計算化学のための等値面可視化コアプログラムの開発

○吉永崇、野口文雄

埼玉大学大学院 理工学研究科

【目的】

原子軌道やフェルミ面などは、式を見ただけでその形状を捉えることは困難である。また、数値シミュレーションを行うと、結果として膨大な数値データを得られるが、これらのデータから計算結果を把握することも困難である。そのため、これらの形状をコンピュータ上で可視化することは非常に有意義である。これらの結果が曲面として表現可能な場合、その表現には面上にある点の空間座標を複数求め、これらの点を頂点とする三角形を割り当てる必要があるが、算出された数万個の点をもとに適切な三角形のパッチを作成し、割り当てることは容易ではない。そこで本研究では曲面の可視化に必要となる三角パッチのデータを作成するためのコアプログラムを作成した。

【方法】

① パッチデータの作成

前述のとおり、無数の点の集合から曲面を生成するのは困難である。そこで、空間を小さく切り分け、その微小空間内の少数の点から三角形のパッチを作成して曲面を可視化するアルゴリズムを採用した。Fig.1 - a に示すように、まず空間を微小六面体(voxel)に切り分けた。そして、その中の voxel を一つずつ順々に取り出し(Fig.1 - b)、取り出された voxel の辺と曲面との交点を算出した(Fig.1 - c)。このとき、交点が 3 つ以上だった場合、これら数個の点をもとに voxel の内部で三角パッチを作成し、曲面に割り当てた。この操作を全ての voxel について行うことで曲面の可視化が可能になる。また、voxel 内部に複数の曲面を含まないことを仮定し、算出された交点を一度平面に投影した後に Delaunay 三角分割 [1] を用いて三角パッチを作成した。

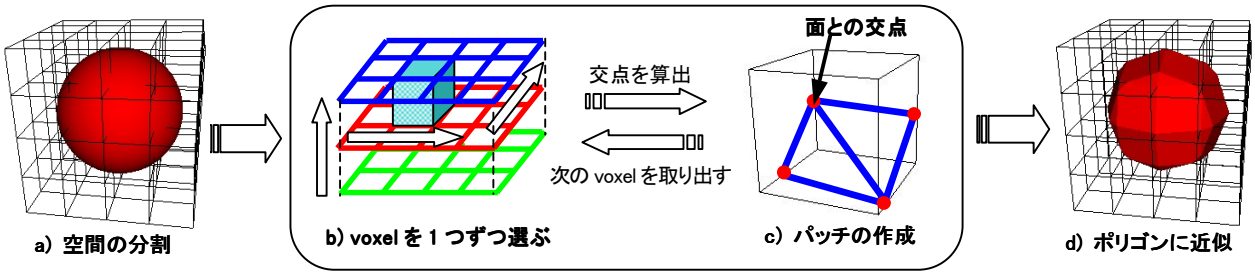

Fig. 1 空間の切り分けと三角形の割り当て

② 交点の算出

voxel の辺と曲面との交点の算出については今回、2 通りの方法を用いた。方法 I は、まず、Fig.2 に示すような格子状データを作成し、それらをもとにしてパッチを作成する方法である。この方法では端点の値が既知である各辺において、辺上の値が線形に変化すると仮定する。すると、辺の端点のうち,

一方を v_1、もう一方を v_2、可視化したい曲面での値(しきい値)を v_{surf} としたときに、$v_2 < v_{surf} < v_1$ または $v_1 < v_{surf} < v_2$ となる場合にはその辺に曲面が交差すると考えることができる。各辺において、このような条件に当てはまったときに線形補間を用いて曲面と voxel の辺との交点を算出した。

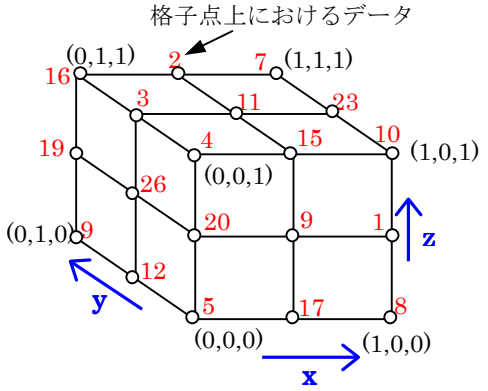

Fig. 2　格子状データの例

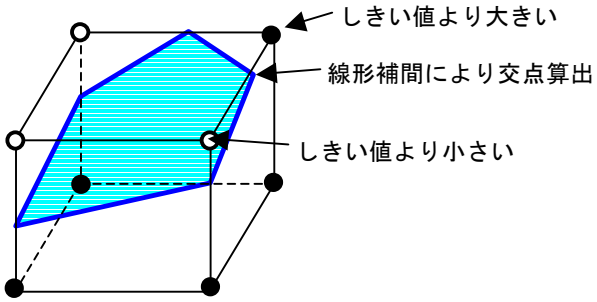

Fig. 3　方法Ⅰの例

また方法Ⅱは、$F = f(x,y,z)$ のように数式で表現された曲面について、voxel の各辺との交点を反復計算によって近似的に求める方法である。具体的には、voxel を構成する辺を 1 つずつ取り出し、その辺上で $f(x,y,z) = v_{surf}$ となる座標(x,y,z)を、正割法等の数値解析の手法を用いて算出した。この方法では、全ての辺について交点の算出のための計算を行うため、Fig.4 のように辺の両端がしきい値より小さい場合でも曲面と voxel との交差を検出可能である。

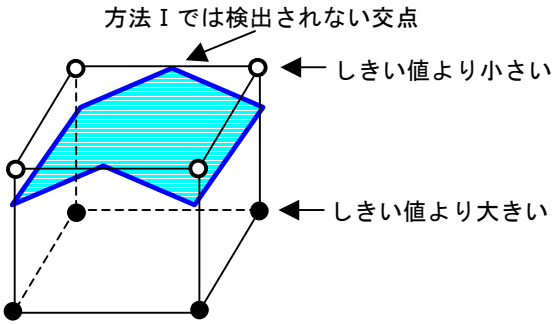

Fig. 4 方法Ⅱの例

③ 曲面の平滑化

本手法では、空間を切り分ける voxel サイズを制御することで、曲面の可視化に用いる三角パッチの数や、大きさを制御できる。つまり、voxel を小さくすれば、その内部で作成される三角パッチのサイズが小さくなり、多数のパッチで正確に曲面を表現することが可能である。逆に voxel を大きくすればそれだけパッチは大きくなり、曲面に割り当てる数は少なくなる。曲面の形状がさほど複雑でない場合には、ある程度大きなパッチを作成して、それらを曲面に割り当て、シェーディングを用いて滑らかにすれば十分である。しかし、形状の未知の物や複雑な形状の曲面を可視化する場合は、メッシュを細かくしてパッチを作成する必要がある。ところが、始めから小さい voxel を作成すると、それらの情報を保持するためのメモリ使用量が多くなる上、計算の無駄が多くなる。特に方法Ⅱの場合は、曲面と交差する、しないに関わらず、全ての辺に対して交点算出の反復計算を行うため、三角パッチのデータの作成に多くの時間を要する。そこで、この計算の無駄を極力減らすために、Fig.5 のように曲面付近のメッシュを主に分割するプログラムを作成した。具体的には次に示すとおりである。

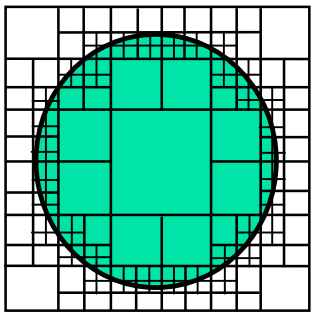

Fig.5 メッシュ分割のイメージ

まず最初は小さすぎない程度の voxel メッシュを作成した。そして、メッシュごとに曲面との交点の有無を調べ、交点を有する voxel についてのみ各辺を二等分し、新たに 8 個の voxel を作成した。この操作を、ユーザーがあらかじめ指定した voxel のサイズになるまで再帰的に繰り返し行い、分割終了後に voxel 内で三角パッチを作成するプログラムを実装した(Fig.6)。

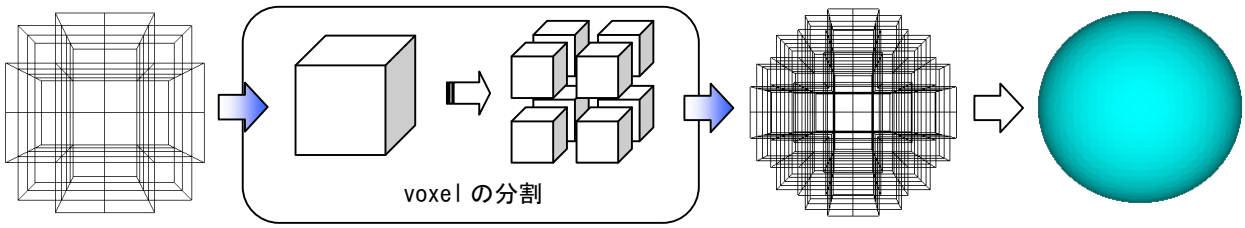

Fig. 6 曲面可視化の過程

【結果】

① 数値シミュレーションの結果の可視化

数値シミュレーションの結果として得られた格子状データをもとに、方法 I で計算結果を可視化した例を示す。

1）デンドライト成長シミュレーション

純金属の過冷却を伴う凝固過程などで見られるデンドライト成長を、簡単に再現できるとされているフェーズフィールド法を用いてシミュレーションし、その形態のパターンを可視化した。モデル式は以下の式で、式(1)は界面運動方程式、式(2)は熱拡散方程式であり、これらを連立させることで各ステップにおけるデンドライトの形状をシミュレーションできる。Fig.7 に示すように、立体可視化することにより枝分かれしたデンドライトの形状が視覚的に捉えられるようになった。

$$\tau \frac{\partial p}{\partial t} = \varepsilon^2 \nabla^2 p + p(1-p)\left(p - 0.5 + m(T, -\nabla p)\right) \quad \cdots(1)$$

$$\frac{\partial T}{\partial t} = \nabla^2 T + K \frac{\partial p}{\partial t} \quad \cdots(2)$$

p:相を表す関数	t:時間
τ:正の定数	T:温度
ε:相境界層の厚さ	K;潜熱
m(T,－∇p):熱駆動力関数	

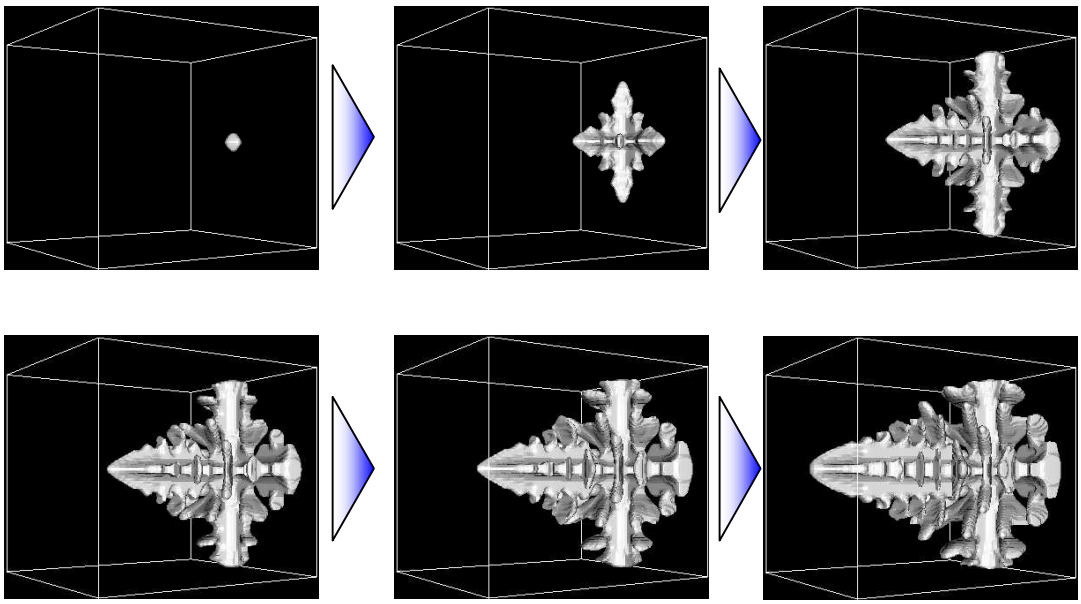

Fig. 7 デンドライト成長シミュレーションの結果

2）スピノーダル分解

高温で固溶していた多成分系の化合物(合金等)が低温でミクロサイズの相に相分離する過程をスピノーダル分解といい、その分解過程は、モデル式(3)で表される[2)]。

$$\frac{\partial c}{\partial t} = M(c)\left(\frac{\partial^2 \chi}{\partial x^2} + \frac{\partial^2 \chi}{\partial y^2} + \frac{\partial^2 \chi}{\partial z^2}\right) + \left(\frac{\partial M}{\partial c}\right)\left(\frac{\partial c}{\partial x}\frac{\partial \chi}{\partial x} + \frac{\partial c}{\partial y}\frac{\partial \chi}{\partial y} + \frac{\partial c}{\partial z}\frac{\partial \chi}{\partial z}\right) \cdots(3)$$

c:濃度
M:易動度
χ:拡散ポテンシャル

Fe－Cr 合金のスピノーダル分解シミュレーション結果の断面図表示と立体表示をそれぞれ Fig.8 に示した。この図より、立体表示することによって三次元にランダムに形成された組織が容易に把握できることがわかった。

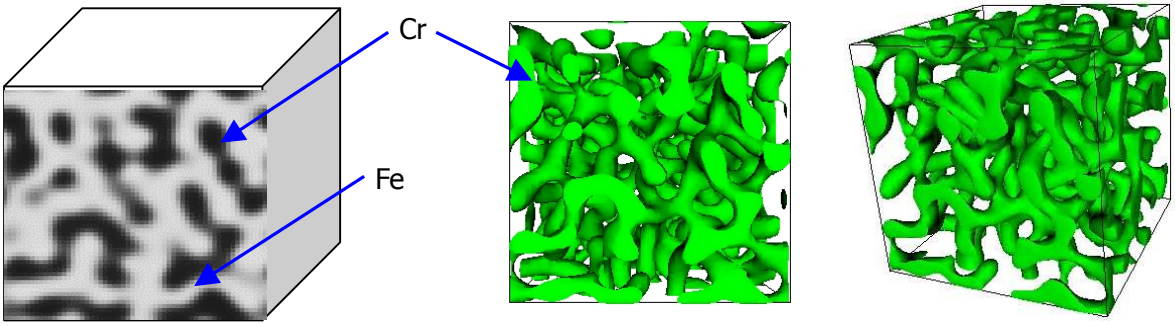

Fig.8 Fe－40at% Cr 合金のスピノーダル分解シミュレーションの結果

② 数式表現された曲面の可視化

方法Ⅱを用いて曲面と voxel との交点を算出し、関数表現された4種類の曲面に対する等値面可視化を行った。式(4)は、三つ穴のトーラス、(5)は水素原子の 4f　$5z^3-3zr^2$ 軌道、(6)は sp 混成軌道、(7)は面心立方格子(fcc)のフェルミ面を表現する数式である。これらの可視化結果はそれぞれ Fig.9～Fig.12 に示す。

$$F(x,y,z) = 27^4 z^2 - \left(1 - \left(\frac{x}{31}\right)^2 - \left(\frac{y}{22}\right)^2\right)\left((x-23.4)^2 + y^2 - 36\right)\left(x^2 + y^2 - 36\right)\left((x+23.4)^2 + y^2 - 36\right) \cdots(4)$$

$$F(x,y,z) = \left(\frac{1}{3072\sqrt{5\pi}}\right)\exp\left(-\frac{r}{4}\right)z\left(5z^2 - 3r^2\right) \cdots(5)$$

$$F(x,y,z) = \frac{1}{\sqrt{2}}\left(\frac{1}{4\sqrt{2\pi}}\right)\left(3.25^{3/2}\right)\left(2 - 3.25r\right)\exp\left(-\frac{3.25r}{2}\right) + \frac{1}{\sqrt{2}}\left(\frac{1}{4\sqrt{2\pi}}\right)\left(3.25^{5/2}\right)\exp\left(-\frac{3.25r}{2}\right)x \cdots(6)$$

$$F(x,y,z) = -\alpha - 4\gamma\left\{\left(\cos\frac{1}{2}ya\right)\left(\cos\frac{1}{2}za\right) + \left(\cos\frac{1}{2}za\right)\left(\cos\frac{1}{2}xa\right) + \left(\cos\frac{1}{2}xa\right)\left(\cos\frac{1}{2}ya\right)\right\} \cdots(7)$$

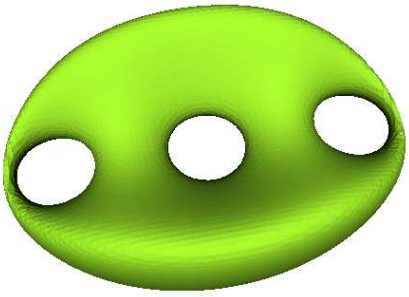

Fig. 9　三つ穴のトーラス

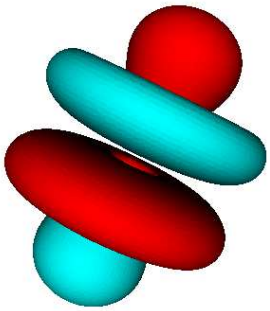

Fig. 10　水素原子の原子軌道
4f　$5z^3-3zr^2$

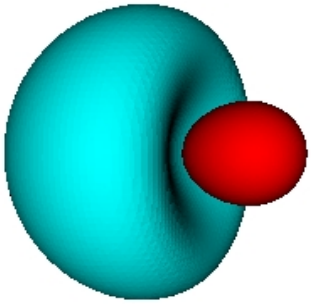

Fig. 11 sp 混成軌道

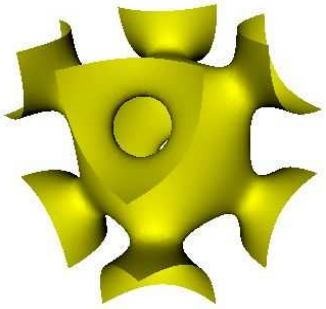

Fig. 12 fcc のフェルミ面

③外見の調整

本プログラム内の VRML を出力する際の設定を変更することで、可視化する曲面の外見の調整が可能である。したがって、PC 上で VRML の出力を見ながら、可視化したオブジェクトを効果的に表現できるように、ある程度見た目を調整した上で CAVE 装置にて可視化が行えることが期待できる。

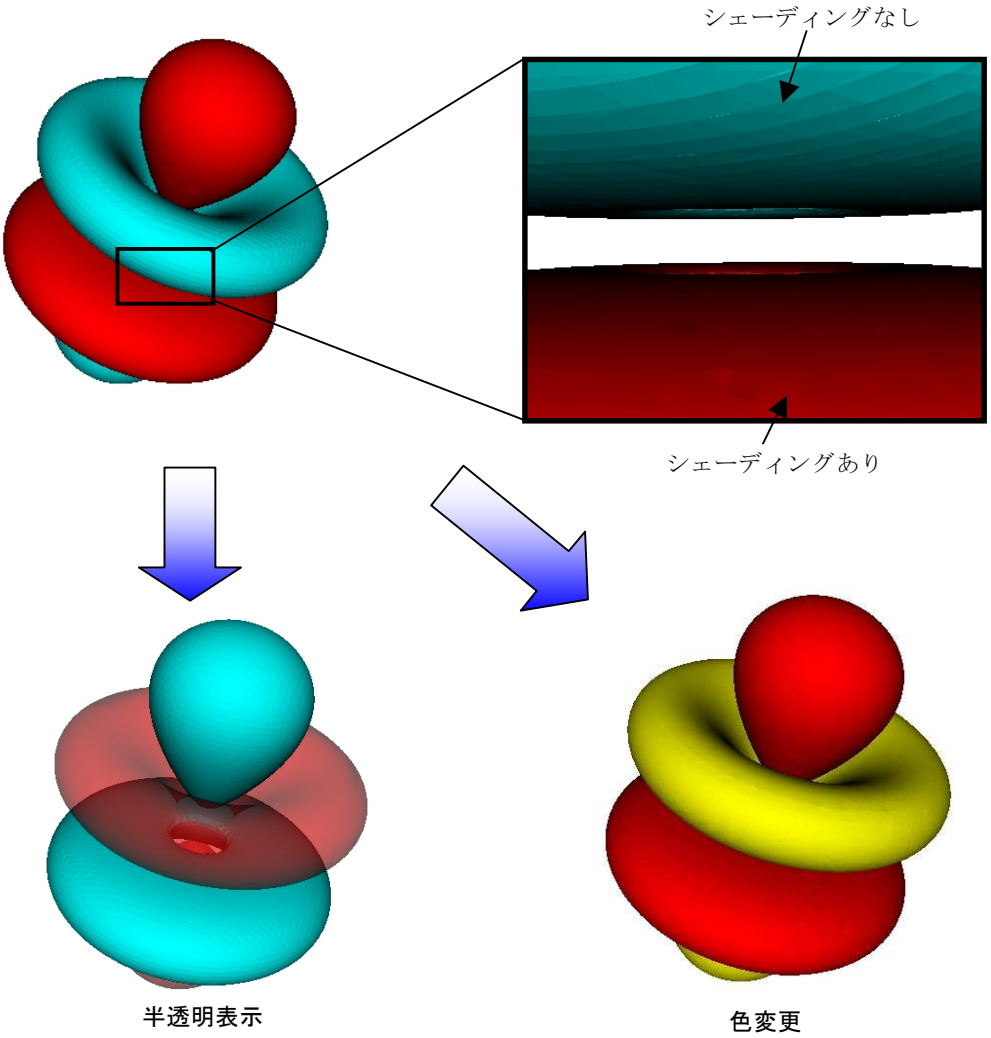

Fig. 13　可視化オブジェクトの外見の調整例

【結論】

膨大な量の点から三角パッチを作成しなければならない問題を、空間を微小空間に分割することで、各微小空間の内部にある少数の点からパッチを作成するだけの簡単な問題に置き換ることができた。そのため、容易に三角パッチを作成し、曲面を可視化できた。また、可視化に最も必要となる三角パッチデータを作成できたことにより、OpenGL や DirectX、VRML、Java3D など様々なグラフィックスライブラリを用いて可視化できる。そのため、CAVE による可視化はもちろん、今後当研究室で開発される科学技術計算プログラムの結果の可視化への適用に期待が持てる。

【今後の展望】

① パッチ作成における問題点

現在、本プログラムでは1つの微小空間には1つの曲面しか入っていないことを仮定しているため、Fig.14 に示すように二つの曲面が同じ微小空間内にあると、それぞれの点がどの曲面に属しているかを判別できない。そのため、Fig.14 の case1 と case2 のどちらの等値面が正しいかを判別できず、正確に曲面を表現できないケースがある。したがって、今後このようなケースでのパッチ作成の際に生じる曖昧さを解消するためのルーチンを作成し、実装する必要があると考えている。

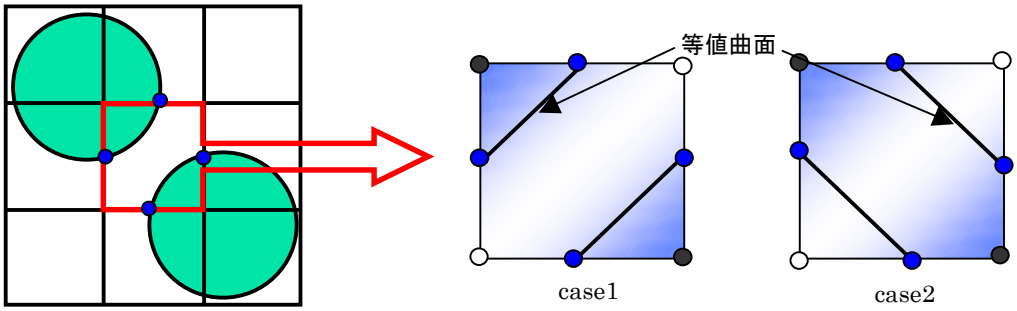

Fig. 14 等値面のあいまいな判別の例

② 処理時間について

今回、曲面とvoxelとの交点の算出に2つの方法を用いて曲面を可視化した結果、方法Ⅱのように反復計算を用いて交点の算出を行い、曲面を可視化した場合と、線形近似を用いた方法Ⅰで交点の算出を行い、曲面を可視化した場合とでは、線形補間を用いた方法Ⅰの方がはるかに処理が速いことがわかった。方法Ⅱには、方法Ⅰでは検出できない曲面との交点を検出できるという利点もあるが、メッシュが十分に小さければ曲面の近似精度も良くなるため、方法Ⅰと方法Ⅱとの間にはさほど大きな差は出ないと考えられる。従って、今後は方法Ⅰをベースに開発を行い、空間の切り分け方法などを工夫して、より効率的に曲面の可視化を行えるプログラムを開発する予定である。また、方法Ⅰに類似した手法で、MarchingCubes 法 [3]という手法がすでに開発されており、この方法ではvoxel内でのパッチの張り方について、数種類のパターンに分類し、それに基づいてパッチを作成している事を先日知った。したがって、このことについても文献調査を進め、今後のプログラム開発に活用したいと考えている。

【参考文献】

1) Jonathan Richard Shewchuk.　Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator

2) 小山敏幸　Phase-field 法に基づく組織形成過程の計算機シミュレーション（2004）

3) C.Montani, R.Scateni, R.Scopigno.　Discretized Marching Cubes　(1994)