

再帰関数を用いた複雑境界モデルによる流路形状の自動最適化

寺井 優 晃[†], 松澤 照 男^{††}

実数型格子ガス法において、複雑な流路形状を扱うための実装モデルを提案する。提案するモデルの妥当性をキャビティおよびチャネル形状を用いて検討し、その応用例として流路形状の自動最適化を試みた。実数型格子ガス法は、従来の格子ガス法や格子ボルツマン法と同様に多数の仮想的な粒子の運動を発生させることで流体運動を再現する解析手法である。これらの手法の特徴として、複雑な流路や界面を形成する流れに対して多く適用されていることがあげられる。本稿では初めに実数型格子ガス法で複雑な流路形状を扱うためにセルの概念を導入し、空間を分割した。境界形状はセルの辺を単位とした直線で構成し、小粒度のセルを用いることで複雑な流路形状が扱えるようになる。また、再帰関数をあわせて用いることで判定方法が煩雑になりにくく、ソルバの拡張性が確保できる。以上、提案した複雑境界モデルのベンチマークテストとして流路形状の最適化問題に適用した。最適化対象の形状は8本のチャネルを持つハニカム形状を模したモデルで、最適化アルゴリズムには遺伝的アルゴリズムを採用した。最適化計算の結果、チャネル間の流量の偏りを緩和させることができた。

A Complex Boundary Model by Using Recursive Function for Optimizing Automatic Design

MASAAKI TERA[†], and TERUO MATSUZAWA^{††}

A complex boundary model is proposed. This model is implementation method of boundary detection in continuous-velocity lattice-gas model. A continuous-velocity lattice-gas model is flow analysis by using particle motion, resembles primitive lattice-gas model and lattice-Boltzmann model. These models were applied to boundaries and interfaces with complex geometry, frequently. In this study, to treat boundaries with complex geometry in a continuous-velocity lattice-gas model, the authors introduce cells to divide a system into sub space. In two-dimensional system, boundaries are consist of edges of cells, therefore small grained cells can approximate boundaries with complex geometry. Moreover by using recursive function can keep implementation simply, and to ensure extension of this solver. The proposed model was developed and was applied optimizing automatic design to estimate benchmark test. This system in benchmark test has 8 channels which are honeycomb structure. This system was optimized by GA algorithm. As a result, in the difference of flow rate between channels, optimal channels are smaller than before optimal channels.

1. はじめに

複雑な流路は工業製品の製造過程および製品自体に見ることができる。大きなものは各種化学プロセスが行われる装置などに、小さなものは電子機器に用いられるマイクロヒートパイプやヒートシンクなどで見られる。これらの複雑な流路は目的こそ違っているが、

一般には流路内の流体が均一になるように設計されることが望ましい。しかし、対象とする流路のスケールが多岐にわたっているため、連続体として見なせる流体と分子論的な特徴が無視できない流体を、1つの数値解析手法で扱うことは難しい。

流動現象の解析では、Navier-Stokes 方程式に基づいた差分法や有限要素法を用いることが一般的である。これを巨視的な条件に基づいた手法と呼ぶ。反面、複雑な流路や界面を形成する流れ^{1),2)} などでは、境界条件の設定が困難な場合が多く、微視的な条件に基づいた手法が用いられる。代表的なものは、格子ガス法や格子ボルツマン法^{3),4)} である。

1997年にMalevanetsらによって提案された実数型格子ガス法^{5),6)} は、従来の格子ガス法と同様な微視的な立場から流れを解析する手法である。特徴とし

[†] 北陸先端科学技術大学院大学情報科学研究科博士後期課程
School of Information Science, Japan Advanced Information Science and Technology (JAIST)

^{††} 北陸先端科学技術大学院大学情報科学センター
Center for Information Science, Japan Advanced Science and Technology (JAIST)
現在、独立行政法人理化学研究所情報基盤センター
Presently with Advanced Center for Computing and Communication, RIKEN

て、DSMC (Direct Simulation Monte Carlo) 法に類似した衝突則を採用している。これにより、粒子の並進方向に制限がなくなり、粒子の速度から温度場を求めることができる。そのため従来の解析手法が必要であったエネルギー方程式を陽に解く必要はない。また、多相流モデル^{7),8)} に関しても報告されており、相間の界面をともなうような複雑な流れモデルも扱える。ただし、複雑な流路形状に関する報告は多くはない。

本稿では、実数型格子ガス法において複雑な流路形状を扱う際の境界条件の実装モデルについて示す。これに関する報告例はない。ここでは、複雑な流路形状の実装が可能となることをセルによる複雑境界モデルを用いて示す。本来、実数型格子ガス法ではセルは必要ない。また一般的に、境界条件の設定は煩雑なコーディングになることが多いため、再帰関数を導入しアルゴリズムの単純化を図った。セル単位で境界形状を扱う実装の特徴を活かすために、画像を入力にし、ピクセルごとの色情報を基に流路形状を設定するシステムを構築した。

実装した複雑境界モデルの応用例として、チャンネル構造を持つ流路の流体計算および形状最適化を行った。日岡⁹⁾ は複雑な流路としてチャンネル構造をあげ、その流路における流動状態の不均一さを解消することは、化学プロセスにおける触媒および吸着反応の効率化につながるとしている。そして、装置性能を高めることを最終的な目的とし、格子ガス法を用いて流路形状の最適化について報告している。一般的に流体問題の最適化は非常に複雑な問題である^{10),11)}。本稿では、実数型格子ガス法と遺伝的アルゴリズムを併用することで流路形状の最適化を行った。ここでは、日岡らと同様に八二カム形状を模した吸着装置内のチャンネル流路をモデルとした。

2. 実数型格子ガス法

実数型格子ガス法は、従来の格子ガス法と同様に並進と衝突を分離して計算を行う。ここでは、2次元理想気体の条件に従う仮想的な粒子を考える。

系に N 個の粒子を与え、その i 番目の粒子の座標を x_i 、速度を c_i とする。各粒子は、1 タイムステップごとに、現在の座標 x_i に速度 c_i を加え、次のステップにおける座標 x_i' となる。肩記号のプライムは更新後の物理量を表す。並進は式 (1) で与えられる。

$$x_i' = x_i + c_i \quad (1)$$

粒子間の衝突は、セルの中心に設けたサイト上で行われる。粒子の座標は実数値で、サイトの座標は整数値である。よって衝突するサイトを決定するために、

橋本らが提案した乱数を用いる方法⁷⁾ を採用する。衝突では運動量 (速度)、運動エネルギーのみ交換し、衝突の前後で座標は不変とした。

すべての粒子について、衝突を考慮するサイトを決定した後に、各サイト上の平均速度ベクトル V を求める。ここではすべての粒子の質量は規格化されているものとする。1 つのサイトの平均速度ベクトル V は式 (2) で求められ、 M はそのサイト上で衝突する粒子数である。回転行列を σ とすることで、衝突則は式 (3) で与えられる。

$$V = \frac{1}{M} \sum_i^M c_i \quad (2)$$

$$c_i' = V + \sigma(c_i - V) \quad (3)$$

回転行列 σ を用いることで、粒子間の運動量の交換 ($m = 1$ なので速度と等価) を、単純な幾何変換 (質量重心による速度ベクトルを回転) を用いて表現できる。この衝突側により、衝突前後の運動量保存および運動エネルギーは保存される。また粒子群の平均速度ベクトル V を持つ仮想的な粒子と、その中のそれぞれの粒子との衝突を考慮することで、粒子群の多体衝突をモデル化している。この回転角により動粘性係数 ν が決定される。ランダムな回転角を与えた場合の動粘性係数は式 (4) となる^{6),7)}。 ρ_0 は初期条件で与えるサイトあたりの粒子数密度、 T_0 は初期条件で与えた温度である。式 (4) は、温度と粒子数密度を大きくすることで動粘性係数 ν が大きくなることを示しており、気体の特徴を示している。動粘性係数 ν 、代表長さ L および代表速さ U を用いることで、レイノルズ数は $Re = L \cdot U / \nu$ として与えられる。

$$\nu = \frac{1}{12} + T_0 \frac{1 + \rho_0 - e^{-\rho_0}}{2(\rho_0 - 1 + e^{-\rho_0})} \quad (4)$$

サイト間の距離は 1 に規格化し、等間隔に配置した。速度、密度、温度、圧力などの状態量はこのサイト上で観測する。

3. 再帰関数による境界モデル

3.1 セルを意識した境界モデル

格子ガス法は、多数の仮想的な粒子の運動を時間的に発展させることにより、流体としての状態量 (速度、密度、温度、圧力) を得る手法である。各々の仮想的な粒子は、それぞれ独立した座標、運動量を持つ。実数型格子ガス法では、サイトと呼ばれる観測点を代表点として、近傍にある粒子の物理量を空間および時間的に平均化することで、巨視的に観測される状態量を得る。

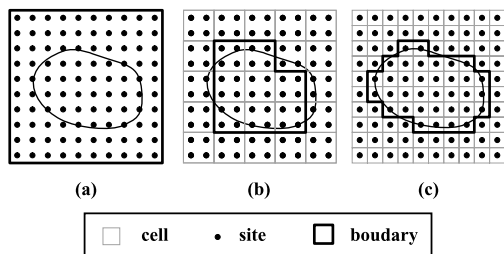


図 1 複雑な境界形状をセルによって近似 . (a) セルが導入されていない場合 . (b) セルが粗い場合 . (c) セルが細かい場合 (セルとサイトの数が等しい場合)

Fig. 1 The model using cells to approximate complex geometry. (a) Using no cells. (b) Coarse grain cells. (c) Small grain cells, its number is equal to number of sites.

図 1 のように、サイトを取り囲むようにセルを配置する . (a) はセルが導入されていない場合 , (b) は系を比較的粗めのセルを用いて分割した場合である . セルの辺部分に境界条件を設定する場合 , 従来のセルはサイトに比べて粗く , 単純な形状にのみ適用されてきた . 本稿では , セルとサイトが 1 対 1 で対応するような (c) が最も複雑な流路形状の実装に適していることを示す .

粒子とセルの関係を実装の面からとらえると , 衝突過程において , 粒子は所属するセル以外の粒子とは依存関係が発生しない . また並進過程においても , 当該粒子が境界を越えて隣接セルに移動しない限りにおいては , 所属するセル以外との依存関係は発生しない . そのため , セルの辺部分に境界条件を設定することで , 各セルは他のセルとは独立した小さなキャピティと見なすことができる . この局所的な構造は , セルオートマトンに特有のものであり , セル間でネットワークを形成することにより系全体が表現される . ここではセルの集合によって形状を表すモデルのことを複雑境界モデルと呼ぶ .

図 2 に , 系が 4 つのセルで構成される場合の複雑境界モデルを示す . 2 次元モデルの場合 , 1 つのセルは 4 つの境界端を持つ . 4 つの境界端には , 物理的な境界条件オブジェクト (physical object) , あるいは他セルとの境界端 (connected object) が接続される . 物理的な境界条件オブジェクトには , 滑りありオブジェクト , 滑りなしオブジェクト , 流入流出オブジェクトを用いる . これらはオブジェクトの名前が示すとおり , 滑りあり , 滑りなし , 流入流出境界条件を課するための関数群である .

コード中では , 各セルの境界端にどのようなオブジェクトが接続されているか , 記憶させておく必

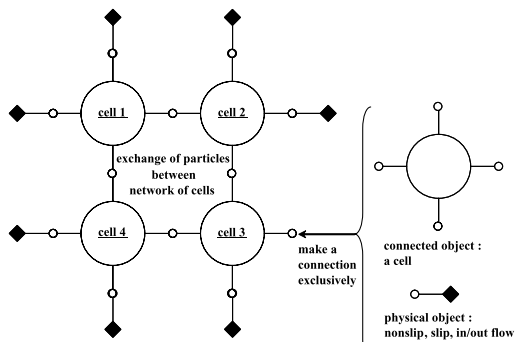


図 2 複雑境界モデル

Fig. 2 Complex boundary model.

```

struct cell_data {
    int id;
    int stat;
    int xsize;
    int ysize;
    double offset_x;
    double offset_y;
    // pointer to adjacent cell data
    struct cell *dest_top;
    struct cell *dest_bottom;
    struct cell *dest_left;
    struct cell *dest_right;
    // pointer to boundary function(physical/connect)
    int (*boundary_function_top)();
    int (*boundary_function_bottom)();
    int (*boundary_function_left)();
    int (*boundary_function_right)();
    // body of particle data
    struct particle_data part[number of particles];
}
    
```

図 3 セルを定義する構造体の詳細

Fig. 3 Data structure to define a cell.

要がある . これら記憶するパラメータを属性値と呼ぶ . 各セルの属性値は構造体のメンバによって管理する . 構造体のメンバは , セルの種類 (stat) , 大きさ (xsize , ysize) , セルに含まれるサイトのオフセット座標 (offset_x , offset_y) , 隣接セルへのポインタ (dest_top など) , 境界条件を処理する関数へのポインタ (boundary_function_top など) などである . 図 3 にセルを定義する構造体を示す . 隣接セルがない場合は dest_top などに NULL を代入させておく . 物理的な境界条件オブジェクトがない場合も boundary_function_top などに NULL を代入させておく .

3.2 再帰関数による境界条件判定

ここでは , 図 1 (c) のような小粒度セルによる複雑境界モデルを実装する . 系全体はセルの集合によって

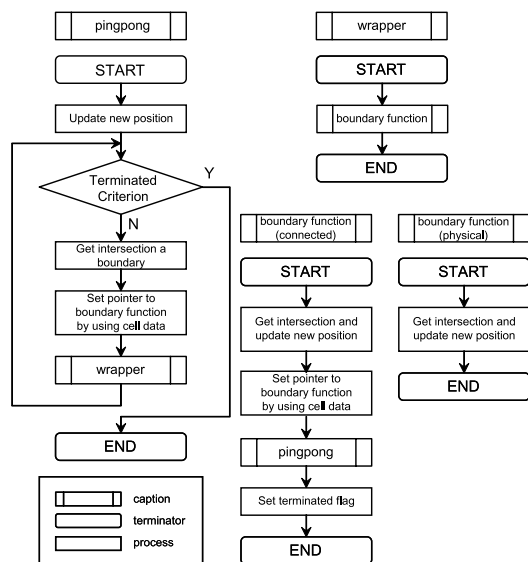


図 4 境界判定処理のフローチャート

Fig. 4 Flowchart to calculate translational motion of a particle at a boundary.

表されるため、粒子が並進することによって所属セルから隣接セルに移動させる処理が必要となる。粒子の速度がセルの辺の大きさよりも十分に大きい場合、1 タイムステップ中にセルの境界を越える処理が複数回必要となる。このような場合、複雑境界モデルを導入しない場合と同じ距離だけ並進することを保障するために、各粒子に1 タイムステップによって規格化された無次元化時間(=1)を持たせる。各粒子は並進開始時に無次元化時間が与えられ、当該セルのセル境界まで並進した距離とセル境界を越えて並進した距離の比に応じて時間を減らし、最終的に無次元化時間が0になった場合に、新しい所属セルが決定する。当該セルの境界を越えた場合に、セルの属性値に従って境界条件処理を行う。境界判定をセル単位に繰り返し行うことにより、処理の単純化が行える。本稿ではこの繰り返し処理の類似性に着目し再帰関数を用いる。これによって、コードの煩雑さが抑えられ、また単位セルに接続する境界条件関数を追加するだけで系全体に適用可能な新しい物理的な境界が設定でき、ソルバの拡張性が確保できる。

図4に境界判定処理のフローチャートを示す。関数 pingpong および境界条件処理関数 boundary_function, それを呼び出すための関数 wrapper から構成される。境界判定と、セルの辺(境界)を越えた場合に適用される境界条件処理を別にする事で、アルゴリズムに汎用性を持たせた。図3の構造体には、辺ごとの境界条件処理関数へのポインタ *boundary_function が定

義されており、ポインタの先には実際の処理関数を与える。辺ごとの判定部分に境界条件処理関数を直接記述すると、コードが煩雑になりやすい。そこで境界条件処理関数のポインタをラッパー関数 wrapper の引数に与えて呼び出すことで、大量のセルに対する境界条件が柔軟に設定できる。

関数 pingpong は、1 粒子のデータ(座標や速度)および所属するセルの属性値を引数に、並進処理の主関数から呼び出される。更新後の座標が当該セルの境界外ならば、再帰による座標補正処理が行われる。再帰呼び出し先の pingpong で当該セルの境界内であると判定されれば、呼び出し元の boundary_function に戻り、終了フラグがセットされる。そして、はじめに呼び出された pingpong に戻ることにより、補正された座標と所属する新しいセルが確定し、当該粒子の並進処理が終了する。このフローチャートに示す関数群は、1 粒子に対する処理を示しており、これをすべての粒子に対して行うことで系全体の並進が終了する。

関数 boundary_function は、境界条件処理関数である。境界条件のプログラム上の本質的な違いはほとんどないため、ここでは抽象化した名称を用いる。ただし再帰処理は、他セルとの境界端が接続された場合に行われ、この場合 boundary_function (connected) を用いる。一方、物理的な境界条件オブジェクトの場合、再帰は行わずに pingpong の反復処理のみで座標補正を行い、boundary_function から pingpong を呼び出すことはない。この場合 boundary_function (physical) を用いる。

なお pingpong が自分自身を呼ぶのではなく、子ルーチンの boundary_function が pingpong を呼び出す構造をしているが、アルゴリズムの構造としては再帰である。

ここで処理の流れを図5に簡単に示す。

- (a) 当該粒子について、現在の座標に速度を加える。この場合 cell 1 の境界外に出るので、cell 1 と cell 2 の境界における座標を求める。関数 pingpong で境界と粒子の軌跡の交点が右境界であることを判定し、右境界のセル属性値に基づき境界条件処理関数を呼び出す。この場合、関数 boundary_function (connected) の境界条件処理関数が呼び出される。その後、子ルーチンから再帰的に pingpong が呼び出される。
- (b) 再帰的に pingpong が呼び出されたことにより、当該粒子は cell 2 に移動したことになる。速度に残り時間をかけた移動量を求め、新座標に加える。この場合、cell 2 の境界外に出るので、(a) と同

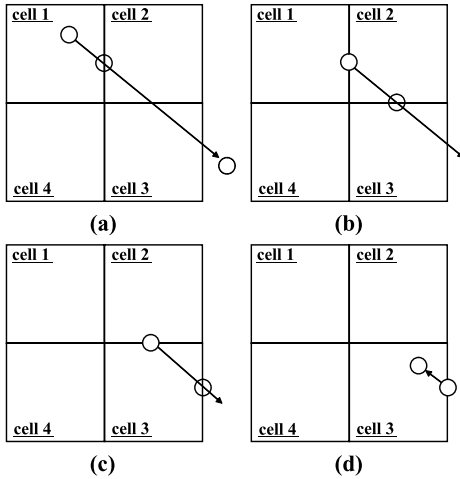


図 5 境界判定処理の動作
Fig. 5 Behavior of boundary process and translation between cells.

様に cell 2 と cell 3 の境界と軌跡の交点を求め、適切な境界条件処理関数を呼び出す。その後、再び pingpong を呼び出す。

- (c) 再帰的に pingpong が呼び出されたことにより、当該粒子は cell 3 に移動したことになる。速度に残り時間をかけた移動量を求め、新座標に加える。この場合、cell 3 の境界外に出るので、交点を求める。その後、境界条件処理関数を呼び出す。
- (d) 関数 boundary_function (physical) の境界条件が呼び出され適切な補正位置に収まる。この例では滑りなし境界条件を示した。呼び出し元の pingpong に戻り、終了フラグをセットする。ここで、当該粒子の終了状態が確定する。

4. 検証

4.1 強制対流キャビティ

提案した複雑境界モデルを導入し、それぞれのセル境界が独立して機能しているか妥当性を示すために強制対流キャビティを用いた検証を行った。図 6 のキャビティ上面は移動壁となっており、それを構成するセルの上辺は滑りありの境界条件を用いている。キャビティ左下を原点とし、横方向を x 成分、縦方向を y 成分とし、 x 成分速さを u 、 y 成分速さを v とした。滑りあり境界条件にはコサイン散乱を用いた。セル単位の移動速さを u_c 、キャビティ上面での平均速さ U で表す。なお、以後用いる物理量はすべて無次元化されているものとする。(a) は x 方向に $u_c = 0.5$ で移動し、(b) は隣り合うセルが反対の方向に $u_c = 0.5$ あるいは $u_c = -0.5$ で移動する。キャビティの代表長

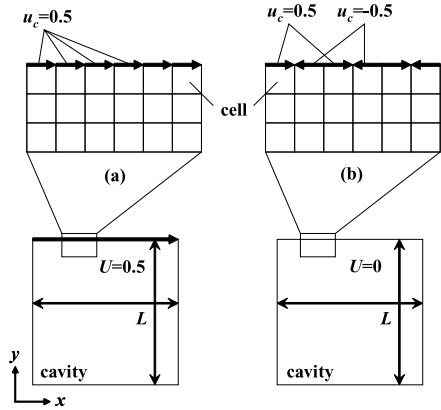


図 6 検証の計算条件 (a) キャビティ上辺の平均速度 $U = 0.5$. セル単位では $u_c = 0.5$. (b) キャビティ上辺の平均速度 $U = 0$. セル単位では $u_c = 0.5, -0.5$ を交互に繰り返す
Fig. 6 Conditions. (a) Average velocity of upper edge is $U = 0.5$. Velocity in each cell is $u_c = 0.5$. (b) Average velocity of upper edge is $U = 0$. Velocity in each cell is alternately $u_c = 0.5, -0.5$.

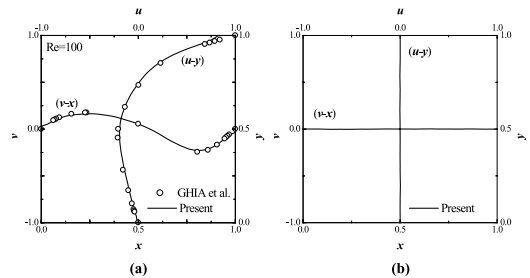


図 7 速度プロファイル . (a) $U = 0.5$, (b) $U \approx 0.0$
Fig. 7 Velocity profiles. (a) $U = 0.5$, (b) $U \approx 0.0$.

さは $L = 128$ で粒子数密度 $\rho = 64$ 、移動壁および系に初期条件で与える温度 $T = 1.0$ を使い、レイノルズ数 Re は 100 程度と見積もられる。

図 7 は、検証結果と Ghia らの結果¹²⁾ を比較したものである。速度プロファイルは、キャビティ中央を通る縦方向と横方向の断面における速度 u, v の大きさを表している。各成分とも壁移動速度 0.5 で正規化した。(a) の条件では、複雑境界モデルを用いない場合と同様にキャビティ上面の平均移動速さは $U = 0.5$ となる。一方 (b) は、セル単位では $u_c \neq 0$ であるが、キャビティの上面全体の平均速さ $U \approx 0$ になるため、対流は発生しないことが、図 7 から分かる。以上より、それぞれのセルに独立した境界条件が与えられ、巨視的な流れに反映されていることが確認できる。なお (b) は検証のための条件であって、特定の物理的な対象は想定していない。

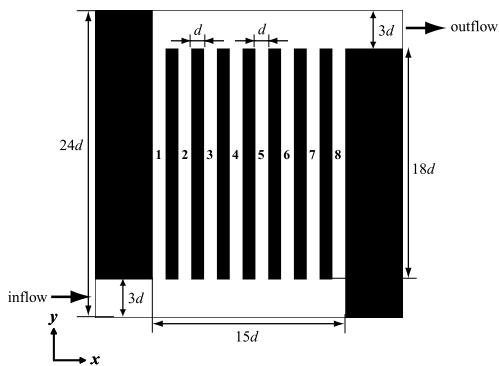


図 8 チャンネル形状

Fig. 8 A model in the shape of channels.

4.2 チャンネル形状

形状はハニカム形状の吸着装置内の流路をモデル化したものを用いた．形状を図 8 に示す．左下境界から流入させ，8 本のチャンネルのいずれかを經由し，右上境界から流出させる．流出境界から出た粒子は破棄し，その流出した粒子数と同数の粒子を流入境界側に一定の速度を与え配置した．系全体の粒子数は一定とした．図 8 のチャンネル中央に書かれている番号はチャンネル番号である．ここでは左から右に昇順で番号付けをした．黒色で塗りつぶされた部分は滑りなしの壁であり，粒子は通過することはできない．粒子数密度 $\rho_0 = 32$ ，温度 $T = 1.0$ ，流入境界に与えた速度 $U_{in} = (0.35, 0.0)$ とし，横方向を x 成分とし，縦方向を y 成分とした．このモデルを用いて日岡らの結果⁹⁾と比較検討を行う．

図 9 に速度ベクトルと y 成分速度コンターを示す．レジェンドはコンターのものである．11000 ステップ計算し，1000 ステップの時間平均をとったものを結果とした．ここではチャンネル幅 $d = 20$ として，通常の粘性条件を用いた．チャンネルの主流方向の長さ $L = 18d$ を代表長さにとると，レイノルズ数は $Re = 110$ 程度と見積られる．またチャンネル内の流速は，ほとんど y 成分速度であることが分かる．流入部よりも幅が小さなチャンネル内では，流速が大きくなっていることが分かる．また番号の大きなチャンネルほど流速が大きく分布に偏りがあり，右下および左上の角で流速がほとんどなくなっている部分があることが分かる．

図 10 に各チャンネルの流量比を示す．流量は一般的には長さ（面積）と流速の積によって定義されるが，流路形状によっては流路の幅（面積）が特定しにくくなる．よって本稿では各チャンネル領域のセルあたりの流量，つまり各チャンネルの平均流速を流量の定義に用いた．なお定量的な比較のために，日岡が行った格

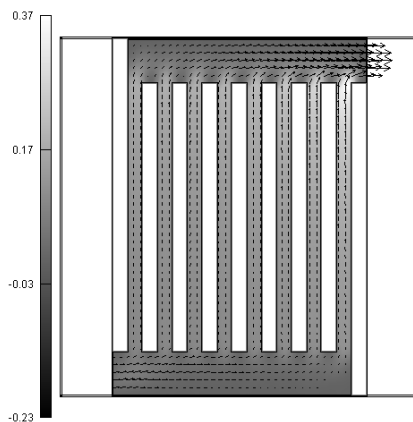


図 9 速度ベクトルおよび y 成分速度コンター

Fig. 9 Velocity vector and y -component velocity contour.

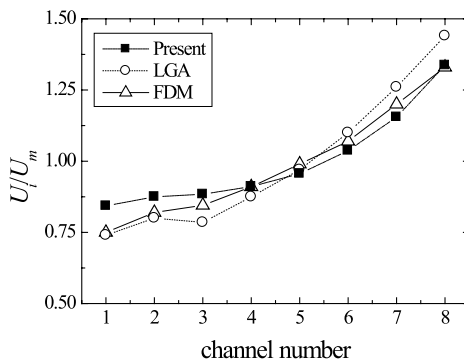


図 10 流量比 (■ 実数型格子ガス法, ○ 格子ガス法, △ 差分法)
Fig. 10 Flow rate ratio. (■ Present, ○ LGA, △ FDM)

子ガス法 (LGA) および差分法 (FDM) の結果⁹⁾を引用した．横軸はチャンネル番号であり，縦軸は各チャンネルの流速 U_i をチャンネル全体の平均流速 U_m で除したものである．系のほぼ中央のチャンネル 5 が平均流速と同程度であり，流出境界に近いチャンネル (6, 7, 8) ほど大きく，流入境界に近いチャンネル (1, 2, 3) ほど小さくなっており，チャンネル間の流量比が均一でないことが分かる．これは日岡らの結果と同じ傾向を示している．

5. 最適流路形状の自動設計

複雑流路モデルの応用例として，流路形状の自動最適化を行う．最適化に用いる流路形状としては，日岡らが用いたチャンネル形状をもとにした流路形状を用いた．

ここではチャンネル形状を変えることで，チャンネル間の流量を均一にする．ここでは実装の容易さから，遺伝的アルゴリズムを採用し最適化を行った．遺伝的アルゴリズムは，解の候補を単純な変数（染色体）で表

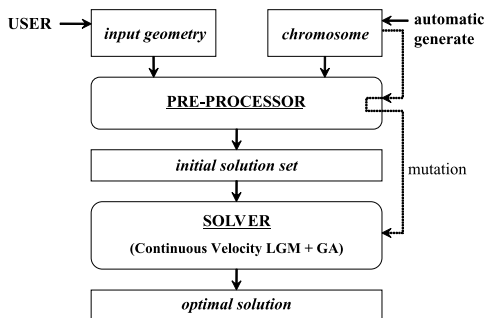


図 11 システムの概要
Fig. 11 Outline of the system.

現し、これを生物の進化に見られる過程を模倣し、選択、交叉、突然変異などの操作を繰り返すことで最適解の近似値を求めるものである。本稿では1つの世代は複数の流路形状（個体）からなる。その中から流路形状の設計図である染色体を2つ選択し、交叉したものを次の世代の解候補に加える。

複雑境界モデルの境界判定はセル単位で行っており、セルの種類（滑りなし壁、滑りあり壁、空間など）は属性値によって決定される。よって画像ファイルが持つピクセルごとの色情報とセル属性値を対応づけることで、境界条件を含めた複雑な境界形状の作成が比較的簡単にできる。システムの概要を図11に示す。最適化するオリジナルの形状ファイル（input geometry）と染色体（chromosome）を前処理部（PRE-PROCESSOR）にあわせて与えることで、最適解候補である形状を画像として生成する。これを複雑境界モデルを実装したソルバに投入し評価計算を行い、その結果に基づいて最適化が行われ、最適解を得ることができる。

5.1 染色体と流路形状

最適化対象の形状は、8つのチャンネルとそれらを分離する7つの壁を持つ。各チャンネルを分離している壁形状を変えることで流体に対する抵抗を変え、チャンネル間の流速を一定にすることを試みる。図12に染色体と生成されるチャンネルの例を示す。壁形状は四角形をしており、ここでは四角形の上辺と下辺の幅を変えることで形状の最適化を試みる。四角形（台形）の壁形状の上辺と下辺の幅（それぞれ L_{top} , L_{bottom} ）を染色体とすることで、長方形の幅のみを変更する場合よりも柔軟に流路形状の調整が可能になり、また壁間に複雑なチャンネル形状が形成されるため複雑境界モデルの良い適用例となる。

チャンネルを分離する壁の中心位置は固定し、チャンネル形状生成時に壁を横方向に伸縮することで四角形（台形）を形成する。上辺と下辺の幅の上限値と下限

make a chromosome :
A chromosome consists of width from center of each wall.

$$L_{bottom/top}(\text{wall number}) := \text{width}$$

chromosome
 $L_{bottom}(1) L_{top}(1) L_{bottom}(2) L_{top}(2) \dots L_{bottom}(7) L_{top}(7)$
 (example) 3148325764832734

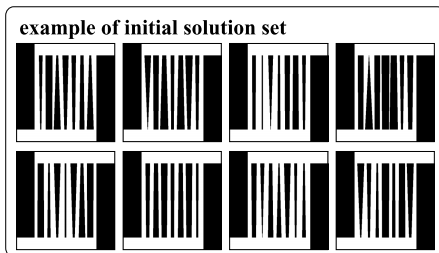
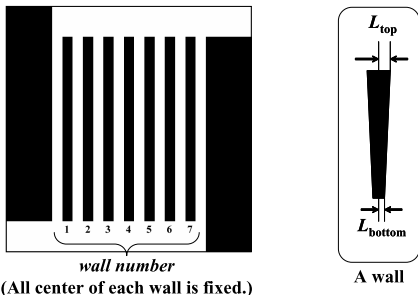


図 12 染色体をもとに生成されるチャンネルの例
Fig. 12 Example of channels based on chromosomes.

値はあらかじめ与えておき、染色体決定の際に、その範囲に収まる値を乱数を用いて決定する。乱数は整数値である。ここでは $2 \leq L_{top}, L_{bottom} \leq 10$ の値を生成した。チャンネルを分離する7つの壁の上辺および下辺それぞれの幅を決定する値が必要となるため、1つの染色体は14個の値からなる。

壁は次の手順で形成する。上辺と下辺の幅が染色体により与えられ、壁の上辺と下辺を生成する。次に上辺の左端と下辺の左端をプレゼンハムアルゴリズムを用いて直線を生成し、四角形の左辺とする、同様に右辺についても上辺の右端と下辺の右端を直線で結び四角形の右辺とする。その線に囲まれた領域を壁とすることにより四角形（台形）が生成される。

5.2 選 択

本稿では各チャンネルの流速が均一になるような形状を得るために、流速の変動係数 C_V を用いて評価値 score を求める。変動係数を用いる方法¹³⁾については伊集院らが提案しており、標準偏差を平均で割ることにより、標準偏差の感度が平均量によって変化し、流量が大きく標準偏差が小さいものが得られることが期待される。チャンネル間の流速の偏りが小さい流れが優良形状であるから、変動係数が小さいものを残すよう

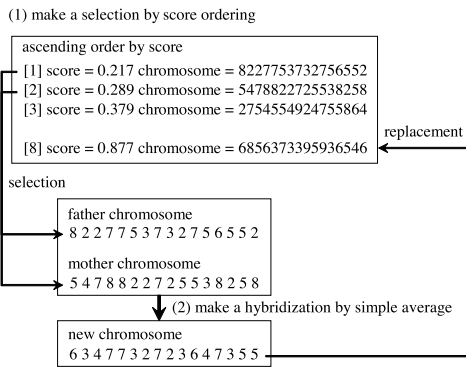


図 13 選択と交叉

Fig. 13 Selection and crossover.

に選択を行う。式 (5) に変動係数を示す。\$N (= 8)\$ はチャンネル数、\$U_i (1 \le i \le N)\$ は \$i\$ 番目のチャンネルの平均流速、\$U_m\$ はチャンネル全体の平均流速である。ただし制約条件として流速が負になる場合（逆流）を考慮し、変動係数にペナルティ値を加えたものを評価値 *score* とする。ペナルティ値 *penalty* (\$\gg 1\$) は十分に大きな値をとる。

$$C_V = \frac{1}{N} \sqrt{\sum_{i=1}^N (U_i - U_m)^2} \quad (5)$$

$$\text{score} = \begin{cases} C_V & u_i \geq 0 \\ C_V + \text{penalty} & \text{otherwise} \end{cases} \quad (6)$$

以上、すべての染色体に対して評価値を求める。図 13 の (1) に示すように、評価値のリストを昇順にソートし、最も評価値が良い染色体（最良形状）と 2 番目に評価値が良い染色体（次点の最良形状）を選択する。

5.3 交 叉

図 13 の (2) に染色体を用いた選択と交叉について示す。選択された 2 つの染色体の平均値を、新しい染色体とする。他の染色体と同様に整数値にするために、平均値の端数は切り捨てる。新しく生成された染色体を、選択の際にソートされたリスト中で最も評価値が悪い染色体（最悪形状）と置き換える。新しい染色体から流路形状を生成し、流体計算を行うことで新しい染色体の評価値を得る。その後、新しく得られた評価値をリストに加えて再ソートし直し、選択と交叉を繰り返す。

5.4 突然変異

通常の交叉の際に、5%の確率でランダムな値を持つような染色体を採用することで突然変異を実現する。この際、最良形状と次点の最良形状の平均値による交

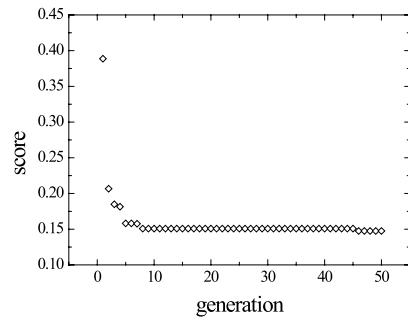


図 14 世代ごとの評価値（評価値は世代ごとの最小値）

Fig. 14 Generation vs. score. (A score is minimum value in each generation).

叉は採用しない。

5.5 結 果

以上、最適化アルゴリズムに遺伝的アルゴリズムを用いて計算を行った。1 世代あたり 8 個の形状（個体）からなり、計算開始時にすべての形状の評価値を求める。これを第 0 世代とする。その後、1 個の最悪形状の更新を 1 世代とし 50 世代の計算を行った。1 世代あたりの評価計算で用いた計算条件は、\$d = 10\$、\$\rho = 16\$、\$T = 1.0\$、流入速度 \$U_{in} = (0.35, 0.0)\$ とした。この場合の動粘性係数は \$\nu = 0.65\$ である。代表長さをチャンネルの主流方向の長さ \$L = 18d\$ とすると、レイノルズ数は \$\text{Re} = 96\$ と見積られる。1100 ステップの流体計算を行い、100 ステップの時間平均をした後に評価値を求めた。4 章の検証と同様に、チャンネル形状は横方向を \$x\$ 成分、縦方向を \$y\$ 成分とし、流量に関してもセルあたりの平均流速を用いた。

進化にともなう評価値の変化を図 14 に示す。1 つの世代において最小の評価値を持つ形状を最適解とし、横軸を世代、縦軸を最良評価値としてプロットした。評価値は 0 に近いほど優れている。当初 0.37 程度あった評価値が、最適化の結果 0.14 まで小さくなっていることが分かる。各世代における最適解が進化によって消えることなく、次の世代に継承されており、最適化が正しく行われていることが分かる。

図 15 に最適化前と最適化後の形状を用いて流体計算を行った結果を \$y\$ 成分速度コンターを用いて示す。(a) は最適化前の形状を用いたもので、右チャンネルほど流速が大きくなっていることが分かる。(b) は最適化後の形状を用いたもので、チャンネルを分離する壁の形状が台形になっていることが分かる。また、流速分布にも変化が表れていることが分かる。最適化前はチャンネルが右側ほど流速の偏りが見られたのに対して、最適化後は中央部のチャンネルの流速が大きくなっている。一方、チャンネル間のばらつきが依然あることが分かる。

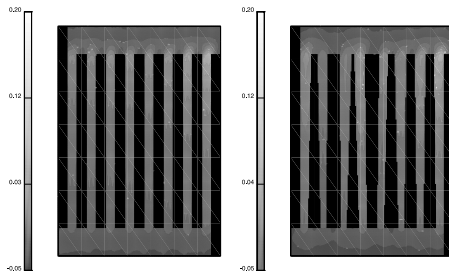


図 15 y 成分流速分布 . (a) 最適化前, (b) 最適化後
Fig. 15 y -component velocity contour. (a) before optimization. (b) after optimization.

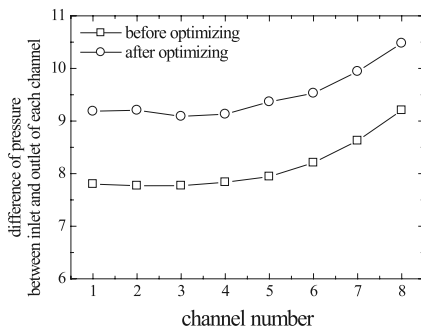


図 16 チャンネル入口・出口間の圧力差 . (a) 最適化前, (b) 最適化後
Fig. 16 Difference pressure between inlet and outlet of each channel. (a) before optimization. (b) after optimization.

今回、壁の厚さを染色体に用いたが、より経験的に最適化が進みやすい方法を模索し、その染色体に適切な交叉のアルゴリズムを用いる必要があると考える。ただし評価値算出を行う流体計算は発散することなく行っていることから、実用上は最適化計算が有効な適用例であることが期待される。

今回の最適化の評価方法に圧力による影響は考慮されていないが、流路設計においては流量均一化とともに、チャンネル入口と出口の間の圧力差の増加を最小限にとどめることが重要である。図 16 に、チャンネル間の圧力差について示す。(a) が最適化前の形状を用いたもので、(b) が最適化後の形状を用いたものである。最適化前に比べて、最適化後の圧力差は流路が狭くなったことで、1.2 倍程度大きくなっている。

流体の最適化問題では、評価値を求めるための計算時間が問題になるが、今回の計算では 1 世代あたりの計算時間は 10 分程度と比較的短時間で得られた。ただし遺伝的アルゴリズムに用いる交叉の条件を複雑にした場合、増大する可能性がある。これに関しては、求める解の精度に応じて、系の大きさや交叉のアルゴリズムなどを調整する必要があるものと思われる。ま

た、並列化を用いることで短縮は可能であると考えられる。

以上、評価値による最適化はおおむね期待したとおりに行われており、完全な 2 次元問題であったとしても、評価関数が小さくなるようなアルゴリズムを適用する限りにおいては最適化計算としては問題はないと考える。

遺伝的アルゴリズムの特徴は、ロバストで大域最適化ができることに特徴がある。一方、実数型格子ガス法は、格子ボルツマン法などの従来の手法に比べると、密度比が著しく異なる 2 つの相が混在するような系においても安定して計算が行える手法である。このような、ロバストなスキームどうしを組み合わせることで、最適化計算あるいは流体計算の適用範囲を拡大できるものと期待する。

6. 結 言

実数型格子ガス法を用いて複雑な形状を扱うための、複雑境界モデルを提案した。複雑境界モデルは、小粒度のセルと再帰関数を併用することにより、煩雑になりやすい境界判定処理を単純化できる。また境界形状を画像ファイルにより設定できるシステムとして実装することが可能である。実験の結果、提案した複雑境界モデルは、強制対流キャビティの差分法による結果、およびチャンネル流路を差分法と格子ガス法で計算した結果と比較することで定性的な妥当性を得た。複雑境界モデルの応用例として、遺伝的アルゴリズムを用いた最適化を行った。本稿では、ハニカム形状を模した吸着装置内をモデル流路とし、流路形状の自動最適化を示すことで提案モデルの汎用性を示すことができた。

参 考 文 献

- 1) 稲室隆二, 小林一暁, 山岡由樹, 荻野文丸: 二相系ボルツマン法による微小隙間内に進入する二流体界面の数値解析, 日本機械学会 2003 年度年次大会講演会論文集 (II), p.283 (2003) .
- 2) 山本和弘: 簡略化したモデルによる多孔質内流れ解析, 日本機械学会 2003 年度年次大会講演会論文集 (II), p.281 (2003).
- 3) d'Humieres, F.U., et al.: Lattice gas hydrodynamics in two and three dimensions, *Complex Systems*, Vol.1, pp.649-707 (1987).
- 4) McNamara, G. and Zanetti: Use of the Boltzmann equation to simulate lattice-gas automata, *Phys. Rev. Lett.*, Vol.61, pp.2332-2335 (1989).
- 5) Malevanets, A. and Kapral, R.: Continuous-velocity lattice-gas model for fluid flow, *Europhys. Lett.*, Vol.44, No.5, pp.552-558

- (1998).
- 6) Malevanets, A. and Kapral, R.: Statistical mechanics of hydrodynamics lattice gases, Thesis for the degree of Ph.D, Dept. of Chemistry, Univ. of Tront (1997).
 - 7) 橋本康弘, 溝上伸也, Chen Yu, 大橋弘忠: 実数型格子ガス法を用いた3次元混相流解析, 流体マイクロシミュレーションの大規模体系への適用研究, 日本原子力学会報告書 (2000).
 - 8) Hashimoto, Y., Chen, Y. and Ohashi, H.: Immiscible real-coded lattice gas, *Computer Phys. Comm.*, Vol.129, pp.56-62 (2000).
 - 9) 日岡英一, 松隅洋介, 井上元, 峯元雅樹: 格子ガスオートマトン法を用いた装置内流動の均一化の試みと並列計算への適応性評価, 化学工学論文集, Vol.29, No.3, pp.421-426 (2003).
 - 10) 大野貴雄, 戸田和之, 山本誠: 最適歯角を有するラビリンスシール内流れに関する数値的研究, 第18回数値流体力学シンポジウム (2004).
 - 11) 陳宏, 大岡龍三, 加藤信介: 対流・放射連成解析と遺伝的アルゴリズム (GA) を用いた建物の最適配置による屋外温熱環境の改善に関する研究, 第17回数値流体力学シンポジウム (2003).
 - 12) Ghia, U., Ghia, K.N. and Shin, C.T.: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method, *J. of Comp. Phys.*, Vol.48, pp.387-411 (1982).
 - 13) 伊集院幸久, 松隈洋介, 峯元雅樹: 格子ボルツマン法を用いた八二カム充填材内の流速分布均一化に関する研究, 機械学会第18回計算力学講演

会, pp.593-594 (2005).

(平成17年11月22日受付)

(平成18年3月4日再受付)

(平成18年8月3日再々受付)

(平成18年8月30日採録)



寺井 優晃 (正会員)

1978年生. 2006年北陸先端科学技術大学院大学情報科学研究科博士後期課程(情報科学)修了. 現在, 独立行政法人理化学研究所情報基盤センターに所属. バイオインフォマティクス, ハイパフォーマンス・コンピューティング分野のシステム運用技術に関心. 博士(情報科学).



松澤 照男 (正会員)

1948年生. 1973年信州大学大学院工学研究科修士課程修了. 同年同大学医学部助手. 1986年沼津工業高等専門学校助教. 1991年北陸先端科学技術大学院大学情報科学センター助教. 1995年同教授. 現在に至る. 数値流体力学, ハイパフォーマンス・コンピューティング, グリッド・コンピューティングに従事. 医学博士. 日本機械学会, 日本流体力学学会等各会員.