二次元多重連結領域内における構造安定な非圧縮流れの木表現の可視化手法

2016SE024 亀谷拓磨　2016SE076 田島嘉人　2016SE090 渡辺康平
指導教員：横山哲郎

1　はじめに

1.1　流体力学の背景

流体力学は，気体や液体の運動について取り扱う力学の主要な研究分野である．流体力学の歴史は古く，この学問が確立される以前から人々の生活に即したものとして発展してきた．近年では，コンピュータの発達により流体方程式から数値解を求め流れを精密に計算することが可能となった．この手法は数値解析と呼ばれ，事故や災害などの実験不可能な課題も計算できる利点を持った近年の主要な流体力学の解析手法である．

1.2　流体の離散解析

離散解析は流体解析の一手法であり，前節で述べた数値解析に対照的である．離散解析は流れをトポロジカルな観点から解析する．トポロジカルな観点に着目することは大域的な構造に着目するということを意味し，これにより流れの本質的な構造を抜き出すことが可能となる．つまり流れの大枠に着目した解析を行いたい場合，離散解析を用いることで効率的に流れの解析を進められるということである．この解析法の代表的なアプローチとして，流体をその構造安定性に着目して解析する方法がある．構造安定性は，力学系に小さな乱れが加わっても流れのトポロジーが変化しない性質のことである．トポロジーは位相幾何学の用語であり，この立場に立てば，連続的に変形できる図形は同じ形である．例えば，四角形と三角形は同じ形である．構造安定性は一般の流体にはあり得ないが，文献 [1] によれば，有限の制度や有限の時間における挙動に着目したり，本質的な構造に着目するなどの現実的な目的の範囲内であれば，現実の流体にもこの考え方が適応できる．構造安定性に着目すれば，構造安定性から流れの変化を考察することができるようになる．例えば，語表現の研究では流れのトポロジーに対して対応した文字列を定義することで，二つの流体構造の中間構造を文字列の変化として考察することが可能となった [2].

1.3　本研究の目的

本研究は，離散解析手法の一つである木表現に対して，図上への可視化手法を与える．ここで，木表現には種類がいくつか存在するが，本研究で扱う木表現は [3] によって与えられた手法である．また，以下で木表現とはこのアルゴリズムを指す．木表現は流線構造を代数的に扱い流れの解析を行うことができるが，その表現から直感的に二次元上の流れの形状を把握することは困難である．そのため，二次元の形状を得たい場合は解析者がその都度木表現を組み合わせて図化することになるが，木表現が複雑になればなるほどその変換も煩雑になり，ともすれば途中で間違った変換を行ってしまうこともあり得る．本研究は木表現を図に自動的に変換する方法を与えることで，解析者の効率的で確実な木文法による流れの解析に寄与するものである．

2　関連研究

2.1　語表現の研究

本研究で扱う木表現は，語表現の研究を発展させたものである．語表現の研究では，流線の位相的な構造を分類し，それぞれの流線構造を系統的に文字列で表現するアルゴリズムを与えた [2]．以下では，語表現とは [2] によって与えられたアルゴリズムを指すこととする．語表現を用いた例として，翼の揚抗比の時間変化を語表現によって表した研究がある [4]．語表現は，有界な多重連結領域上で非圧縮かつ非粘性で構造安定な流れを代数的に表すことができる．多重連結領域とは複数の障害物が含まれている領域のことである．ここでは非圧縮・非粘性という理想流体を仮定しているが，この仮定は現実の流体に対しての直接の適用に制限を与える．しかし，現実への応用に関しては文献 [4] でその適用方法が考えられている．流線構造を文字列で表現することで，流線構造を数学的に厳密に分類できるようになり流れの変動を特徴付けて捉えることができるようになる．また，代数的に扱うことができるため，流線の構造の特徴を説明するための共通言語として用いることもできる．文献 [2] によって与えられた語表現には同じ流線に複数の語表現を与えることができるという問題があった．そのため，文献 [5] によって自然な語表現を与えるアルゴリズムが与えられた．また，文献 [6] によって流れの向きを考慮した場合の語表現を与えるアルゴリズムが与えられた．

2.2　木表現の研究

木表現もその前提条件は語表現と共通であり，有界な多重連結領域上で非圧縮性かつ非粘性で構造安定である．木表現はその特性から，語表現より細かい流れの分類を可能とする．そのため流れの特徴を語表現より，多く捉えることを期待されている．実際，円盤状の非圧縮流の反転の解析を木表現によって行った研究では，語表現と比べた木表現の表現力の高さが確かめられた [7].

2.3　その他の離散解析手法

語表現および木表現と関連のある離散解析手法として，コンレイ・モース分解，グラフクラスタリングがある [1]．これらの手法は，流体に対してそれぞれ異なるアプローチを持つが，流体をその構造安定性に着目して解析するという点では一致している．

2.4　図を描画する手法

2.4.1　グラフ描画アルゴリズム

グラフ描画アルゴリズムは，コンピュータによって作成されるグラフの視覚性を向上するための技術である．一般的なグラフ描画アルゴリズムは，あるグラフを形成する点と線それぞれに対し力学的な力を計算することで二次元上での位置関係を見やすいものに変更する．この技術は多くの場合グラフのレイアウトを整える際に使用されるが，それ以外にも文献 [8] のように，見た目を整えたい図をグラフとしてモデル化しグラフ描画アルゴリズムを適用することができる．

3　自動可視化への準備

3.1　可読性の定義

作成するプログラムには，木表現で作成されたトポロジーを確実に図に再現することが求められるが，さらに，再現されたトポロジーは可読性が高いものである必要もある．本研究では，可読性の高さを定義する上でグラフ描画アルゴリズムにおける可視性の高さの基準 [9] を参考にプログラムに求められる可読性を定義した．

- 線が重ならない
- 線の間が適切な距離を保つ
- 線が滑らかである

3.2　文法の定義

本研究では，既存の木文法の研究である [3]，[7] を参考にしている．そのため以降の詳細については [3] によって提案された木文法を取り扱う．木文法 G=(S，N，F，R) は以下のように定められる．S は開始記号，V は非終端記号の集合，F は終端記号の集合，R を生成規則とする．このとき，N= { S，A，B_+，B_-，C_+，C_-，C_+^*，C_-^*，} ，F = { $a_\phi()$，$b_\phi(,\{\})$，$b_{\phi-}(,\{\})$，$a_+()$，$a_-()$，$a_2()$，$b_{++}\{,\}$，$b_{+-}\{,\}$，$b_{--}\{,\}$，$b_{-+}\{,\}$，$\beta_+\{\}$，$\beta_-\{\}$，$c_+(,)$，$c_-(,)$，l，λ，cons(，)} とする．

流れの生成規則　R

S → $a_\phi\ (A^*) \mid b_{\phi+}\ (B_+,\{C_-^*\}) \mid b_{\phi-}\ (B_-,\{C_+^*\})$
A → $a_+(B_+) \mid a_-(B_-) \mid a_2(C_+^*, C_-^*)$
A^* → l | cons(A ,A^*)
B_+ → l | $b_{++}(B_+,B_+) \mid b_{+-}\ \{B_+,B_-\} \mid \beta_+\ \{C_+^*\}$
B_- → l | $b_{--}(B_-, B_-) \mid b_{-+}\ \{B_-,B_+\} \mid \beta_-\ \{C_-^*\}$
C_+ → $c_+(B_+,C_-^*)$
C_- → $c_-(B_-,C_+^*)$
C_+^* → λ | cons(C_+,C_+^*)
C_-^* → λ | cons(C_-,C_-^*)

図 1　文法規則

3.3　構文解析

本研究では，python での実装を行うため python 専用の構文解析ライブラリ PLY を利用し構文解析を行った．PLY とは，Python Lex-Yacc の略であり，python 専用の構文解析ライブラリである．lex.py では，入力されたテキストを正規表現により定義されたデータ構造に分解し，字句解析を行う．その後，yacc.py により，自由文法によって定義された構文を評価して抽象構文木を作成し構文解析を行う．

3.4　デザインパターン

ソフトウェア設計者の経験から同じような問題は，典型的に同じパターンの解決策になることが発見され，それらのパターンをカタログ化したものをデザインパターンという．オブジェクト指向において，様々なプログラムで利用でき，目的に応じて 23 種の中から選択して扱うことができる．主にオブジェクト指向言語で再利用性が高いクラスやライブラリを作成する際に利用される．本研究では，デザインパターンの中のインタプリタパターンを取り扱う．インタプリタパターンについては，[10] を参考にした．インタプリタパターンは，形式的に記述された記号列を解析した結果に則って処理したい場合に利用されるデザインパターンである．目的に応じた言語を作成することで，通訳の意味を果たすプログラムを用意することにより，素早く処理することが可能になる．再帰的な構造を持ち，全ての要素に共通な処理を持つ抽象クラスを定義することで構造の変更を容易にしたり，見た目をシンプルにすることが可能である．本研究では，木文法より定義された生成規則によって，字句解析と構文解析を行い，構文木を作成する．そのため，構文木に則った処理を実現するのに最適なインタプリタパターンを利用した．また，作成するクラスが多くいため，オーバーライドを利用することで機能の追加が容易になり，修正を減らすことができる．

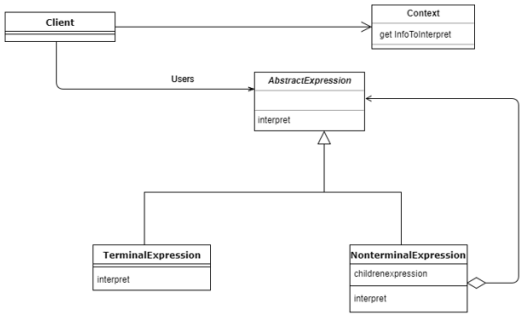

図 2 インタプリタパターンクラス図

3.5 アプローチ

本研究の目的の達成には，3.1 の定義を満たすトポロジーな流れの図を二次元上の画面に表示することを必須としている．アプローチとしては，コンピュータのディスプレイ上に作成した図を可視化し，画像ファイルとして保存することを可能にするプログラミング言語を本研究で使用した．私達は様々な分野での実績があり多様な環境での信頼性が高い python とベクタ形式の描画を提供し LaTeX への組版が可能な Asyptote の 2 つプログラミング言語をアプローチ方法とした．

3.6 Asymptote

コンピュータ上で図や画像を表現する際には，ビットの集合体で表現するラスタ形式と，数式で作成した図形の集合体で表現するベクタ形式の大きく分けて，２つの表現方法が存在する．ラスタ形式は一つ一つのビットで図を構成しているので，写真などの複雑な画像を描画する際は適しているが，図の拡大縮小を行うと配置にずれが生じたり，ジャギーが発生し画質が落ちてしまう．それに対し，ベクタ形式は拡大縮小に合わせて数式を変更すれば容易に行え，且つ画質が落ちることもない．画質の劣化は本研究の目的に支障をきたすので，ベクタ形式で描画する必要がある．Asymptote はベクタ形式のプログラミング言語であり，他のグラフィックプログラムと比べても高品質のスクリプトと GUI の機能を活用できるので，流体の図を作成する際にも適している．

4 Asymptote による実装

Asymptote では，木表現に対しての 3.1 の定義を満たす図を作成するためのライブラリを提供し，実際に存在する流体をトポロジー的に再現することができれば本研究の目的を満たすことを示す．

4.1 実行方法

Asymptote ではプログラムを実行する方法として，Ubuntu の端末や Windows のコマンドプロンプト上に直接コードを記述する interactive mode(対話モード) という方法が存在する．そこでライブラリを読み込み，画面上に表示させたい流線図に対応する木表現を記述することで，画面上にトポロジー化された流線図を表示させる．

4.2 実行例

本研究では非圧縮流体を題材として扱っているが，トポロジー的な流れの図を描画できるかは，非圧縮流体でなくても実証は可能である．さらに，非圧縮流体以外の実際に存在する流れを例題に示すことで本研究の多様性が高まるなどが期待ができる．本研究では，カルマン渦を実例として挙げた．カルマン渦については，文献 [11] を参照した．

図 3 カルマン渦

ある範囲の速度の流れの中に円筒状の障害物が存在すると，その後流側に規則正しく左右交互に並んだ 2 列の渦が発生するこの渦をカルマン渦と呼ぶ．一部のカルマン渦をトポロジー的に表した図を Asymptote で再現する．

図 4 トポロジー化したカルマン渦

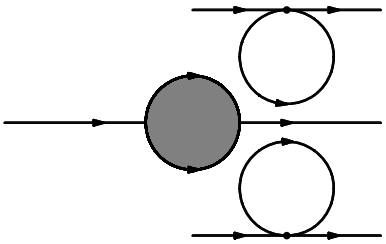

図 5 作成したカルマン渦

カルマン図は a 系のみで構成され，2 つ a-と 1 つの a2 で再現した．

4.3 提供する関数

void ap(real d,pair s)
void am(real d,pair s)
void a2(real d, pair s)
void bpp(real d, pair s)
void bmm(real d, pair s)
void bp(real d, pair s)
void bm(real d, pair s)

void bpm(real d, pair s)
void bmp(real d, pair s)
void cp(real d, pair s)
void cm(real d, pair s)

以上が木文法の生成規則に準じた主な流れの関数である．だが，描画する際に描画対象の図によって流れの向きを調節する必要があるため，図が右向き (Right) か左向き (Left)，上向き (Up) か下向き (Down) の頭文字を関数の語尾につける．例えば，右向きかつ上向きの a+ の流れを描画するには void apRU(real d,pair s) となる．real 型 d は流線の大きさを表し，pair 型 s で流線の座標を合わせる．これらの関数を組み合わせ，パラメータを調節することにより目的の流れを描画できる．

4.4 python による実装

python で流線図を描画するにあたり，python 用のグラフ描画ライブラリである matplotlib を使用した．

5 おわりに

5.1 python

現在検討し作成している手法により，この先プログラムを C 系に対し拡張したとしても 3.1 により定められた定義を満たす図を製図できると考える．しかし，C 系の流れにはループが存在し，その部分の要素は無限に増やすことが可能である．そのため，要素数や要素の種類に合わせ柔軟に生成される図を変更できることが求められる．これを解決するためグラフ描画アルゴリズムを利用することを考えているが，まだ，実装には至っていない．今後は，C 系の描画手法を考え実装し，作成したプログラムの有用性に関して実証していきたい．

5.2 Asymptote

ライブラリを提供することで，ある範囲の中での流れの図の作成を支援することは可能であるが，Asymptote では本研究の目的である入力を完全な木文法で描画することが現状ではできなかった．字句解析を対応させ他の言語内で抽象的データ型にすることで可能になると考察した．今後は，描画の効率や画質の質を上げるために改良を施していきたい．

参考文献

[1] 荒井迅．トポロジカルな流れ構造の理解へ向けて．ながれ：日本流体力学会誌, Vol. 33, No. 1, pp. 23–28, feb 2014.

[2] T Yokoyama and T Sakajo. Word representation of streamline topologies for structurally stable vortex flows in multiply connected domains. *Proc. Roy. Soc. A*, Vol. 469, No. 2150, pp. 1–18, 2013.

[3] 横山哲郎, 横山知郎．多重連結領域上の非圧縮流を表す木文法の深化．プレプリント．

[4] 坂上貴之, 横山知郎, 澤村陽一．二次元多重連結領域内における構造安定な非圧縮流れの文字列表現アルゴリズム．数理解析研究所講究録, Vol. J101-D, pp. 11–25, 2014.

[5] 横山哲郎, 横山知郎．ハミルトン曲面流に対応する語の列挙アルゴリズム．電子情報通信学会論文誌 D, Vol. 100, No. 10, pp. 892–894, 2017.

[6] 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. 101, No. 8, pp. 1220–1222, 2018.

[7] 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.

[8] 丸山貴志子, 谷崎正明, 嶋田茂．デフォルメマップ生成のための道路形状正規化モデルとそのシステム評価．電子情報通信学会論文誌. A, 基礎・境界 = The transactions of the Institute of Electronics, Information and Communication Engineers. A, Vol. 87, No. 1, pp. 108–119, jan 2004.

[9] 土井淳, 伊藤貴之．力学モデルを用いた階層型グラフデータ画面配置手法の改良手法とウェブサイト視覚化への応用．芸術科学会論文誌, Vol. 3, No. 4, pp. 250–263, 2004.

[10] 結城浩．Java 言語で学ぶデザインパターン入門．Java 言語で学ぶデザインパターン入門. SB クリエイティブ, 2001.

[11] Morten Brøns, Bo Jakobsen, Kristine Niss, Anders V Bisgaard, and Lars K Voigt. Streamline topology in the near wake of a circular cylinder at moderate reynolds numbers. *Journal of Fluid Mechanics*, Vol. 584, pp. 23–43, 2007.