

# 流れ関数を用いた二次元多重連結領域上の構造安定な非圧縮流の可視化

2015SE055 永田 翔也 2016SE006 江崎 昂 2016SE031 加藤 晴海

指導教員：横山 哲郎

## 1 はじめに

流体力学は流体の運動を研究対象とする力学の一分野であり、その解析方法の1つに離散解析がある。離散解析には二次元多重連結領域上の流れをトポロジーによって分類するものが知られている。これは流体の構造を大まかな見方で分類するものである。しかし、完全な分類が可能となり、直観的に理解しやすい。離散解析の研究の1つにトポロジーを極大語によって分類する研究がある。語表現により多重連結領域上の構造安定な非圧縮流を記号列で表す方法は区別しにくい流線のトポロジーを分類することに優れている。より多くの流れの向きを考慮した木表現が語表現の発展形としてある。語表現には1つの語表現に対して複数の流線パターンが存在する欠点があるのに対して、木表現は1つの流れのトポロジーに対し1つの記号列を対応させる。語表現よりも高い表現力を持つといえる。

本研究では、二次元多重連結領域上の流れのトポロジーを可視化する手法を提案する。本研究では、木表現から形状の把握をしやすい流線の描画を目指す。与えられた深さに対する木表現の数を上下からそれぞれ抑えられる数を評価する。また、流れの遷移の構造を解析する。

流れを可視化する手法として木表現 流れ関数 可視化の手順を踏む。木表現は既存研究のものを使用する。まず同パターンの流線を可視化に用いないよう木表現の列挙を行い、重複の調査をした。流線の可視化をする際、異なる木が同じ流線パターンを描画する可能性があるためである。次に、木で表した流線の流れ関数で表現する。得られた流れ関数を用いて流線の描画をし、結果を評価した。構造安定な非圧縮流の変換規則を木文法によって定義する。定義された変換規則に従い流れの遷移グラフを作成し、流れの遷移の可視化を目標とする。流れの遷移を可視化することで、流れの遷移を直観的に理解することができる。

## 2 関連研究

### 2.1 前提条件

本研究では多重連結領域上で非圧縮性をもつ構造安定な流れを仮定する。例えば、障害物をもたない領域を単連結領域、障害物をもつ領域のことを多重連結領域という。また、障害物、停留点、及び渦構造の数の和を  $M$  とする。非圧縮性は連続体の密度が変形の前後で変化せず常に一定である性質である。構造安定性は小さな乱れが加わっても構造が変化しない性質で流れの研究においては流れのトポロジーが変化しないことをいう。トポロジーは形を連続変形をさせても変わらない性質のことである。例えば、浮輪と

カップは同じで、浮輪とポールは異なる。

### 2.2 木表現の初期パターン

初期パターンとは、 $M = 0, 1$  の構造安定な流れの初期構造となるもので  $M = 0$  の構造安定な流れはパターン  $I$  とパターン  $II$  の2つがある。また、吸い込み湧き出し対を持たない流れの他に  $M = 1$  となる構造安定なパターン  $O$  がある。以上の3つを初期パターンとする。初期パターンは一樣流  $a_0$ 、反時計回りの流れ  $b_{0+}$ 、時計回りの流れ  $b_{0-}$  を表す。

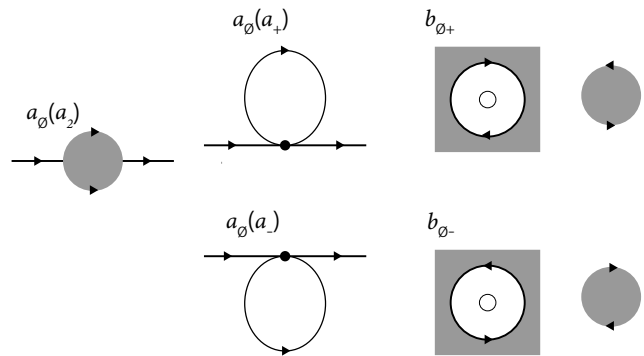


図1 初期パターン

### 2.3 木表現の5つの操作

構造安定な流れの流線に対して  $M$  を1つとそれに伴う流れの構造を追加する操作は  $A_0, A_2, B_0, B_2, C$  の5つのみである。木文法では  $A_0$  を  $a_+, a_-$ ,  $A_2$  を  $a_2$ ,  $B_0$  を  $b_{++}, b_{--}, b_{+-}, b_{-+}$ ,  $B_2$  を  $\beta_+ \{c_+(l, \lambda)\}, \beta_- \{c_-(l, \lambda)\}$  とし、 $C$  を  $c_+(l, \lambda), c_-(l, \lambda)$  と表す。木表現で5つの操作を表した流線を示す。5つの操作は初期パターンの流れや吸い込み湧き出しの有無によって適応できる操作に条件がある。操作  $C$  を適応する場合には操作  $B_2$  が操作  $B_0, B_2$  を適用する場合には操作  $A_0, C$  が存在しなければならない。

### 2.4 木文法

開始記号  $S$ , 非終端記号の集合  $N = \{S, A, A^*, B_+, B_-, C_+, C_-, C_+^*, C_-^*\}$ , 終端記号の集合  $F = F_e \cup F_A \cup F_B \cup F_C \cup \{l, \lambda, cons(,)\}$ , 生成規則  $R$  である。終端記号  $F$  は  $F_e = \{a_0(), b_{0+}(\{\}), b_{0-}(\{\})\}$ ,  $F_A = \{a_+(\cdot), a_-(\cdot), a_2(\cdot)\}$ ,  $F_B = \{b_{++}\{\cdot\}, b_{+-}(\cdot), b_{--}\{\cdot\}, b_{-+}(\cdot), \beta_+\{\cdot\}, \beta_-\{\cdot\}\}$ ,  $F_C = \{c_+(\cdot), c_-(\cdot)\}$  とその要素から生成される木の根にあたる記号の種類で分類でき、それぞれが初期パターン、

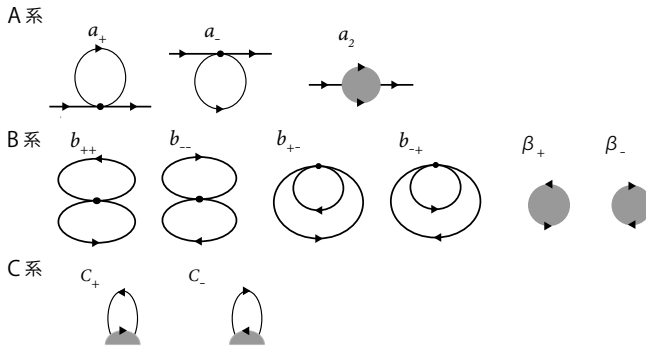


図2 木文法の5つの操作に現れる流線

A系, B系, C系の操作によって現れる流線を表している. 生成規則は以下のように記述する.

$$\begin{aligned}
S &\rightarrow a_0(A^*) \mid b_{0+}(B_+, \{C_+^*\}) \mid b_{0-}(B_-, \{C_+^*\}) \\
A &\rightarrow a_+(B_+) \mid a_-(B_-) \mid a_2(C_+^*, C_-^*) \\
A^* &\rightarrow \lambda \mid \text{cons}(A, A^*) \\
B_+ &\rightarrow l \mid b_{++}\{B_+, B_+\} \mid b_{+-}(B_+, B_-) \mid \beta_+\{C_+^*\} \\
B_- &\rightarrow l \mid b_{--}\{B_-, B_-\} \mid b_{-+}(B_-, B_+) \mid \beta_-\{C_-^*\} \\
C_+ &\rightarrow c_+(B_+, C_+^*) \\
C_- &\rightarrow c_-(B_-, C_-^*) \\
C_+^* &\rightarrow \lambda \mid \text{cons}(C_+, C_+^*) \\
C_-^* &\rightarrow \lambda \mid \text{cons}(C_-, C_-^*)
\end{aligned}$$

### 3 流れの可視化

本章では, 流れの可視化手法について考える. 関連研究で示した流れの初期パターンに5つの操作をした流線を流れ関数で表現し, それを用いて描画を行う. 流れ関数を用いた描画方法として  $\log$  を用いた流れ関数を利用した方法と半球状の流れ関数を利用した方法の2つを考え, 流線の描画に適している後方で描画を行った. さらに可視化をする前段階として木表現の列挙をし重複する流れのパターンを予め取り除くことで, 重複する流れを可視化してしまうことを防ぐ. また, 流れの変換規則について調べる.

#### 3.1 木表現の列挙

流れの構造は再帰的に分解をすることが可能なため木の列挙は木の葉の部分になり得る. B系から2.4節の生成規則に従い開始記号に向けて行う. あるノードで生成される木の数から漸化式で深さ  $n$  のときに生成される木の数を上から抑えることができる. 木の列挙を実際に行うに当たって注意する点がある. 流れの構造が2つ以上の要素を持つ場合に円順序を流線図において同一視することができるならば, その流れの構造に波括弧を用いる. よって, 生成規則に従って生成された木は波括弧内の要素は反転させても同じものとなる. 例として, 流れの構造の要素のラベルを考慮しない場合には  $x, y$  という任意の要素を流れの構造にもつ木表現  $b_{++}\{x, y\}, b_{++}\{y, x\}$  は, トポロジ的には

同じ流れ構造をもつ木となる. 同様に波括弧を用いる  $b_{--}$  と2つ以上の要素を持つ場合には  $\beta_+, \beta_-$  も同様である. この流線図上で流れの構造を同一視できることで列挙した木には重複した木が含まれる. この重複した木を部分木にもつた木もまた重複した木となり木の対称性により同一視の重複もある.

#### 3.2 流線の描画

流線の描画方法として木表現を流れ関数で表現し, それを描画し可視化する手法を用いる. 初期パターンに5つの操作をした流線を半球状の流れ関数で表現し, それを組み合わせた流れを等高線図上に描画する. 等高線図では, 高いポテンシャルほど暖色になり, 低いポテンシャルになるほど寒色になる. 流れの向きは, 右手でより低いポテンシャルに触れながら左手でより高いポテンシャルに触れて進む向きである. 等高線図に流れを描画する際, 流れ関数は渦を表現する場合  $x^2 + y^2 = r$  を使用する. また, 入力値に  $(x + a, x + b, s)$  を用いて  $a, b$  の値を設定して平行移動させたものを使う.  $s$  は渦のスケールとする. 等高線図で流れを表現する際の問題点として, 2つの流れが重なりあう場所で  $z$  軸の値が互いに干渉してしまい歪な渦の形になるというものがあった. その問題の解決方法として MATLAB の  $\max$  関数を使用した. 例えば  $b_{++}$  の中に  $b_{+-}$  がある流れがあったとする.  $b_{++}$  内に  $b_{+-}$  を描画させるため,  $b_{+-}$  の式を  $\max$  関数で表現し, 最大要素とすることで  $b_{++}$  の式を足しあわせても  $b_{--}$  には影響が無いようにした. このように流れの内側に来る流れを最大要素とすることで異なる流れの干渉を防ぐ. また, 5つの操作のA系やC系には障害物を含んだ流れがあり, 等高線図での障害物の表現方法を考える必要がある. 等高線図では決めた範囲の塗りつぶしが難しいため, 今回は障害物として決めたい範囲の  $z$  軸の値を固定し流線の中の空洞になっている部分を障害物として判断する.

次に MATLAB で描画するためのソースコードを次に示す.

```

x=linspace(-10,10);
y=linspace(-10,10);
(X,Y)=meshgrid(x,y);
syms f(x,y,s);
syms R(x,y,s);
syms L(x,y,s);
f(x,y,s)=piecewise(x.^2+y.^2<=s,(x.^2+y.^2)/s,0)
R(x,y,s)=piecewise(x.^2+y.^2<=s,2,0);
L(x,y,s)=piecewise(x.^2+y.^2<=s,0.5,0);
Z=contour(X,Y,Z); % 流れを表現する関数
colorbar;

```

流線を描画する範囲を  $x$  軸と  $y$  軸が  $-10$  以上  $10$  以下とし,  $x, y$  の二次元グリッド座標とする. シンボリック関数  $f, R, L$  を定義する.  $f$  は流線を表す関数  $f(x, y, z)$  とし,  $x$  と  $y$  部分で渦の中心点を表し,  $z$  部分で渦の大きさを表す. 関数  $R$  と関数  $L$  は障害物を表現をする. 初期パターンに5つの操作を行った流線に現れる障害物には必ず障害物に沿う流線が存在する. 関数  $R$  は, 障害物と障害物に右

回りで沿うようにできる流れを表現し、関数  $L$  は障害物と障害物に左回りで沿った流れを表現する。

最後に初期パターンに 5 つの操作を加えた流線を描画できるような式を定義する。A 系の流線の  $a_+$  と  $a_-$  の流線はひとつの渦と一様流で表現が可能であり、描画したものを図 3 と図 4 に示す。渦を半球状の流れ関数、一様流を  $Y./10$  とし足し合わせる。 $a_2$  は障害物に沿った流れと一様流で表現できる。障害物に沿った流線を関数  $R$  で表し、一様流は  $a_+$ 、 $a_-$  と同様に表現する。一様流の  $Y./10$  は  $y$  軸を等間隔に平行に流線が流れている様子であり  $a_2$  を描画したものを図 5 に示す。また、A 系の全ての式を以下に示す。

$$\begin{aligned} Z &= -f(X-0.5, Y-2.4) + Y./10 && \%a+ \\ Z &= f(X-0.5, Y-2.4) - Y./10 && \%a- \\ Z &= R(X, Y, 1) + Y./10 && \%a2 \end{aligned}$$

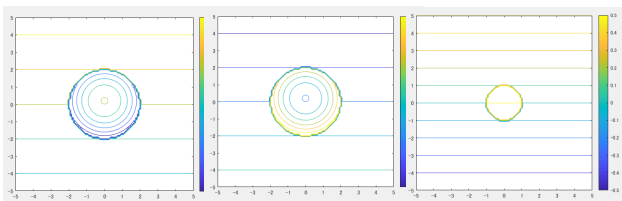


図 3  $a_+$

図 4  $a_-$

図 5  $a_2$

次に B 系の流線を描画する。B 系の流線は 2 つの渦の組み合わせで表現できるため、2 つの半球状の流れ関数を組み合わせる。 $\beta_+$  と  $\beta_-$  は障害物に沿った流線なので関数  $R$  と関数  $L$  で表現可能となり描画したものは図 8 である。B 系の全ての式を以下に示し、 $b_{++}$  を図 6、 $b_{-+}$  を図 7 に示す。

$$\begin{aligned} Z &= f(X-0.5, Y+1.9, 4) + f(X-0.5, Y-1.9, 4) && \%b++ \\ Z &= f(X-0.5, Y+1.9, 4) - f(X-0.5, Y-1.9, 4) && \%b-- \\ Z &= \max(f(X, Y-2.1, 1.0), 0) - f(X, Y-3, 4) && \%b+- \\ Z &= \max(f(X, Y-2.1, 1.0), 0) + f(X, Y-3, 4) && \%b-+ \\ Z &= L(X, Y, 1) && \%beta+ \\ Z &= R(X, Y, 1) && \%beta- \end{aligned}$$

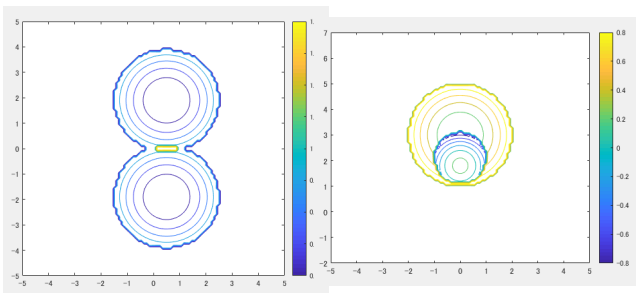


図 6  $b_{++}$

図 7  $b_{-+}$

C 系の流れは障害物に沿うように出来る流線と渦の組み合わせで表現できる。C 系の式を以下に示し、描画したものを  $c_+$  は図 9、 $c_-$  は図 10 とする。

$$Z = \max(R(X, Y, 1), f(X, Y+1.9, 4)) \quad \%c+$$

$$Z = \max(L(X, Y, 1), f(X, Y+1.9, 4)) \quad \%c-$$

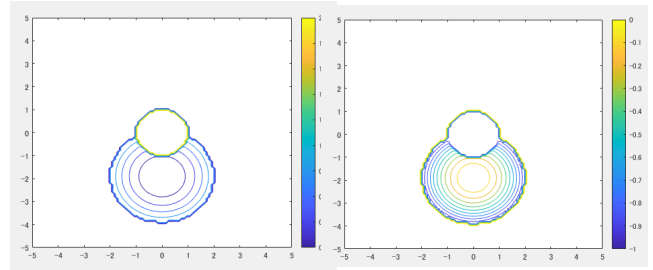


図 8  $c_+$

図 9  $c_-$

定義した A 系と B 系と C 系をもとに  $a_0(\text{cons}(a_+(\beta_+\{c_+(l, \lambda)\}), \lambda))$  を表す関数をつくる。その関数に位置情報を与え、 $a_0(a_+(\beta_+\{c_+(l, \lambda)\}))$  をうまく描画できるように調整する。

$$\max(R(X, Y+4, 1), f(X, Y+5.9, 4)) + f(X, Y+4, 18) + Y \quad (1)$$

式 1 をもとに流線を描画したものが図 14 となる。これで

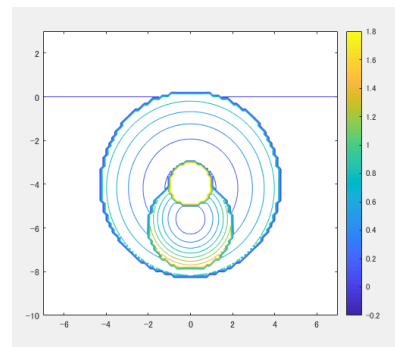


図 10  $a_0(a_+(\beta_+\{c_+(l, \lambda)\}))$

log を用いた描画の問題点である渦同士を足し合わせると歪んでしまう点を解決した流線を描画に成功した。渦の一部分が重なりあう流線の場合には、 $\max$  関数を使用することで流線の最大値を表示させている。これらから A 系、B 系、C 系をひとつずつ含む流線を関数化させ描画することが可能ということがわかった。さらに、流線の要素を増やしたとしても上記の手順の応用で描画可能と予想できる。

### 3.3 流れの遷移

閉円板上の構造安定な非圧縮流の変換規則を木文法によって定める。本節は停留点の個数が保存されるようなトポロジーの変化のみを考えているので、停留点同士が結合や分裂により停留点の数が増減する場合は考えない。しかし一般に、閉円板上の非圧縮流の遷移は、停留点同士が結合すること、または異なる 2 つの停留点を繋ぐ軌道が結合することで起こる。流れのトポロジーを変化させるには、ある構造安定な流れを構造安定でない流れに変化した後、他の構造安定な流れにしなければならない。内部に境界をもたない有界な流れの変換規則は 4 つ挙げられる。本節で

は図 11 の変換規則 1 のみを使う．トポロジーの変化のみを考えるため，ノードを流線図，エッジを遷移とした流れの遷移グラフを作成する．

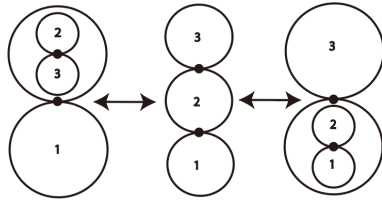


図 11 変換規則 1

各ノードをラベリングした場合は，図 12 のようにノード数が 3 の完全グラフとなり，任意の流線図からどの流線図へも 1 つのエッジで遷移できる．1 つのノードから出るエッジの数は 2 である．図 13 のグラフは，ノード数が 15，任意の流線図から任意の流線図に最大 3 つのエッジで遷移できる．1 つのノードから出るエッジの数は 4 になる．図 12，図 13 において 1 つのノードから出るエッジの数はそれぞれ全て等しい．

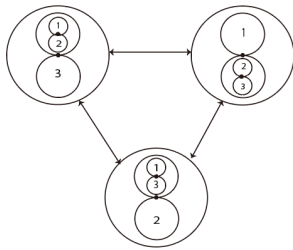


図 12 ノード数 3 のラベリングされた流れの遷移グラフ

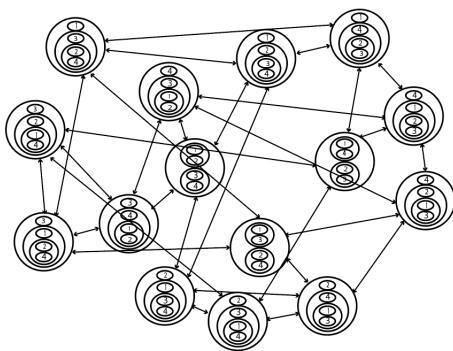


図 13 ノード数 4 のラベリングされた流れの遷移グラフ

各ノードをラベリングしない場合は，ノード数 3 と 4 の流線図はそれぞれ図 14 と図 15 の構造のみである．ラベリングをしない場合，流線図中でノード数が 3 から 4 に 1 つ増えると構造の違う流線図が 1 つ増えた．同様の方法で B 系の流線図に他の変換規則を用いることで，性質や法則を見つけることが期待できる．

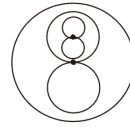


図 14 ノード数 3 のラベリングされていない流線図

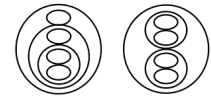


図 15 ノード数 4 のラベリングされていない流線図

#### 4 おわりに

木表現から流れの可視化するための手法として木表現を流れ関数で表現し，その流れ関数をもとに可視化する方法を考えた．木表現の 5 つの操作に現れる流線を全て流れ関数で表現し，その関数を元に描画をするという手順で描画を行う．描画には MATLAB を使用した．まず，渦を log 関数で表し描画する方法を考えたがうまく描画することができなかった．1 つの渦の描画は可能だったが 2 つ以上の渦を同時に描画する際に問題が生じたからである．別の描画方法として，半球状の流れ関数を利用する方法を考え，木表現の 5 つの操作に現れる流線を生成規則に従って組み合わせたものであれば全て描画可能となった．また，等高線図を利用したことにより  $z$  軸の値の変化で流線の色が変化し，流線の向きや A 系 B 系など異なる要素の流れ構造が視覚的に理解しやすくなった．木の列挙からあるノード  $n$  で生成される木の数を漸化式によりある程度抑えることができた．流れの変換規則を木文法によって定義した．さらに，流れの遷移グラフを作成したことにより， $b_{++}\{l, b_{++}\{l, l}\}$  と  $b_{++}\{b_{++}\{l, l\}, b_{++}\{l, l\}\}$  の遷移を可視化することができた．作成した遷移グラフを元に性質や法則，ある流線図から最も遠いパスで遷移する流線図を探した．

#### 参考文献

- [1] Sakajo, T. and Yokoyama, T.: Transitions between streamline topologies of structurally stable Hamiltonian flows in multiply connected domains, *Physica D*, Vol.307, pp.22–41 (2015).
- [2] 加藤 舞，内藤綾香：多重連結領域上の安定非圧縮流の解析，南山大学 2018 年度卒業論文 (2019) .
- [3] 坂上貴之，横山知郎，澤村陽一：二次元多重領域内における構造安定な非圧縮流の文字列表現アルゴリズム，数理解析研究所講究録，Vol.1900，pp.11–25 (2014) .
- [4] 内藤綾香，加藤 舞，横山哲郎ほか：円板上の非圧縮流の反転の解，情報処理学会 第 81 回全国大会講演論文集，pp.319–320 (2019) .
- [5] 横山哲郎，横山知郎：ハミルトン曲面流に対応する流れの向きを考慮した極大語の列挙アルゴリズム，電子情報通信学会論文誌 D，Vol.J101-D，No.8，pp.1220–1222 (2018) .