

離散的ソレノイダル場のスプライン補間による力線の可視化

清水 徹† 鵜飼 正行†

非圧縮流速場や磁場のようなソレノイダル場を可視化するとき、力線（や流線）がよく用いられ、その位相学的性質が重要な意味を持つ場合がたびたびある。しかし、三次元の数値シミュレーションで得られる離散ベクトルデータから、ソレノイダル条件 ($\text{div } \mathbf{V}=0$) を厳密に満たすように力線を作成する有効な方法は知られていない。この問題に対し、本論文では次のような手法を提案する。(i)まず離散ベクトルデータから数値積分により離散的な流束関数を定義する。(ii)それをスプライン補間し、微分することで連続的ベクトル場を生成する。(iii)このベクトル場を追跡して力線を作成する。このようにして得られた力線は、ソレノイダル条件を厳密に満たしている。最後に、ここで提案された手法と一般的な力線作成法を比較して、この手法の有効性が具体的に示される。

Visualization of Lines of Force by Spline Interpolation of a Discrete Solenoidal Field

TOHRU SHIMIZU† and MASAYUKI UGAI†

Lines of force are often used in order to visualize solenoidal fields, such as incompressible flow and magnetic field, and the topological features are frequently of great importance in many situations. At present, there is no effective method to generate such lines of force, which exactly satisfy the solenoidal condition ($\text{div } \mathbf{V}=0$), on the basis of a discrete vector data obtained from three dimensional numerical simulations. The present paper proposes the following method for this problem. (i) First, a set of discrete flux functions is defined by numerical integrations of the discrete solenoidal vector data. (ii) Second, these functions are interpolated by B-spline, and then differentiations of these lead to a continuous vector field. This vector field exactly satisfies the solenoidal condition at any spatial point. (iii) Finally, lines of force are visualized by tracing the continuous solenoidal vector field. The effectiveness of the present method is demonstrated in comparison with the conventional method.

1. はじめに

近年の計算機のめざましい発展は、複雑で大規模な理工学シミュレーションを可能にしている^{1),2)}。これに伴い、大量の出力データから注目する物理的情報をどのように効果的に抽出し、直感的に可視化するかという問題が重要となっている。

一般的に流体力学や磁気流体力学 (MHD) の数値シミュレーションでは、圧力や密度のようなスカラーデータと磁場や流速場のようなベクトルデータの2種類の離散データが扱われる。これらのデータを可視化するとき、前者は等高線や等高面で表現され、後者はしばしば力線（や流線）として描かれる。ここで後者は普通、ベクトルデータをもとに力線を追跡して得ら

れる³⁾。この追跡において、我々は力線上でベクトル場を連続的に知る必要があるから、離散ベクトルデータを何らかの方法で補間しなければならない。

特に、我々が非圧縮流速場や磁場のようなソレノイダル場における力線の位相学的性質に注目する場合、可視化には注意が必要である。例えばMHDの場合、磁力線の位相学的構造はプラズマの輸送メカニズムに密接に関係しており、その位相が変わるとその意味する物理が大きく変わってしまう^{4),5)}。このような場合、離散ベクトルデータを補間して得られる連続的ベクトル場（と力線）は、ソレノイダル条件を満たさなければならない。ソレノイダル場の数値シミュレーションで離散データを生成し、それをもとに力線を得るためには、次の二つの点に注意すべきであると考えられる。

まず、もともとなる離散ベクトルデータは（離散的な意味で）ソレノイダル条件を満たしていなければならない。本論文で考える離散データは、数値シミュレ

† 愛媛大学工学部情報工学科
Department of Computer Science, Faculty of
Engineering, Ehime University

ションにおいて差分計算の結果として得られるものである。このため、その条件は後で見ると各差分格子内に入出入りする流束の総和が0になることで定義される。

次に、そのようなデータを補間して（連続的な意味で）ソレノイダル条件を満たすベクトル場を生成しなければならない。これは、上述の離散データから流束関数を定義し、それをスプライン補間して得られる。

本論文はつぎのように構成されている。まず第2章でソレノイダルベクトルデータを生成する有限差分スキームとそのデータに関する（離散的な）ソレノイダルの意味が議論される。第3章ではここで用いられるBスプライン関数と補間の手続きが示される。第4章は具体的応用例として力線追跡の結果を示し、そして第5章はまとめとなっている。

2. ソレノイダル場

2.1 離散的ソレノイダル場の生成

本論文で扱うベクトルデータは、数値シミュレーションにおいて有限差分方程式から生成されるものであり、すなわち離散的なものである。ゆえにソレノイダル（発散なし）の性質も離散的な意味で考える必要がある。

今、次のような偏微分方程式について考える。

$$\mathbf{U} = \text{rot } \mathbf{F}. \quad (1)$$

これは \mathbf{U} がソレノイダル場であることを意味し、つまり、

$$\text{div } \mathbf{U} = 0 \quad (2)$$

が成立する。非圧縮流の場合、 \mathbf{U} は流速で \mathbf{F} が流れ関数に対応する。また \mathbf{U} を \mathbf{V} の時間微分で置き換えると

$$\frac{\partial \mathbf{V}}{\partial t} = \text{rot } \mathbf{F} \quad (3)$$

となり、これは $-\mathbf{F}$ を電場、 \mathbf{V} を磁場とすればファラデーの法則になる。もし \mathbf{V} が初期状態($t=0$)でソレノイダルならば、式(3)はその後の \mathbf{V} も常にソレノイダルであることを意味する。実際、このような式はMHD方程式の中に現れ、ソレノイダルということが磁場の基本的性質となっている。

ところで式(3)を数値的に解くには、有限差分式への何らかの変換が必要である。空間についての差分演算子を D_x, D_y, D_z で表すと、式(3)に対する有限差分式は一般的に

$$\begin{aligned} V_x^{n+1} &= V_x^n + \delta t (D_y F_z - D_z F_y), \\ V_y^{n+1} &= V_y^n + \delta t (D_x F_z - D_z F_x), \end{aligned}$$

$$V_z^{n+1} = V_z^n + \delta t (D_x F_y - D_y F_x) \quad (4)$$

と書ける。例えば、空間の打ち切り誤差が二次精度の場合、格子点 (i, j, k) のまわりに $D_x F_x \equiv ((F_x)_{i+1, j, k} - (F_x)_{i-1, j, k}) / (2\delta x)$ である。ここで、添え字 i, j, k と n は x, y, z 方向の格子点番号と時間ステップ数、 δx と δt はそれぞれ空間と時間の差分きざみである。また、式(4)の $\mathbf{F} (= (F_x, F_y, F_z))$ は $\mathbf{V}^n (= (V_x^n, V_y^n, V_z^n))$ から計算される時、時間について前進差分になるが、 $\mathbf{V}^{n+1/2}$ や \mathbf{V}^{n+1} から計算されるなら2-step Lax-Wendroff法などの多段スキーム⁶⁾も含むことになり、多くの差分法が式(4)の形になる。

2.2 離散的ソレノイダル条件

式(4)の差分演算子によってソレノイダル条件式(2)の離散形を次のように定義する。

$$D_x V_x + D_y V_y + D_z V_z = 0. \quad (5)$$

ただし、 \mathbf{U} は \mathbf{V} で置き換えた。ここで、もし初期状態 \mathbf{V}^0 で式(5)が成り立ち、かつ差分演算子について交換則(例えば $D_x D_y = D_y D_x$)が成り立てば、式(5)は時間発展後の \mathbf{V}^n についても成り立つ。それは \mathbf{V} を $n+1$ ステップの値とした式(5)へ式(4)を代入すれば明らかである。以下ではこの条件を満たす差分演算子として、一般的に

$$D_x (V_x)_{i, j, k} = \sum_{l=1}^L d_l ((V_x)_{i+l, j, k} - (V_x)_{i-l, j, k}) \quad (6)$$

となるものを考える。 D_y, D_z も同様である。これは中心差分型であり、 \mathbf{V} の線形結合になっている。例えば、 $L=1$ で係数 $d_1=1/(2\delta x)$ のときは空間二次精度になり、 $L=2$ で $d_1=2/(3\delta x)$ 、 $d_2=-1/(12\delta x)$ なら四次精度のスキームとなる。つまり、一般に用いられる中心差分法のほとんどは式(6)の形になる。

ところで、式(5)は離散式なので、式(2)のような微分方程式とはソレノイダルの意味が当然異なる。そもそもソレノイダル条件というのは、そのベクトル場で湧き出しや吸い込みがないということであって、そのベクトル \mathbf{V} で定義される流束の任意領域への出入り量は常に差引き0でなければならない。しかし、離散データでは差分格子を単位とした立体セルより小さい領域への流束の出入りを考える意味がない。そこで差分格子に基づいた立体セルへの流束を考える。このために、図1(a)へ $z = \text{const.}$ の xy 二次元面を示し、縦横の実線で差分格子を表す。つまり、その各格子点に離散ベクトルデータ (V_x, V_y, V_z) がのっているとする。

簡単のため、式(6)で $L=1$ の場合、すなわち典型

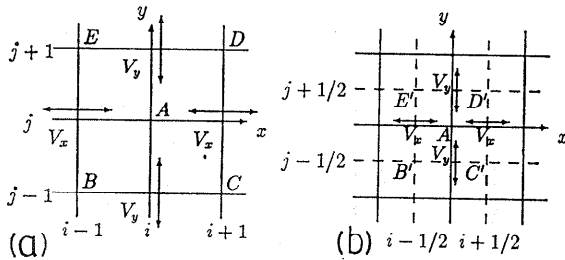


図1 差分格子上の離散ベクトルデータに関するソレノイダル条件:

- (a) 領域 BCDE での流束の出入り
- (b) 領域 B'C'D'E' (単位セル) での流束の出入り

Fig. 1 Solenoidal condition for discrete vector data on grid points, where the flux flows across (a) the region BCDE and (b) the unit cell region B'C'D'E'.

的な二次精度差分スキームの場合を考える。このとき、点Aを (i, j, k) とし、そこでの式(5)に $8 \delta x \delta y \delta z$ を掛けると

$$4\delta y \delta z ((V_x)_{i+1,j,k} - (V_x)_{i-1,j,k}) + 4\delta x \delta z ((V_y)_{i,j+1,k} - (V_y)_{i,j-1,k}) + 4\delta x \delta y ((V_z)_{i,j,k+1} - (V_z)_{i,j,k-1}) = 0 \quad (7)$$

と書ける。例えば、この $4\delta y \delta z (V_x)_{i+1,j,k}$, $4\delta x \delta z (V_y)_{i,j+1,k}$ は、図1(a)においてそれぞれ面CDとDEを通り抜ける流束と見なせる。このように考えれば、式(7)は領域 BCDE へ出入りする流束の和が0であることを示している。ただし、この領域 BCDE は、三次元的にみれば差分格子を単位として体積が8の立体セルからなっている。

次に、このようなベクトルデータから、体積が1の立体セルごとにソレノイダル条件を満たすベクトルデータを構成する。これは差分格子に対し、流束を考える時の最小単位の領域である。この結果得られる離散ベクトルデータを次節の補間処理に用いる。図1(b)にこの立体セルを破線で示す。例えば、面C'D'やD'E'を通り抜ける流束は、その面上にデータ点がないので、隣接データ点の値の平均をとり、

$$(\bar{V}_x)_{i+1/2,j,k} = 0.5((V_x)_{i+1,j,k} + (V_x)_{i,j,k}),$$

$$(\bar{V}_y)_{i,j+1/2,k} = 0.5((V_y)_{i,j+1,k} + (V_y)_{i,j,k}) \quad (8)$$

等とおく。こうすると式(7)から

$$\delta y \delta z ((\bar{V}_x)_{i+1/2,j,k} - (\bar{V}_x)_{i-1/2,j,k}) + \delta x \delta z ((\bar{V}_y)_{i,j+1/2,k} - (\bar{V}_y)_{i,j-1/2,k}) + \delta x \delta y ((\bar{V}_z)_{i,j,k+1/2} - (\bar{V}_z)_{i,j,k-1/2}) = 0 \quad (9)$$

が成り立ち、これは体積が1の立体セルB'C'D'E'への流束の出入りが0であることを意味する。

以上の議論は、もっと一般的な差分式(6)の場合へも拡張できる。そのためには、式(8)のかわりに

$$(\bar{V}_x)_{i+1/2,j,k} = \frac{\sum_{l'=1}^L d_{l'} \sum_{i'=1}^{l'} (V_x)_{i+l',j,k} + (V_x)_{i+l'-1,j,k}}{\left(\sum_{l'=1}^L 2l' d_{l'} \right)} \quad (10)$$

とすればよい。 \bar{V}_y , \bar{V}_z についても同様である。これで、式(6)の差分演算子について式(5)を満たすベクトルデータから、各単位立体セルごとにソレノイダル条件式(9)を満たす流束が定義できる。

3. 補間方法

3.1 スプライン補間

離散データから滑らかな関数を求めたいとき、スプライン関数が広く用いられている。それは区分多項式であり、ある離散データに対し、適当な組のBスプライン関数を基底関数として生成されるものである。スプライン関数は、節点とデータ点のとり方や関数の次数など、多くの設定条件がある。本補間法において、あまり本質的な意味を持たない条件(節点のとり方等)もあるが、ここでは後述するベクトル場の補間の説明を明瞭かつ簡潔にするため、具体的にx軸上の点 x_i におけるスカラーデータ $\phi_i (i=1 \dots I)$ を補間して、 $\Phi(x)$ という連続関数を生成することを考え、その表式をはっきりさせておく。

x_i はデータ点と呼ばれる。ここで、二つの端点 x_1 と x_I では、 ϕ_1 , ϕ_I 以外その微分値のような境界条件は与えない。さらにBスプラインをつくるのに必要な節点 ξ_i の数を I' とする。簡単のため、これらを図2のように等間隔におく。図中の m はスプラインの次数 $(m-1)$ を決める変数である。もしこれが大きいとラグランジュ補間に近くなり、データ点間距離以下の波長のノイズを含むようになる。したがって、それは必要な連続性を考慮した上で可能な限り小さくすべき

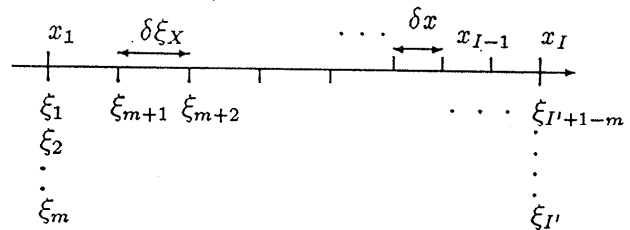


図2 スプライン補間におけるデータ点と節点のとり方
Fig. 2 Positioning of data points and knots for spline interpolation.

である。ここでは後で述べるように、スプライン関数の一階微分が連続になることが必要なので $m=3$ とする (つまり二次)。また、 m 個の節点をそれぞれ両端点で重ねる。こうすると節点間隔は $\delta\xi_x=(x_1-x_1)/(I'+1-2m)$ となり、Bスプライン関数の数は $(I'-m)$ となる。

結局、これらを用いて得られるスプライン関数は、

$$\Phi(x) = \sum_{i=1}^{I'-m} a_i B_{i,m}(x) \tag{11}$$

と書ける。なお、 $B_{i,m}(x)$ は $m-1$ 次のBスプラインである。ここで、 $(I'-m)$ が I に等しくなるように節点をとれば、 I 個の未知係数 a_i は、 I 個のデータ (x_i, Φ_i) を式(11)へ代入して得られる線形連立方程式から唯一に求まる。次節の具体的計算では、 $B_{i,m}(x)$ は deBoor-Cox の算法で求め、連立方程式は LU 分解法で解く。

3.2 流束関数の補間

ここでは、第2章で求めたソレノイダル条件 (式(9)) を満たす離散ベクトルデータを補間する。まず離散的流束関数 Φ_x, Φ_y, Φ_z を数値面積分の形で以下のように定義する。

$$\begin{aligned} (\Phi_x)_{i,j,k} &= \sum_{j'=1}^j \sum_{k'=1}^k (\bar{V}_x)_{i+1/2,j',k'} \delta y \delta z, \\ (\Phi_y)_{i,j,k} &= \sum_{i'=1}^i \sum_{k'=1}^k (\bar{V}_y)_{i',j+1/2,k'} \delta x \delta z, \\ (\Phi_z)_{i,j,k} &= \sum_{i'=1}^i \sum_{j'=1}^j (\bar{V}_z)_{i',j',k+1/2} \delta x \delta y. \end{aligned} \tag{12}$$

こうすると、式(9)から $\Phi(\equiv \Phi_x + \Phi_y + \Phi_z)$ は、

$$\begin{aligned} &\Phi_{i+1,j+1,k+1} - \Phi_{i+1,j+1,k} - \Phi_{i,j+1,k+1} - \Phi_{i+1,j,k+1} \\ &+ \Phi_{i+1,j,k} + \Phi_{i,j+1,k} + \Phi_{i,j,k+1} - \Phi_{i,j,k} = 0 \end{aligned} \tag{13}$$

を満たしている。

式(11)に y, z 方向のBスプライン関数 $C_{j,m}(y), D_{k,m}(z)$ を追加し、式(12)の離散データ Φ_x, Φ_y, Φ_z を補間すると

$$\begin{aligned} \Phi_x(x, y, z) &= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K a_{i,j,k} B_{i,m}(x) C_{j,m}(y) D_{k,m}(z), \\ \Phi_y(x, y, z) &= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K b_{i,j,k} B_{i,m}(x) C_{j,m}(y) D_{k,m}(z), \\ \Phi_z(x, y, z) &= \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K c_{i,j,k} B_{i,m}(x) C_{j,m}(y) D_{k,m}(z) \end{aligned} \tag{14}$$

となる。もしデータサイズ (I, J, K) が大きいと、未知係数 $a_{ijk}, b_{ijk}, c_{ijk}$ を直接解くには莫大な計算が必要になる。そこで実際には、この計算は3ステップに分けて行う。例えば式(14)の Φ_y については、まず式(12)の $(\Phi_y)_{ijk}$ から $\hat{a}_i \left(= \sum_{j=1}^J \sum_{k=1}^K a_{i,j,k} C_{j,m}(y) D_{k,m}(z) \right)$ を求め、それより $\hat{a}_{i,k} \left(= \sum_{j=1}^J a_{i,j,k} C_{j,m}(y) \right)$ を得て、最後に a_{ijk} が求められる。 b_{ijk}, c_{ijk} についても同様である。こうすると、解くべき連立方程式の一つ一つが小さくなり、大規模な多次元データへの適用も可能となる。

結局、任意の点でのベクトル量 \mathbf{V} を求めるには、この連続な流束関数を空間微分すればよい。これは、Bスプラインの微分公式⁷⁾を用いて、次式のように書ける。

$$\begin{aligned} V_x(x, y, z) &= \frac{\partial^2 \Phi_x}{\partial y \partial z} = \\ &\sum_{i=1}^I \sum_{j=2}^J \sum_{k=2}^K (a_{i,j,k} + a_{i,j-1,k-1} - a_{i,j-1,k} - a_{i,j,k-1}) \\ &\cdot B_{i,m}(x) C_{j,m-1}(y) D_{k,m-1}(z) (m-1)^2 / (\delta\xi_y \delta\xi_z). \end{aligned} \tag{15}$$

V_y, V_z についても同様である。

ところでソレノイダル条件式(2)は式(15)で用いたのと同じ微分公式により、

$$\begin{aligned} \text{div } \mathbf{V} &= \frac{\partial^3 \Phi}{\partial x \partial y \partial z} = \\ &\sum_{i=2}^I \sum_{j=2}^J \sum_{k=2}^K A_{i,j,k} B_{i,m-1}(x) C_{j,m-1}(y) D_{k,m-1}(z) \\ &= 0 \end{aligned} \tag{16}$$

となる。ただし、

$$\begin{aligned} A_{i,j,k} &= \\ &((a+b+c)_{i,j,k} + (a+b+c)_{i-1,j-1,k} \\ &+ (a+b+c)_{i-1,j,k-1} + (a+b+c)_{i,j-1,k-1} \\ &- (a+b+c)_{i-1,j,k} - (a+b+c)_{i,j-1,k} \\ &- (a+b+c)_{i,j,k-1} - (a+b+c)_{i-1,j-1,k-1}) (m-1)^3 \\ &/ (\delta\xi_x \delta\xi_y \delta\xi_z) \end{aligned}$$

とした。式(16)は、係数 A_{ijk} と3つのBスプライン関数から成っている。もしすべての (i, j, k) に対して $A_{ijk}=0$ なら、式(16)は (x, y, z) によらず成り立つ。実際、補間される離散データ式(12)の Φ_x, Φ_y, Φ_z に対し式(13)が成り立つ時、式(16)は常に成り立つことが示せる (付録)。この時、式(15)から得られる \mathbf{V} は (式(2)の意味で) ソレノイダル場となる。

4. 力線の可視化

ここでは式(9)を満たす離散ベクトルデータをもとに、力線を作成する方法について考える。二次元の場合これは簡単である。すなわち、離散データ $\mathbf{V} = (V_x, V_y)$ を数値積分して $V_x = -\partial\phi/\partial y$, $V_y = +\partial\phi/\partial x$ を満たすスカラーポテンシャル $\phi(x, y)$ を求めれば、その離散的スカラー場 ϕ の等高線が力線となる³⁾。この力線はソレノイダル条件を満たす。しかし、一般的な三次元の場合、このようなスカラーポテンシャルが定義できないので、等高線を描くような方法が使えない。そこで、力線追跡法による可視化を行う。

4.1 力線追跡法

非ソレノイダル場の場合も含めた一般的な力線の可視化法として、有限差分法を用いたベクトル場の力線追跡法が用いられる。これは常微分方程式の初期値問題であり、二次元の場合、

$$\frac{\partial x(s)}{\partial s} = V_x(x, y), \quad \frac{\partial y(s)}{\partial s} = V_y(x, y) \quad (17)$$

と書ける。この解 $x(s)$, $y(s)$ が力線となり、 s はパラメータである。これを解くには格子点以外でも連続的に \mathbf{V} の値が必要であり、何らかの補間が必要となる。また、式(17)を数値的に解くには差分化も必要である。

4.1.1 直接補間と流束補間

通常よく使われる補間法³⁾は、離散ベクトルデータの直接的な線形補間(一次のスプライン補間)である。例えば二次元の場合、 V_x は次のようになる。

$$V_x(x, y) = R_x R_y (V_x)_{i+1, j+1} + (1-R_x)(1-R_y)(V_x)_{i, j} + (1-R_x)R_y (V_x)_{i, j+1} + R_x(1-R_y)(V_x)_{i+1, j} \quad (18)$$

ここで

$$R_x = (x - x_i) / (x_{i+1} - x_i) \quad (x_{i+1} \geq x > x_i), \\ R_y = (y - y_j) / (y_{j+1} - y_j) \quad (y_{j+1} \geq y > y_j)$$

である。三次元の場合も同様である。以下ではこれを直接補間と呼ぶことにする。これは極めて簡単であるが、得られる連続的ベクトルデータ \mathbf{V} は、式(15)と異なり一般にソレノイダルとならない(式(2)を満たさない)。

この直接補間法と、第3章で紹介した流束関数の補間(以下では流束補間と呼ぶ)法(式(15))が、以下で比較される。

4.1.2 差分スキーム

一方、式(17)の差分計算法には多くのバリエーションがある。しかし、差分ステップサイズ(δs とする)

を十分小さくすれば、追跡される力線はどんな差分式を用いても一致するはずである。ゆえに、ここでは比較的簡単な二種類の差分スキームを選び、その追跡精度の比較を行う。その一つを次に示す。

$$x^{n+1} = x^n + \delta s V_x^n / V^n, \quad y^{n+1} = y^n + \delta s V_y^n / V^n. \quad (19A)$$

ただし、

$$V_x^n = V_x(x^n, y^n), \quad V_y^n = V_y(x^n, y^n), \\ V^n = \sqrt{(V_x^n)^2 + (V_y^n)^2}$$

である。これは最も簡単なもので一次精度を持っている。もうひとつは、よく知られたかえる跳びスキームで二次精度になっている。

$$x^{n+1} = x^{n-1} + 2\delta s V_x^n / V^n, \quad y^{n+1} = y^{n-1} + 2\delta s V_y^n / V^n. \quad (19B)$$

式(19B)の初期ステップ($n=0$ から 2)は計算上の初期条件として1ステップ先($n=1$)の点データが余計に必要である。ゆえに、最初の1ステップ目だけ式(19A)を用いる。

4.2 二次元の力線作成(その1)

まず簡単な例として、次式で表される二次元の円形スカラーポテンシャルに基づくソレノイダル場を考える。

$$\phi = \exp(-\alpha(x^2 + y^2)). \quad (20)$$

離散ベクトルデータは適当な差分格子を設定し、各格子点 (i, j) で次式を用いて得られる。

$$(V_x)_{i, j} = -(\phi_{i, j+1} - \phi_{i, j-1}) / (2\delta y), \\ (V_y)_{i, j} = +(\phi_{i+1, j} - \phi_{i-1, j}) / (2\delta x). \quad (21)$$

ここで、 $\delta x = \delta y = 0.04$ 、格子数は $I=J=25$ とした。このようにして得られたベクトルデータ \mathbf{V} は常に式(7)の意味でソレノイダル場になっている。

図3(a)は式(20)で表されるポテンシャル関数 ϕ ($\alpha=50$ とする)のある等高値における等高線である。ただし、 ϕ を上述の差分格子上に離散化し、それに基づいて等高線を作図した。すでに述べたとおり、これはソレノイダル条件を満たす力線である。等高線であることから、力線は必ず閉じなければならない。勿論、図3(a)の力線は閉じている。

図3(b), (c), (d)は力線追跡の結果である。いずれの場合も追跡は $\delta s = 0.1\delta x$ で $n=5000$ ステップまで行った。このうち、図3(b)の力線は、直接補間法(式(18))と差分スキーム(19A)を用いて、図3(a)の等高線上の点を初期位置とした追跡結果である。明らかに、この追跡力線は閉じていない。この場合は、 δs をもっと小さくすれば力線が閉じるようになる。

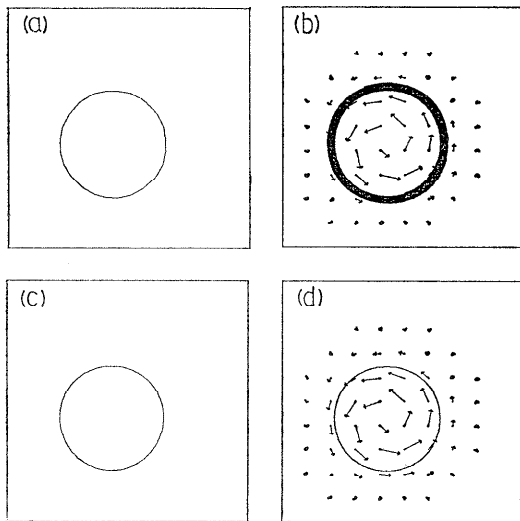


図 3 力線追跡 (二次元円形ポテンシャル [式(20)]):
 (a) ポテンシャルの等高線
 (b) 直接補間 [式(18)]と一次精度スキーム [eq. (19A)] を用いた場合
 (c) 直接補間 [式(18)]と二次精度スキーム [eq. (19B)] を用いた場合
 (d) 流束補間 [式(15)]と二次精度スキーム [eq. (19B)] を用いた場合

Fig. 3 Example of the line tracing (the case of a two dimensional circle potential function [eq. (20)]):
 (a) A contour line of the potential,
 (b) The scheme [eq. (19A)] and the direct interpolation [eq. (18)],
 (c) the scheme [eq. (19B)] and the direct interpolation [eq. (18)], and
 (d) the scheme [eq. (19B)] and the flux interpolation [eq. (15)], are used.

一方、これと同じ補間式(18)と δs で、スキーム(19B)を用いたのが図3(c)である。その力線は一回転で完全な円形リミットサイクルになっており、さらに約10サイクルの追跡に渡って力線は完全に重なった(しかもこの場合、 δs はさらに数倍大きくてもよい)。以上のことから、同じ δs であれば、差分スキームとしては式(19A)より(19B)の方がよいことがわかる。

また、流束補間法(式(15))と差分スキーム(19B)を用いた結果を図3(d)に示す。 δs や追跡距離は図3(b)や(c)と同じである。これも図3(c)と同様に力線は閉じ、正確な力線(図3(a))と一致した。

ところで、図3(b)には式(21)から得た離散ベクトルデータを間引きして矢印で表した。一方、図3(d)の矢印は、式(15)で得られた格子点でのベクトルを

表している。両者は、ほぼ同じに見える。勿論、図3(d)は流束関数に対する補間なので、もとの離散ベクトルデータ(図3(b))と格子点で完全に一致するとは限らない。この不一致は、もとの離散ベクトルデータを格子点での流速と見なすのか(図3(b))、それとも格子点付近の平均流速と見なすのか(図3(d))の違いに基づいている。本論文では、第2章で述べたように、このデータに格子の面積を掛けて流束を求めているから、後者の考え方に従っている。

4.3 二次元の力線作成 (その2)

次に、補間式(15)、(18)による違いを見るために、もっと複雑なポテンシャル関数の場合について調べる。このポテンシャルは次式のように二つの山を持っている。

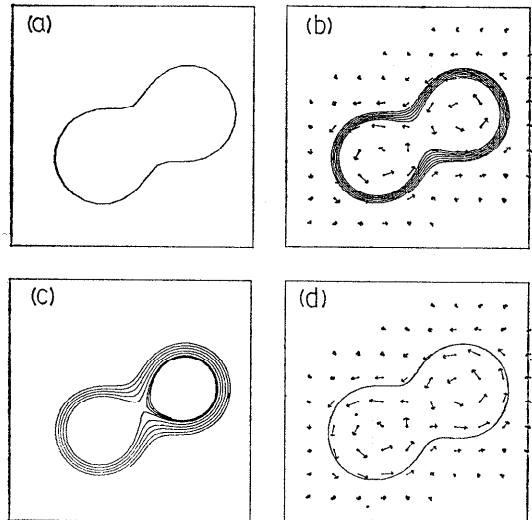


図 4 力線追跡 (二つ山の二次元ポテンシャル [式(22)]):
 (a) ポテンシャルの等高線
 (b) 直接補間 [式(18)]と一次精度スキーム [eq. (19A)] を用いた場合
 (c) 直接補間 [式(18)]と二次精度スキーム [eq. (19B)] を用いた場合
 (d) 流束補間 [式(15)]と二次精度スキーム [eq. (19B)] を用いた場合

Fig. 4 Example of the line tracing (the case of a two dimensional potential function with two peaks [eq. (22)]):
 (a) A contour line of the potential,
 (b) The scheme [eq. (19A)] and the direct interpolation [eq. (18)],
 (c) the scheme [eq. (19B)] and the direct interpolation [eq. (18)], and
 (d) the scheme [eq. (19B)] and the flux interpolation [eq. (15)], are used.

$$\begin{aligned} \phi = & \exp(-\alpha((x-\gamma_1)^2 + (y-\gamma_2)^2)) \\ & + \exp(-\alpha((x-\gamma_3)^2 + (y-\gamma_4)^2)). \end{aligned} \quad (22)$$

離散ベクトルデータの生成方法や格子条件, 追跡条件 ($\delta s = 0.1\delta x$, 5000 ステップ) は前例と同じである。ただし, $\alpha = 50$, $\gamma_1 = \gamma_2 = 0.35$, $\gamma_3 = 0.65$, $\gamma_4 = 0.5$ とした。

図 4 (a) に ϕ の等高線を示す。ポテンシャルの二つの山が隣接しているため, それはひょうたん形になっている。前例と同じく二次元のソレノイダル場なので, 力線は閉じる。

図 4 (b), (c), (d) は力線追跡の結果である。このうち, まず図 4 (b), (c) について考える。これらは図 3 (b), (c) と同じく直接補間式 (18) を用い, 差分スキームはそれぞれ式 (19A), (19B) である。いずれも力線は閉じない。しかも, この状況は δs をさらに小さくしても改善されない。これは, いかなる高精度計算を行っても, 力線が閉じないことを意味する。この理由は式 (18) で定義された連続的ベクトル場が式 (2) の意味でソレノイダル場になっていないからで, 図 3 (c) が良好な結果を与えたのは特殊な (真円ポテンシャル) データであったからにほかならない。つまり, 式 (18) で追跡された力線は湧き出しや吸い込みが起り得て, ソレノイダルに基づく位相学的性質を満たさない。

これに対し, 図 4 (d) は流束補間式 (15) と差分スキーム (19B) の場合である。図中の矢印は, 式 (15) による補間後のベクトル場を示す。図 3 と同様, これは補間前のベクトル場を示す図 4 (b) とほぼ同じである。また追跡された力線は, 等高線 (図 4 (a)) と同じ形状となり, 閉じている。これは式 (15) が厳密にソレノイダル条件を満たしているからである。勿論, この場合も δs は望む精度を得るために十分に小さくすべきである。しかし, 差分格子間隔の数分の一より小さな δs についてその追跡力線はほとんど変わらない。なぜなら, その線のなめらかさの程度は, B スプラインの次数と節点間隔でほとんど決まってしまうからである。

4.4 三次元力線追跡

次に三次元ベクトル場の力線追跡について考える。

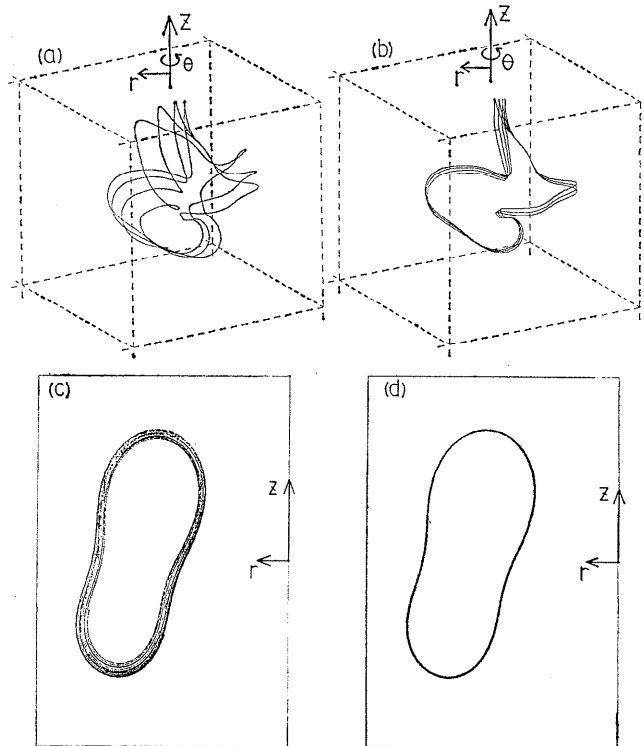


図 5 三次元力線追跡 (ベクトルポテンシャル [式 (23)]: すべて二次精度スキーム [(19B)] で追跡

- (a), (c) 直接補間を用いた場合
- (b), (d) 流束補間を用いた場合
- (c), (d) 力線のトーラス管断面への投影

Fig. 5 Three dimensional line tracing (the case of the vector potential [eq. (23)]).

- (a), (c) The direct interpolation [eq. (18)] and the flux interpolation [eq. (15)] are used with the scheme [eq. (19B)].
- (c), (d) The projection of the line onto a cross section of the torus tube.

この場合, スカラーポテンシャルが存在しないので, 力線追跡以外の方法は使えない。例としてトーラス形のベクトル場の場合を図 5 に示す。これは, 円柱座標を用いて, 次のようなベクトルポテンシャル ϕ から生成される。

$$\begin{aligned} \phi_r = & \alpha_1 z (1 - \cos(0.1\pi r)), \\ \phi_\theta = & \alpha z \exp(-\beta((r-\gamma_1)^2 + (z-\gamma_2)^2)) \\ & + \alpha z \exp(-\beta((r-\gamma_3)^2 + (z-\gamma_4)^2)), \\ \phi_z = & 0, \quad r = \sqrt{x^2 + y^2}. \end{aligned} \quad (23)$$

ただし, ここでは式 (23) を座標変換でデカルト座標に直し, デカルト座標の格子点上において $\mathbf{V} = \nabla \times \phi$ の差分式から式 (21) と同様の手順で, 離散データを生成

する。格子条件は $\delta x=1.0$, $\delta y=0.9$, $\delta z=0.8$, データサイズは $I=J=K=50$ である。また、式(23)で $\alpha_1=0.25$, $\alpha_2=8$, $\alpha_3=5$, $\beta=0.025$, $\gamma_1=6$, $\gamma_2=-5$, $\gamma_3=9$, $\gamma_4=5$ とする。

追跡条件を $\delta s=0.05\delta x$, 10000 ステップとして、得られた追跡力線の三次元的形状を、直接補間および流束補間について、それぞれ図5(a), (b)に示す。このうち、図5(a)は補間式(18)を三次元化したもの、図5(b)は補間式(15)を用いた。また、いずれの場合も差分スキームは(19B)を三次元化して用いた。結果はどちらもトーラス形になっているものの、少し違いが見られる。

二次元の場合と違って、そのままでは等高線との比較ができないので、ここではトーラス管断面に注目する。つまり、ベクトルポテンシャルの ϕ_θ 成分の等高線は rz 面上で管断面となり、それは閉じた線となる。すなわち、力線は位相学的にその面上で閉じなければならない。

図5(a), (b)に対応して rz 面上に投影した追跡力線をそれぞれ図5(c), (d)に示した。このうち、図5(c)は補間式(18)を三次元化したもので追跡しており、力線は閉じない。つまり、トーラス管断面が θ 方向に回転するにつれ単調(増加または減少)変化し、式(23)で定義したトーラスとは位相学的に別のものになっている。この結果は δs をさらに小さくしても変わらない。これとは対照的に補間式(15)を用いた図5(d)の力線は閉じており、トーラス形状に要求される位相学的性質を満たしている。

以上の結果より、三次元においても二次元の場合と同様、式(2)の意味でソレノイダル条件を満足するように補間することが、力線の位相学的性質を保持する上で極めて重要であると結論づけられる。実際、ソレノイダル条件を満足しない補間式(18)を用いた場合、 δs をどんなに小さくして(つまり高精度計算をして)も力線の位相学的性質は保持されない。

5. ま と め

本論文では、離散ベクトルデータ式(9)(または式(7))の意味でソレノイダル場である時、式(2)の意味でソレノイダルな連続的ベクトル場を生成する補間方法を示し、これがそのような位相学的性質を持った力線の可視化(特に三次元)に対して有効な方法であることを示した。この手法においてスプライン関数は重要な役割を担っている。すなわち、補間で得られる連

続的流束関数式(14)は、スプラインの微分公式により容易に連続的ベクトル場に変換できた。また、スプラインはBスプライン関数を基底とする線形補間式なので、 Φ_x , Φ_y , Φ_z を別々に補間しても、それらを足し合わせた Φ が式(13)を満たしていれば、得られる連続的ベクトル場(式(15))は常にソレノイダル条件式(16)を満足している(付録)。

本論文で考えた式(9)を満たす離散ベクトルデータは、非圧縮性流速場や磁場などのソレノイダル場の数値シミュレーションにおいてしばしば現れる。第2章では、数値シミュレーションで一般的に用いられる有限差分演算子(式(6))から、このようなベクトルデータが生成されることを示した。これは多くの高精度スキームや多段スキームに対しても、本手法が適用できることを示している。

特に、この手法は三次元 MHD 数値シミュレーションにおける磁場の位相学的な解析において重要である。通常、高導電率プラズマ中では磁力線の位相学的構造は変化せず、プラズマは磁力線に沿った移動しかできない。しかし、プラズマ中に電気抵抗が存在すると、磁力線のつながりによってその位相学的構造が変化し、プラズマの輸送過程は大きく変化する。普通、このような数値シミュレーションでは磁場を離散化するから、磁場データから磁力線を可視化する手段として、本手法は有効であると考えられる。

本手法の問題点として、以下に二点指摘しておく。まず、式(14)の未知係数を求めるには線形連立方程式を解かなければならない。大規模な数値シミュレーションで扱う離散データは大きくなるので、この計算に要する時間が問題となる。例えば、図5で用いた三次元問題のデータサイズは $50 \times 50 \times 50$ で、これを LU 分解法によって FUJITSU M-1800 で行うと約 50 秒を要した。この計算行列はかなり疎であるから、反復法等の算法によって高速化できるかもしれない。また、シミュレーション空間で位相学的性質が重要となる部分が限られているなら、本手法で扱うデータサイズも小さくて済むであろう。

もう一つの問題点は、Bスプラインの次数を三次以上にすると、ラグランジュ補間と同様の短波長ノイズが現れやすくなることである。これはスプライン補間自体の持つ欠点である。本手法は補間関数(流束関数)の一階微分をとって連続的ベクトル場を得るため、その連続性を保証する上でBスプラインは二次以上であればよい(このとき、ベクトル場は最低一次と

なる)。しかし、ベクトル場をさらに微分したいとき(例えば、MHD では磁場の回転微分は電流である)、三次以上を必要とし、その際のスプライン補間は節点のとり方等に注意が必要である。

また、本手法の問題点ではないが、それぞれの理工学問題に応じて要求される位相学的構造を離散ベクトルデータ自体が持つために、そのデータの生成過程(シミュレーション技法)にも注意が必要である。例えば、数値シミュレーションでは計算を安定化させるために、人工的散逸²⁾がしばしば加えられるが、これに起因する計算誤差がベクトルデータの位相学的構造に影響を与えることがある。MHD 問題を例にすれば、この人工的散逸は先述の電気抵抗(物理的散逸)と同様に磁力線をつなぎ変えてしまう。ゆえに、本手法を有効に利用するには、注目する位相学的構造に配慮したデータ生成(数値シミュレーション)を行うことも大切である。

謝辞 本稿をまとめるにあたり、貴重なご助言をいただいた和歌山大学 吉本富士市教授、愛媛大学情報工学科 天野要助教授に感謝いたします。

参 考 文 献

- 1) 鶴飼正行, 津田孝夫: 陽的差分による 3 次元電磁流体シミュレーションのスーパーコンピューティング, 情報処理学会論文誌, Vol. 31, No. 1, pp. 16-23 (1990).
- 2) Ugai, M.: MHD Simulations of Fast Reconnection Spontaneously Developing in a Current Sheet, *Comput. Phys. Commun.*, Vol. 49, No. 1, pp. 185-192 (1988).
- 3) Ugai, M.: Computer Simulations of Field-Aligned Currents Generated by Fast Magnetic Reconnection in Three Dimensions, *J. Geophys. Res.*, Vol. 96, A 12, pp. 21173-21181 (1991).
- 4) Schindler, K., Hesse, M. and Birn, J.: General Magnetic Reconnection, Parallel Electric Fields, and Helicity, *J. Geophys. Res.*, Vol. 93, A 6, pp. 5547-5557 (1988).
- 5) Priest, E. R. and Forbes, T. G.: Magnetic Flipping: Reconnection in Three Dimensions without Null Points, *J. Geophys. Res.*, Vol. 97, A 2, pp. 1521-1531 (1992).
- 6) Roache, P. J. 著, 高橋亮一ほか訳: 流体力学(下), 構造計画研究所, pp. 54-76 (1984).
- 7) 吉村和美, 高山文雄: パソコンによるスプライン関数, 東京電機大学出版局, p. 125 (1988).
- 8) Ugai, M. and Tsuda, T.: Magnetic Field Line Reconnection by Localized Enhancement of Resistivity. Part 1, Evolution in a Compressible MHD Fluid, *J. Plasma Phys.*, Vol. 17, Part 3,

pp. 337-356 (1977).

付 録

次式を満たす三次元離散データ Φ_{ijk} ($i=1\cdots I; j=1\cdots J; k=1\cdots K$) について考える。

$$\begin{aligned} & \Phi_{i+1,j+1,k+1} - \Phi_{i+1,j+1,k} - \Phi_{i,j+1,k+1} - \Phi_{i+1,j,k+1} \\ & + \Phi_{i+1,j,k} + \Phi_{i,j+1,k} + \Phi_{i,j,k+1} - \Phi_{i,j,k} = 0. \end{aligned} \quad (a1)$$

ここで、 (i, j, k) は座標系 (x, y, z) における格子点番号とする。 Φ_{ijk} をスプライン補間すると、得られる関数 $\Phi(x, y, z)$ は

$$\Phi(x, y, z) = \Phi_1(y, z) + \Phi_2(x, y) + \Phi_3(x, z) \quad (a2)$$

となる三つの連続関数 Φ_1, Φ_2, Φ_3 の和で表せることを以下で示す。この場合、必然的に次式が成り立つ。

$$\frac{\partial^3 \Phi}{\partial x \partial y \partial z} = 0. \quad (a3)$$

これは式(13)を満たす離散データに対し式(16)が常に成り立つことを示している。

本文中では、この離散データ Φ_{ijk} を適当に分割した三つの離散データ Φ_x, Φ_y, Φ_z (式(12))について別々に補間を行った。しかし、スプライン関数は B スプラインを基底とする線形結合型の補間関数なので、このように別々にスプライン補間した関数の和は Φ_{ijk} をそのままスプライン補間したものに一致し、すなわち式(a3)が成り立つ。言い替えれば、 Φ が式(a3)を満たせば、スプライン以外の補間法でもよい。

A.1 二次元の場合

まず簡単のため、次式を満たす一組の二次元離散データ Φ_{ij} ($i=1\cdots I; j=1\cdots J$) について考える。

$$\Phi_{i+1,j+1} - \Phi_{i+1,j} - \Phi_{i,j+1} + \Phi_{i,j} = 0. \quad (a4)$$

これは次のように書き直せる。

$$\Omega_{i,j+1} - \Omega_{i,j} = 0. \quad (a5)$$

ただし、

$$\Omega_{i,j} = \Phi_{i+1,j} - \Phi_{i,j} \quad (a6)$$

である。式(a5)は Ω_{ij} が j に依存しないことを表しているから、 Ω_{ij} を Ω_i で書き換えてもよい。こうすると、式(a6)から

$$\Phi_{i+1,j} = \Phi_{i,j} + \sum_{p=1}^j \Omega_p \quad (a7)$$

が得られる。この右辺第一項は j にのみ依存し、第二項は i にのみ依存する。したがって、これら二つの項を別々な離散データと見なせば、それぞれを補間した二つのスプライン関数の和として得られる関数 $\Phi(x, y)$ は、 $\Phi_1(y) + \Phi_2(x)$ の形で与えられる。

A.2 三次元の場合

次に、以上の議論を三次元に拡張する。そのために

$$\gamma_{i,j,k} = Q_{i,j+1,k} - Q_{i,j,k}, \quad (\text{a 8})$$

$$Q_{i,j,k} = \Phi_{i+1,j,k} - \Phi_{i,j,k} \quad (\text{a 9})$$

とおくと、式(a 1)は $\gamma_{i,j,k+1} - \gamma_{i,j,k} = 0$ と書ける。このとき、式(a 4)の Φ_{ij} は式(a 8)の Q_{ijk} で i を無視した二次元データと形式的に一致している。したがって、式(a 7)と同様に Q_{ijk} は (i, j) および (i, k) に依存する二つの項に分けられ、

$$Q_{i,j,k} = \omega_{i,j} + \lambda_{i,k} \quad (\text{a 10})$$

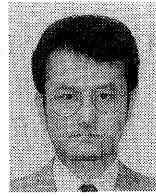
とおける。結局、式(a 9)、(a 10)から

$$\Phi_{i+1,j,k} = \Phi_{i,j,k} + \sum_{p=1}^i \omega_{pj} + \lambda_{pk} \quad (\text{a 11})$$

を得る。右辺の三つの項はそれぞれ (j, k) 、 (i, j) 、 (i, k) にも依存する。したがって、スプライン補間によって得られる関数 $\Phi(x, y, z)$ は、 $\Phi_1(y, z) + \Phi_2(x, y) + \Phi_3(x, z)$ の形となる。

(平成4年12月7日受付)

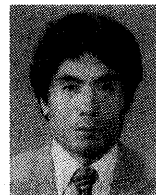
(平成5年5月12日採録)



清水 徹 (正会員)

昭和37年生。昭和62年東北大学大学院修士課程(応用物理)修了。同年高知工業高等専門学校機械工学科助手を経て、現在愛媛大学工学部情報工学科助手。工学修士。計算物理、

特にプラズマの計算機シミュレーションによる研究に従事。数値計算法、流れの可視化等に興味をもつ。地球電磁気地球惑星圏学会会員。



鞆飼 正行 (正会員)

昭和23年生。昭和46年京都大学工学部電子工学科卒業。昭和48年同大学院修士課程修了。同年愛媛大学工学部助手。現在同大学工学部情報工学科助教授。工学博士。計算物理の基礎的研究に従事。現在は数値シミュレーション、特に数値解析、スーパーコンピューティング、グラフィックス(アニメーション)等に興味をもつ。電気学会等会員。

特に数値解析、スーパーコンピューティング、グラフィックス(アニメーション)等に興味をもつ。電気学会等会員。