卒業論文

格子欠陥の視覚化

関西学院大学理工学部

情報科学科 西谷研究室

0514 平硲 直樹

平成 26 年 2 月 17 日

概 要

物理学の理論，数式を実際の現象において考えるには，そのモデル化が重要になるが，模式的なモデルではその理解が困難となる．本研究室で主に取り扱っている物理現象として，LPSO 構造の生成機構，Al の対称傾角粒界などが挙げられる．これらには結晶中に格子欠陥が存在する．1 次元欠陥である転位およびその部分転位に挟まれた積層欠陥は，模式図で理解することは容易であるが，現在の原子位置から理解するのは難しい．また，第一原理計算で求めるエネルギー値も文面だけでは理解することが困難である．そこで，格子欠陥とエネルギーの変化をイメージすることができれば研究の助けになると考えた．本研究では，物理現象を 3 次元的に視覚化することで，より明確なイメージを与え，直感的に理解できるようにする目的で行った．

視覚化ツールとして，Processing を使用した．Processing は無料でダウンロードすることができ，自分でコード開発ができるので，格子欠陥を視覚化することに関して適している．しかし，VASP の計算で求められた原子位置入力ファイル POSCAR をそのまま使用することができない．そこで，Ruby を使用し，Processing で読み込むことが出来る tsv ファイルに変換し，Processing で結晶構造やエネルギーの視覚化を行った．格子欠陥に色付けをし，拡大・縮小などの機能も追加して，見やすくした．

さらに，積層欠陥や部分転位について，様々な視点（方向）からの視覚化を行った．近年は，高分解能顕微鏡像で原子の様子を見ることは可能であるが，その視点は固定されている．今回の研究結果によって，様々な視点（方向）からの確認が可能となり，理論面での学習を行う際にはとても有効なものなった．

目 次

第 1 章　序論 **3**

1.1　背景 3

1.2　格子欠陥 3

1.2.1　結晶構造と格子定数 3

1.2.2　ミラー指数 4

1.2.3　hcp 構造と fcc 構造における積層周期 6

1.2.4　格子欠陥 8

1.2.5　転位 8

1.2.6　バーガース・ベクトルとバーガース・サーキット 8

1.2.7　部分転位 10

1.2.8　積層欠陥 10

第 2 章　視覚化 **13**

2.1　視覚化ツール 13

2.1.1　Maya 13

2.1.2　VESTA 14

2.1.3　Processing 14

2.2　視覚化のアルゴリズム 15

2.2.1　Ruby 16

2.2.2　Processing 20

第 3 章　実行結果 **22**

3.1　視覚化 22

第 4 章　考察 **30**

4.1　LPSO 構造 30

4.2　Al の小傾角粒界 31

4.2.1 転位の視覚化 31
4.2.2 エネルギーの視覚化 32

第 5 章 総括 **34**

第1章　序論

1.1　背景

金属などの物質を原子単位で見たときに，原子球が3次元的に規則正しく並んでいるわけではなく，実際の結晶には何らかの乱れが含まれている．それらを総称して，格子欠陥と呼ぶ．格子欠陥には点欠陥や，線欠陥，面欠陥がある．線欠陥は特に転位と呼ばれ，材料の変形を左右する重要な欠陥である．また hcp 構造中に入った転位は，部分転位に分解することが知られている．この2本の部分転位は広がり，その間に積層欠陥と呼ばれる面欠陥を形成する．これらの格子欠陥は，模式図で理解するのは容易であるが，3次元的な原子モデルから理解するのは非常に困難である．そこで本研究では，原子モデル上のどの位置に転位等の格子欠陥が存在するのか視覚化するシステムの開発を目的とした．そして本研究室で研究対象としている Mg の長周期積層欠陥（LPSO）構造や，Al の小傾角粒界における格子欠陥の視覚化を行った．

1.2　格子欠陥

本節では，格子欠陥を理解する上で必要となる結晶構造や，ミラー指数の専門用語について記述する．

1.2.1　結晶構造と格子定数

図 1.1 に代表的な結晶格子を示す．図 1.1(a) には最も単純な規則配列を示した．ちょうど赤線で造られる立方格子の格子点に原子が配置されているので，この結晶構造を単純立方格子（simple cubic lattice）という．また大きな結晶構造を造る際に，積み上げる単位ブロック（ここでは赤線で示したブロック）を単位胞（unitcell）という．図 1.1(b) には立方格子の格子点と重心に原子が配置されている体心立方構造（body ceteted cubic:bcc）を示した．鉄（Fe）やタングステン（W）の結晶がこの構造をとる．図 1.1(c) には立方格子の格

子点とすべての面の中心に原子が配置されている面心立方構造（face centered cubic:fcc）を示した．この構造では，原子がすべて同じ大きさの真球である場合に最も密な原子配列をとるため，立方最密格子ともいう．例えば，Al，金（Au），銀（Ag），銅（Cu），イットリウム（Y）の結晶がこの構造をとる．図 1.1(d) には六方最密充填構造（hexagonal closed package:hcp）を示した．この構造は fcc 構造と同様に最密構造であるが異なる構造である．例えば，Mg，亜鉛（Zn）がこの結晶構造をとる．hcp 構造と fcc 構造は LPSO 構造を考える上で重要な構造であるため，1.2.3 小節で詳しく解説する．

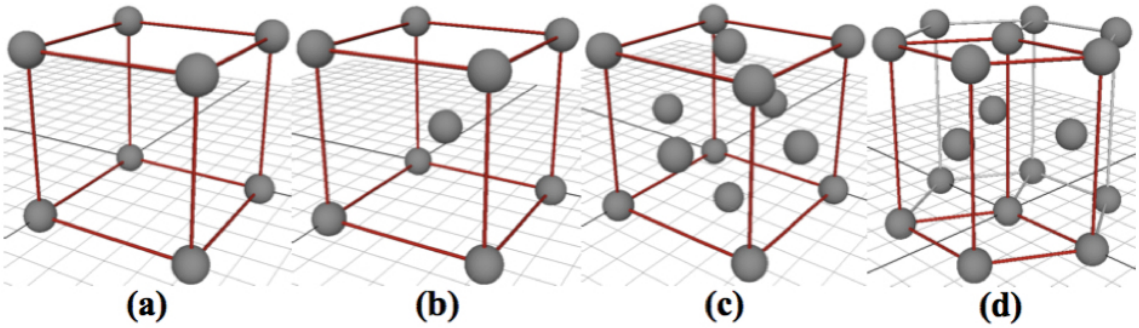

図 1.1: 代表的な結晶構造の格子の形と原子配置の様子．(a) 単純立方格子 (simple)．(b) 体心立方構造 (bcc)．(c) 面心立方格子 (fcc)．(d) 六方最密充填構造 (hcp)．

結晶は，格子に内包する原子に応じて特定の結晶軸の長さや軸間角度をとる．その単位胞の各軸間の角度 α，β，γ と角軸の長さ a，b，c を表す 6 個の定数を格子定数という．また格子定数は，3 次元空間におけるベクトル $\boldsymbol{a}$，$\boldsymbol{b}$，$\boldsymbol{c}$ で表すこともできる．格子定数 a，b，c，α，β，γ が表す値を図 1.2(a) に示した．立方格子の場合，$a = b = c$，$\alpha = \beta = \gamma = 90°$ が成り立つため，格子定数 a のみで表すことができる．六方晶における格子定数が表す値を図 1.2(b) に示した．hcp 構造の場合，$a = b \neq c$，$\alpha : \beta : \gamma = 90 : 90 : 120°$ が成り立つ．hcp 構造を議論する際，a 軸と c 軸の長さの比を表す c/a を用いることが多い．補足だが，原子を真球であるとき $c/a = 1.63$ が成り立ち，この比を c/a の理想軸比と呼ぶ．

1.2.2 ミラー指数

結晶内の方向や面における構造の異方性を考えるためにミラー指数が用いられる．格子面が $(a : b : c)$ 軸上で切る点 $(x : y : z)$ に対し，ミラー指数は $(1/x : 1/y : 1/z)$ で定まる．図 1.3 にミラー指数で表される面を示した．図 1.3(a) は (111) 面を示した．$(x : y : z)$ は $(1 : 1 : 1)$ であり $(1/1, 1/1, 1/1) = (111)$ となる．図 1.3(b) は (100) 面を示した．(100) 面は

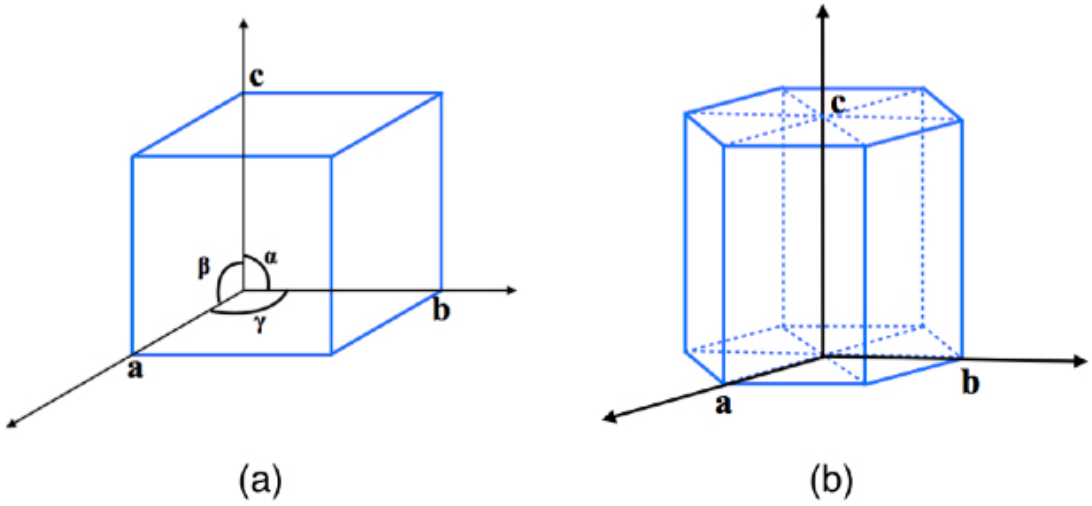

図 1.2: (a) 立法構造，(b)hcp 構造における格子定数.

$a = 1$ は定めるが b, c は無限に広がっていると考える．そのため，$(x : y : z)$ は $(1 : \infty : \infty)$ であり $(1/1, 1/\infty, 1/\infty) = (100)$ となる．図 1.3(c) は六方晶における (0001) 面を示した．六方晶では格子面を $(a1 : a2 : a3 : c)$ で考え，ミラー指数は $(1/a1 : 1/a2 : 1/a3 : 1/c)$ で定める．以上の示したミラー指数以外に，軸を負の側で切る場合がある．その場合は指数の上にマイナス記号をつけ $(\bar{h} : k : l)$ のように表示する．また，ミラー指数 (hkl) で表される面に垂直な方向を [hkl] と表す.

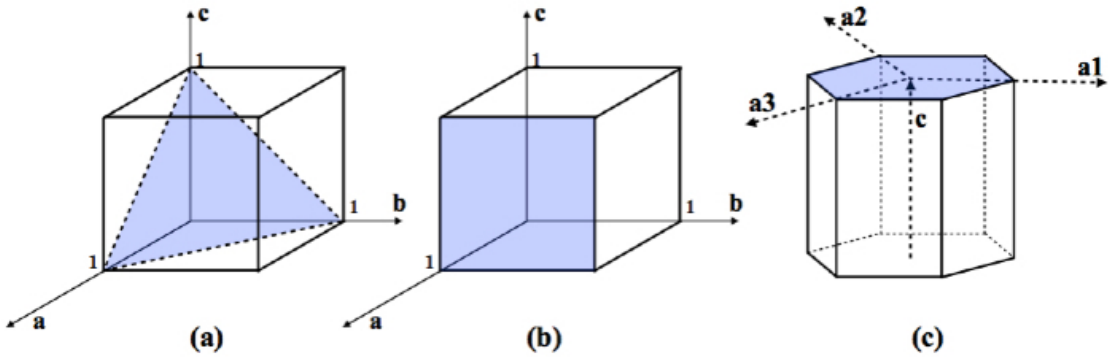

図 1.3: (a) 立方形の (111) 面．(b) 立方形の (100) 面．(c) 六方形の (0001) 面.

1.2.3 hcp 構造と fcc 構造における積層周期

hcp 構造と fcc 構造は LPSO 構造を考えるにあたって重要な構造である．hcp 構造と fcc 構造はともに最密構造であり，それぞれ (111) 面，(0001) 面に最密面を有する．これらの構造の違いは最密面の積み方，つまり積層順序の違いである．

まずは，積層順序の違いについて説明する．最密面上に最密面を積む方法を図 1.4 に示した．青丸で示した最密面上に最密面を積むパターンは，図 1.4(a) もしくは図 1.4(b) がある．それぞれ，図 1.4(c) に黒線で示した菱型内にある 2 つの正三角形上の，どちらかの重心に原子が積まれていることがわかる．よって青の最密面を A 面とすると，A 面の上に詰める面は緑の最密面である B 面，もしくは赤の最密面である C 面である．また，B 面に積める面は A 面と C 面であり，C 面に積める面は A 面と B 面である．このように積層順序は，最密面の相対的な位置で A,B,C 面を特定することができる．

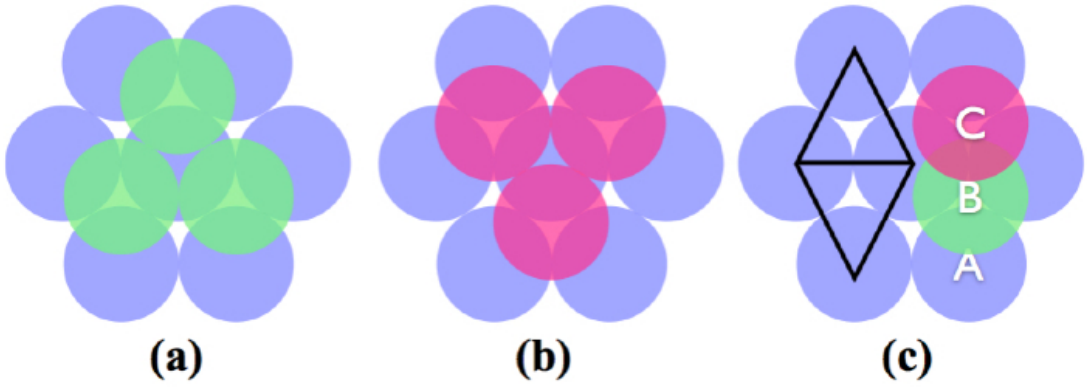

図 1.4: 最密面の積み方の様子．最密面上への積層は，(a),(b) のいずれかである．(c) に先の 2 通りの最密面の積層を示す．(a) は (c) に示す黒いひし形の逆三角形の重心にあたり，(b) は正三角形の重心にあたる．最密面の積層順序は相対的な位置で A,B,C を区別できる．

次に，hcp 構造と fcc 構造の積層順序の違いについて説明する．図 1.5(a) に hcp 構造，(b) にその $(11\bar{2}0)$ 面を示した．この図より，hcp では [0001] 方向に最密面である A 面，B 面の 2 種の層が交互に積層していることがわかる．また，図 1.6(a) に fcc 構造，(b) にその $(1\bar{2}0)$ 面を示した．この図の (a) と (b) における原子の色は対応しており，fcc 構造における立方格子は $(1\bar{2}0)$ 面に示したように位置している．この図より，fcc 構造の [111] 方向に最密面である A 面，B 面，C 面の 3 種の層が順に積層していることがわかる．以上のことから，hcp 構造は...|AB|AB|... の 2 層 1 周期の構造であるのに対し，fcc 構造は...|ABC|ABC|... の

3 層 1 周期の構造であるという違いがわかる．

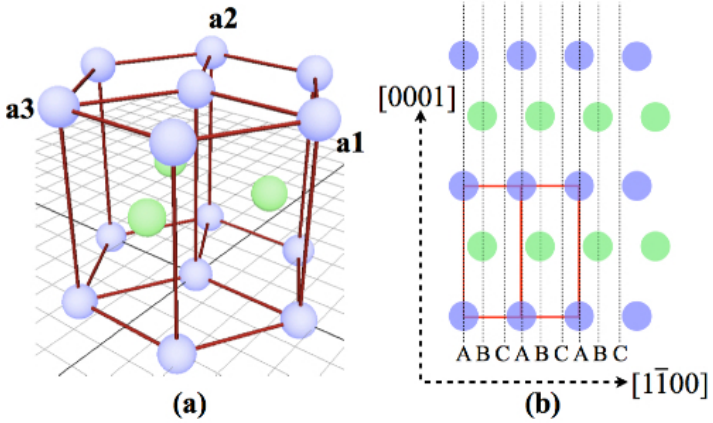

図 1.5: hcp 構造の積層順序．(a) は hcp 構造を示し，(b) は (11$\bar{2}$0) 面から見た hcp 構造を示している．

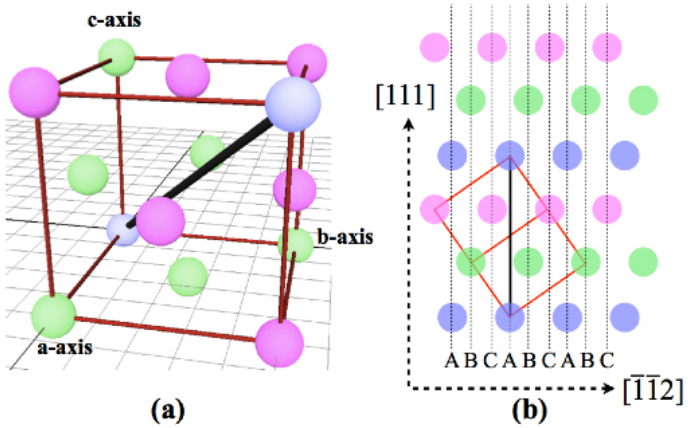

図 1.6: fcc 構造の積層順序．(a) は hcp 構造を示し，(b) は (1$\bar{1}$0) 面から見た hcp 構造を示している．

この hcp 構造と fcc 構造の違いから，結晶格子において ABA のような積層順序の箇所を hexagonal 構造 (“hexagonal” closed package)，ABC のような積層順序となる箇所を cubic 構造（face centerd “cubic”）とみなすことができる．また，最密構造は hcp 構造，fcc 構造の他にも存在する．例えば，...|ABCB|... のように 4 層 1 周期である 4H 構造，...|ABCACB|... のように 6 層 1 周期である 6H 構造が存在する．他にも，18 層 1 周期である 18R 構造も存在し，この構造は本稿で扱う LPSO 構造の 1 種である．このような結晶多形の表記法は

「Ramsdell の表記法」と呼ばれ，先の数字が 1 周期における層数，後のアルファベットは結晶系の頭文字「C:立方晶 (cubic)，H:六方晶 (hexagonal)，R:菱面体 (rhombohedral)」を示す．

1.2.4 格子欠陥

金属材料はミクロレベルで見ると三次元的に構成原子が規則正しく並んだ完全結晶であるが，実際の結晶には何らかの規則の乱れが含まれている．それらを総称して格子欠陥と呼ぶ．格子欠陥には 0 次元的な欠陥の点欠陥や，1 次元的な欠陥の線欠陥や 2 次元的な欠陥の面欠陥がある．線欠陥は転位と呼ばれ，材料の変形を左右する重要な欠陥である．代表的な転位として刃状 (edge) 転位とらせん (screw) 転位などがある．

1.2.5 転位

刃状転位について述べる．図 1.7 には 1 原子間距離のすべりがすべり面の左半分で起こっているが，右半分では起こっていないような単純立方格子を示す．図 1.7 の FE を転位という．転位の位置は，図 1.8 に示されるように，結晶の上半分に余分に挿入された垂直な半平面をなす原子面の端で示される．転位の近くでは結晶は大きい歪を受けている．単純な刃状転位は，すべり方向に垂直に，すべり面上をどこまでも続いている．

1.2.6 バーガース・ベクトルとバーガース・サーキット

転位の性質は，すべり方向への大きさ b を持つベクトル $\mathbf{b}$ によって定義づけられる．このベクトルのことをバーガース・ベクトルという．図 1.9(a) に示す刃状転位を含む実際の結晶を考える．まず，転位線の向きを決めるために，転位線と平行に単位ベクトルを定義する．次に，結晶格子の任意の原子 (S) から，時計回りに転位線の回りを回って同じ位置 (F) に戻る閉じた回路を作る．これをバーガース・サーキットという．図 1.9(b) は格子欠陥を全く含まない理想結晶である．この理想結晶内で，図 1.9(a) のようにして作成した実際の結晶のバーガース・サーキットと対応する回路を作ってみる，始点 S と終点 F が一致しない．転位のバーガース・ベクトルの定義は，F と S を結ぶベクトルである．

結晶内に閉曲線かまたは両端が結晶の表面で終わっている開曲線を考える．(a) この曲線を周縁とする任意の単純曲線に沿って結晶を切断する．(b) この面の一方の側の結晶を他方の側に対しベクトル $\mathbf{b}$ だけずらす．この $\mathbf{b}$ をバーガース・ベクトルという．

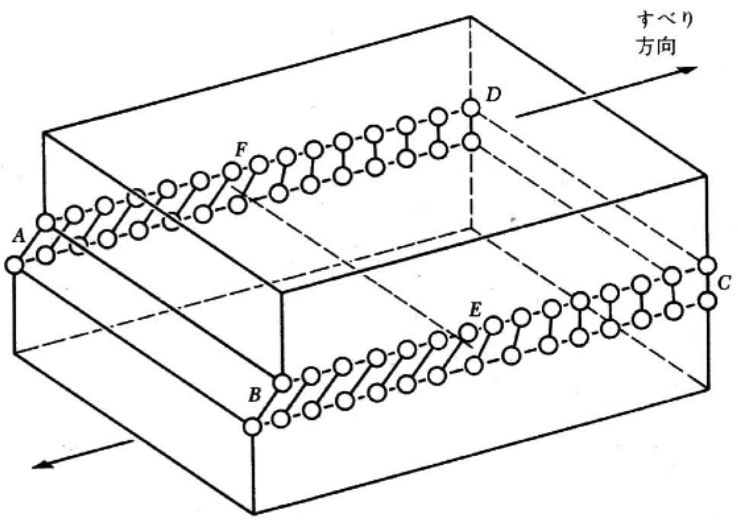

図 1.7: すべり面 ABCD にある刃状転位 EF．原子が格子定数の半分以上変異しているようなすべりの生じている領域 ABEF と，変異が格子定数の半分以上であるすべりの生じていない領域 FECD が示されている [1].

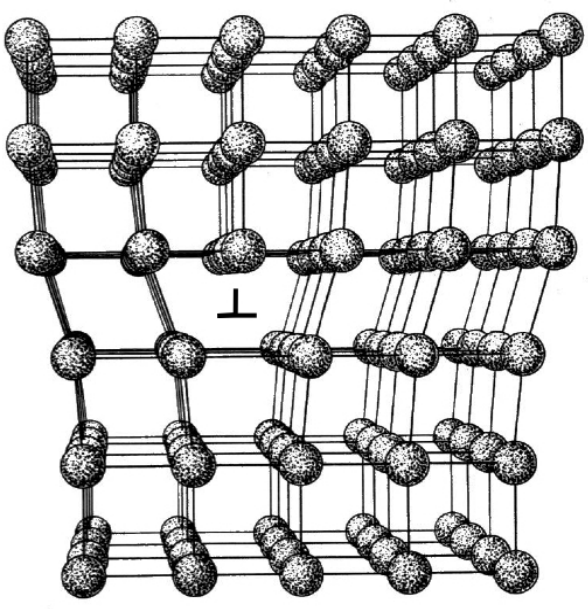

図 1.8: 刃状転位の構造．y 軸の上半分に余分の原子価を挿入したために，変形していると考えられる．この挿入で，上半分の原子は圧縮され，下半分の原子は引き伸ばされる [1].

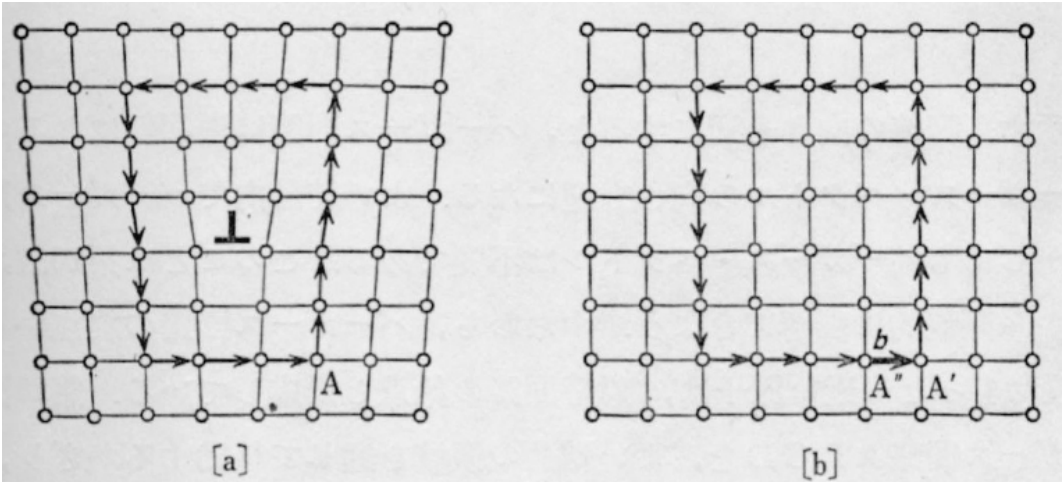

図 1.9: 実際の結晶のバーガースサーキットと理想結晶の対応サーキット [2].

1.2.7 部分転位

すべり面上の原子がずれた領域とまだずれていない領域の境界線が，転位として存在する．転位がその結晶格子のベクトルと等しいベクトル $\mathbf{b}$ のようなバーガース・ベクトルをもつとき，その転位を完全転位と呼ぶ．また一致しない転位を部分転位と呼ぶ．図 1.10 は完全転位の模式図を示した．ベクトル $\mathbf{b}$ があるため，完全転位となっている．図 1.11 は部分転位の模式図である．積層欠陥 $\mathbf{d}_0$ の端に部分転位 $\mathbf{b}_{pa}$，$\mathbf{b}_{pb}$ がある．

1.2.8 積層欠陥

積層欠陥とは，積層順序の連続性が局所的に乱れた欠陥である．例えば hcp 金属における部分転位の間は，fcc 構造をもつ局部領域となっており，積層欠陥を形成している．hcp 構造における積層欠陥の様子を表した模式図を図 1.12 に示した．hcp 構造ではこの図で示すように [0001] 方向に最密面が ABAB と積層している．この時，赤枠で囲った原子を赤矢印の方向にずらすと，積層順序が ABCA となる．そして hcp 構造上に発生した積層欠陥面の上下の白丸で示した層を中心とした積層順序を考えると，それぞれ ABC,BCA となっていることがわかる．このことから hcp 構造において積層欠陥が発生すると cubic 構造である fcc 構造が導入されることがわかる．積層欠陥は，結晶成長時に生じたり，また一本の転位が二つの部分転位に分離して拡張転位を作るときに生じる．この二つの部分転位で囲まれた領域が積層欠陥である．

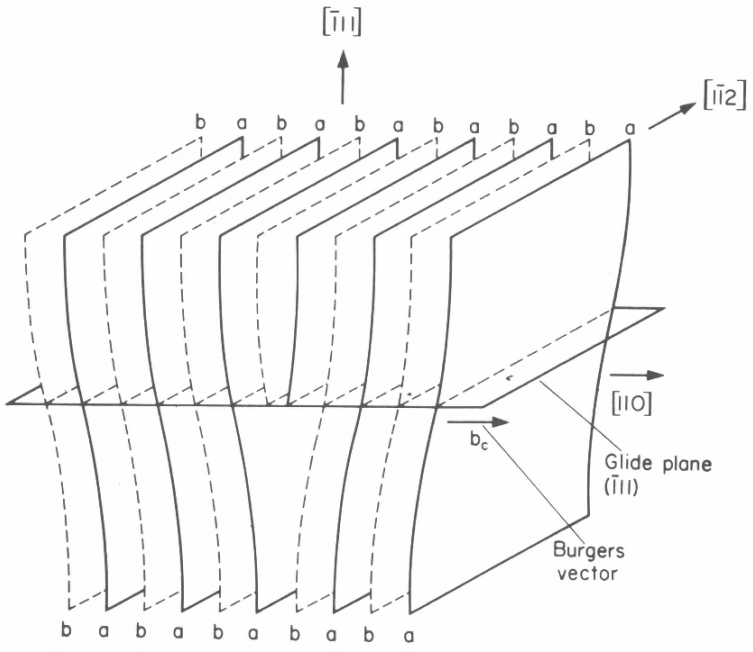

図 1.10: 完全転位の模式図 [3].

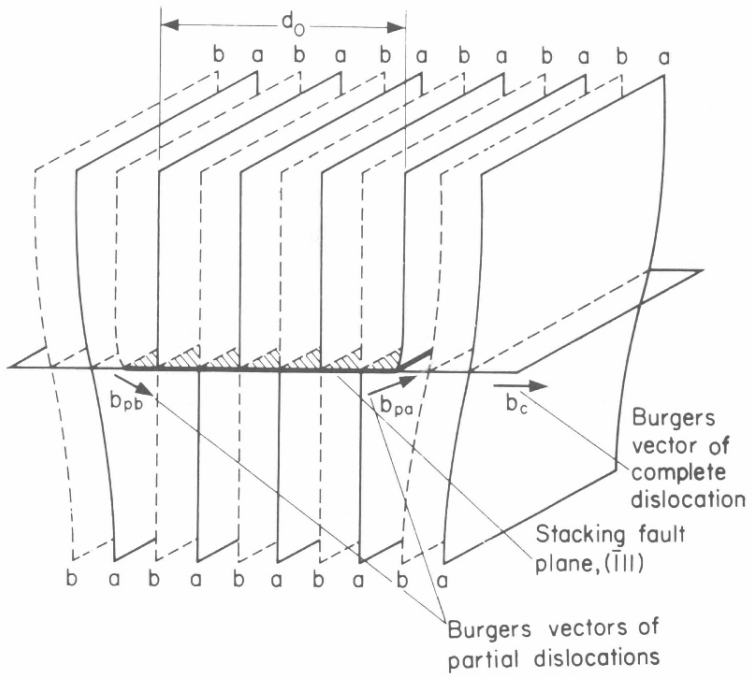

図 1.11: 部分転位の模式図 [3].

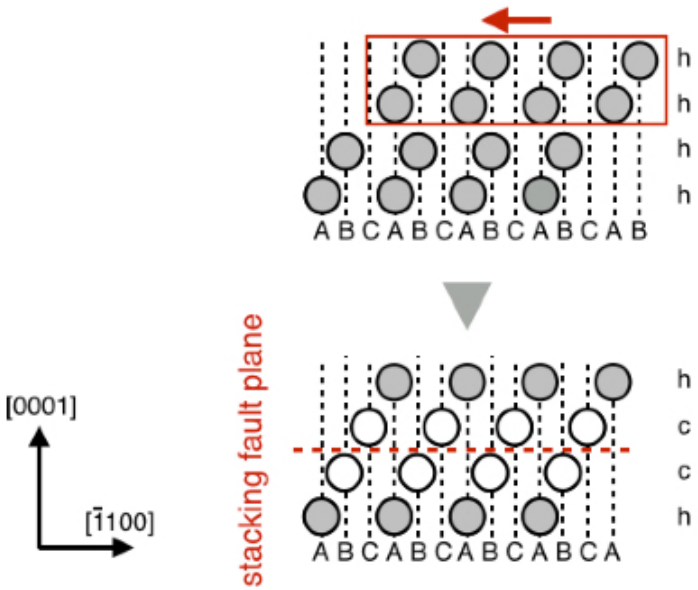

図 1.12: hcp 構造における積層欠陥の様子を表した模式図. グレーの丸は hcp 構造，白丸は fcc 構造を示している．また赤の破線部は積層欠陥面である．

第2章　視覚化

本研究室では，原子の位置やエネルギーを第一原理計算ソフト VASP を用いて計算している．そのため，VASP の計算で求められた原子位置入力ファイル POSCAR をそのまま使用してシミュレーション結果を視覚化できるソフトとして，結晶構造描画ソフト VESTA を使用してきた．しかし，VESTA は自分でコードを書き換えることが困難であるため拡張性に欠け，積層欠陥や転位部分に色付けすることができない．そこで結晶をより詳しく見たいときには結晶モデル構築ソフト Maya を使用していた．しかし，Maya は 3D アニメーションに関連する機能は充実しているが，ライセンス料がとても高価である．一方，Processing は無料でダウンロードすることができ，自分でコード開発することが出来るソフトである．Processing とはグラフィック機能を中心とした Java を単純化した言語で，画像処理やアニメーションに特化しているといった特徴を持っている．　以上の 3 つの視覚化ツールの特徴について詳しく説明する．

VASP: 密度汎関数法による平面波・擬ポテンシャル法を用いた第一原理計算プログラムパッケージである．擬ポテンシャル法は原子の内殻電子を除いた価電子だけを考慮する手法である．そのため，全電子を計算するフルポテンシャル方に比べ比較的高速な計算が可能となる．また，内殻電子は化学結合や物性に影響を与えることが少ないため，擬ポテンシャル法であっても十分な精度で計算できる．

2.1　視覚化ツール

2.1.1　Maya

Maya は，オープンソースアーキテクチャを基板とした強力な統合型の 3D モデリング，アニメーション，レンダリングソリューションである．多くのフィルムやビデオアーティスト，ゲーム開発者，マルチメディアデザイナー，3DCG に関わる SOHO デザイナーなどが使用しているプロ仕様のハイエンドソフトである．グラフィックス性に優れ，非常に高

度な機能を有しているため，自由度も高く，アニメーション機能が充実している．また，Maya はその GUI の全てを MEL というスクリプト言語で実行可能となっている．しかしライセンス料が非常に高価であるため，Web に上げても Maya が入っていないパソコンから見ることが出来ない．読み込むプログラムの量が多いと途中で勝手に改行され，正常に読み込まれない場合があるなどコード開発が非常に困難である．

MEL: MEL(Maya Embedded Lanquage) とは Maya で使用できるスクリプト言語であり，Maya の GUI の機能の全てをまかなうことが可能となっている．MEL のみで CG の作成，カスタマイズを行うことができるが，Maya 専用のスクリプトエディタ上でしか実行できない．また，線形計算などの数値計算に向いておらず，構文も複雑な構成となっている．

2.1.2 VESTA

西谷研究室では，第一原理計算を行うソフトとして VASP を使用している．VASP には原子位置入力ファイルとなる POSCAR を用いている．その POSCAR を VESTA に直接入力し，原子位置を簡単に確認することが出来る．他にも VESTA は無料でダウンロードすることができ，高度な専門知識抜きで使いこなせ，広い学術分野をカバーしている. 結晶構造，電子・各密度分布等の 3 次元可視化プログラムで，20 種以上のファイル形式から構造データの入力ができる．詳細な英文マニュアルを備えていることから，世界的に普及し，数多くの研究に貢献している．しかし，ソフトが完成しているので，自由にコードを書き換えることが困難であるため，自由度がとても低い．ソフトにある機能しか出来ないので，拡張性に欠ける．

2.1.3 Processing

Processing とは Ben Fry 氏と Casey Reas 氏が開発したコンピュータ上で簡単にグラフィックを扱って動かすことができるプログラム開発環境である．誰でも公式サイトから無料でダウンロードして利用することができる．シンプルで軽く，初めての人でも使いやすい環境である．sphere で表示するとプログラムが重くなってしまい，box にしなければいけない．そのため box で表示すると動きやキレイさでは Maya や VESTA の方が優れている．しかしシンプルなプログラムで実装することができ，自由度も高く，Web から無

料でダウンロードすることができる．そして web に上げれば，web 上で動かすことも可能になる．

これら 3 つのツールの特徴を以下の表にまとめた．

表 2.1: 比較表

	Maya	Processing	VESTA
動き	◎	○	◎
キレイさ	◎	○	◎
自由度	○	○	×
安価	×	○	○
Web 対応	×	○	×
コード開発	○	◎	×

本研究では，格子欠陥を視覚化することが目的であるため，格子欠陥の色分けをする機能が備わっていない VESTA では格子欠陥を詳しく見ることが出来ない．また，視覚化したモデルを Web 上で共有したい．そのため，ライセンス料が高価である Maya は共有できる PC が限られてしまうことから，本研究では，格子欠陥の色分けや安価である Processing を使用する．

2.2 視覚化のアルゴリズム

Processing で POSCAR を読み込ませるためには，プログラムが複雑になる．そのため，処理にとても時間がかかってしまう．そこで効率良くするために，Ruby で POSCAR を tsv ファイルに変換や格子欠陥の色分けの処理を行ってから，Processing で表示させる方法を使用する．

- Ruby
  - POSCAR 読み込み．
  - 絶対座標変換．
  - ネイバーリスト作成．
  - 原子の色分け．
  - tsv ファイル出力．

- Processing
  - tsv ファイル読み込み.
  - 3D 表示.

2.2.1 Ruby

POSCAR 読み込み

POSCAR とは，本研究室で以前から使っている原子位置入力ファイルである．図 2.1 が POSCAR ファイルの例である．青枠は格子定数の倍率，赤枠は格子ベクトル，黒丸は原子数，緑枠は格子内における原子の相対座標となっている．赤枠の 1 行目が a 軸，2 行目が b 軸，3 行目が c 軸，1 列目が x 軸，2 列目が y 軸，3 列目が z 軸に対応する．緑枠の 1 列目が a 軸，2 列目が b 軸，3 列目が c 軸に対応する．

```
New structure
      1.0000000000000000
        3.200709999999999944   0.000000000000000000   0.000000000000000000
       -1.600354999999999750   2.771896170146890714   0.000000000000000000
        0.000000000000000000   0.000000000000000000  15.446886666666665988
Mg
   6
Direct
        0.000000000000000000   0.000000000000000000   0.000000000000000000
        0.333333333333333315   0.666666666666666630   0.166666666666666657
        0.666666666666666630   0.333333333333333315   0.333333333333333315
        0.000000000000000000   0.000000000000000000   0.500000000000000000
        0.333333333333333315   0.666666666666666630   0.666666666666666630
        0.666666666666666630   0.333333333333333315   0.833333333333333370
```

図 2.1: POSCAR ファイル.

まず格子ベクトルと相対座標を入れる配列 (prim_vector，opp_vector) を定義する．そして POSCAR を読み取る．図 2.1 のように，POSCAR の 5 行目までは決まった配置になっているので，2 行目から 4 行目の格子ベクトルを配列 (prim_vector) に，Direct を見つけたらその次から相対座標なので配列 (opp_vector) に格納する．

POSCAR 読み取り

```
1: prim_vector=[]
2: opp_vector=[]
3: check=false
4: count=0
5: File.open(filename).each do |line|
6:   if count>1 && count<5 then prim_vector << line.chomp.split(" ") end
7:   if /Direct/ =~ line then check=true
8:   elsif line==" \n" then check=false
9:   elsif check then opp_vector << line.chomp.split(" ") end
10:  count=count+1
11: end
```

絶対座標変換

prim_vector と opp_vector に格納したものは文字とみなされているため，数字に変換する．そしてそれらをかけて絶対座標に直す．例えば図 2.1 の計算の方法としては以下の行列のように計算する．

$$vector = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \end{pmatrix} \begin{pmatrix} a_i \\ b_i \\ c_i \end{pmatrix}$$

その絶対座標を新たな配列 (vector) に格納する．

絶対座標変換

```
1: def ch_prim(prim_vector,opp_vector)
2:   x=prim_vector[0][0].to_f*opp_vector[0].to_f+prim_vector[0][1].to_f
      *opp_vector[1].to_f+prim_vector[0][2].to_f*opp_vector[2].to_f
3:   y=prim_vector[1][0].to_f*opp_vector[0].to_f+prim_vector[1][1].to_f
      *opp_vector[1].to_f+prim_vector[1][2].to_f*opp_vector[2].to_f
4:   z=prim_vector[2][0].to_f*opp_vector[0].to_f+prim_vector[2][1].to_f
      *opp_vector[1].to_f+prim_vector[2][2].to_f*opp_vector[2].to_f
5:   vector = [x,y,z]
6:   return vector
7: end
```

絶対座標格納

```
1: def absolute_coordinate(prim_vector, opp_vector)
2:   vector=[]
3:   opp_vector.length.times do |i|
4:    vector << ch_prim(prim_vector,opp_vector[i])
5:  end
6:  return vector
7: end
```

ネイバーリスト

ネイバーリスト (以下 n_l) とは近接している原子の集合を意味する．再近接原子を計算するにあたり全ての原子間の相互作用を考慮すると計算に時間がかかりすぎるので，影響力の弱い，原子から遠い位置にある原子との相互作用を遮断する．図 2.2 のように赤色で示す 1 個の原子から青色で示す規定の範囲内にある原子を n_l に登録する．ここでは再近接原子よりも 1 つ奥にある原子の手前までを n_l 範囲に指定することで再近接原子のみを n_l に登録する．図 2.2 は単純立方格子の例である．

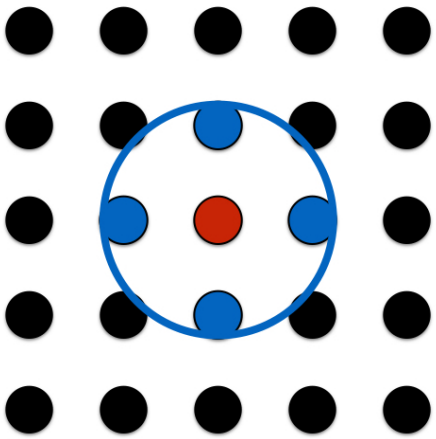

図 2.2: 単純立方格子の例で再近接原子のイメージである．青丸が再近接原子である．

原子 i と他の原子 j の原子間距離を求める．その原子間距離が再近接原子間距離の範囲内にあるか調べる．もし範囲内にあれば，n_l に格納していく．本研究では，Mg 結晶における視覚化の場合は 3.20Å，Al 結晶の場合は 2.85Å と設定している．また他の結晶の視覚

化を行う場合は，そのたびに再近接原子間距離を変更する．

原子間距離

```
1: def atom_distance(a,b)
2:   return sqrt((a[0]-b[0])**2+(a[1]-b[1])**2+(a[2]-b[2])**2)
3: end
```

ネイバーリスト

```
1: def make_n_l(bond_length, abst_coord)
2:  n_l=[]
3:  for i in 0..abst_coord.length-1 do
4:   atom=[]
5:    for j in 0..abst_coord.length-1 do
6:     if i==j then next end
7:     if atom_distance(abst_coord[i], abst_coord[j]) <= bond_length*1.1 then
8:      atom << j
9:     end
10:   end
11:   n_l << atom
12:  end
13:  return n_l
14: end
```

原子の色分け

原子モデル内において，格子欠陥を見分けるために原子の色分けを行った．hcp 中の積層欠陥は fcc である．このような構造の違いを判別し，積層欠陥を色分けする．図 2.3(a), (b) に hcp，(c) に fcc のそれぞれ模式図を示した．[0001] 方向において，hcp の構造は上下の原子の軸が同じである．例えば，図 2.3(a)，(b) のモデルにおいて，上下両方の原子が A となっている．一方，fcc の構造は上下の原子の軸が異なっている．例えば，図 2.3(c) のモデルにおいて，上下の原子が A，C となっている．そのため，モデル内のある原子がどちらの構造になっているのか判別するには，上下の原子の軸を比較すればよい．このように構造の判定を行い，積層欠陥である fcc の原子に色付けを行った．

図 2.4 の赤で囲った原子を見ると，他の原子より配位数が 1 つ少ないことがわかる．こ

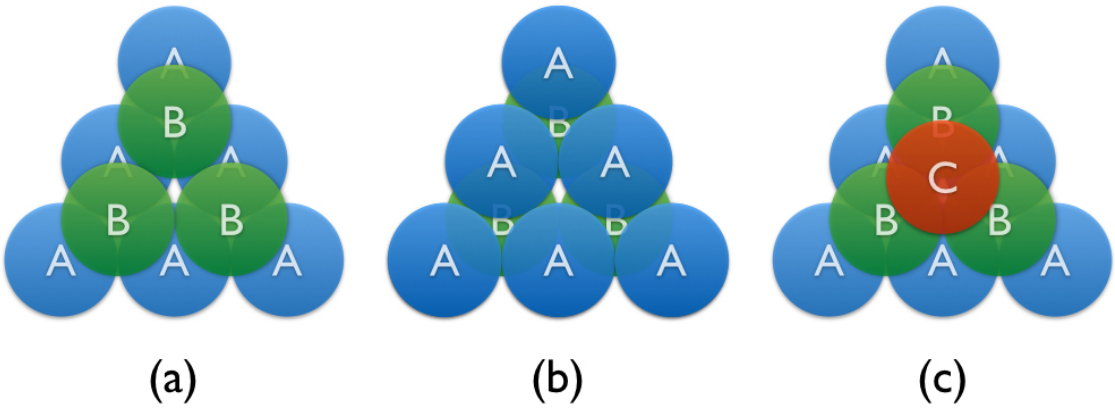

図 2.3: fcc と hcp の最密原子の面の積み重なり方．(a)，(b) は hcp，(c) は fcc である．

れより，転位部分は配位数が少ないことが示唆される．そこで，配位数の違いで色分けを行った．ある原子の n⊥l の再近接原子を数える．そして，hcp と fcc の配位数はそれぞれ 12 であるので，積層欠陥部分で配位数が 12 より少なければ原子の色を変える．これらを元に tsv ファイルに積層欠陥を 0，転位を 2，その他を 1 と出力し，色分けを行った．

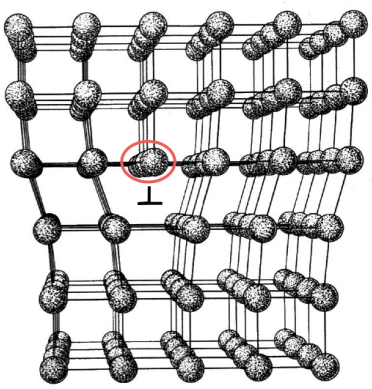

図 2.4: 刃状転位の構造 [1]．

2.2.2 Processing

Ruby で書きだした tsv ファイルを読み込み，原子モデルを表示させた．表示の仕方として，3D 表示を選定した．実際の原子は 3 次元的である．そのため格子欠陥を見る上で，実際の原子に近づける必要があったからである．原子を表示させる際に本来では元々メソッドとして備わっている 3 次元的に球を表示させる sphere() を用いたかったが，動作が

重くなり，滑らかに動かなかった．そこで 3 次元的に立方体を表示させる box() を原子の表示に使用した．また，再近接原子との位置関係などをより詳しく見るために拡大させ，構造の全体を見るために縮小の機能を追加した．さらに，拡大・縮小は座標における中心でしか行えない．そこで，確認したい場所を中心に持ってくるために上下左右に移動させる機能を追加した．

第3章　実行結果

3.1　視覚化

- Ruby
  - POSCAR 読み込み.
  - 絶対座標変換.
  - ネイバーリスト作成.
  - 原子の色分け.
  - tsv ファイル出力.
- Processing
  - tsv ファイル読み込み.
  - 3D 表示.

以上のことを行った結果を以下に示す.

tsv ファイル

POSCAR を読み込んで，出力した結果である．1 列目の data は配列に格納されない要素なので適当な文字を出力している．2 列目の x は x 座標，3 列目の y は y 座標，4 列目の z は z 座標，5 列目の color は色分けを決める数字を出力している.

tsv ファイル

```
data     x     y     z     color
data     20    -250  -103  1
data     12    -263  -103  1
data     4     -277  -103  1
data     -4    -291  -103  1
               :
               :
               :
data     -44   -175  -38   2
data     -52   -189  -38   2
data     20    -37   -38   1
data     12    -50   -38   0
data     4     -64   -38   0
data     -4    -78   -38   0
               :
               :
               :
```

Processing のメソッド

2 行目に書かれている background() で背景の色を変更した．このメソッドは 0∼255 の間でグレースケールを設定できる．また，回転の中心は 3 行目の translate() で決定している．まずは，画面の真ん中を回転の中心とした．そして，矢印キーによって s と r の変数を変更することで，上下左右に回転の中心を動かした．

メインルーチン

```
1: void draw(){
2:     background(0);
3:     translate(width/2+s,height/2+r);
4:     fill(255);
5:     loop();
6:     noStroke();
7: }
```

原子の色は以下のように決めている．積層欠陥部分が 1 で，転位部分が 2 で，他の部分が 0 とした．fill() は塗りつぶしのメソッドであり，RGB で色を決定することができる．

原子の色

```
1:  if(atom==0){fill(255,255,255);}
2:  else if(atom==1){fill(0,0,255);}
3:  else if(atom==2){fill(255,0,0);}
```

3D 表示

図 3.1 では，6 原子で構成された fcc の POSCAR を読み込んで，3D 表示させた．マウスで自由な角度に動かした．図 3.1(a) を回転させると図 3.1(b) になる．直感的に原子とわかるように 2.2.2 小節で述べた sphere で表示させた．

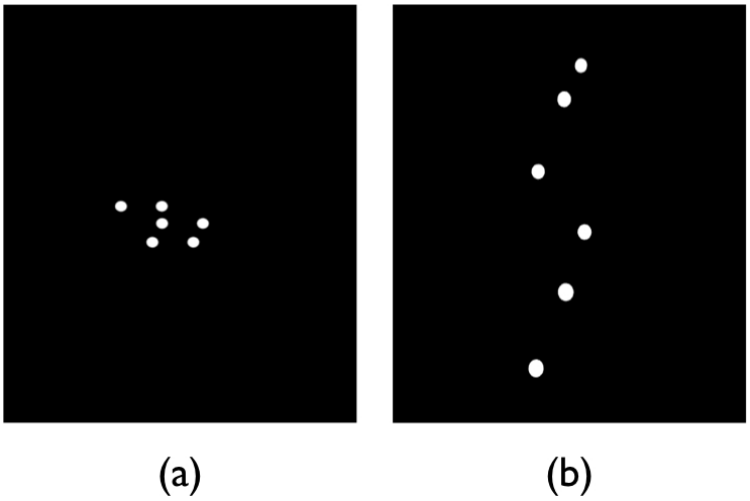

図 3.1: 初期 3D 表示.

積層欠陥表示

図 3.2 は実際の原子位置を仮想して作った POSCAR を読み込んで，積層欠陥を青色で表示した結果である．しかし，実行の処理時間に非常に時間がかかり，滑らかに動かない．

sphere() → box()

図 3.3 のように sphere を box に変えることで処理時間が早くなり，滑らかに動くようになった．

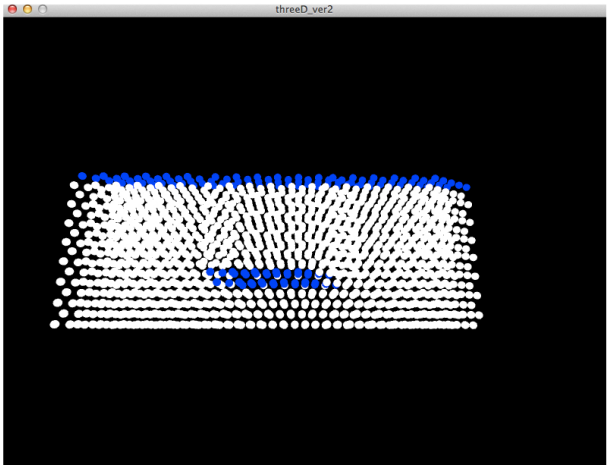

図 3.2: 積層欠陥を表示させた原子.

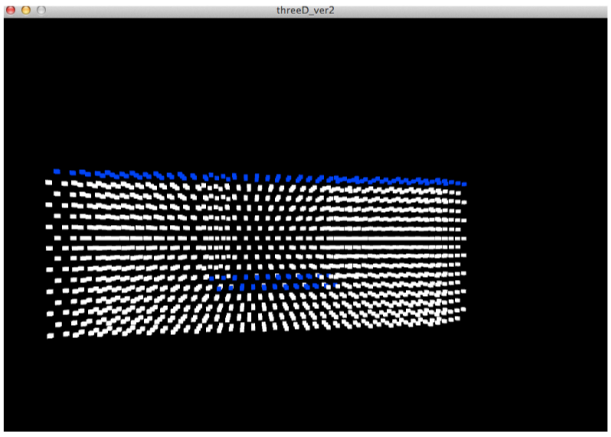

図 3.3: box 変換.

転位表示

図 3.4 転位部分に赤色を表示させようとした結果である．転位を探すメソッドを作り，積層欠陥とそうでない部分に 0,1 の番号をそれぞれ振り分けた．そして 0 と 1 の変わり目の部分を転位の部分にしようとした．しかし転位でない端の原子すべて赤色に変化してしまい失敗した．この方法では転位は表示出来ない．

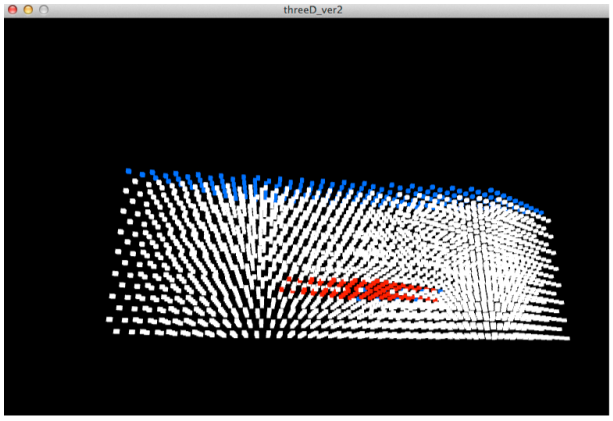

図 3.4: 転位表示.

そこで，図 3.5 では積層欠陥で配位数が違うものだけを赤色に変えた結果である．1 番上にも赤色が出てしまったが，周期的境界条件を考慮すると転位を赤色に表示することができた．

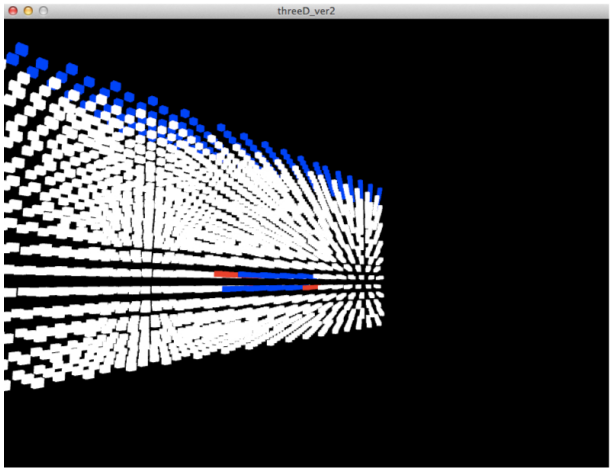

図 3.5: 周期的境界条件を考えた転位表示.

本研究では，顕微鏡で映しだした平面的な図と比較するために投影法を使用した．投影法を使用した結果が図 3.6 である．投影法は 3 次元的なものを平面的に表示させるため，顕

微鏡などの平面的な図と見比べやすい．

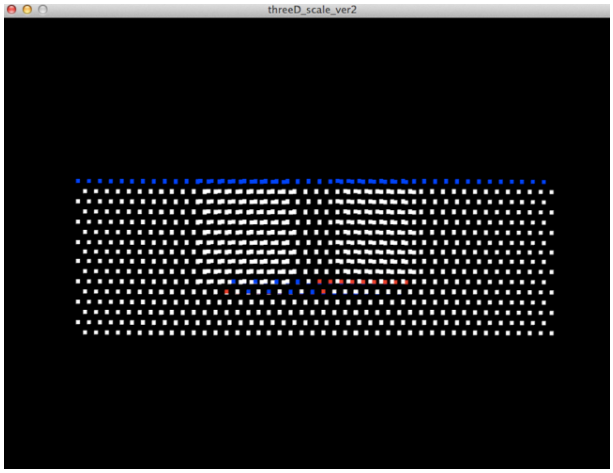

図 3.6: 投影法を使用した図.

また Mg の転位の視覚化以外に Al の小傾角粒界モデルの視覚化を行った．図 3.7 にその様子を示した．小傾角粒界には転位が存在している．この図においても，結晶の真ん中に転位が存在している様子が確認できる．そのため，原子間距離さえわかればどの原子も視覚化することが出来る．

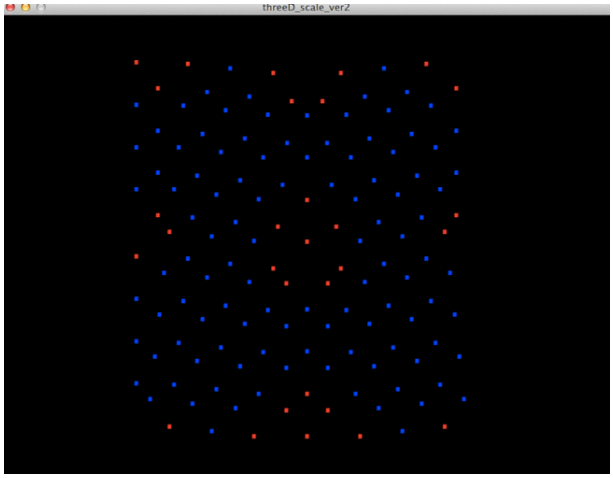

図 3.7: Al の転位表示.

エネルギー表示

転位周りの原子はエネルギー的に不安定である．そこで，転位周りの原子がどれほどエネルギー的に不安定であるか調べるために原子の色とエネルギーを対応させた視覚化を行った．

図 3.8 では，エネルギーの違いによって原子の色を変えた小傾角粒界モデルを示した．この図では，エネルギーのしきい値を設定して，その値を超えると赤，超えなければ青で表示している．しかしこの図では，最も安定な原子に比べて少しだけ不安定な原子を確認することが出来ない．

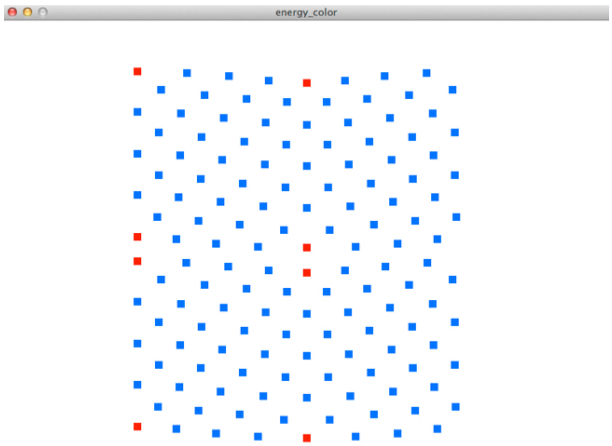

図 3.8: エネルギー表示. エネルギーが高ければ赤，低ければ青．

そこで，図 3.9 では，赤と青だけで表示するのではなく，グラデーション表示をした．この図では，エネルギーが高くなっていくに連れて，赤色に近づく．このようにグラデーション表示したことで，より詳細にモデルからエネルギーの強弱を確認することができるようになった．

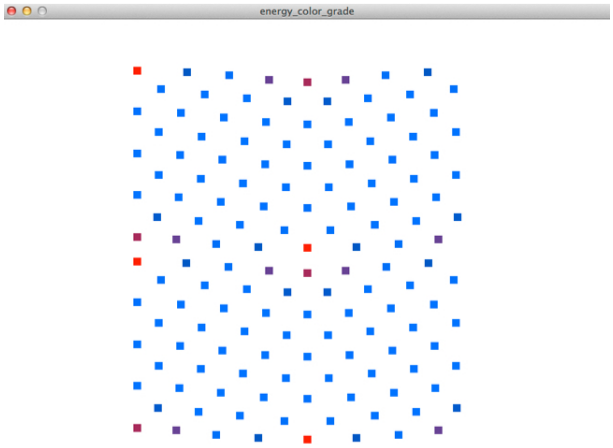

図 3.9: エネルギーをグラデーション表示した．高ければ赤で，低ければ青に近づいていく．

第4章　考察

4.1　LPSO構造

本研究室では，以前から母相hcp相に対して長周期積層欠陥 (LPSO) 構造の研究をしてきた．LPSO構造とは，積層欠陥が周期的に並ぶ構造である．しかしLPSO構造の生成機構は未だ解明されていない．現状ではLPSO構造の生成機構は，18Rから14Hに変わるとされている [5]．図4.1では東北大学木口らが観察したMg合金のHAADF-STEM像を示した [6]．HAADF-STEM像とは電子顕微鏡像の一種で重い原子ほど明るく映しだす特徴を持っている．この図において，映しだされた原子をなぞって書かれている赤丸はfcc構造，青丸はhcp構造を示している．この図において，右の白枠で囲った18Rの積層欠陥の間が一部8層になっていることがわかる．そして積層欠陥の端に黄色の丸で囲った部分転位が存在していることから積層欠陥の端には転位が存在することも示唆される．またこの図からは18Rから14Hに変わることが推測される．

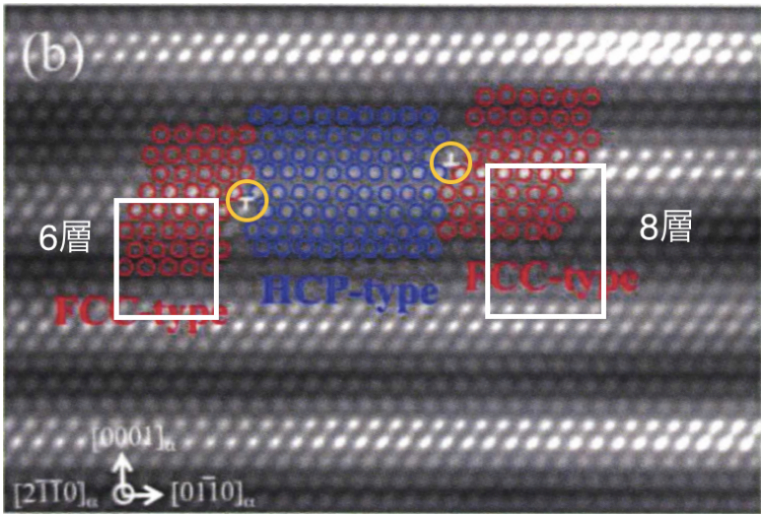

図 4.1: Mg-Zn-Y合金のHAADF-STEM像である．この図の白枠の右側において，積層欠陥の下から2番目と3番目の間が8層となっている．積層欠陥の端には黄色の丸で囲った部分転位が存在していると考えられる [6]．

しかしこの図からでは，転位がどの方向に存在しているかはわからない．そこで，本研究で開発した視覚化システムを用いて視覚化した．

図 4.1 を参考に POSCAR を作り，それを視覚化した．図 4.2 でわかるように，図 4.1 とは少し構造が異なっている．図 4.2(a) の緑枠に注目すると，緑枠のところは他の部分に比べると詰まっていたり大きく離れていたりする．そこで，図 4.2(b) で表した緑枠を矢印の方向に動かすと詰まっているところがなくなるが，積層欠陥部分の配列がずれる．配列がずれると，hcp もしくは fcc とは違う構造になってしまうことがわかった．

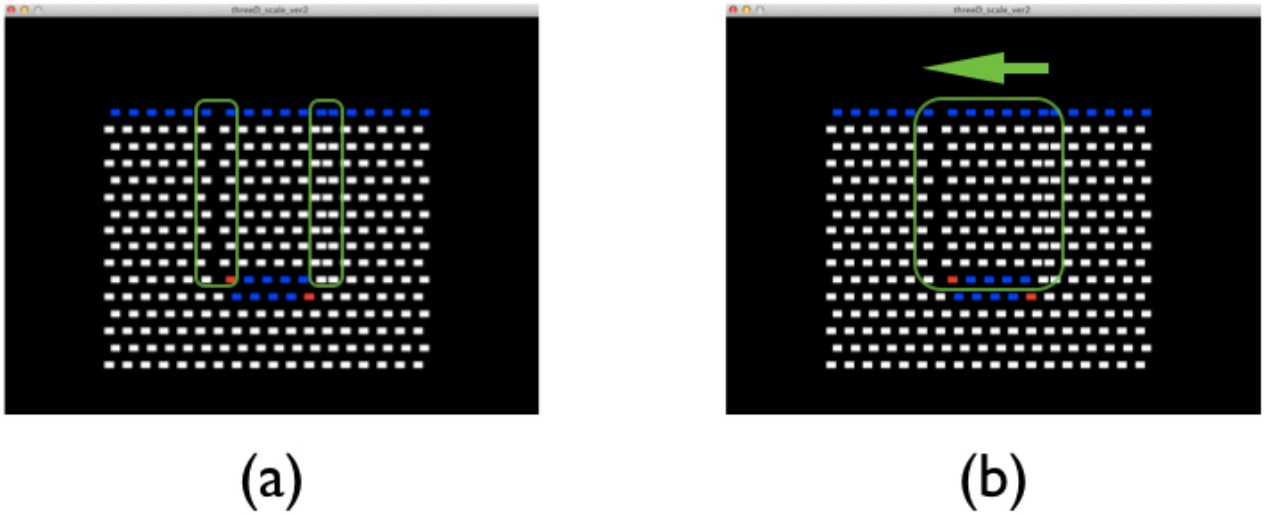

図 4.2: 木口らが観察した LPSO を視覚化した図.

4.2 Al の小傾角粒界

本研究室では，Al の小傾角粒界を対称とした研究を行っている．対称傾角粒界の模式図を図 4.3 に示した．対称傾角粒界の角度が 0~30° 程度であるものを小傾角粒界という．この図は立方晶で示したものであり，Al の小傾角粒界の構造が未だ解明されていない．そこで，本研究室で行っている第一原理計算とともに視覚化も行うことにした．

4.2.1 転位の視覚化

図 4.4 は緩和前のモデルである．緩和とは結晶モデル内の原子を 1 つずつバラバラに動かして最安定位置を求める手法である．赤色は転位部分を表している．転位周りを白の破

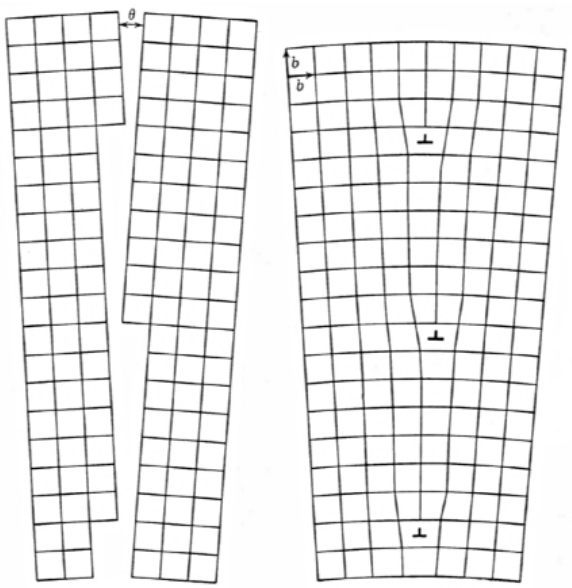

図 4.3: 対称傾角粒界の模式図.

線で囲うと，バーガース・サーキットができた．これは 1.2.6 小節で示したように転位の回りがバーガース・サーキットであることを確認することが出来た.

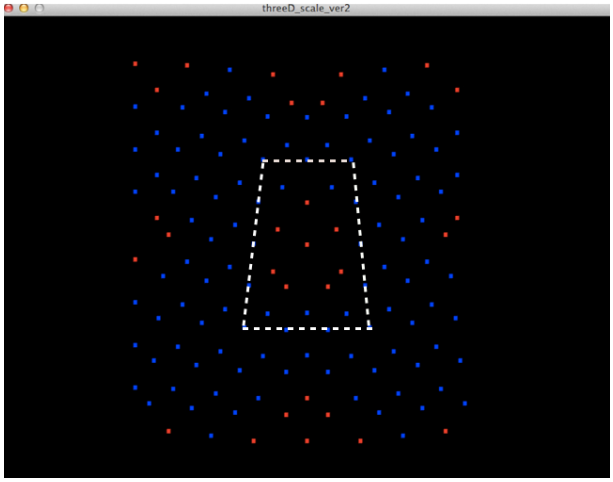

図 4.4: 緩和前のモデル.

4.2.2　エネルギーの視覚化

図 4.5 はエネルギーの高低をグラデーションした結果である．緑枠のところが赤色に近づいていることから，帯状にエネルギーが高い原子が連なっていると示唆される．そこで

同じ POSCAR を使用して，転位を表示した．その結果が図 4.5 である．図 4.5 の緑枠のところが転位部分であると言える．

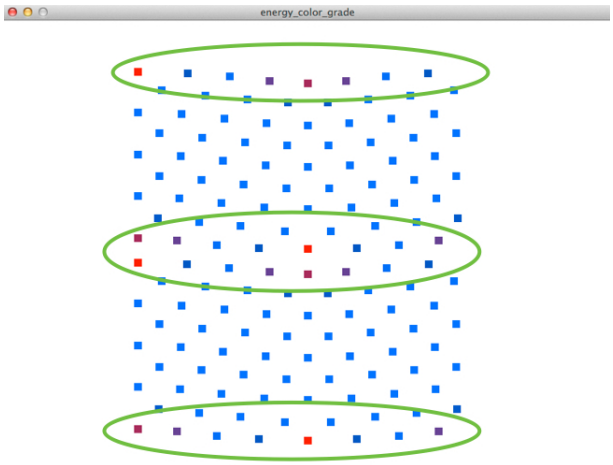

図 4.5: エネルギーをグラデーション表示した図．

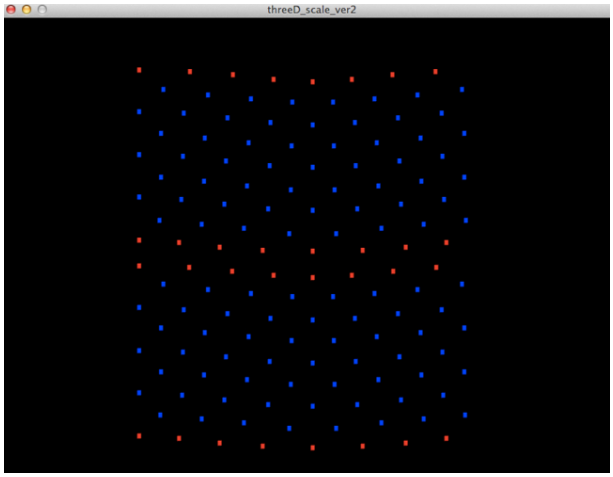

図 4.6: 図 4.5 を転位表示した結果．

第5章　総括

本研究では，西谷研究室で行われているMgのLPSO構造とAlの対称傾角粒界のエネルギー表示を視覚化することを目的とした．Processingで積層欠陥と転位が自動で表示され，自分で自由に動かし，確認することが出来る視覚化システムを作成した．

Processingでの視覚化システムの成果を以下に記す．

- LPSO構造の原子レベルでの様子を，3Dで視覚化した．Processingでの表示は様々な角度からみることができ，積層欠陥が入った時の原子の位置を確認することができる．積層欠陥や転位部分を色を変えて表示することによって，構造を確認しやすくなった．この視覚化システムによって，視覚的観点から補うことが可能となり，研究効率が高まった．

- 転位について，様々な視点（方向）からの視覚化を行った．視覚化することによって直感的に確認することが可能となった．現在では顕微鏡などで原子の様子を見ることは可能であるが，あまりはっきりとは映っていない．その点今回の研究によって，さまざまな視点（方向）からの確認が可能となり，理論面での学習を行う際にはとても有効なものとなった．

- 対称傾角粒界の転位などをエネルギーの違いによって，視覚化した．これにより，転位周りのエネルギーがそれ以外のエネルギーより高いことがわかった．こちらも3Dで表示しているため，奥の場所のエネルギーの違いも確認することが出来る．文面上やグラフでエネルギーを確認するよりも，視覚的観点で捉えやすくなった．

参考文献

[1] C.Kittel,「キッテル固体物理学入門」, 丸善株式会社,(2005).

[2] 鈴木秀次,「転位論入門」, 株式会社アグネ,(1967).

[3] D.Hull, D.J.Bacon 著,「Introduction to Dislocations 3rd Edition」, Pergamon Press, Oxford,(1984).

[4] H.Yokobayashi, et al., Acta Mater. 59, 7287(2011).

[5] A. Ono, E. Abe,T. Itoi, M. Hirohashi, M. Yamasaki and Y. Kwamura: Mater. Trans. 49, 990(2008).

[6] 木口賢紀.「シンクロ型 LPSO 構造の材料科学-次世代軽量構造材料への革新的展開-」(平成 24 年度報告)82-88.

謝辞

本研究を遂行するにあたり，終始多大なる有益なご指導およびご丁寧な助言を頂き，関西学院大学理工学部 情報科学科 西谷滋人教授 に深く感謝するとともに心より御礼を申し上げます．また，本研究の進行に伴い，西谷研究室に在籍しております皆様にも様々な知識の提供，ご協力を頂き，本研究を大成することができました．最後になりましたが，この場をお借りして心から深く御礼申し上げます．