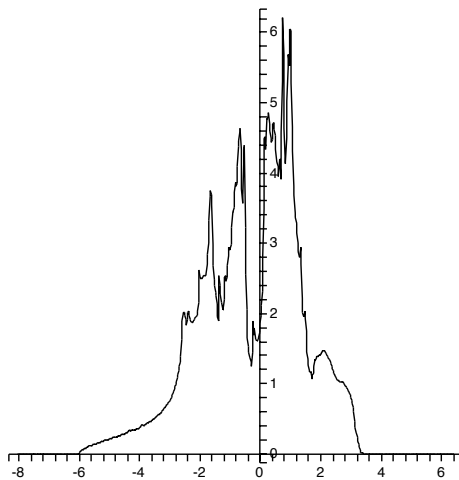


図 A.1: DOS のグラフ.

A.5 VASP マニュアルの和訳

本研究において使った部分を和訳したので, それを掲載する. [7]

- 6.18 NSW
NSW はイオンのステップ数を表している. NELM の中では SC ループが実行されている. Hellmann-Feynman 力と圧力はそれぞれのイオンステップにより計算される.

- 6.19 NBLOCK and KBLOCK

NBLOCK はイオンステップの相互関係を表しているその DOS は計算されイオンの構造は XDATCARfile に書かれる。それに加えて NBLOCK はもし SMASS=-1 なら運動エネルギーを何回増やすのかをコントロールできる。これらの仕事による CPU のコストは NBLOCK=1 にすることでかなり減らすことができる。平均的な相互関係のとれた KBLOCK*NBLOCK の繰り返しの後に DOS は PCDAT と DOSCAR に値が表記される。

- 6.20 IBRION,NFREE

IBRION はイオンがどのように置き換えられ動かされるのかを決めるパラメーターである。他のすべてのアルゴリズムが局所的な最小エネルギーを緩和することになっているのに対して IBRION=0 は分子力が機能される。難しいリラクゼーションの問題を扱う場合は共役勾配法 (IBRION=2) を使用することを薦める。現在、バックアップのとれる最も信頼できるルーチンである。始まりがよくないときに最急降下法 (IBRION=3) はしばし役立つ。局所的な最小値の近似を行う RMM-DIIS (IBRION=1) がうまく動く。

- 6.22 ISIF

ISIF は応力テンソルを計算するかどうかを決めるパラメーターである。応力テンソルの計算は比較的時間がかかる。それゆえ MD's はもともと消されている。力はいつも計算されている。それに加えて ISIF は自由度 (イオン, 結晶数, 結晶の形) を変更することができる。現在セルの変更は唯一リラクゼーションを実行できるが原子力シミュレーションはできない。

そして、並列計算について、調べたこともあるので、それも掲載する。

- 3.5.15 MPI

デフォルトのままだと並列構成が生じる。再びすべてのファイルをコンパイルする必要がある。並列環境にはマシンに MPI が install されている必要がある。そして、ライブラリのパスは makefile に明記する必要がある。MPI を動かすには mpif.h を含むファイルが必要という小さな問題がある。それゆえその mpif.h というファイルを VASP.4 のディレクトリからコピーする必要がある。そして、f90 スタイルを変換し、mpif.h に名前を変える必要がある。変換というのは F77 の fortran ファイルを F90 フォーマットに変える。そして、VASP.4 ディレクトリから供給される。

- 3.5.16 MPI-CHAIN

しなやかな帯構造を保ちコンパイルするバージョンである。mpif.h ファイルは上で説明したものと同一方法で作る。ほとんどのファイルはシリアルバージョンと同じ方法でコンパイルする必要がある (例えば何の並列もない FFT サポートは必要)。この場合、IMAGE タグと同じ数のノードで走らせなければ

ばならない。このバージョンはシリアルバージョンと同じぐらい早く (フル MPI バージョンよりも早い), とても効率よくワークステーションのクラスターで走らせることができる。

- 3.5.17 use-collective

LAMやMPICHのようなMPIパッケージのほとんど役に立たない手段があるから、VASP4.5はMPIが使えるVASPのバージョンで集合的なコミュニケーションを避ける。(SGI基点とT3E上で)他方では、MPIルーチンの共同体は高精度で最適化される。MPIルーチンがアーキテクチャによって最適化されるときならいつでもuse-collectiveはこれらのシステムに明記される。

- 3.5.18 MPI-BLOCK

現在、VASPは即座にMPIを送信し、小さなデータよりも先に大きなデータを呼び出し受け取るという作業を分割している。大きな障害が100Mbitもしくは1GbitイーサネットにリンクしたLINUXクラスターに大きな帯域幅の収縮を起こしている。MPI-BLOCKはブロックサイズを決定する。もし、use-collectiveが使われるなら、MPI-BLOCKは唯一広範囲のルーチンが使える。

- 3.5.20 scaLAPACK

VASPはLU分解と対角行列を用いたLAPACKの代わりにscaLAPACKを使うだろう。これらの演算はシリアルバージョンなのでとても早いですが、多くの電荷によって大規模に並列計算機のシステムが詰まる。もしscaLAPACKが並列計算機にインストールされているなら(T3E,SGI,IBM,SPX)というスイッチが使える。scaLAPACKは3.0.1.0環境のプログラミングによってT3Eが使える。追加スイッチのT3Dは明記されている、少なくともscaLAPACKバージョンはテストされている(T3DのscaLAPACKは普通のscaLAPACKルーチンとは互換性がない)。穏やかなネットワークとPCクラスター(100Mbitイーサネットと1Gbitイーサネットでさえ)でscaLAPACKを使うことは薦められない。動作の改良は取るに足らなく、scaLAPACKはLAPACKより遅い。もし、それがほしいなら、www.netlib.org/SCALAPACKからソースをダウンロードできる。編集は簡単にできるが、MPIフォートラン、C、UNIXに精通することが必要だ。(いつも根本的なBLACSルーチンは正確にうごいているのは明確だ)。scaLAPACKはINCAR.fileのLSCALAPACK=.FALSE.と定義すれば動く。scaLAPACKの問題にぶつかったときにこれを代替策として使用する。さらにscaLAPACKを基本としたLU分解はシリアルLU分解よりも遅い。それゆえINCAR fileにLSCALU = .FALSE.と定義すれば並列LU分解にスイッチすることが可能だ。(規則正しい部分空間はこの場合scaLAPACKによってなされる。)

謝辞

本研究を遂行するにあたり，終始多大なる御指導および御教示を賜りました西谷滋人教授に深く感謝の意を表すとともに，厚く御礼申し上げます。そして，もっとも身近で研究に協力して頂きました同研究室の皆様に心から感謝の意を表します。

参考文献

- [1] 「Cu 添加極低炭素鋼における Cu 析出の初期段階」
佐野直幸 前原泰裕 著
- [2] wikipedia
<http://ja.wikipedia.org/wiki/RSS>
- [3] important glossary for electronic structure calculations
<http://www.geocities.co.jp/Technopolis/4765/INTRO/yogo.html>
- [4] 「ニューメリカルレシピ in C-C 言語による数値計算レシピ」
W.H.Press. B.P.Flannery S.A.Teakolshy W.T.Vetterling 著,
丹慶勝市, 奥村晴彦, 佐藤俊朗, 小林減, 訳
1993(技術評論社 専字)
第 10 章 6 節
- [5] 「ニューメリカルレシピ in C-C 言語による数値計算レシピ」
W.H.Press. B.P.Flannery S.A.Teakolshy W.T.Vetterling 著,
丹慶勝市, 奥村晴彦, 佐藤俊朗, 小林減, 訳
1993(技術評論社 専字)
第 10 章 5 節
- [6] 「ニューメリカルレシピ in C-C 言語による数値計算レシピ」
W.H.Press. B.P.Flannery S.A.Teakolshy W.T.Vetterling 著,
丹慶勝市, 奥村晴彦, 佐藤俊朗, 小林減, 訳
1993(技術評論社 専字)
第 10 章 7 節
- [7] VASP the GUIDE
<http://cms.mpi.univie.ac.at/VASP/>