

FreeFem++ と Paraview による非線形現象の可視化 —反応拡散系と流体力学への応用—

大森 克史・山口 範和

Visualization of Nonlinear Phenomena with FreeFem++ and Paraview —Application to Reaction-Diffusion System and Fluid Dynamics—

Katsushi OHMORI and Norikazu YAMAGUCHI

E-mail: ohmori@edu.u-toyama.ac.jp; norikazu@edu.u-toyama.ac.jp

概要 様々な自然現象は非線形偏微分方程式により数理モデル化される。現時点では非線形偏微分方程式に対する万能な方法はない。従って、数値計算は非線形偏微分方程式の解の様相を理解する手段として有効な方法の一つである。本稿では、有限要素法による数値解析の為にオープンソフトウェア FreeFem++ を用いた非線形偏微分方程式の数値計算と可視化ソフトウェア Paraview を用いた数値解の可視化について述べる。

キーワード: 非線形偏微分方程式, 可視化, 有限要素法, FreeFem++, Paraview

Keywords: Nonlinear Partial Differential Equations, Visualization, Finite Element Method, FreeFem++, Paraview

1. はじめに

本稿では、有限要素法による数値解析の為にオープンソフトウェア FreeFem++ を用いて非線形偏微分方程式に対する数値計算を行い、その結果を Paraview というソフトウェアを用いて可視化する方法を述べる。また、幾つかの具体的な非線形問題に対して実際に数値計算および可視化を行う。

本論に入る前に、背景や既存の手法について述べる。

1.1. 数値計算の必要性

様々な自然現象は「偏微分方程式」を用いて数理モデル化される。例えば、流体の運動における Navier-Stokes 方程式や K-dV 方程式、形態形成における Gierer-Meinhardt 系、化学反応におけるオレゴネータ、Gray-Scott モデル、古典力学やゲーム理論における Hamilton-Jacobi 方程式等、例を挙げれば枚挙に暇がない。

多くの場合、現象のモデル方程式として得られる偏微分方程式は「非線形」である。現時点では全ての非線形偏微分方程式に対して有効な統一的な理論はない為、数学的には各論的に扱わざるを得ない。

数理物理等に現れる微分方程式の研究では「適切性」という概念が重要である。適切性は Hadamard によって導入された概念であり、与えられたデータ（初期条件や境界条件）の下での問題に対する (i) 解の存在, (ii) 解の一意性, (iii) 解のデータ連続依存性の三つを総称した概念である*1。

*1 例えば、非圧縮性粘性流体の運動を支配する基礎方程式である Navier-Stokes 方程式においては、運動エネルギー有界な任意の初期速度に対する時間大域的適切性はかれこれ 80 年近く未解決問題であり、Clay 数学研究所の指定する 21 世紀における 7 つ

何らかの手段により、問題の適切性や解の漸近的なプロファイルを純粋数学的な方法によって得ることが出来たととしても、モデル方程式が支配する現象の全てを数学的に理解できる訳ではない。特に、解析学的な手法は何らかの極限的な状態の理解には有効に働くが、中間帯域における状況での取り扱いには不向きである。例えば、反応拡散系と呼ばれる非線形偏微分方程式の解析ではパターン形成が一つの大きなテーマであるが、パターンが形成される過程や与えたデータに対して形成されるパターンの詳細を把握する事は数学的な方法だけでは極めて難しい。

そこで「数値計算」が必要となる。現時点では数値計算は非線形偏微分方程式の解を目に見える形にするほとんど唯一の方法と言って良い。

1.2. 微分方程式の数値計算

微分方程式の解を計算機で厳密に計算することは出来ない。何故ならば計算機では極限操作を扱うことが出来ない為である。

そこで、微分方程式を数値的に解くには微分方程式における「微分」を四則演算で表現し直す必要がある。そのような定式化を「離散化」という。離散化手法には様々な方法があり、代表的な方法としては有限差分法 (FDM)、有限要素法 (FEM)、有限体積法 (FVM)、境界要素法 (BEM) 等がある。扱う問題によって得手不得手があり、万能な方法はない。何れにせよ、離散化を行えば微分方程式は代数方程式（多くの場合は連立 1 次方程式）へ近似的に帰着される。従って、そうした代数方程式の解を数値

の未解決問題の一つである。詳しくは以下の URL を参照。
<http://www.claymath.org/>

的に求めることが出来れば、微分方程式の近似解が求まる。本論文では空間変数の離散化には有限要素法を用いる。更に、時間発展問題を数値的に解く場合には時間変数の離散化には差分法を用いる。

さて、微分方程式を何らかの方法により離散化し、得られた代数方程式の解（数値解）が求められたとしよう。得られた数値解は元の微分方程式の近似解である。従って、精度良く近似された問題に対して得られた数値解を調べれば、元の微分方程式の解の様相をおおよそ知ることが出来る。しかし、注意しなければならないのは数値計算の結果として得られる「数値解」とは単なる数値の羅列に過ぎない事である。微分方程式の数値計算では得られる数値解のみを眺めていても近似解の様子を把握する事は難しい。数値の羅列でしかない数値計算結果を分かり易いものにする手続きが「可視化」である。可視化により、数値の羅列に過ぎない数値解は初めてグラフのように目で見て理解が可能な形となる。

1.3. 有限要素法と FreeFem++

有限要素法は偏微分方程式を数値計算する際に強力な離散化手法であり、偏微分方程式を直接扱うのではなく、その弱形式を用いた離散化を行う方法である。楕円型偏微分方程式に対する変分原理に基づく方法^{*2}に馴染みがあれば、その原理は理解し易い。弱形式を元にしていて、関数解析との相性が良く、数学的にも整備されている方法である。

有限要素法は元々は構造解析において生まれた計算手法であるが、近年では構造解析以外の分野、例えば流体力学等への応用も盛んに研究がなされている（例えば、[32, 11]）。有限要素法は差分法と比べた場合に、複雑な形状の領域における問題を面倒な座標変換等をする事なく取り扱う事が出来る事、Neumann 型の境界条件の扱いが易しい事、等の利点がある。

有限要素法には菊地 [21, 22]、田端 [42]、Ciarlet [4]、Brenner & Scott [2]、Quarteroni & Valli [33] 等定評のある入門書が数多く出版されている。では、こうした入門書で有限要素法の仕組みを理解したら、直ぐにでも有限要素法により数値シミュレーションが出来るかと問われれば、答えは否である。有限要素法に限らず、数値計算法の仕組みを理解する事とそれを元にした数値計算を実際に行う事の間には大きな隔りがある。例えば、C や FORTRAN でプログラムを作成するのであれば、これらのプログラミング言語に関する知識が必要であるし、計算を行う事が出来るようになったとしても、可視化という別の手続きが残っている。

非線形現象の数値計算を行いたい研究者や学生は必ずしも数値計算の専門家である訳ではない。従って、非線形現象そのものに興味がある者にとって重要な事は数値計算プログラムの開発に割く時間をなるべく少なくし、確

実な数値計算を行えるようになることである。その為には、自前でプログラムを作成するよりも最先端の数値計算ツールを使う方が良い。そうしたツールには数値計算の専門家たちが長年の研究で蓄えてきた様々なノウハウが組み込まれている為である^{*3}。

本稿で扱う FreeFem++ はフランスのパリ第 6 大学の J.-L. Lions 研究所の数値解析グループによって開発されている有限要素法により偏微分方程式を数値的に解くためのオープンソフトウェアあり、正に上で述べたような発想から生まれたソフトウェアである^{*4}。実際に、FreeFem++ を用いれば C や FORTRAN といったプログラミング言語に精通していなくとも、有限要素法による数値シミュレーションを行うことが出来る。

FreeFem++ は有限要素法による流体解析の第一人者である Pironneau 教授によって開発が始められた。実際、[32, Appendix] には FreeFem++ の前身である MacFEM という古い Macintosh 上で動作する有限要素解析ソフトウェアが紹介されている。そこでは開発の動機も述べられているが、まさしく上で述べたような事であり、それから長い時間をかけて改良を重ねられたものが現在の FreeFem++ である^{*5}。

FreeFem++ は C++ に似たプログラミング言語であり、こうした言語を少しでも学習したことがあれば慣れるのは容易い。FreeFem++ には次のような特徴がある。

- 空間 2 次元、空間 3 次元の偏微分方程式を扱うことが出来る。
- 問題は弱形式の形そのまま表現することが出来る為、わざわざ弱形式を基にした連立 1 次方程式を導出する必要がなく、数学的な記述をそのまま用いる事が出来る。
- 連立 1 次方程式に関する GMRES, CG 法, Crout 等の高機能なソルバーが複数用意されており、実際に連立 1 次方程式を数値計算するにはソルバーを指定するだけで済む。
- 計算メッシュの生成は Delaunay-Voronoi アルゴリズムにより自動的に行われる。また、アダプティブ・メッシュを用いた計算等も組み込みの命令を利用する事で簡単に実現可能である。
- $P1, P2, P3, P4, P1b, P1dc, P1nc, RT0$ 等、複数種類の有限要素を利用することが出来る^{*6}。
- 可視化機能も備えている^{*7}。
- Windows/Mac OS X/Linux といった代表的な OS で動作するマルチプラットフォームソフトウェアである。

^{*3} 精度保証付き数値計算の第一人者である大石によれば、欧米の大学では Matlab 等の数値計算ツールを使う事がもはや常識になっているようである [29]。

^{*4} <http://www.freefem.org/ff++/> から入手可能。

^{*5} 現在の開発責任者はパリ第 6 大学の Hecht 教授に移っている。

^{*6} 詳しくは [14, Chapter 6] を参照

^{*7} 但し、本稿では FreeFem++ に組み込みの可視化機能は使わない。

^{*2} 例えば、Brezis [3, Chapter 8,9] を参照。

こうした特徴から、FreeFem++ を用いることで有限要素法による数値計算プログラムの作成に必要な時間は大幅に削減する事が出来る*8。

FreeFem++ の詳細についてはマニュアル [14] を参照されたい。また、日本ではつい最近、大塚・高石 [28] による解説書が出版された。本稿と同じく流体力学や反応拡散系に対する数多くの応用がコードと共に掲載されているので合わせて参照されたい。Hecht [13] や Sadaka [34] にも各種の応用が掲載されている。

なお、FreeFem++ 用の IDE として FreeFem++-cs*9がある。開発からプログラム実行までを一つのソフトウェアで行うことが出来る (図 1.1)。コマンドラインでの計算機の取り扱いに不慣れな場合は FreeFem++-cs を用いるとよい*10。

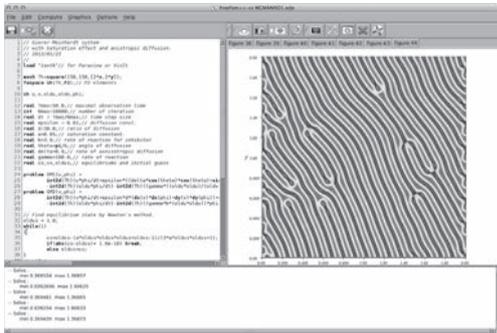


図 1.1: FreeFem++-cs の操作画面

1.4. Paraview と可視化

既に述べたように FreeFem++ には組み込みの可視化機能があるため、数値解を単に可視化するだけならば特別な操作も道具も必要ない*11。しかし、FreeFem++ の可視化機能では得られるグラフィクスを細かく制御するのは大変である。また、時間発展問題に対する動画を作成すると、そのような機能は FreeFem++ には用意されていない為、別のソフトウェアに頼らざるを得ない。そこで、本稿では FreeFem++ は原則的に計算の為だけに用いることにし、可視化は Paraview というソフトウェアを用いて行う。これには、一つの計算データから複数種類の可視化を試すことが出来るようになる、動画の作成が可能となる等の利点がある。

Paraview は米国商務省の ASCII Views 計画による資金提供を受け、2000 年にアメリカのロスアラモス国立研究所と Kitware Inc. の共同プロジェクトとして開発が開

始されたソフトウェアであり、名称は Parallel View Processor から来ている。2005 年には Paraview 3.0 の開発が Kitware 株式会社とサンディア国立研究所、CSimSoft の共同プロジェクトとして開発が始められた。現在では Paraview 4.1 がリリースされている*12。

Paraview は BSD ライセンスによるフリーソフトウェアであるが、商用のソフトウェアである AVS*13や TecPlot*14等と比べても遜色ないほどに高品質な可視化が可能であり、高機能なフィルターを用いて様々な可視化表現を試したり、動画を作成したりすることが出来る。また、FreeFem++ と同様に Windows/MacOS X/Linux の全てで動作するマルチプラットフォームなソフトウェアでもある。

Paraview は様々な形式のデータを取り扱うことが可能であるが、ここでは VTK*15形式のデータを用いる。FreeFem++ では iovtk というパッケージを読み込む事で、VTK 2.0 形式でのデータ保存を簡単に行うことが可能であり、Paraview との相性が良い。

本稿では、数値計算結果を可視化するという事に要点を絞って Paraview の操作方法を説明する。Paraview の基本的な使用法は林 [12] を参照して頂きたい。



図 1.2: Paraview の操作画面

1.5. 本論文の目的および構成

本稿の目的は FreeFem++ を用いて非線形偏微分方程式の数値計算を行い、得られる数値解を Paraview を用いて可視化する一連の方法を紹介することである。実際に、反応拡散系と流体力学に現れる幾つかの具体的な非線形偏微分方程式に対する数値計算を行い、その結果を紹介する。

本稿の構成は以下の通りである。

第 2 節では楕円型偏微分方程式の典型例の一つである Poisson 方程式の境界値問題を題材に、FreeFem++ による数値計算結果を Paraview により可視化する方法の基本を述べる。特に、どのようにして計算データを保存し、計算データを Paraview 上でどのように扱えば良いかを

*8 実際に、プログラミング言語を用いてプログラムを組むことに比べると書かなければならないコードの量は 1/20 程度の量で済む。

*9 <http://www.ann.jussieu.fr/~lehyaric/ffcs/> から入手可能。

*10 FreeFem++-cs には FreeFem++ がバンドルされているので、この IDE を用いる場合は FreeFem++ を別に入手する必要はない。但し、最新版に比べてバンドルされているものが古い可能性はある事を注意しておく。

*11 plot 命令を使うだけでよい。

*12 <http://www.paraview.org/> から入手可能。

*13 <http://www.avs.com/>

*14 <http://www.tecplot.com/>

*15 Visualization Tool-Kit の略。Paraview の開発を行っている Kitware Inc. により提供されている。[12, 付録 A] 等を参照。

詳述する。

第3節では放物型偏微分方程式の典型例の一つである線形拡散方程式を題材に時間発展問題の扱い方を述べる。特に計算結果をアニメーションとして出力する方法を述べる。

第4節および第5節では非線形現象の例として反応拡散系により記述される現象と非圧縮性粘性流体の平面的な運動を扱う。

第4節では形態形成に関する Gierer-Meinhardt 系と呼ばれる反応拡散系を用いた Turing パターンの数値的再現を行う。更に、BZ 反応のモデル方程式の一つであるオレゴネータを用いて、BZ 反応に特有のスパイラルパターンを数値的に再現する。

第5節では流体现象に固有な移流項の取り扱い方を簡単に説明し、Navier-Stokes 方程式の初期値・境界値問題を数値的に解くことで、障害物に一樣流がぶつかる場合の物体後方における Karman 渦列の生成を可視化する。また、複雑な流れ問題の例として気液二相流に対するレベルセット法による数値計算結果を紹介する。

なお、本稿に掲載した数値計算結果は 3.4GHz Intel Core i7 プロセッサ、メモリ 32GB を搭載した Appli 社製 iMac (Late 2012) および 2.8GHz Intel Core i7 プロセッサ、メモリ 16GB を搭載した Apple 社製 MacBook Pro (Late 2013) を用いて行ったものである。

紙数の都合もあり、第2節から第5節では数値計算の収束性や安定性等についての説明は割愛した。

1.6. 幾つかの注意

本稿では、FreeFem++ 3.30 および Paraview 4.1 を用いた。これらはいずれも本稿執筆時点での最新の安定版である。Paraview の操作画面のスクリーンショットは MacOS X で動作する Paraview を用いて作成したものである。Windows や Linux 上で Paraview を使用した場合にはメニューの配置等に多少の差異があるが、本質的には変わらない。

FreeFem++ の各ソースコードには行番号を付しているが、行番号はすべて論理行の番号であり、物理行の番号ではない事に注意されたい。

2. 数値計算と可視化の基本 (1) —Poisson 方程式を例に—

本節では、楕円型偏微分方程式の典型例である Poisson 方程式の境界値問題を題材に FreeFem++ で数値計算を行い、計算結果を Paraview で可視化する方法を述べる。

2.1. Poisson 方程式の境界値問題

まずは、ある程度一般的な条件の下で問題を定式化しよう。 $\Omega \subset \mathbb{R}^2$ を有界領域とする。その境界 $\partial\Omega$ は区分的に滑らかであり、 Γ_D と Γ_N に分けられているとしよう。即ち、 $\partial\Omega = \Gamma_D \cup \Gamma_N, \Gamma_D \cap \Gamma_N = \emptyset$ とする。このとき、次の

Poisson 方程式の境界値問題を考える。

$$-\Delta u = f, \quad \mathbf{x} \in \Omega, \quad (2.1a)$$

$$u = b, \quad \mathbf{x} \in \Gamma_D, \quad (2.1b)$$

$$\frac{\partial u}{\partial \mathbf{n}} = q, \quad \mathbf{x} \in \Gamma_N. \quad (2.1c)$$

ここで、 $u = u(\mathbf{x}) = u(x, y)$ が未知関数であり、 $f = f(\mathbf{x}), b = b(\mathbf{x}), q = q(\mathbf{x})$ は既知関数 (データ) である。 Δ は \mathbb{R}^2 における Laplacian である。また、(2.1c) で $\mathbf{n} = (n_1, n_2)$ は境界 Γ_N 上での単位外法線ベクトルであり、

$$\frac{\partial u}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla u = n_1 \frac{\partial u}{\partial x} + n_2 \frac{\partial u}{\partial y}, \quad \mathbf{x} \in \Gamma_N \quad (2.2)$$

は Γ_N 上の法線方向の微分を表わす。以下では、 $\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right)$ で u の勾配 (gradient) を表わす。

Poisson 方程式は数理物理の様々な場面に登場する。例えば、それ自身が静電場におけるスカラーポテンシャルを記述する。また、(2.1a) は熱伝導方程式

$$\frac{\partial u}{\partial t} - \Delta u = f(\mathbf{x}) \quad (2.3)$$

の解 $u = u(t, \mathbf{x})$ のうちで時間変数 t に無関係なものを考えていると見做せば、解は Ω 上の熱伝導の定常状態を与える。同様に波動方程式の定常状態と見做せば、 Ω 上に張られた膜の形状を表しているとも捉えられる。これら以外にも、(2.1a) については様々な物理的な解釈を与えることが出来る。

境界条件について述べておく。(2.1b) は、 Γ_D 上での関数 u の値が関数 b によって決定されるという境界条件である。このようなタイプの境界条件は Dirichlet 境界条件と呼ばれる。特に、有限要素法の理論では基本境界条件と呼ぶ。(2.1c) は、 Γ_N で関数 u の法線方向の微分が q によって決定されるという境界条件である。これは境界を通した物質の流出入に対する境界条件であり、このようなタイプの境界条件は Neumann 境界条件と呼ばれる。特に、有限要素法の理論では自然境界条件と呼ぶ。既に述べたように有限要素法では Neumann 境界条件の扱いは差分法と比べて易しい。

2.2. 弱形式の導出

有限要素法は弱形式に基づく離散化手法であり、既に述べたように FreeFem++ では弱形式化した問題を元にプログラムを作成する。そこで、境界値問題 (2.1) に対する弱形式を導出しよう。

$\varphi \in \{\varphi \in C^\infty(\Omega) \mid u|_{\Gamma_D} = 0\}$ を試験関数として、(2.1a) の両辺に φ をかけて Ω 上で積分する。Gauss の発散定理より、

$$-\int_{\Gamma_N} \frac{\partial u}{\partial \mathbf{n}} \varphi d\sigma + \int_{\Omega} \nabla u \cdot \nabla \varphi dx = \int_{\Omega} f \varphi dx \quad (2.4)$$

を得る。 Γ_N 上での境界条件 (2.1c) に注意すれば

$$(\nabla u, \nabla \varphi) - (f, \varphi) - \langle q, \varphi \rangle = 0 \quad (2.5)$$

を得る^{*16}. これが (2.1) に対する弱形式である. ここで, (\cdot, \cdot) で Ω 上の L^2 内積を, $\langle \cdot, \cdot \rangle$ で Γ_N 上の L^2 内積を表わした^{*17}.

あとは, この弱形式に基づいて離散化を行えば, (2.1) に対する有限要素法による近似問題を作ることが出来る. C や FORTRAN を用いて自前でプログラムを作成する場合には (2.5) から解くべき連立 1 次方程式を導かなければならないが, FreeFem++ ではそうした部分を気にする必要はない. 偏微分方程式と対応する弱形式さえ求めれば, 後はそれを元に簡単にプログラムの作成を行うことが出来る.

2.3. FreeFem++ によるインプリメンテーション

境界値問題 (2.1) において Ω, f, b, q を具体的に与えて, 数値解を FreeFem++ で計算してみよう.

例として領域 Ω は単位円盤 $\Omega = \{x \in \mathbb{R}^2 \mid |x| < 1\}$ とし, その境界は $\Gamma_D = \{x \in \partial\Omega \mid y \geq 0\}$, $\Gamma_N = \partial\Omega \setminus \Gamma_D$ とする. また, データ f, b, q については, $f(x, y) = \sin(3\pi(x+y))$, $b(x, y) \equiv 0$, $q(x, y) \equiv 0$ とする.

有限要素空間 $U_h \subset H^1(\Omega)$ を区分的 1 次要素の空間とすると, FreeFem++ により (2.1) の境界値問題の数値解を計算するプログラムのソースコードは以下のようになる.

Program 2.1: Poisson.edp

```

1 // BVP of Poisson eq.
2 load "iovtk"
3
4 border C1(s=0, pi){x=cos(s);y=sin(s);}
5 border C2(s=pi, 2*pi){x=cos(s);y=sin(s);}
6 mesh Th=buildmesh(C1(50)+C2(50));
7 fespace Uh(Th, P1);
8 Uh u, phi;
9
10 solve Poisson(u, phi) =
11   int2d(Th)((dx(u)*dx(phi)+dy(u)*dy(phi)))
12   - int2d(Th)(sin(3*pi*(x+y))*phi)
13   - int1d(Th, C2)(0*phi)
14   + on(C1, u=0);
15 // plot(u, wait=1, value=1);
16 savevtk("Poisson.vtk", Th, u, dataname="u");

```

以下, Program 2.1 の詳細を説明する.

- 1 行目はコメントである. C/C++ 等と同様に // 以降は行コメントとして扱われる. なお, /* */ で囲んだ部分はブロックコメントとして扱われる.
- 2 行目は VTK 形式のファイルの出力を行うために必要なパッケージの読み込みを行っている. 可視化を Paraview で行う場合には VTK ファイルでの出力が必要となるので, 2 行目は Paraview を併用する場合のおまじないだと思えば良い.
- 4 行目, 5 行目では単位円のパラメータ表示 $x = \cos s, y = \sin s$ ($0 \leq s < 2\pi$) を用いて境界を定義し

ている. C1 が Γ_D , C2 が Γ_N に対応する.

- 6 行目が 4 行目, 5 行目で定義した境界を用いたメッシュの生成と定義である. Th は生成するメッシュの名前であり, 名前は自由に付けてよい.
- 7 行目が有限要素空間 U_h の定義であり, \mathcal{T}_h 上の区分的 1 次要素 (P1 要素) の空間を考えている. 区分的 2 次要素を使う場合は P1 を P2 に書き直すだけで良い. 第 1 節でも述べたように, 他にも様々な要素を使うことが出来る.
- 8 行目は $u, \varphi \in U_h$ の宣言である.
- 10 行目から 14 行目で, Poisson 方程式の境界問題 (2.1) を弱形式により表現し, その近似問題を解くような指示を与えている.
 - `int2d(Th)(...)` は \mathcal{T}_h 上での積分であり, 11 行目が $(\nabla u, \nabla \varphi)$ に対応し, 12 行目が $-(f, \varphi)$ に対応する.
 - 13 行目の `int1d(Th, C2)` は境界 Γ_N 上での線積分であり, Neumann 境界条件を表現している. いまの場合, $q = 0$ であるから, この行はあってもなくても関係がないが, (2.5) の表現と合わせるために敢えて明示的に無意味な行を入れてある.
 - 14 行目が Dirichlet 境界条件であり, 境界 Γ_D 上で $u = 0$ としている部分に対応する.
- 15 行目のコメントアウトを外せば, FreeFem++ 組み込みの可視化機能を用いて数値解のグラフが描かれる.
- 16 行目では計算結果を `Poisson.vtk` という名称で保存している. 第 1 引数は保存するファイル名, 第 2 引数はメッシュ, 第 3 引数はデータ, 第 4 引数はデータの名称である.

Program 2.1 を `Poisson.edp` という名称で保存し^{*18}, ターミナル上で以下のようにすれば, 数値計算が実行される^{*19} ^{*20}.

```
% FreeFem++ Poisson.edp ↩
```

実行が上手く行けば, `Poisson.vtk` という VTK ファイルが生成されているはずである.

試しに生成された VTK ファイルの中身をテキストエディタで開いてみると, 以下の様になっている^{*21}.

```

1 # vtk DataFile Version 2.0
2 Poisson.vtk, Created by Freefem++

```

^{*18} 拡張子 `edp` は *Équation aux Dérivées Partielles* の頭文字から来ている. これはフランス語で偏微分方程式 (Partial Differential Equations) を表わす言葉である.

^{*19} `Poisson.edp` が保存されているディレクトリにワーキングディレクトリを移動して操作を行うこと. もし, `FreeFem++` という命令が見つからないというエラーが出る場合は, `FreeFem++` の実行ファイルがあるディレクトリにパスが通っているかに注意されたい.

^{*20} `FreeFem++-cs` を用いる場合は, プログラムを保存し, メニュー `Compute` から `Run` を選べば良い.

^{*21} 実際にはこの計算結果のファイルは 4,000 行以上ある. 実はこれは少ない方で, 実際に本稿で扱った非線形問題の数値計算では各計算ステップで生成されるファイルが 30MB 程度あり, 1 回のシミュレーションで数 GB から数十 GB のディスク容量を要する.

^{*16} 実際には無理に (左辺) = 0 の形に書き直す必要はないが, FreeFem++ ではこのような表現を用いて方程式の弱形式を考えるので, ここでもそのような形を採用した.

^{*17} 以下でも同様の表現を用いる.

```

3 ASCII
4 DATASET UNSTRUCTURED_GRID
5 POINTS 947 float
6 -0.9921147 -0.12533323 0
7 -1 1.2246468e-16 0
8 -0.99802673 -0.06279052 0
9 -0.99802673 0.06279052 0
10 .....

```

1行目から5行目までが VTK ファイルのヘッダ部分である。3行目ではファイルが ASCII コードで書かれている事を宣言している。4行目ではデータが非構造格子で書かれていることを明示しており、5行目では節点数が全部で 947 点あり、データは実数型であることを明示している。6行目以下が数値解のデータである。

冒頭でも述べたように、数値計算の結果得られる数値解とはこのような数値の羅列に過ぎないので、これを分かりやすい形に変換する必要がある。その手段として Paraview を用いる。

2.4. Paraview による可視化

2.4.1. VTK ファイルの読み込み

Paraview を起動し、Poisson.vtk を読み込み、Apply ボタンを押す^{*22}。次に、Coloring で u を選ばば (図 2.1) , カラースケールを用いて数値解が可視化されるが見栄えが悪い。そこで、以下では読み込んだデータにフィルターを適用し、見栄えが良くなるように加工する。

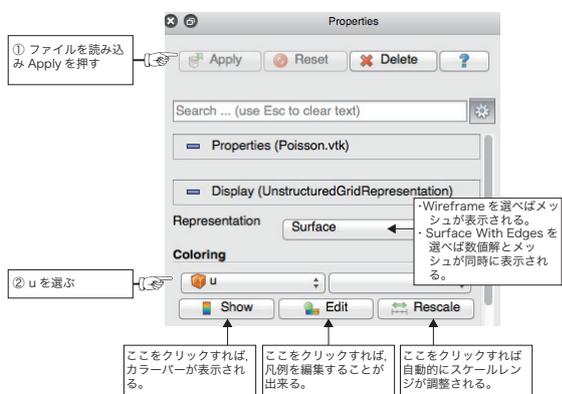


図 2.1: 読み込んだデータの Properties

2.4.2. 幾つかのフィルターの適用

フィルターを用いることで、計算結果から様々なスタイルでの可視化が可能になる。ここでは幾つかのフィルターを用いて、Poisson.vtk より数値解 u_h の可視化を行う。なお、ここで紹介するもの以外にも様々なフィルターがある。詳しくは、林 [12] を参照されたい。

■ **Cell Data to Point Data** まずは、画面上部のメニューで Filter→Alphabetical と進み^{*23}、Cell Data to Point Data を選び適用する。これにより、周りの点の値を用いた平均化がなされる為、数値解が滑らかになる (図 A.1(a)) .

^{*22} 実際には毎回 Apply を押すのは面倒だという場合は、設定画面の “Auto Apply” にチェックを入れておくと良い。

^{*23} このメニューには Paraview で使用可能な全てのフィルターが表示される。

また、画面左部の Pipeline Browser にはフィルターの適用により、CellDatatoPointData1 という新たなデータが加わる^{*24}。Pipeline Browser の各データ左に表示される目のアイコンは濃く表示されているものが現在の Layout に用いられているデータであり、半透明のものは現在の Layout に表示されていないことを表わす (図 2.2) .



図 2.2: Pipeline Browser

■ **Warp By Scalar** Pipeline Browser で上で作成した CellDatatoPointData1 を選び、フィルター Warp By Scalar を適用する。これにより、3D のグラフが作成される。このままでは z 方向の変動が小さすぎるので、WarpByScalar1 の Properties において Transforming における z 成分の値を操作する (図 2.3) .

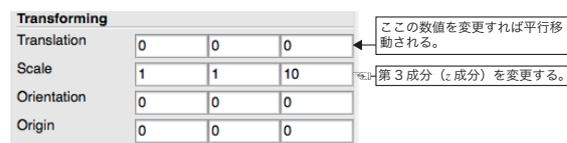


図 2.3: Transforming

実際に、作成された 3D のグラフを表示するには Layout の表示を 3D 表示に切り替えれば良い。計算メッシュと得られた 3D のグラフを同時に描いたものが図 A.1(b) である。

■ **Contour** 数値解をもとに等高線を描く。CellDatatoPointData1 にフィルター Contour を適用する。Contour By で u を選ばば数値解の等高線が作成される。描く等高線の値や本数は Isosurfaces で自由に設定することが出来る (図 2.4) .

3. 数値計算と可視化の基本 (2) —熱伝導方程式を例に—

本節では線形の熱伝導方程式 (拡散方程式ともいう) を題材に、時間発展問題の扱いを述べる。

3.1. 熱伝導方程式

$\Omega \subset \mathbb{R}^2$ を区分的に滑らかな境界 $\partial\Omega$ をもつ有界領域とする。 Ω において Fourier の熱伝導方程式の初期値・境

^{*24} 名称は自由に変更する事が出来る。

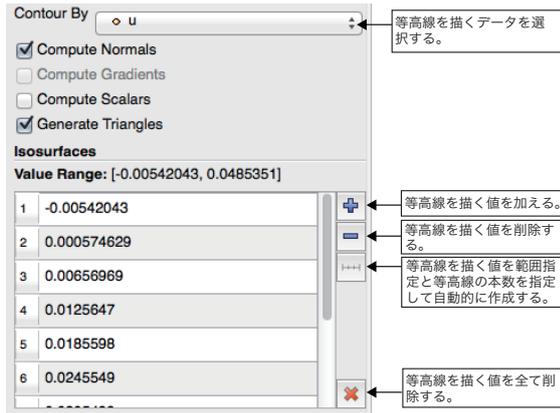


図 2.4: Contour

界値問題を考えよう。

$$\frac{\partial u}{\partial t} = d\Delta u + f, \quad \mathbf{x} \in \Omega, 0 < t \leq T, \quad (3.1a)$$

$$u = 0, \quad \mathbf{x} \in \partial\Omega, 0 < t \leq T, \quad (3.1b)$$

$$u(0, \mathbf{x}) = a(\mathbf{x}), \quad \mathbf{x} \in \bar{\Omega}. \quad (3.1c)$$

$u = u(t, \mathbf{x})$ は温度分布を表す未知関数; $a = a(\mathbf{x})$ は時刻 $t = 0$ での温度分布, $f = f(t, \mathbf{x})$ は熱源であり a, f は共に既知とする. $d > 0$ は媒質の熱拡散率を表す定数であり, $T > 0$ は系の最大観測時刻である. 境界条件 (3.1b) は単純の為, 斉次 Dirichlet 条件とした. 即ち, $\partial\Omega$ 上での温度は一定温度 $u = 0$ に保たれている.

3.2. 離散化

(3.1a) は時間発展方程式であるから, 数値計算をするには空間変数のみでなく, 時間変数も離散化する必要がある. 我々は時間離散化に関しては有限差分法を用いる.

最初に, 時間方向についてのみ半離散化を行う. $N \in \mathbb{N}$ を時間方向の分割数として任意に固定する. このとき, $\tau = T/N$ によって時間ステップサイズを定め, $n = 0, 1, \dots, N$ に対して, $t_n = n\tau$ とおく.

$u = u(t_n, \mathbf{x})$ を (t_n, \mathbf{x}) における真の解とし, $u^n = u^n(\mathbf{x})$ を時間方向のみ離散化した半離散近似とする. (3.1a) 左辺の u の t 変数に関する偏導関数を後退 Euler 近似を用いて近似をすると, 以下の偏微分・差分方程式を得る.

$$\frac{u^{n+1}(\mathbf{x}) - u^n(\mathbf{x})}{\tau} = d\Delta u^{n+1}(\mathbf{x}) + f(t_n, \mathbf{x}), \quad \mathbf{x} \in \Omega, n = 0, 1, \dots, N-1 \quad (3.2)$$

ここで, 第 n ステップにおける u^n が既知であるすれば, u^{n+1} を求めるには, 以下の楕円型偏微分方程式を解けば良い.

$$\tau^{-1}u^{n+1} - d\Delta u^{n+1} = \tau^{-1}u^n + f(t_n, \mathbf{x}) \quad (3.3)$$

(3.3) で右辺が既知であるとするれば, その弱形式化は前節で述べた Poisson 方程式に対する方法とほとんど変わらない. 実際, $\varphi \in C_0^\infty(\Omega)$ を試験関数とすると, 弱形式は

$$\tau^{-1}(u^{n+1}, \varphi) + d(\nabla u^{n+1}, \nabla \varphi) - \tau^{-1}(u^n, \varphi) - (f(t_n, \cdot), \varphi) = 0 \quad (3.4)$$

となる.

u^0 は初期状態 a に他ならないので, n に関する繰り返し処理により順次 u^n を $n = N$ となるまで計算すれば良い.

3.3. FreeFem++ による数値計算

以下の Program 3.1 は (3.1) で $\Omega = (0, 1) \times (0, 1)$, $f = 0$, $a = 16x(1-x)y(1-y)$, $d = 0.1$ として数値計算を行うものである.

Program 3.1: Heat.edp

```

1 // Linear Heat eq.
2 load "iovtk"
3
4 int N=100;
5 real T=1.0;
6 real tau=T/N; // time step size
7
8 mesh Th=square(100,100);
9 fespace Uh(Th,P2);
10 Uh u,oldu,phi,f;
11
12 real d=0.1;
13 f=0;
14
15 problem Heateq(u,phi) =
16   int2d(Th)(u*phi/tau+d*(dx(u)*dx(phi)+dy(u)
17   *dy(phi)))
18   -int2d(Th)(oldu*phi/tau)
19   -int2d(Th)(f*phi)
20   +on(1,2,3,4,u=0);
21
22 int n=0;
23 oldu=16*x*(1-x)*y*(1-y); // initial data
24 savevtk("Heat"+n+".vtk",Th,oldu,dataname =
25 "Temperature");
26
27 for(n=1;n<=N;n++){
28   Heateq; // Solve (3.4)
29   savevtk("Heat"+n+".vtk",Th,u,dataname =
30 "Temperature");
31   oldu=u; // Refresh states
32 }

```

- 8 行目ではメッシュの生成を行っているが, 長方形領域の場合には `square` が利用できる. ここでは各辺を 100 個の点を用いて分割している.
- 10 行目における `u` と `oldu` がそれぞれ u^{n+1}, u^n に対応する. メモリを節約する為に, n の増加に伴い古い情報は状態更新 (29 行目) を行うことで捨てる.
- 15 行目から 19 行目が初期値・境界値問題 (3.1) に対応する問題である. 19 行目が境界条件である.
- 22 行目で初期条件 $u^0 = a$ を与えている.
- 25 行目から 29 行目で (3.4) に対する近似問題を繰り返し解き, その都度計算結果を VTK ファイルとして保存する. 28 行目で各 n に対して `Heatn.vtk` が保存される. 全ての計算が終われば, `Heat1.vtk, ..., Heat100.vtk` が生成される. 文字列の連結には `+` を用いている事に注意して欲しい.

3.4. Paraview によるアニメーションの作成

まずは, Program 3.1 を実行して得られる Heat0.vtk から Heat100.vtk までの 101 個のファイルを Paraview で開く*25. Paraview 操作画面上部にある Play ボタンをクリックすれば, Paraview 内で連続的にデータの内容が表示される.

3.4.1. アニメーションの作成

計算結果をアニメーションとして保存する方法を述べる*26. 最初に Animation View で Mode を Sequence に変更し, No. Frames でフレーム数を決定する (図 3.1). 今の場合には 101 にすれば良い*27.

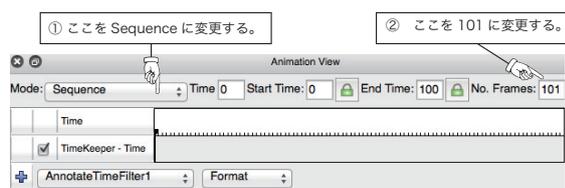


図 3.1: Animation View における操作

次にメニューの File から Save Animation を選ぶ. すると図 3.2 のようなダイアログが表示されるので, 適当に設定をしてから Save Animation ボタンをクリックする. 後は保存するファイル名と動画の形式を選んで保存すれば計算結果がアニメーションとして保存される.

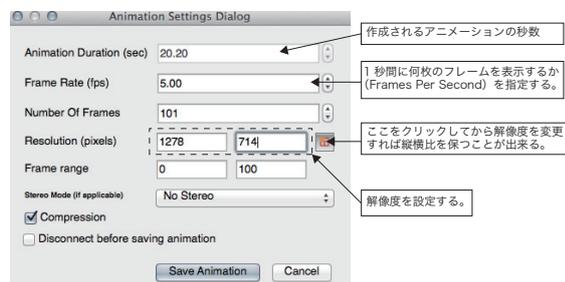


図 3.2: アニメーション設定のダイアログ

3.4.2. 時刻の表示

計算ステップ n や時間 t_n をアニメーションに表示する場合にはフィルター Annotate Time を用いる. このフィルターを適用して現れる Annotate TimeFilter1 の Properties で Format や Scale を適宜変更すれば良い. Format は特に何も設定しなくても良いが, Scale は行った数値計算に合わせておくべきである. 最大観測時刻 T を (No. Frames-1) で割った数を入れておけばよい.

なお, 挿入する文字列には LaTeX による制御綴りを利用する事も出来る. 実際に図 3.3 では表示される数値の前に " $t_n =$ " という文字列を表示するようにしてある.



図 3.3: Annotate TimeFilter

4. 反応拡散系

4.1. 反応拡散系とは

次の形の非線形偏微分方程式を反応拡散系 (Reaction-Diffusion system) という.

$$\frac{\partial \mathbf{u}}{\partial t} - D\Delta \mathbf{u} = \mathbf{f}(\mathbf{u}), \quad \mathbf{x} \in \Omega, t > 0. \quad (4.1)$$

$\Omega \subset \mathbb{R}^N$ は N 次元の領域. $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$ が未知関数である. $D = \text{diag}(d_1, \dots, d_n)$ は拡散係数行列であり, $d_j > 0$ である. 右辺の \mathbf{f} は \mathbf{u} に関する n 次元ベクトル値関数であり, 反応項と呼ばれる.

上に挙げた反応拡散系は半線形放物型偏微分方程式系に分類される. 冒頭でも述べたように偏微分方程式の研究では適切性が一つの基本的な問題となるが, 半線形の反応拡散系に関しては小さい初期データに対する時間大域的な適切性, または任意の大きさの初期データに対する時間局所的な適切性は Laplacian の分数中あるいは, 半群 e^{tA} に対する L^p - L^q 評価等を用いて証明が出来る場合が多い. また, 特別な反応項を持つ系であれば比較原理等を援用することで, 任意の大きさの初期データに対して時間大域的適切性すら証明できる場合すらある. こうした観点からの数学解析については, 例えば, Henry [15] や Smoller [39] を参照されたい.

適切性とは全く別の観点から反応拡散系は非常に面白い非線形偏微分方程式である *28. 熱伝導方程式 (3.1) の解を眺めていると, 拡散は空間非一様な状態を空間一様な状態へと移す効果を持っているように見える. 実際に, 線形の熱伝導方程式ではそのようになる. しかし, 非線形となると話は別である.

1952 年に Turing は拡散が空間非一様な状態を促進する効果を持っていることを示した [46]. Turing は今日では拡散誘導不安定性 (Turing 不安定性とも言う) と呼ばれる概念を用いて形態形成の説明を試みたが, 形態形成分子 (モルフォゲン) の存在が確認されていなかった当時, 彼の理論は数学者による机上の空論に過ぎないとして, 当時の生物学者たちには受け入れられる事はなかった. Turing の発見からおよそ 40 年後に Kondo & Asai [23] はタテジマキンチャクダいの体表に現れる模様が Turing の理論を用いなければ説明できないことを観察実験によって明らかにした.

また, Turing と同じ 1952 年に Hodgkin & Huxley は拡散は波を作り出す効果を持っているという Turing とは

*25 Paraview のファイルダイアログを用いれば簡単に複数のファイルを同時に開くことが出来る.

*26 ここで述べる方法はアニメーションを細かく制御する一つの方法に過ぎない.

*27 データ数が多い場合は適宜変更するとよい.

*28 詳しくは, 三村 [26] を参照されたい.

別の拡散のパラドックスを示した [16]. 彼らはヤリイカを実験材料し、孤立した興奮パルス信号が波形を崩さずに一定の速度で神経軸索を伝わっていく仕組みを理解する為に、神経膜を出入りするイオンに関して一つの仮説を提唱し、その考えに基づき数理モデルを導出した. 詳細は省くが、得られたモデルは反応拡散方程式と非線形常微分方程式の系であり、Hodgkin-Huxley モデルとして知られている*29.

当時は形を保ったまま伝播する波（進行波）を記述するには波動方程式で表現しなければならないと考えられていたが、彼らは得られた数理モデルを数値計算することで、波動方程式でなくともそのような解を持ち得ることを示したのである.

以下では、形態形成に関する Gierer-Meinhardt 系と BZ 反応の数理モデルであるオレゴネータの数値解を FreeFem++ で求め、その結果を Paraview により可視化する.

4.2. Gierer-Meinhardt 系と拡散誘導不安定性

Gierer-Meinhardt 系はヒドラの形態形成を説明する為に [9] で導入された反応拡散系である. 以下では、飽和効果を取り入れた以下の Gierer-Meinhardt 系 ([9], [38] を参照せよ) の数値解を見る.

$$\frac{\partial u}{\partial t} = \varepsilon \Delta u + \gamma \left(-u + \frac{u^2}{v(1+au^2)} \right) \quad (4.2a)$$

$$\frac{\partial v}{\partial t} = \varepsilon d \Delta v + \gamma b(u - v) \quad (4.2b)$$

(4.2) は活性・抑制型の反応拡散系である*30. $u = u(t, \mathbf{x})$, $v = v(t, \mathbf{x})$ はそれぞれ活性因子, 抑制因子の濃度を表わす未知関数である. $\varepsilon > 0$ は u に対する拡散係数であり, $d > 0$ は u と v の拡散係数の比である. 以下では $d \gg 1$ とする. 即ち、抑制因子の拡散速度は活性因子のそれと比べて十分に速いとする.

a, b, γ は全て正の定数であり, $a > 0$ が飽和効果の強さ*31, b は活性反応と抑制反応の反応率の違い, γ は反応のレートを制御するパラメータである. γ は数値計算の都合で入れてあるパラメータである.

拡散誘導不安定性 (Turing 不安定性) とは、反応のみの系*32では平衡点 (u_s, v_s) は漸近安定であるが、拡散が入ることにより漸近安定性が崩れ不安定化する事をいう. この不安定化により平衡点 (u_s, v_s) に対して与えた微小摂動は時間発展に伴い増大し、Turing パターンを形成する (詳しくは三村 [26] 等を参照されたい).

反応拡散系の数値計算を行い拡散誘導不安定性のようなものを見たい場合には無闇にパラメータを決定してもそれを数値的に再現する事は難しい. 例えば、(4.2)

であれば、 $d \gg 1, \gamma > 0$ を固定した際に、どのようにパラメータ a, b を選べば良いかある程度は見積もっておく必要がある. その為には、Turing 空間と呼ばれるパラメータ空間を調べておくと良い. そのようなパラメータ空間は (4.2) の平衡点周りでの線形化問題として得られる無限次元の常微分方程式において、少なくとも一つの固有値の実部が正となるようなパラメータの組のなす集合である. 実際に、(4.2) に対して数値計算を行えば $d = 30.0, \gamma = 100.0$ の場合には図 4.1 が得られる.

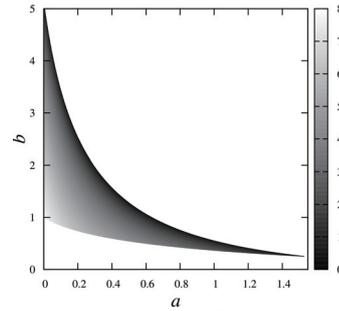


図 4.1: $d = 30.0, \gamma = 100.0$ の場合のパラメータ空間

4.2.1. 幾つかの Turing パターンの形成

まずは、 $\Omega = (0, 2) \times (0, 2)$ として数値計算を行う. 境界条件は u, v 共に斉次 Neumann 境界条件を課す. (4.2) を時間方向にのみ半離散化したものとして、次を採用する. 但し、 τ は時間ステップサイズを表わす.

$$\frac{u^{n+1} - u^n}{\tau} = \varepsilon \Delta u^{n+1} + \gamma \left(-u^{n+1} + \frac{(u^n)^2}{v^n(1+a(u^n)^2)} \right), \quad (4.3a)$$

$$\frac{v^{n+1} - v^n}{\tau} = \varepsilon d \Delta v^{n+1} + \gamma b(u^n - v^{n+1}). \quad (4.3b)$$

ここで、反応項の近似にのみ若干の工夫を施してある. (4.3) の弱形式を求めてみればわかるが、双線型形式の前の符号が正となる. 上の半離散化に基づき弱形式を求めれば (4.2) に対する全離散化を得る. 求めた弱形式を元に FreeFem++ のプログラムは例えば次のようになる.

Program 4.1: GM.edp

```

1 // Gierer-Meinhardt with Saturation
2 //
3 load "iovtk"
4
5 real Tmax=50.0;
6 int Nmax=10000;
7 real tau=Tmax/Nmax;
8
9 mesh Th=square(150,150,[2*x,2*y]);
10 fespace Uh(Th,P2);
11 Uh u,v,oldu,oldv,phi;
12
13 real eps=0.01;
14 real d=30.0; // ratio of diffusion
15 real a=0.05;
16 real b=3.9;
17 real gamma=100.0;
18 real us,vs,oldus; // for Newton method
19

```

*29 簡略化したモデルに FitzHugh-Nagumo モデルがある.

*30 線形化行列の各成分の符号によって型が分類されている.

*31 $a = 0$ とすると、 v が小さい時に u の増加率は極端に大きくなる.

*32 $\varepsilon = 0$ とすることで得られる常微分方程式系.

```

20 problem GM1(u, phi)=
21   int2d(Th)(u*phi/tau+eps*(dx(u)*dx(phi)+dy(u)
22     *dy(phi))+gamma*u*phi)
23   -int2d(Th)(oldu*phi/tau)-int2d(Th)((gamma*((
24     oldu*oldu)/(oldv+a*oldu*oldu))*phi));
25 problem GM2(v, phi)=
26   int2d(Th)(v*phi/tau+eps*d*(dx(v)*dx(phi)+dy(v)
27     *dy(phi))+gamma*b*v*phi)
28   -int2d(Th)(oldv*phi/tau)-int2d(Th)((gamma*b
29     *(oldu*oldu))*phi);
30 // Compute Equilibrium by Newton's Method
31 oldus=1.0; // Initial guess
32 while(1){
33   us=oldus-(a*oldus^3+oldus-1)/(3*a*oldus
34     ^2+1);
35   if(abs(us-oldus)<1.0e-10) break;
36   else oldus=us;
37 }
38 vs=us*us;
39 cout << "Equilibrium_of_u_v=" << us << endl;
40
41 randinit; // Initialize random seed
42 int n=0;
43 oldu=us+0.01*sin(randreal3()); // initialdata
44 oldv=vs; // initialdata
45 savevtk("GM_"+n+".vtk", Th, oldu, oldv, dataname="
46   Activator_Inhibitor");
47
48 for(n=1; n<=Nmax; n++){
49   GM1; // Solve (4.3a)
50   GM2; // Solve (4.3b)
51
52   if(n%50==0){
53     savevtk("GM_"+n/50+".vtk", Th, u, v, dataname="
54       Activator_Inhibitor");
55   }
56   oldu=u;
57   oldv=v;
58 }

```

Program 4.1 について簡単にコメントをしておこう。

- 28 行目から 33 行目は Newton 法により平衡点 (定数定常解) の近似値を計算している。
- 37 行目は乱数の種を初期化する命令である。39 行目で活性因子 u に対する初期値 u_0 を平衡点への微小摂動として与えている。抑制因子 v に対する初期値 v_0 は平衡点 (定数定常解) である。
- 47 行目から 49 行目は計算結果を VTK ファイルとして保存する部分であるが、 $N = 10,000$ と計算回数が多いため、条件分岐を用いて 50 回毎に計算結果を保存するような処理を入れてある。

Program 4.1 を用い、数値計算を $\varepsilon = 0.01, d = 30.0, \gamma = 100.0$ として行う。 (a, b) は図 4.1 を参考に拡散誘導不安定性が起こるような組を選ぶ。ここでは、 $(a, b) = (0.05, 3.9), (0.05, 1.2), (0.3, 1.0)$ という 3 つの組について数値計算を行った^{*33}。その結果、得られた数値解は図 A.2, 図 A.3 のようになり、殆んど空間一様な状態が不安定化し Turing パターンの形成が確認できる。実際、図 A.3(a),(c) のような模様は魚類の体表に見られる。例

^{*33} Program 4.1 の 15 行目, 16 行目を適宜修正する。

えば、メガネモチノウオ等が代表的である [18]。

4.2.2. ダンベル型領域における数値計算

有限要素法の利点の一つは複雑な形状の領域における問題を簡単に扱うことが出来る事である。ここでは、二つの円盤の間を細い通路によって繋いだ領域を考える。このような領域はダンベル型領域と呼ばれる。 $r, 2l, 2h$ はそれぞれ円盤の半径, 通路の長さ, 通路の幅とするとき、

$$B_{r,l} = \{(x, y) | (x - (\pm(l+r)))^2 + y^2 < r^2\}, \quad (4.4)$$

$$C_{l,r} = \{(x, y) | -l < x < l, -h < y < h\} \quad (4.5)$$

を上手くつなげばよい^{*34}。ここでは、スポットパターンを生み出す $(a, b) = (0.05, 1.2)$ を選びダンベル型領域において数値計算をしてみよう。但し、通路の幅 h はスポットの直径よりも小さくなるように選ぶ。また、 $\gamma = 4.0$ とした。初期摂動として左側の円盤の中心にのみ微小摂動を与えて数値計算を行ったところ、図 A.4 を得た。時間発展に伴い、空間全体にスポットが形成されるが、通路内にはスポットが入りきらない為、通路内での解の形状は空間 1 次元問題のそれに近い。

4.3. BZ 反応とオレゴネータ

反応拡散系に関する 2 つ目の題材として Belousov-Zhabotinskii 反応 (BZ 反応) の数理モデルであるオレゴネータを扱う^{*35}。BZ 反応は化学振動反応の典型例として知られている化学反応である。

BZ 反応のメカニズムを数理的に解明しようという試みは 1970 年前後に各地で開始された。1970 年代初頭に Field, Körös, Noyes は BZ 反応における 3 つの反応中間体濃度を未知関数とするオレゴネータと呼ばれる数理モデルを導出した [6, 7]。本稿でも BZ 反応の数理モデルとしてオレゴネータを採用する。

オレゴネータは 3 変数の常微分方程式系であるが、3 つのうち 1 つの変数に関しては準定常近似を施すことで、2 変数の系へ落とすことが出来る^{*36}。

以下では、拡散効果を取り入れた 2 変数のオレゴネータを考える。以下の (4.6) はセリウムを触媒とする BZ 反応の数理モデルとして知られている^{*37}。

$$\frac{\partial u}{\partial t} = \varepsilon \Delta u + \frac{\gamma}{\varepsilon} \left(u(1-u) - f v \frac{u-q}{u+q} \right) \quad (4.6a)$$

$$\frac{\partial v}{\partial t} = \varepsilon d \Delta v + \gamma(u-v) \quad (4.6b)$$

ここで未知関数 u, v はそれぞれ亜臭素酸 HBrO_2 およびセリウムイオン Ce^{4+} の濃度を表わす。 ε, d, f, q は正の定数である。以下では、 u, v に対する境界条件は齊次 Neumann 条件とする。

^{*34} 実際には重なりが生じないように上手く繋ぐ必要がある。

^{*35} BZ 反応およびオレゴネータについては [25] が詳しい。

^{*36} Tyson [47], 三池 他 [25, 2.4.2] を参照。

^{*37} フェロインを触媒とする場合には RZ メカニズムに基づくモデルがある。(4.6a) の反応項の形が変わる。

4.3.1. 離散化と数値計算

オレゴネータに対して数値計算を行う方法は (4.2) のそれと殆んど同じである為、ここでは反応項の近似についてのみ述べる。(4.6) において反応項を以下のように近似する。

$$u(1-u) - fv \frac{u-q}{u+q} \approx u^n(1-u^{n+1}) - fv^n \frac{u^{n+1}-q}{u^n+q}, \quad (4.7a)$$

$$u-v \approx u^n - v^{n+1}. \quad (4.7b)$$

アイデアは (4.2) において反応項を近似した際と同様であり、弱形式を考えた際に双線形式の前の符号が全て正となるような近似を考えている。FreeFem++ のプログラムは Program 4.1 と殆んど同じようにすればよいのでソースコードは紙数の都合もあり割愛する。

4.3.2. 螺旋波の再現

以下では、 $\varepsilon = 0.1, f = 1$ として数値計算を行う。

(4.6) を用いて BZ 反応に現れるターゲットパターンやスパイラルパターンを数値的に再現する為には、まずは $\varepsilon = 0$ とした反応系に関する観察が必要である^{*38}。即ち、反応系が振動解を持つような q を調べておく必要がある。相平面解析等により、 q が $q < 0.079$ 程度に小さければよいことがわかる^{*39}。

最初に初期値 u_0, v_0 を以下のように与えて数値計算を行う。

$$u_0(\mathbf{x}) = u_s + 0.1\chi_{\{x < 5.0, 4.5 < y < 5.5\}}, \quad (4.8a)$$

$$v_0(\mathbf{x}) = v_s + 0.5\chi_{\{y < 5.0\}} v_s - 0.5\chi_{\{y < 5.0\}} v_s \quad (4.8b)$$

ここで u_s, v_s は (4.6) に対する平衡点であり^{*40}、 χ_A は A 上の特性関数を表わす^{*41}。 u_0 については平衡点よりも大きい値をとる部分があり、 v_0 については平衡点よりも $0.5v_s$ だけ大きい部分と $0.5v_s$ だけ小さい部分に分ける。

計算領域は $\Omega = (0, 10) \times (0, 10)$ とし、パラメータは $q = 0.01, d = 0.01, \gamma = 10.0, T = 10.0, N = 20000$ ($\tau = 0.0005$) とし、メッシュ \mathcal{T}_h は次を用いた。このとき、図 A.5 を得た。

```
1 mesh Th=square(150, 150, [10*x, 10*y]);
```

次に Gierer-Meinhardt 系を考えた際と同じように、初期データ (u_0, v_0) を平衡点付近 (u_s, v_s) にランダムな微小摂動を加えたものとしてみよう。但し、 $\gamma = 20.0$ とした。このとき、図 A.6 を得た。

いずれの場合も螺旋波が時間発展に伴い計算領域全体を埋め尽くす事が観察される。実際の BZ 反応では同心円状に広がるターゲットパターンも観察される。ターゲットパターンを再現するには例えば、(4.8b) を $v_0(\mathbf{x}) \equiv v_s$ に変更すればよい。

^{*38} 闇雲に f, q を設定しても意図した数値解は得られない。

^{*39} 実際に、実験から導かれる q の値は 0.079 よりも遥かに小さい為、この数値に関しては現象の立場からも適切である。

^{*40} $u_s = v_s$ となる事は (4.6b) から明らか。

^{*41} FreeFem++ では bool を用いれば特性関数は簡単に表現できる。

5. Navier-Stokes 方程式

本節では非圧縮性粘性流体の運動を扱う。なお、話を簡単にする為に流れは平面流に限定する。

5.1. Navier-Stokes 方程式

非圧縮性粘性流体の運動は次の Navier-Stokes 方程式系によって支配される^{*42}。

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \Delta \mathbf{u} - \nabla p + \mathbf{f}, \quad (5.1a)$$

$$\operatorname{div} \mathbf{u} = 0. \quad (5.1b)$$

$\mathbf{u} = (u_1(t, \mathbf{x}), u_2(t, \mathbf{x}))$ が速度場、 $p = p(t, \mathbf{x})$ が圧力場であり、これらが未知関数である。 $\mathbf{f} = \mathbf{f}(t, \mathbf{x})$ は流体に対して外から働く力であり、これは既知のベクトル場である。例えば、重力のような力が該当する。 $\nu > 0$ は粘性係数と呼ばれるパラメータである。

(5.1a) は流体の運動方程式である。左辺は流体の加速度を表し、右辺が流体に働く力の総和であり、右辺第 1 項から順に粘性項、圧力項、外力項と呼ぶ。Cauchy の応力テンソル $\mathbb{T}(\mathbf{u}, p) = 2\nu \mathbb{D}(\mathbf{u}) - p\mathbb{I}$ を用いれば、 $\nu \Delta \mathbf{u} - \nabla p = \operatorname{div} \mathbb{T}(\mathbf{u}, p)$ と書ける。ここで、 $\mathbb{D}(\mathbf{u})$ は歪みテンソルであり、以下のように定められる^{*43}。

$$\mathbb{D}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + {}^t(\nabla \mathbf{u})) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)_{i,j=1,2} \quad (5.2)$$

(5.1a) の左辺に関しては $D\mathbf{u}/Dt$ と書く場合もある。この表記は質量微分や Lagrange 微分と呼ばれる。(5.1b) は質量保存則から導かれる式であり連続の式と呼ばれる。

Navier-Stokes 方程式の導出や詳しい物理的な背景については、今井 [19] や Serrin [35] を参照されたい。また、数学的な取り扱いについては多方面からの研究がある。例えば、Galdi [8], Giga, Giga & Saal [10, Chapter 2], Ladyzhenskaya [24], 岡本 [30], 柴田・久保 [37], Sohr [40], Temam [45] やこれらの参考文献を参照するとよい。有限要素法による Navier-Stokes 方程式の取り扱いについては、Pironneau [32], Girault & Raviart [11] を参照されたい。

5.2. 非圧縮条件と圧力

Navier-Stokes 方程式は数学的に扱うにせよ、数値的に扱うにせよ非常に難しい方程式である。Navier-Stokes 方程式の取り扱いを難しくさせる要因は幾つかある。方程式の非線形性もその一つであるが、それについては次の小節で述べる。Navier-Stokes 方程式系が反応拡散系のような非線形問題と大きく異なる部分は (5.1a) における圧力項 ∇p の存在と、非圧縮条件 (5.1b) による \mathbf{u} の発散に対する制約の有無である。

(5.1) は圧力 p に関する時間発展方程式ではないので、逐次代入により Navier-Stokes 方程式の数値解を求める

^{*42} 簡単のため密度は $\rho \equiv 1$ とした。

^{*43} $x_1 = x, x_2 = y$ である。

際に各時間ステップにおける圧力 p^n をどのように求めるのかが一つの問題となる。その為には、運動方程式 (5.1a) から p^n に関する情報を取り出せば良い。 (5.1a) に発散を作用すれば p についての Poisson 方程式が得られるが、この事実から p は非局所的である。即ち、ある領域上の一点における圧力の値を知るには領域全体における速度場の情報が必要となる。更に、圧力 p には境界条件は設けないので⁴⁴ Poisson 方程式を解くことは容易ではない。また、数値的には非圧縮条件 (5.1b) を保ち続けることは難しい。時間発展に伴い、数値解は非圧縮性を失っていく。

このように Navier-Stokes 方程式には非圧縮条件と圧力項の存在により、他の非線形問題にはない難しさがある。こうした困難を克服する方法には色々な方法が知られているが、ここでは処罰法 (ペナルティ法) と呼ばれる方法を紹介する⁴⁵。 (5.1b) を以下の \mathbf{u} と p の間の関係式によって取り替える。

$$\operatorname{div} \mathbf{u} = -\varepsilon p \quad (5.3)$$

ここで、 $\varepsilon > 0$ はパラメータである。形式的に $\varepsilon = 0$ とすれば非圧縮条件 (5.1b) を得る⁴⁶ので、十分に小さい $\varepsilon > 0$ を用いれば非圧縮条件 (5.1b) は近似的に満足される。更に、(5.3) を p について解き、(5.1a) に代入すれば \mathbf{u} について閉じた方程式系を得るので、圧力 p が時間発展をしないという問題をも同時に克服する事が出来る。更に、処罰法を用いることで Stokes 方程式の変分構造が変わる為、有限要素法の観点からも Stokes 問題は幾分、扱いやすくなる⁴⁷。但し、処罰法を用いた場合には厳密には Navier-Stokes 方程式ではなく、その近似問題の数値計算を行うことになる点に注意されたい。

5.3. 特性曲線法に基づく質量微分の近似

Navier-Stokes 方程式は速度場 \mathbf{u} に関して 2 次の非線形性を持つ非線形偏微分方程式である。但し、前節で扱った反応拡散系と大きく異なるのは、非線形項が空間微分を伴う非線形項である点である。このような非線形項は移流項と呼ばれる。

離散化手法を問わず、移流項を数値的に上手く扱うには工夫が必要である。FreeFem++ には物質微分に対して、特性曲線法に基づく方法が用意されている。以下、その概要を述べる⁴⁸。

\mathbf{w} を既知のベクトル場とし、スカラー場 $\phi = \phi(t, \mathbf{x})$ の

⁴⁴ 条件が過剰になるため。

⁴⁵ Temam [44] により導入された近似。

⁴⁶ 実際には、連続の式の代わりに (5.3) を採用すれば Navier-Stokes 方程式に対する処罰法を適用した問題の解は ε に依存して決まるため、 $\varepsilon \rightarrow +0$ において $p = p_\varepsilon$ が大きくなるという保証がない。近似の正当化については、Shen [36] や Yamaguchi [48] を参照されたい。

⁴⁷ 詳しくは菊地 [21] を参照されたい。

⁴⁸ 詳細は Pironneau [32, Chapter 3], Hecht [14, §9.5], 田端 [43] を参照せよ。

質量微分を考える。

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \mathbf{w} \cdot \nabla\phi \quad (5.4)$$

$0 < t \leq T$ 上で定義されたベクトル場 $\mathbf{X}(t)$ が常微分方程式

$$\dot{\mathbf{X}}(t) = \mathbf{w}(t, \mathbf{X}(t)) \quad (5.5)$$

を満たしているとする。このとき、合成関数の微分公式と $\mathbf{X}(t)$ が (5.5) の解であることから、 $\phi(t, \mathbf{X}(t))$ を t について微分すると、以下を得る。

$$\begin{aligned} \frac{d\phi}{dt}(t, \mathbf{X}(t)) &= \frac{\partial\phi}{\partial t} + \frac{\partial\phi}{\partial x} \dot{X}_1 + \frac{\partial\phi}{\partial y} \dot{X}_2 \\ &= \frac{D\phi}{Dt}(t, \mathbf{X}(t)) \end{aligned} \quad (5.6)$$

そこで、 $\tau > 0$ を時間ステップサイズとして、上式の左辺を後退 Euler 近似すると以下を得る。

$$\frac{D\phi}{Dt}(t, \mathbf{X}(t)) = \frac{\phi(t, \mathbf{X}(t)) - \phi(t - \tau, \mathbf{X}(t - \tau))}{\tau} + O(\tau) \quad (5.7)$$

$t_n = n\tau$ とする。 $\mathbf{x} = (x, y) \in \Omega$ を任意の点として、初期条件 $\mathbf{X}(t_n) = \mathbf{x}$ を与えた (5.5) の初期値問題を $t_{n-1} < t < t_n$ で考える。このとき、 $t = t_{n-1}$ における (5.5) の真の解 $\mathbf{X}(t_{n-1})$ は (5.5) に対する Euler 法による近似解 $\mathbf{X}^n = \mathbf{x} - \mathbf{w}(\mathbf{x}, t_n)\tau$ を用いて

$$\mathbf{X}(t_{n-1}) = \mathbf{X}^n + O(\tau^2) \quad (5.8)$$

と書ける。従って、Taylor の定理と (5.7), (5.8) より

$$\frac{D\phi}{Dt}(t_n, \mathbf{x}) = \frac{\phi(t_n, \mathbf{x}) - \phi(t_{n-1}, \mathbf{X}^n)}{\tau} + O(\tau) \quad (5.9)$$

が得られる。ここで、誤差項 $O(\tau)$ を無視したものが ϕ の物質微分に対する一つの近似を与える。

FreeFem++ では上で述べたような特性曲線法に基づく近似をするための命令として `convect` が用意されている。

上で述べた質量微分の近似と (5.3) の近似を用いて、(5.1) に対する時間方向の離散化を行うと

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n \circ \mathbf{X}^n}{\tau} = \nu \Delta \mathbf{u}^{n+1} - \nabla p^{n+1}, \quad (5.10a)$$

$$\operatorname{div} \mathbf{u}^{n+1} + \varepsilon p^{n+1} = 0 \quad (5.10b)$$

を得る。熱方程式や反応拡散系を考えた際と $\mathbf{u}^n \circ \mathbf{X}^n$ が既知であるとすれば、(5.10) を数値的に解くことは Stokes 方程式系へと帰着される。そこで、Stokes 方程式系の扱いを小節を改めて述べよう。

5.4. 定常 Stokes 流に対する Cavity-Flow 問題

ここでは Navier-Stokes 方程式の空間離散化に関連し、線形の定常 Stokes 方程式に対して Cavity-Flow 問題を考える⁴⁹。

⁴⁹ 熱伝導方程式や反応拡散系では時間方向の半離散化により得られる問題が Poisson 方程式であったのに対し、Navier-Stokes 方程式では圧力項の存在により Stokes 方程式となる。

$\Omega = (0, 1) \times (0, 1)$ における次の定常 Stokes 方程式系の境界値問題を考える。

$$-\nu \Delta \mathbf{u} + \nabla p = \mathbf{0}, \quad \mathbf{x} \in \Omega, \quad (5.11a)$$

$$\operatorname{div} \mathbf{u} = 0, \quad \mathbf{x} \in \Omega, \quad (5.11b)$$

$$\mathbf{u} = \mathbf{e}_x, \quad \mathbf{x} \in \Gamma_{\text{Top}}, \quad (5.11c)$$

$$\mathbf{u} = \mathbf{0}, \quad \mathbf{x} \in \partial\Omega \setminus \Gamma_{\text{Top}} \quad (5.11d)$$

ここで, $\Gamma_{\text{Top}} = \{\mathbf{x} = (x, y) \in \bar{\Omega} \mid 0 \leq x \leq 1, y = 1\}$ とおいた。Stokes 方程式系は Navier-Stokes 方程式系の自明解の周りでの線形化問題として得られる系であり, 十分に遅い流れを記述する事が出来る^{*50}。(5.11) は非圧縮性粘性流体によって占められた矩形の領域 Ω において, 上側の境界が x 軸に平行に等速 1 で動いている場合の遅い定常流を記述する問題である。Stokes 方程式系に対する Cavity-Flow 問題は非圧縮性粘性流体の数値計算における一つのベンチマークとして用いられている。

以下では, (5.11) の解を FreeFem++ により数値計算し, Paraview により可視化する。

5.4.1. 流れ関数と渦度

数値計算を行う前に, 渦度と流れ関数を導入しておく。流れ関数は後に流線を描くために用いる^{*51}。速度場 \mathbf{u} に対して, 以下で定まるスカラー場 $\omega = \omega(\mathbf{x})$ を渦度という^{*52}。

$$\omega = \operatorname{rot} \mathbf{u} = \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y} \quad (5.12)$$

また, 流れ関数 $\phi = \phi(\mathbf{x})$ を速度場 $\mathbf{u}(\mathbf{x})$ に対して, 以下の関係をみたすものとする。

$$u_1 = \frac{\partial \phi}{\partial y}, \quad u_2 = -\frac{\partial \phi}{\partial x} \quad (5.13)$$

このような ϕ が得られているとすると, 明らかに $\operatorname{div} \mathbf{u} = 0$ を満たす。また, ϕ は以下の Poisson 方程式を満たすことは (5.12) と (5.13) からわかる。

$$-\Delta \phi = \omega, \quad \mathbf{x} \in \Omega. \quad (5.14)$$

もし境界上の流速 $\mathbf{u}|_{\partial\Omega}$ が得られていれば, ϕ に対する境界条件は \mathbf{u} から決定することが出来る。従って, (5.14) を解けば流れ関数 ϕ を求めることが出来る。

(5.11) において連続の式をペナルティ法 (5.3) により近似した問題を弱形式化する。試験関数を (\mathbf{v}, q) とし, Gauss の発散定理を用いて計算すると,

$$0 = \nu(\nabla u_1, \nabla v_1) + \nu(\nabla u_2, \nabla v_2) - (p, \operatorname{div} \mathbf{v}), \quad (5.15a)$$

$$0 = (\operatorname{div} \mathbf{u} + \varepsilon p, q) \quad (5.15b)$$

を得る。また, 流れ関数 ϕ に対する Poisson 方程式の弱形式化は第 2 節で述べた方法と同様である。

^{*50} $\mathbf{u} \cdot \nabla \mathbf{u}$ が $\nu \Delta \mathbf{u}$ に比べて十分に小さいとみなせる程度に遅い流れ。

^{*51} Paraview には Stream Tracer というフィルタが用意されているので, 以下の Poisson 方程式を解かなくとも流線を描くことは出来る。

^{*52} 渦度がスカラー場となるのは空間 2 次元の場合のみ。

この弱形式を元に, (5.11) を数値的に解くための FreeFem++ のプログラムは以下の Program 5.1 のようになる。なお, 速度場 \mathbf{u} については $P2$ 要素を用い, 圧力 p については $P1$ 要素を用いる。この要素の組み合わせは Hood-Taylor 要素と呼ばれ, 下限上限条件を満たす代表的な例として知られている。詳しくは, [32, 42] 等を参照されたい。

粘性係数と処罰法のパラメータは $\nu = 0.01, \varepsilon = 10^{-10}$ とした。

Program 5.1: Stokes.edp

```

1 // Stationary Stokes flow
2 macro div(u) (dx(u#1)+dy(u#2)) //EOM
3 load "iovtk"
4
5 mesh Th=square(50,50,[x,y]);
6 fespace Uh(Th,P2); //
7 fespace Ph(Th,P1); // Hood-Taylor
8
9 real nu=1.0; // Viscosity
10
11 Uh u1,u2,v1,v2,phi,psi;
12 Ph p,q;
13
14 // Stationary Stokes eqs. (penalized)
15 problem Stokes([u1,u2,p],[v1,v2,q])=
16   int2d(Th)(nu*(dx(u1)*dx(v1)+dy(u1)*dy(v1)+dx
17     (u2)*dx(v2)+dy(u2)*dy(v2))
18     -p*div(v)-q*div(u)-1e-10*p*q)
19   +on(1,2,4,u1=0,u2=0)
20   +on(3,u1=1,u2=0);
21 // Poisson eq. for Stream function
22 problem Poisson(phi,psi)=
23   int2d(Th)(dx(phi)*dx(psi)+dy(phi)*dy(psi))
24   -int2d(Th)((dx(u2)-dy(u1))*psi)
25   +on(1,2,3,4,phi=0);
26 Stokes;
27 Poisson;
28
29 savevtk("Stokes.vtk",Th,[u1,u2,0],p,phi,
30   dataname="Velocity_Pressure_StreamFunction");

```

- 2 行目は \mathbf{u} に対して $\operatorname{div} \mathbf{u}$ を与えるマクロである。マクロの宣言は `macro` で行い, `//` で終える^{*53}。`u#1` は u_1 , `u#2` は u_2 に対応する。
- 30 行目が VTK 形式でのデータ保存であるが, ベクトル場を保存する場合には 3 次元ベクトルとして保存しなければならない。今の場合には 2 次元流を扱っているため流れの z 成分は 0 としている。

5.4.2. ベクトル場の可視化

ここではベクトル場を可視化する方法を述べる。Program 5.1 を実行して得られる `Stokes.vtk` を Paraview に読み込み, まずフィルター `Cell Data to Point Data` を適用する。更に, 得られるデータに対してフィルター `Glyph` を適用すれば, 速度ベクトルを描くことが出来る。

表示されるベクトル場を整形するにはフィルターの適用により得られる `Glyph` の各数値等を編集すれば良い。

^{*53} 行末の `//` を書かないと動かないので注意。

表示されるベクトルが大きすぎたり、小さすぎたりする場合には Set Scale Factor を編集する (図 5.1). 更に, Coloring を速度場の大きさに応じて表現すれば, ベクトルの大きさをカラースケールとベクトルの大きさで直感的に理解できるようになる. これまでに述べた可視化手法を用いて, Program 5.1 の計算結果を可視化したものが図 A.7 である.

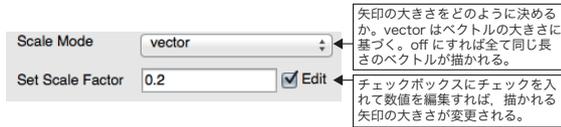


図 5.1: Glyph の Scale に関する部分

5.5. Karman 渦列の形成

平面上に設置された柱状の物体の外側での流れを考える. $B \subset \mathbb{R}^2$ を閉曲線によって囲まれた有界領域とし, $D = \mathbb{R}^2 \setminus \overline{B}$ によって領域 D を定める^{*54}. D は B に対する外部領域と呼ばれる. B を剛体で出来た障害物とし, その外部領域 Ω は非圧縮性粘性流体によって占められているとする. B に一様流 \mathbf{u}_∞ がぶつかるときの物体周囲での流れを考えればよい^{*55}.

一様流 \mathbf{u}_∞ がある程度の速さを超えると, 物体 B の後方に Karman 渦列と呼ばれる渦列が形成される. 実際に, 韓国の済州島の付近ではしばしば Karman 渦列状の雲が観測される. 図 5.2 は河野 [20] に掲載されている 2012 年 10 月に済州島南部に発生した Karman 渦列の衛星写真の一部を切り取ったものである.

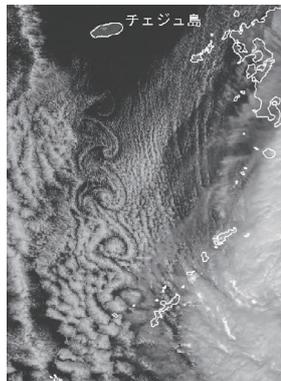


図 5.2: 済州島南部に発生した Karman 渦列 ([20] より)

ここでは Navier-Stokes 方程式を FreeFem++ で数值的に解くことで, Karman 渦列の生成を見よう. そこで, Navier-Stokes 方程式の初期値・境界値問題を次の条件の下で考える.

$$\mathbf{u} = \mathbf{0}, \quad x \in \partial D, t > 0, \quad (5.16a)$$

$$\lim_{|x| \rightarrow \infty} \mathbf{u} = \mathbf{u}_\infty. \quad (5.16b)$$

^{*54} B は単連結でなくともよい.

^{*55} 非圧縮粘性流体中を一定速度で並進する物体周囲の流れについても同様の定式化が可能である.

(5.16a) は物体 B の表面での粘着条件であり, (5.16b) は無限遠方では一様流 \mathbf{u}_∞ に収束する事を要求している. ここで, 一様流 $\mathbf{u}_\infty \in \mathbb{R}^2$ は定数ベクトルである. Navier-Stokes 方程式は回転不変であるから, \mathbf{u}_∞ は x_1 軸に平行になるように選んでも一般性を失わない. そこで, 以下では一様流は x 軸に平行であるとし, $\mathbf{u}_\infty = (U_\infty, 0)$ とおく.

有限要素法では外部問題のような非有界領域における問題を厳密な意味で取り扱うことは出来ない. そこで, Ω を $B \Subset \Omega$ となる位に十分に大きい有界領域であるとして, $\Omega \setminus B$ における問題を考える. $\partial\Omega$ には人工的な境界条件を上手く設定することで実際の外部問題に近い状況を作り出す.

以下では, $\partial\Omega$ は矩形領域とし, 以下のような境界条件を設定する. 一様流と平行な境界 $\Gamma_{\text{Top}}, \Gamma_{\text{Bottom}}$ では自由滑り条件

$$\mathbf{n} \cdot \mathbf{u} = 0, \quad \mathbb{T}(\mathbf{u}, p)\mathbf{n} - (\mathbf{n} \cdot \mathbb{T}(\mathbf{u}, p)\mathbf{n})\mathbf{n} = \mathbf{0}$$

を課す. これは $\Gamma_{\text{Top/Bottom}}$ 上で接方向に応力が働かないことを意味する. 一様流に垂直な境界のうち左側の境界 Γ_{Left} では流入条件として $(u_1, u_2)|_{\Gamma_{\text{Left}}} = (U_\infty, 0)$ を与え, 右側の境界 Γ_{Right} では流出条件として $\mathbb{T}(\mathbf{u}, p)\mathbf{n}|_{\Gamma_{\text{Right}}} = \mathbf{0}$ を課す^{*56}. これは Γ_{Right} で法線方向に応力が働かない条件である.

5.5.1. B が円盤の場合

まずは, B が円盤である場合に数値計算をしてみよう. $\Omega = (0, 100) \times (0, 50)$, $B = \{(x - 30)^2 + (y - 25)^2 < 4\}$, $\nu = 0.01$, $U_\infty = 50.0$ とする.

いま Ω は有界領域であるから, 圧力 p については平均が 0 となるように補正することが出来る. 実際に, \bar{p} を定数とするとき $p = p(t, \mathbf{x})$ が Navier-Stokes 方程式を満足するならば, $p - \bar{p}$ も Navier-Stokes 方程式を満たす. そこで,

$$\bar{p}(t, \mathbf{x}) = \frac{1}{\mathcal{L}(\Omega \setminus B)} \int_{\Omega} p(t, \mathbf{x}) dx \quad (5.17)$$

により得られる計算領域上の平均圧力を用いて $p - \bar{p}$ を改めて p とおくことで圧力の平均を 0 にとることが出来る. なお, $\mathcal{L}(A)$ は A の面積を表わす. ここではこのような補正を用いる.

以下の Program 5.2 は (5.10) と (5.15) に基づき得られる弱形式に基づいたプログラムの一例である.

Program 5.2: KarmanVortex.edp

```
1 // KarmanVortex
2 //
3 macro div(u) (dx(u#1)+dy(u#2)) // EOM
4
5 load "iovtk"
6 real Tmax=10.0;
7 int Nmax=200;
8 real tau=Tmax/Nmax;
```

^{*56} 右側の境界についても \mathbf{u}_∞ で流出させる事も考えられるが, Ω が相当に大きくなければ B の影響を受けて不自然な流れとなる.

```

9
10 real W=100; // Width of Artificial Domain
11 real H=50; // Height of Artificial Domain
12 border C1(s=0,W){x=s;y=0;} // Bottom
13 border C2(s=0,H){x=W;y=s;} // Right
14 border C3(s=0,W){x=W-s;y=H;} // Top
15 border C4(s=0,H){x=0;y=H-s;} // Left
16 border C5(s=0,2*pi){x=0.3*W+2*cos(s);y=0.5*H
17 +2*sin(s);} // Obstacle
18 mesh Th=buildmesh(C1(80)+C2(40)+C3(80)+C4(20)+
19 C5(-30));
20 fespace Uh(Th,P2); // FE space for velocity
21 fespace Ph(Th,P1); // FE space for pressure
22 Uh u1,u2,uold1,uold2,v1,v2;
23 Ph p,q,vortex;
24 real Uinf=50.0; // Velocity of Uniform flow
25 real nu=0.01; // Viscosity Constant
26 real meanp; // average of the pressure
27
28 problem NS([u1,u2,p],[v1,v2,q])=
29 int2d(Th)((u1*v1)/tau+(u2*v2)/tau+nu*(dx(u1)
30 )*dx(v1)+dy(u1)*dy(v1)+dx(u2)*dx(v2)+dy(u2)*
31 dy(v2)+div(u)*div(v))-p*div(v)-q*div(u)-1e
32 -10*p*q)
33 -int2d(Th)(convect([uold1,uold2],-tau,uold1)
34 )*v1/tau+convect([uold1,uold2],-tau,uold2)*
35 v2/tau)
36 +on(C1,C3,u2=0) // Free slip boundary
37 +on(C4,u1=Uinf, u2=0) // Inflow condition
38 +on(C5,u1=0,u2=0); // nonslip boundary
39
40 int n=0;
41 uold1=0;
42 uold2=0;
43
44 for(int n=1;n<=Nmax;n++){
45 NS; // Solve Problem NS
46 vortex=dx(u2)-dy(u1); // 2D Vorticity
47 meanp=int2d(Th)(p)/int2d(Th)(1.0);
48 p=p-meanp; // averaging pressure
49
50 Th=adaptmesh(Th,[u1,u2],p,hmin=0.02,nbvx
51 =20000); // AMR
52
53 savevtk("KarmanVortex_"+n+".vtk",Th,[u1,u2
54 ,0],p,vortex,dataname="velocity_pressure_v
55 ortexicity");
56 uold1=u1; // Refresh
57 uold2=u2; // Refresh
58 }

```

- 18 行目では Ω から \mathcal{B} をくり抜くために、 $C(-30)$ としている。曲線の向きに注意されたい。
- 29 行目の $(\operatorname{div} \mathbf{u}, \operatorname{div} \mathbf{v})$ は処罰法を用いて厳密な非圧縮性が失われている為に入れた補正項である。この他にも粘性項から生じる弱形式を $(\mathbb{D}(\mathbf{u}), \mathbb{D}(\mathbf{v}))$ とする考え方もある。
- 42 行目, 43 行目で圧力平均を 0 にしている。
- 46 行目ではアダプティブ・メッシュの計算を行っている。流れが急激に変化する領域では細かなメッシュを採用し、そうでない領域では粗いメッシュを採用する。このような方法は AMR^{*57} と呼ばれる。

*57 Adaptive Mesh Refinement の略。

FreeFem++ ではアダプティブ・メッシュの計算はいとも簡単に実現可能であり、このように命令 `adaptmesh` を用いるだけでよい。この命令には様々なオプションがあり、最適化メッシュを細かく制御する事が出来る。詳しくは [14, §5.5] を参照して欲しい。

図 A.8 は、Program 5.2 を用いて数値計算を行った結果を元に可視化したものである。Karman 渦列が再現されていることがわかる。

5.5.2. \mathcal{B} が楕円盤の場合

次に \mathcal{B} が楕円盤である場合を考える。細長い楕円盤を擬似的な翼に見立て、迎角 θ を変更して障害物後方に現れる Karman 渦の様子を見よう。

$a > b > 0$ として、楕円の方程式 $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ を $x = a \cos s, y = b \sin s$ ($0 \leq s < 2\pi$) によってパラメータ表示する。更に、回転行列 $R(\theta)$ を用いて x, y を θ だけ回転することで、回転した楕円のパラメータ表示を得る。従って、Program 5.2 の 16 行目をこのような楕円盤を描くように変更すればよい。実際に、粘性係数 ν と一様流 U_∞ は円柱周りの流れを計算した際と同じものを採用し、迎角 θ を $0, \pi/24, \pi/12, \pi/6$ と変えて数値計算を行うと $t = 10.0$ で図 A.9 のような計算結果を得る。

5.5.3. 複数の障害物を配置した場合

上で述べた 2 つの数値計算結果はいずれも Ω 内に配置された障害物が 1 つの場合であった。ここでは、同一形状の障害物を複数配置した場合の流れを考える。具体的には、上と同じ Ω 内に半径 2 の円盤を等間隔で 3 つ配置し、数値計算を行ったところ、図 A.10 を得た。この計算が本稿に掲載したものである。この計算が本稿に掲載したものである。この計算が本稿に掲載したものである。この計算が本稿に掲載したものである。計算に用いた節点数は平均して 450,000 程度である。

5.6. 気液二相流体の運動

熱対流やプラズマの運動等の複雑な流れは流体の運動のみで閉じていない為、Navier-Stokes 方程式系と別の非線形偏微分方程式系の連立系を考える必要がある。例えば、Boussinesq 方程式系や MHD 系等である。

ここではそうした複雑な流れ問題の例として気液二相流を考える。液体中を上昇する泡の挙動や気体中を落下する雨粒の挙動等を数学的に捉えるには気体と液体の運動および界面の変形を同時に考える必要がある。気体は厳密には圧縮性流体であるが、流れは音速を超えない程度に穏やかであるとして、非圧縮性流体として近似的に取り扱うことにしよう。

時刻 t において領域 Ω は界面 $\Gamma(t)$ によって液体の領域 $\Omega_1(t)$ と気体の領域 $\Omega_2(t)$ に分けられているとする^{*58}。 $\Omega_1(t)$ 内では液体の運動、 $\Omega_2(t)$ 内では気体の運動を考えることになる。最大の問題は界面 $\Gamma(t)$ が流体の運動に伴い時々刻々変化する事であり、界面 $\Gamma(t)$ をどのように捉

*58 $\Gamma(t)$ は複数の閉曲線となる場合もある。

えるかである。

ここではレベルセット法^{*59}を用いることで界面の位置や形状を捉える。紙数の都合もあり詳細には立ち入らないが、 $\phi = \phi(t, \mathbf{x})$ を符号付き距離関数^{*60}するとき、界面は $\Gamma(t) = \{\phi(t, \mathbf{x}) = 0 | \mathbf{x} \in \Omega\}$ と書け、 Ω は ϕ の符号により以下のように分けられる。

$$\Omega(t) = \begin{cases} \Omega_1(t), & \phi > 0, \\ \Gamma(t), & \phi = 0, \\ \Omega_2(t), & \phi < 0, \end{cases} \quad (5.18)$$

また、レベルセット関数 ϕ は以下に記す移流方程式 (5.19c) により駆動される。

レベルセット法による定式化を用いると、領域 Ω における流体の運動および界面 $\Gamma(t)$ の運動は次の Navier-Stokes 方程式と移流方程式の連立系によって記述される。

$$\rho(\phi) \frac{D\mathbf{u}}{Dt} = \text{div}(2\nu(\phi)\mathbb{D}(\mathbf{u}) - p\mathbb{I}) + \sigma\kappa\delta(\phi)\mathbf{n}|_{\Gamma} - \rho(\phi)g\mathbf{e}_y, \quad (5.19a)$$

$$\text{div } \mathbf{u} = 0, \quad (5.19b)$$

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = 0. \quad (5.19c)$$

粘性係数 $\nu = \nu(\phi)$ および密度 $\rho = \rho(\phi)$ は ϕ の符号により以下のように定める。

$$\nu(\phi) = \begin{cases} \nu_1, & \phi > 0, \\ \nu_2, & \phi < 0, \end{cases} \quad \rho(\phi) = \begin{cases} \rho_1, & \phi > 0, \\ \rho_2, & \phi < 0 \end{cases} \quad (5.20)$$

ν_i, ρ_i ($i = 1, 2$) は定数である。また、(5.19a) の右辺における σ は表面張力定数、 κ は界面 $\Gamma(t)$ の曲率、 $\delta(\phi)$ は $\phi = 0$ である点のみに台を持つ Dirac の delta 関数、 $\mathbf{n}|_{\Gamma}$ は界面 $\Gamma(t)$ 上の外向き単位法線ベクトルである。 $g > 0$ は重力加速度を表す定数であり、 $\mathbf{e}_y = (0, 1)$ である。

5.6.1. 気液二相流の数値計算

領域 Ω を設定し、(5.19) を数値的に解くことが出来れば界面 $\Gamma(t)$ の運動や領域内での流体の運動について知見を得ることが出来る。ここでは話を大幅に簡単にする為に、表面張力を無視して数値計算を行う。即ち $\sigma = 0$ とし、簡略化された問題を考える^{*61}。

^{*59} Osher & Sethian [31] により導入された方法であり、気液二相流への応用は Sussman et.al. [41] による差分法による結果が代表的である。

^{*60} 各点で $|\nabla\phi| = 1$ である。

^{*61} 表面張力をどのように数値的に定式化するかは大きな問題である。例えば、[41] では δ に対する以下の近似を用いて表面張力を数値的に表現している。

$$\delta_\alpha(\phi) = \begin{cases} \frac{1}{2\alpha} \left(1 + \cos\left(\frac{\pi\phi}{\alpha}\right)\right), & |\phi| < \alpha, \\ 0, & |\phi| \geq \alpha. \end{cases}$$

ここで $\alpha > 0$ は定数である。 $h > 0$ を $\phi = 0$ が通過するメッシュのサイズとするときに $\alpha \approx 1.5h$ と選ぶ。本稿で扱う計算では表面張力は簡単の為に無視しているが、 $\rho(\phi)$ や $\nu(\phi)$ を連続的に修正する為に、Heviside 関数に対して類似の近似を用いている。

ここでは気液二相流の例として、液体中を気泡が上昇していく様子を (5.19) の数値計算によって再現する^{*62}。数値計算を行う領域は $\Omega = (0, 1) \times (0, 2)$ とする。初期レベルセット関数を

$$\phi(0, \mathbf{x}) = \sqrt{x^2 + (y - 0.5)^2} - 0.25$$

で与える。即ち、初期界面は $\Gamma(0) = \{\mathbf{x} \in \Omega | \phi(0, \mathbf{x}) = 0\}$ である。初期速度は $\mathbf{u}(0, \mathbf{x}) = \mathbf{0}$ とする。また、密度と粘性は $(\nu_1, \rho_1) = (10.0, 1000.0)$, $(\nu_2, \rho_2) = (0.1, 1.0)$ とし、 $g = 0.98$ とする。 Ω の外壁における境界条件は x 軸に平行な境界では粘着条件 $\mathbf{u} = \mathbf{0}$ を課し、 x 軸に垂直な境界では自由滑り条件 $u_2 = 0$ を課す^{*63}。図 5.3 は今回行う数値計算の初期状態の模式図である。

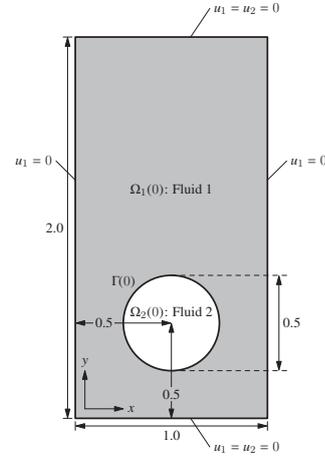


図 5.3: 初期状態の模式図

このとき、AMR を用いて数値計算を行うと液体中を上昇する気泡の挙動が短時間であれば上手く再現される。実際に、 $\tau = 0.002$ として数値計算を行った結果、図 A.11 を得た。

注意 (再初期化). 気液二相流の運動に対して精密な数値計算を行うには、 ϕ が符号付き距離関数となるように $|\nabla\phi| = 1$ を保つような計算が必要である。特性曲線法に基づき (5.19c) を解くと、計算を進めるうちに $|\nabla\phi| = 1$ という性質が時間発展に伴い徐々に失われていく。

そこで、再初期化 (Reinitialization) と呼ばれる処理を組み込む必要があり、[41] にある 1 階双曲型偏微分方程式を用いる方法が主流である。ここでは、最近 Basting & Kuzmin [1] により導入された方法を用いて再初期化を行っている。

6. おわりに

本稿では、非線形偏微分方程式に対する数値計算を FreeFem++ により行い、計算結果を Paraview によって

^{*62} 但し、表面張力を無視している為、気泡の界面の変形はやや不自然である。

^{*63} 同様の設定の数値計算が [17] や [5] で扱われている。

可視化する方法を述べた。また、反応拡散系と流体力学から幾つかの非線形問題を扱い、得られた結果を図 A.2 から図 A.11 に掲載した。

FreeFem++ と Paraview を組み合わせることで、かつては大規模な研究組織でなければ出来なかったような数値計算や可視化を個人レベルでも手軽に試すことが出来るようになった事が伝われば幸いである。

謝辞. 第二著者に対し、様々な可視化技術に関する話題や Paraview の存在をご教授下さいました岡山大学環境理工学部の水藤寛教授に心より感謝申し上げます。

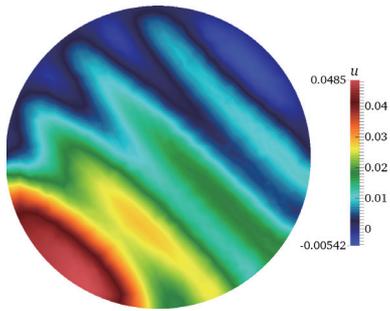
本研究の一部は平成 25 年度富山大学人間発達科学部・学部長裁量経費の援助を受けて行われた。

参考文献

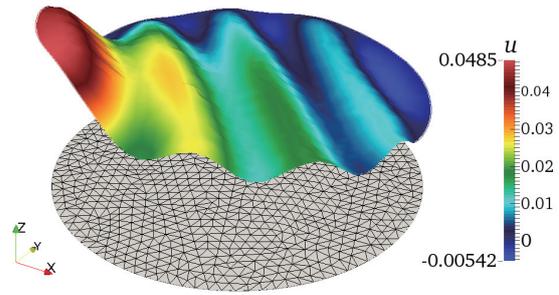
- [1] C. Basting and D. Kuzmin. A minimization-based finite element formulation for interface-preserving level set reinitialization. *Computing*, **95**(1, suppl.):S13–S25, 2013.
- [2] S. C. Brenner and L. R. Scott. *The mathematical theory of finite element methods*. Springer, third edition, 2008.
- [3] H. Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer, 2011.
- [4] P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978.
- [5] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'homme, and M. Ismail. Simulation of two-fluid flows using a finite element/level set method. Application to bubbles and vesicle dynamics. *J. Comput. Appl. Math.*, **246**:251–259, 2013.
- [6] R. J. Field, E. Körös, and R. M. Noyes. Oscillations in chemical systems. II. thorough analysis of temporal oscillation in the bromate-cerium-malonic acid system. *J. Am. Chem. Soc.*, **94**(25):8649–8664, 1972.
- [7] R. J. Field and R. M. Noyes. Oscillations in chemical systems. V. quantitative explanation of band migration in the Belousov-Zhabotinskii reaction. *J. Am. Chem. Soc.*, **96**(7):2001–2006, 1974.
- [8] G. P. Galdi. *An introduction to the mathematical theory of the Navier-Stokes equations*. Springer, second edition, 2011.
- [9] A. Gierer and H. Meinhardt. A theory of biological pattern formation. *Kybernetik*, **12**:30–39, 1972.
- [10] M.-H. Giga, Y. Giga, and J. Saal. *Nonlinear partial differential equations*. Birkhäuser Boston Inc., 2010.
- [11] V. Girault and P.-A. Raviart. *Finite element approximation of the Navier-Stokes equations*. Springer-Verlag, Berlin-New York, 1979.
- [12] 林真. はじめての Paraview. 工学社, 2009.
- [13] F. Hecht. New development in freefem++. *J. Numer. Math.*, **20**(3-4):251–265, 2012.
- [14] F. Hecht. *Freefem++*. Version 3.30. 2014.
- [15] D. Henry. *Geometric theory of semilinear parabolic equations*. Springer-Verlag, 1981.
- [16] A. Hodgkin and A. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, **117**:500–544, 1952.
- [17] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *Internat. J. Numer. Methods Fluids*, **60**(11):1259–1288, 2009.
- [18] 井田斉, 松浦啓一, 藍澤正宏. 魚. 小学館, 2003.
- [19] 今井功. 流体力学 (前編). 裳華房, 1973.
- [20] 河野麻由可. 今月のひまわり画像-2012 年 10 月: 台風第 21 号とカルマン渦. 天気, **59**(12):1074, 2012.
- [21] 菊地文雄. 有限要素法の数値: 数学的基礎と誤差解析. 培風館, 1994.
- [22] 菊地文雄. 有限要素法概説 [新訂版]. サイエンス社, 1999.
- [23] S. Kondo and R. Asai. A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*. *Nature*, **376**:765–768, 1995.
- [24] O. A. Ladyzhenskaya. *The mathematical theory of viscous incompressible flow*. Gordon and Breach Science Publishers, 1969.
- [25] 三池秀敏, 山口智彦, 森義仁. 非平衡系の科学 III: 反応 拡散系のダイナミクス. 講談社サイエンティフィック, 1997.
- [26] 三村昌泰. 反応拡散系への誘い. 三村昌泰 (編), パターン形成とダイナミクス, 非線形・非平衡現象の数値, 第 4 巻, 第 1 章, pp. 1–20. 東京大学出版会, 2006.
- [27] K. Ohmori and N. Saito. Flux-free finite element method with Lagrange multipliers for two-fluid flows. *J. Sci. Comput.*, **32**(2):147–173, 2007.
- [28] 大塚厚二, 高石武史. 有限要素法で学ぶ現象と数値 —FreeFem++ 数値思考プログラミング—. 共立出版, 2014.
- [29] 大石進一. MATLAB による数値計算. 培風館, 2001.
- [30] 岡本久. ナヴィエ-ストークス方程式の数値. 東京大学出版会, 2009.
- [31] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, **79**(1):12–49, 1988.
- [32] O. Pironneau. *Finite element methods for fluids*. John Wiley & Sons Ltd., 1989.
- [33] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Springer-Verlag, Berlin, 1994.
- [34] G. Sadaka. FreeFem++, a tool to solve PDEs numerically. arXiv: 1205.1293v1, 2012.
- [35] J. Serrin. Mathematical principles of classical fluid mechanics. In *Handbuch der Physik (herausgegeben von S. Flügge)*, Bd. 8/1, *Strömungsmechanik I (Mitherausgeber C. Truesdell)*, pp. 125–263. Springer-Verlag, Berlin-Göttingen-Heidelberg, 1959.
- [36] J. Shen. On error estimates of the penalty method for unsteady Navier-Stokes equations. *SIAM J. Numer. Anal.*, **32**(2):386–403, 1995.
- [37] 柴田良弘, 久保隆徹. 非線形偏微分方程式. 朝倉書店, 2012.
- [38] H. Shoji and Y. Iwasa. Labyrinthine versus straight-striped patterns generated by two-dimensional Turing systems. *J. Theoret. Biol.*, **237**(1):104–116, 2005.
- [39] J. Smoller. *Shock waves and reaction-diffusion equations*. Springer-Verlag, second edition, 1994.
- [40] H. Sohr. *The Navier-Stokes equations*. Birkhäuser Verlag, 2001.
- [41] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, **114**:146–159, 1994.
- [42] 田端正久. 偏微分方程式の数値解析. 岩波書店, 2010.
- [43] 田端正久. ガレルキン・特性曲線有限要素法: 理論と応用. 数理解析研究所講義録, **1791**:76–86, 2012.
- [44] R. Temam. Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bull. Soc. Math. France*, **96**:115–152, 1968.
- [45] R. Temam. *Navier-Stokes equations*. AMS Chelsea Publishing, Providence, RI, 2001.
- [46] A. Turing. The chemical basis of morphogenesis. *Phil. Trans. R. Soc. B.*, **237**:37–72, 1952.
- [47] J. J. Tyson. Scaling and reducing the field-koros-noyes mechanism of the belousov-zhabotinskii reaction. *The Journal of Physical Chemistry*, **86**(15):3006–3012, 1982.
- [48] N. Yamaguchi. Mathematical justification of the penalty method for viscous incompressible fluid flows. *RIMS Kokyūroku*, **1830**:127–142, 2013.
- [49] 山口範和. 差分法による流体の運動方程式の数値解析. 第 5 回若手による流体セミナー講義録, pp. 15–50, 2014.

(2014年 5 月 20 日受付)

(2014年 7 月 9 日受理)

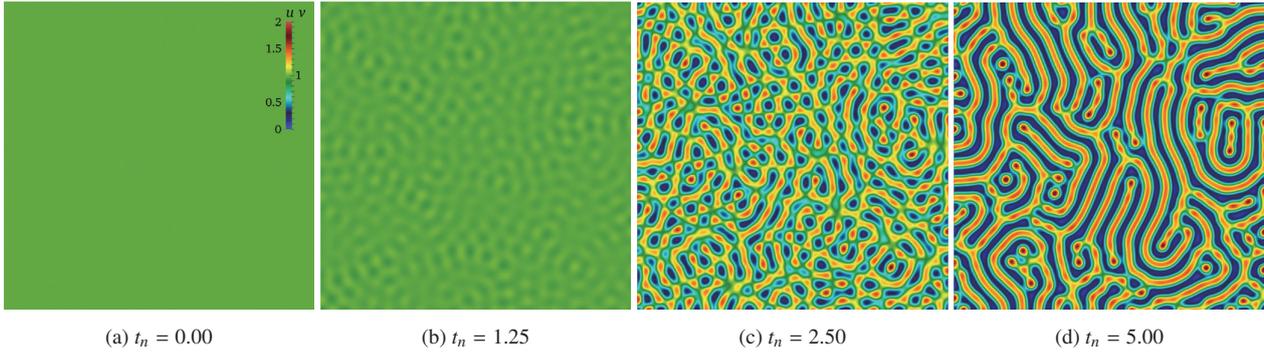


(a) 数値解をカラースケールで可視化したもの.



(b) 数値解を立体的に描いたもの. 同時に計算メッシュも描いてある.

図 A.1: Poisson 方程式 (2.1) の数値解



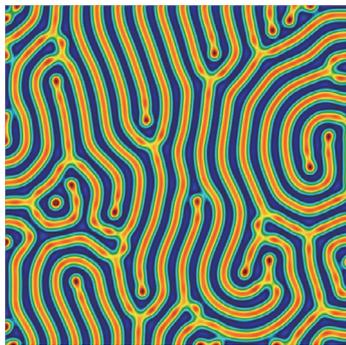
(a) $t_n = 0.00$

(b) $t_n = 1.25$

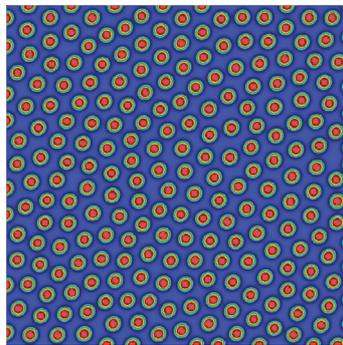
(c) $t_n = 2.50$

(d) $t_n = 5.00$

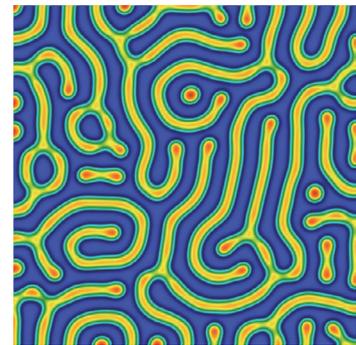
図 A.2: $(a, b) = (0.05, 3.9)$ の場合に (4.2) の活性因子 u の数値解を各タイムステップに関して描いたもの. 初期状態 (a) では殆んど空間一様な状態であるが時間発展に伴い徐々に不安定化し, パターン形成が始まる様子が見取れる. 更に計算を続けると, 図 A.3(a) のようになる. なお, (a) に掲載したカラースケールの凡例は図 A.3, 図 A.4 も共通.



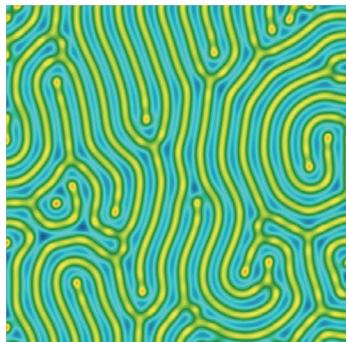
(a) $(a, b) = (0.05, 3.9)$, 活性因子 u



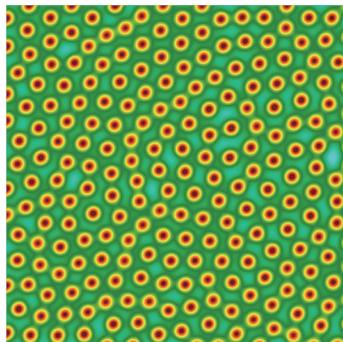
(b) $(a, b) = (0.05, 1.2)$, 活性因子 u



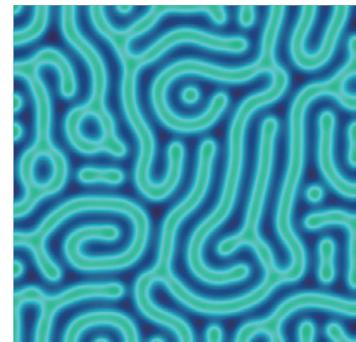
(c) $(a, b) = (0.3, 1.0)$, 活性因子 u



(d) $(a, b) = (0.05, 3.9)$, 抑制因子 v



(e) $(a, b) = (0.05, 1.2)$, 抑制因子 v



(f) $(a, b) = (0.3, 1.0)$, 抑制因子 v

図 A.3: 幾つかのパラメータセット (a, b) に対して同じ初期状態から数値計算を行った結果 ($t_n = 50.0$). パラメータの違いにより形成されるパターンが異なる事が確認できる.

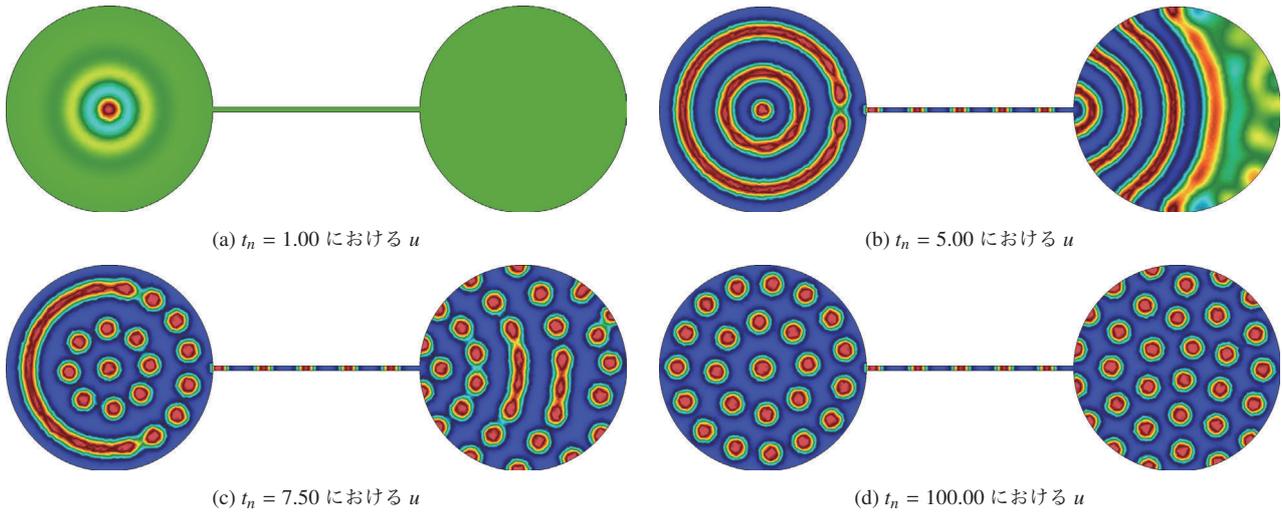


図 A.4: $(a, b) = (0.05, 1.2)$ としたダンベル型領域における (4.2) の数値計算結果. 初期摂動は左側の円盤の中心付近にのみ与えてある. 時間発展に伴い, 通路を通して領域全体が不安定化する. 途中まではストライプ状のパターンを形成するが, 最終的には (d) のようなスポット状のパターンに落ち着く.

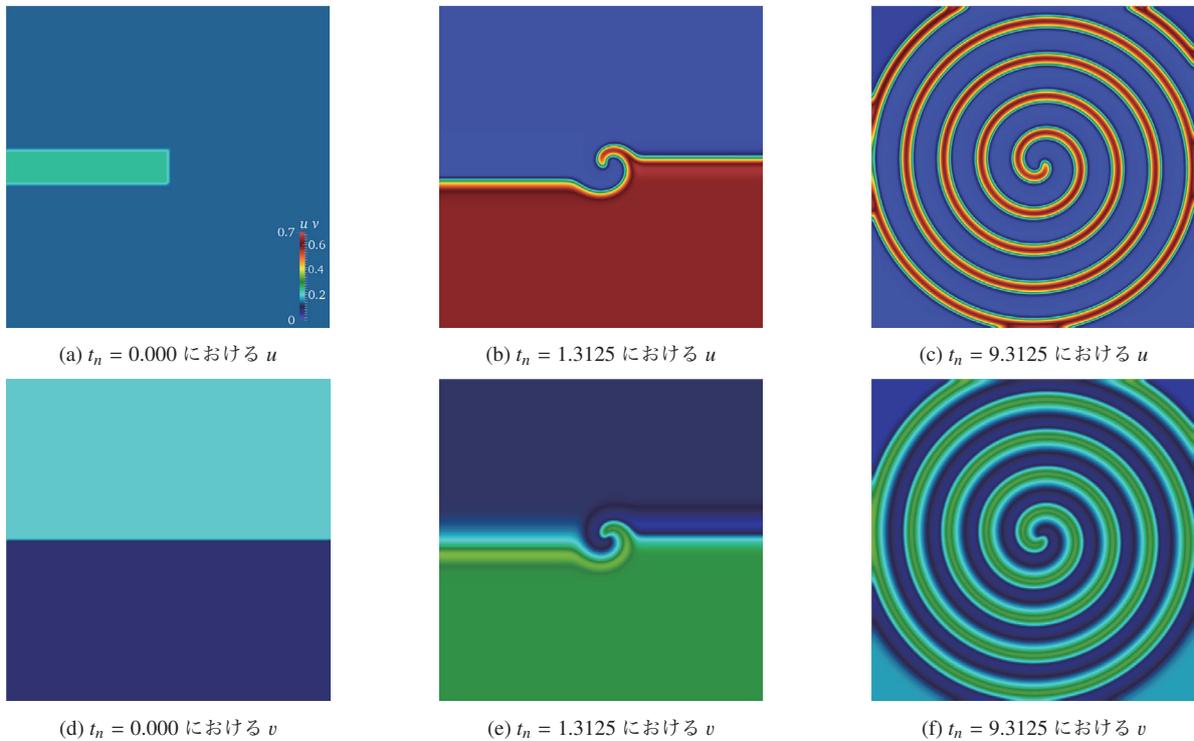


図 A.5: オレゴネータ (4.6) に対する数値解の時間発展の様子. $t_n = 0$ における初期値は u については (a), v については (d) のように与えた. 時間発展に伴い, (b), (d) のように螺旋波が発生し始める. 螺旋は反時計回りに回転しつつ広がり, ある程度の時間が経過すると, (c), (f) のように螺旋が計算領域全体へと広がる. 紙面の都合で割愛するが, v の初期値への摂動を上下で入れ替えれば螺旋が逆向きとなる. なお, (a) の凡例は図 A.6 と共通.

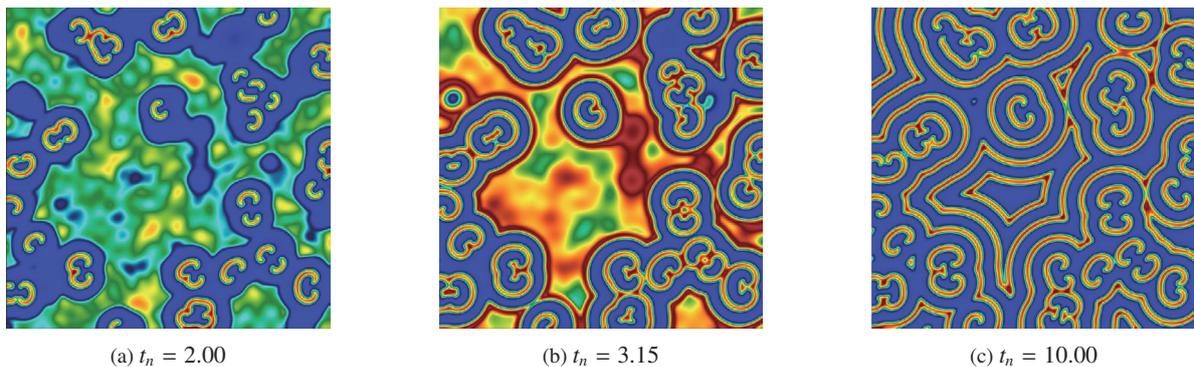


図 A.6: ランダムな初期状態から得られるオレゴネータ (4.6) の数値解 u の様子. ある程度の時刻が経過すると (a) のように各所に螺旋波が現れる. 時間発展に伴い, 螺旋波は領域全体に広がっていき, 最終的には (c) のように複数の螺旋波によって領域全体が埋め尽くされる.

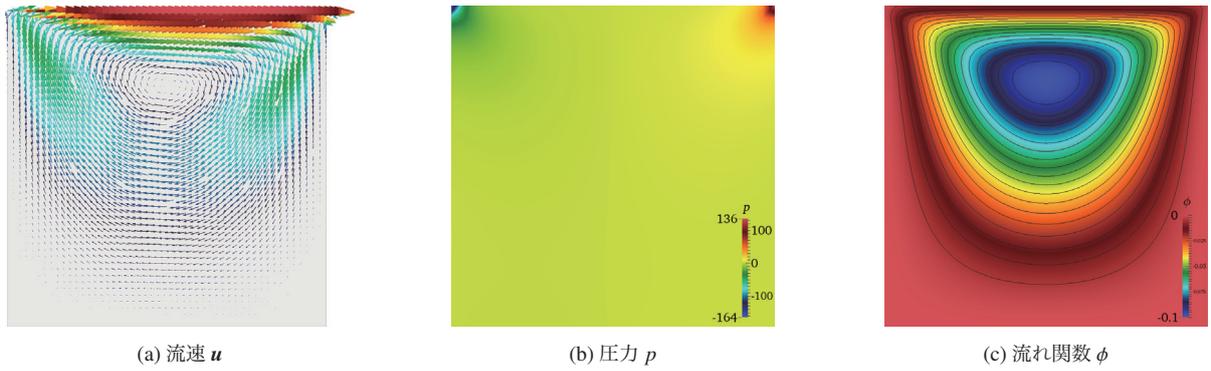


図 A.7: Stokes 方程式系のキャピティ流れの可視化. (a) では速い流れほど大きい矢印で描かれている. (c) では ϕ をカラスケールで描くと同時に等高線を 20 本描いた.

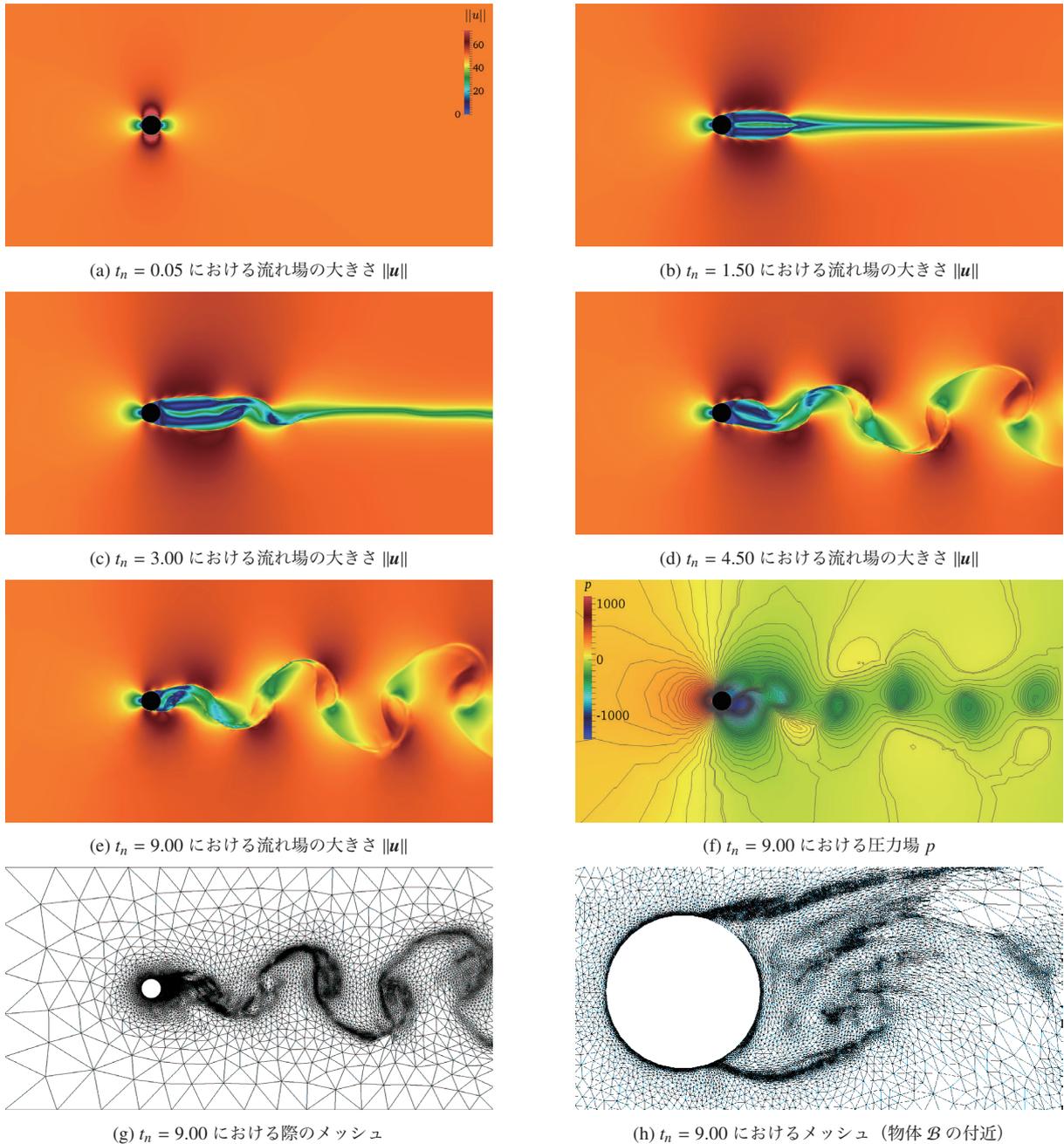


図 A.8: 円柱後方における Karman 渦列の形成. (a) から (e) は速度場 $\|\mathbf{u}\|$ の大きさを用いて可視化したもの. $t_n = 3.00$ 前後で物体後方の流れの対称性が崩れ始め, $t_n = 4.50$ 前後で物体後方に Karman 渦列が形成され, 以後ほぼ周期的な運動を行う. (f) は $t_n = 9.00$ での圧力場 p をカラスケールおよび 100 本の等高線と共に描いた. (g) は $t_n = 9.00$ での計算メッシュを描いたものである. メッシュの形を見るだけでも流れの様子が見て取れる. (h) は特に物体 B の周囲での計算メッシュを拡大して描いたものであり, 物体近傍では非常に細かいメッシュが用いられている様子が見てとれる. なお, 凡例は図 A.9 と共通.

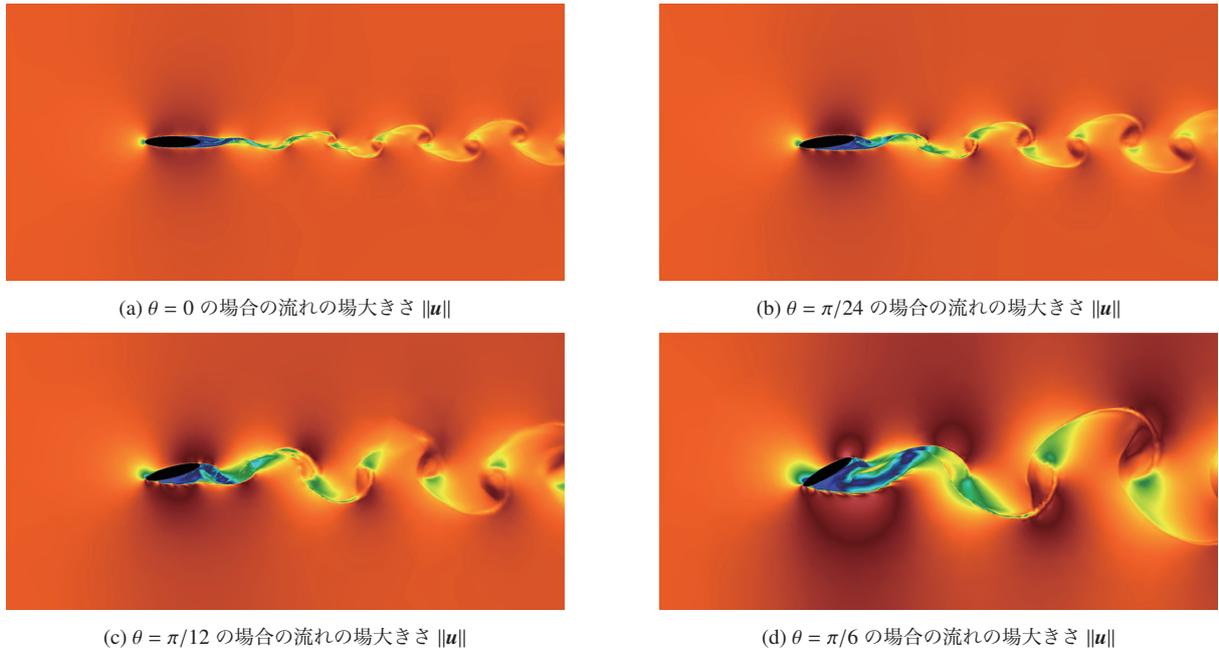


図 A.9: 楕円柱後方における Karman 渦列の形成. (a) から (d) はすべて $t_n = 10.00$ における速度場の大きさ $\|u\|$ を用いて可視化したもの. 迎角 (x 軸と楕円の長軸とのなす角) が大きくなるに連れ, 物体後方には大きな渦が発生する様子が見てとれる.

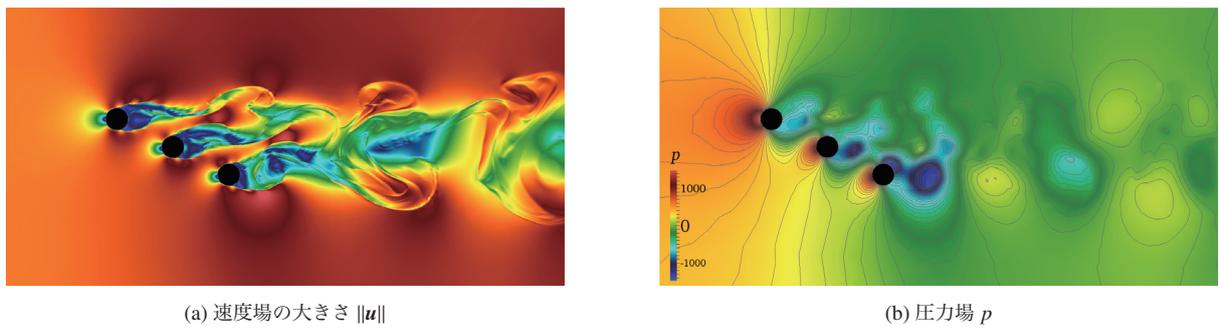


図 A.10: 複数の物体を配置した場合の $t_n = 10.0$ での流れの様子 (a) と圧力場 (b). 物体後方の流れの相互作用により, 非常に複雑な流れになっていることが見える.

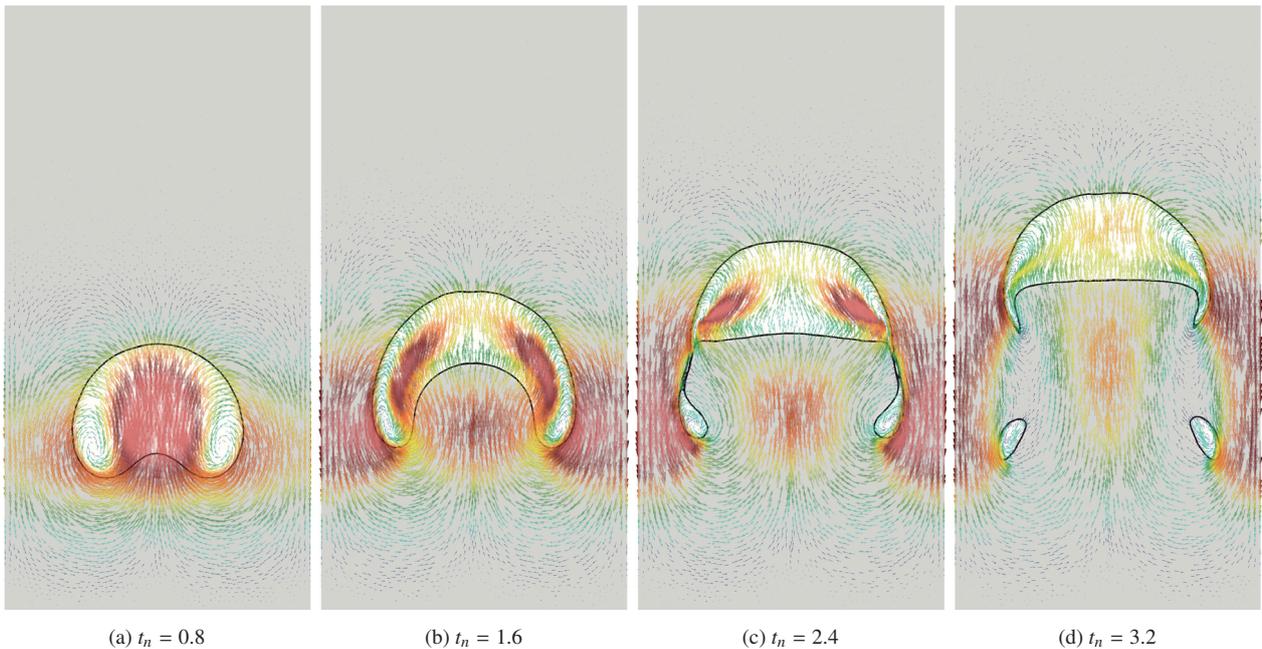


図 A.11: 気液二相流の数値計算結果を流れ場と ϕ の等高線 (ゼロレベルセット) を用いて可視化した結果. 時間発展に伴い, 気泡は変形しつつ上昇し (a), (b), $t_n = 2.4$ 付近で3つの気泡に分かれる (c). 大きな気泡は上昇を続け, 小さな気泡は左右の境界付近に発生した下向きの流れによって, 一旦下へと移動する (d).